2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Fault Estimation and Virtual Sensor FTC Approach for LPV Systems

Saúl Montes de Oca, Damiano Rotondo, Fatiha Nejjari and Vicenç Puig*

***Abstract*— In this paper, a Fault Tolerant Control (FTC) strategy using a virtual sensor for Linear Parameter Varying (LPV) systems is proposed. The main idea of this FTC method is to reconfigure the control loop such that the nominal controller could still be used without need of retuning it. The plant with the faulty sensor is modified adding the virtual sensor block that masks the sensor fault. The suggested strategy is an active FTC strategy that reconfigures the virtual sensor on-line taking into account faults and operating point changes. In order to implement the virtual sensor approach, a fault estimation is required. Here, this fault estimation is provided by formulating it as a parameter estimation problem. Then, a block/batch least square approach is used to estimate additive and multiplicative faults. The LPV virtual sensor is designed using polytopic LPV techniques and Linear Matrix Inequalities (LMIs). To assess the performance of the proposed approach a two degree of freedom helicopter simulator is used.**

***Index Terms*— Fault Tolerant Control, Linear Parameter Varying, Virtual Sensor, Linear Matrix Inequality, TRMS.**

I. Introduction

Fault Tolerant Control (FTC) is a new idea recently introduced in the research literature [1] which allows to maintain current performance close to a desirable one and preserve stability conditions in the presence of component and/or instrument faults. Accommodation capability of a control system depends on many factors such as severity of the failure, the robustness of the nominal system and mechanisms that introduce redundancy in sensors and/or actuators. From the point of view of the control strategies, the literature considers two main groups of techniques: the *active* and the *passive* (see [2] for a review). The *passive FTC techniques* are control laws that take into account the 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. On the other hand, the *active FTC techniques* consist in adapting the control law using the information given by the Fault Detection and Isolation (FDI) block [1]. With this information, some automatic adjustments in the control loop are done after the fault appearance trying to satisfy the control objectives with minimum performance degradation. Most of the FTC methods have been proposed for LTI systems. This paper proposes a FTC approach for non-linear systems that can be approximated by an LPV model. The main advantage of LPV models is that they allow applying powerful linear design tools to complex non-linear models [3]. Various LPV system modeling techniques in the fault-free case are presented in [4]. The LPV theory is mainly used for designing controllers for non-faulty systems, but recently it has also been used for active FTC [5].

Recently, virtual sensors for linear systems have been proposed as a fault accommodation approach (see, e.g. [1]). In particular, this paper extends this virtual sensor approach for FTC to LPV systems. The main idea of this FTC method is to reconfigure the control loop such that the nominal controller could be still used without need of retuning it. The plant with the faulty sensor is modified adding the virtual sensor block that masks the sensor fault and allows the controller to see the same plant as before the fault. The LPV virtual sensor is designed using LMI regions [6] and taking into account the effect of the fault and the operating point. This approach requires to approximate the LPV system in a polytopic way guaranteeing the desired specifications at the expense of introducing some conservatism. As a benefit, controller design can be reduced to solve a convex optimization problem which can be solved using solvers that are very efficient nowadays.

To implement the virtual sensor block, a fault estimation is needed. In the original virtual sensor scheme proposed in [1], the fault estimation method is not provided. This paper proposes a method for computing the fault estimation needed by the virtual sensor that allows to estimate multiplicative and additive sensor faults. The fault estimation is formulated as a parameter estimation problem in such a way that any parameter estimation algorithms (such as least squares, generalized/extended least squares, instrumental variables, maximum likelihood, extended Kalman filter and others) could be used. In general, least-square algorithms can be formulated either in block/batch or recursive on-line forms [7]. The format implemented in this paper is referred to as block least squares following the ideas proposed by [8].

The paper is organized as follows: *Section II* presents the details regarding the proposed FTC strategy based on virtual sensors in the context of LPV systems and the fault estimation procedure. *Section III* presents the LPV controller and a reconfiguration analysis of the proposed strategy. In *Section IV*, the polytopic approximation of a LPV system is presented. *Section V* proposes the design of a virtual sensor by means of LMI pole placement. Finally, *Section VI* describes the two-degree of freedom helicopter used as application example, which shows the performance of the proposed approach.

This work is supported by the Spanish Ministry of Science and Technology through the following projects CYCYT HYFA DPI2008-01996 and CICYT TEOMAPIN DPI2008-00403

S. Montes de Oca, D. Rotondo, F. Nejjari and V. Puig are with Advanced Control Systems Research Group at the Automatic Control Department of Technical University of Catalonia (UPC), Pau Gargallo, 5, 08028 Barcelona, Spain. e-mail: vicenc.puig@upc.edu

II. Fault Estimation and Virtual Sensor for LPV Systems

A. LPV System and Fault Definition

Let us consider an LPV system in state-space form including sensor faults as follows:

$$x_f(k+1) = A(\vartheta_k)x_f(k) + Bu(k) \quad (1)$$

$$y_f(k) = C_f(\gamma_k)x_f(k) + f_y(k) \quad (2)$$

where $x_f(k) \in \Re^{n_x}$ represents the state vector, $u(k) \in \Re^{n_u}$ denotes the control inputs and $y_f(k) \in \Re^{n_y}$ are the sensor outputs including faults. $A(\vartheta_k) \in \Re^{n_x \times n_x}$ and $C_f(\gamma_k) \in \Re^{n_y \times n_x}$ are time-variant matrices while $B \in \Re^{n_x \times n_u}$ is a constant matrix. $f_y(k) \in \Re^{n_y}$ denotes the additive sensor faults and the multiplicative sensor faults are embedded in the matrix $C_f(\gamma_k)$ as follows:

$$C_f(\gamma_k) = diag(\gamma_1(k), \gamma_2(k), \ldots, \gamma_{n_y}(k))C, \quad 0 \leq \gamma_i(k) \leq 1 \quad (3)$$

where γ_i represents the effectiveness of the i^{th} output sensor, such that the extreme values $\gamma_i = 0$ and $\gamma_i = 1$ represent a total failure of the i^{th} sensor and the healthy i^{th} sensor, respectively. For example, when $\gamma_1 = 0.8$, the effectiveness of the first output sensor is 80%. C denotes the faultless output matrix. ϑ_k ($\vartheta_k := \vartheta(k)$) is the system vector of time-varying parameters of dimension n_ϑ that changes with the operating point. This vector is scheduled by some measured system variables p_k ($p_k := p(k)$) that can be estimated using some known function $\vartheta_k = f(p_k)$, named scheduling function.

B. LPV Virtual Sensor

In this paper, the concept of virtual sensor introduced in [1] is extended to non-linear systems that can be approximated by an LPV model. The main idea of this FTC method is to reconfigure the *faulty plant* such that the nominal controller could be still used without need of retuning. The plant with the faulty sensor is modified adding the virtual sensor block that masks the fault and allows the controller to see the same plant as before the fault. The overall scheme includes an *LPV state observer*, an *LPV nominal controller* and the *fault estimation* module.

The reconfiguration structure can be expressed as:

$$y_c(k) = P(\gamma_k)\left(y_f(k) - f_y(k)\right) + C_\Delta(\gamma_k)x_v(k) \quad (4)$$

where $y_c(k)$ is the same (or approximately the same) output as the nominal plant and $x_v(k) \in \Re^{n_x}$ is the virtual sensor state. Matrices $P(\gamma_k)$ and $C_\Delta(\gamma_k)$ are given by:

$$P(\gamma_k) = CC_f(\gamma_k)^\dagger \quad (5)$$

$$C_\Delta(\gamma_k) = C - P(\gamma_k)C_f(\gamma_k) \quad (6)$$

where $C_f(\gamma_k)^\dagger$ is the pseudo-inverse of $C_f(\gamma_k)$. Notice that $C_\Delta(\gamma_k) = 0$ if the following rank condition is satisfied:

$$\text{rank}\left(C_f(\gamma_k)\right) = \text{rank}\begin{pmatrix} C \\ C_f(\gamma_k) \end{pmatrix} \quad (7)$$

for $\vartheta_k \in \Theta$. If such a condition holds (e.g. when the fault has changed the sensitivity of the sensor but the signal is not completely lost), the LPV virtual sensor reduces to a static reconfiguration block scheduled by γ_k. Cases where (7) is not satisfied (e.g. when one or more sensors are completely broken) should be described through values of the matrix C^* such that the following condition holds[1]:

$$C^* = P(\gamma_k)C_f(\gamma_k) \quad (8)$$

In these cases $C_\Delta(\gamma_k) \neq 0$ and $y_c(k)$ depends on the virtual sensor state $x_v(k)$ that is calculated as:

$$x_v(k+1) = A(\vartheta_k)x_v(k) + Bu(k) + L_v(\vartheta_k)\left(P(\gamma_k)(y_f(k) - f_y(k)) - C^* x_v(k)\right) \quad (9)$$

where $L_v(\vartheta_k) \in \Re^{n_x \times n_y}$ is the gain of the LPV virtual sensor.

When the sensor fault appears, the LPV virtual sensor reconstructs the system output vector $y_c(k)$ from the faulty sensor output $y_f(k)$. The faulty plant and the LPV virtual sensor are called the reconfigured LPV plant which is connected to the nominal LPV controller. If the reconfigured LPV plant behaves like the nominal plant, the loop consisting of the reconfigured plant and the LPV controller behaves like the nominal closed-loop system.

C. Fault Estimation

The LPV system (1)-(2) takes into account the multiplicative sensor faults in the matrix $C_f(\gamma_k)$ and the additive sensor faults in the term $f_y(k)$ such that for the i^{th} sensor:

$$y_{f,i}(k) = \gamma_i(k)\sum_{l=1}^{n_y} c_{i,l}x_l(k) + f_{y,i}(k) \quad (10)$$

where $y_{f,i}$ is the output of the i^{th} sensor when a fault occurs and $c_{i,l}$ is l^{th} element of the i^{th} row of the faultless matrix C. In this paper, to estimate the multiplicative and additive sensor faults, a parameter estimation approach is used. Using (10) and taking into account the last N samples obtained from the output sensors leads to:

$$Y_{f,i}(k) = \Psi_i^T(k)f_i \quad (11)$$

where:

$$Y_{f,i}(k) = \begin{bmatrix} y_{f,i}(k-N+1) & y_{f,i}(k-N+2) & \cdots & y_{f,i}(k) \end{bmatrix}^T \quad (12)$$

$$\Psi_i(k) = \begin{bmatrix} c_{i,1}x_1(k-N+1) + c_{i,2}x_2(k-N+1) + \cdots & +c_{i,n_x}x_{n_x}(k-N+1) & 1 \\ c_{i,1}x_1(k-N+2) + c_{i,2}x_2(k-N+2) + \cdots & +c_{i,n_x}x_{n_x}(k-N+2) & 1 \\ \vdots & \ddots & \vdots \\ c_{i,1}x_1(k) + c_{i,2}x_2(k) + \cdots & +c_{i,n_x}x_{n_x}(k) & 1 \end{bmatrix}^T \quad (13)$$

$$f_i = \begin{bmatrix} \gamma_i & f_{y,i} \end{bmatrix}^T \quad (14)$$

Thus, the fault estimation $\hat{f}_i(k)$ is given by:

$$\hat{f}_i(k) = \left(\Psi_i^T(k)\Psi_i(k)\right)^{-1}\Psi_i^T(k)Y_{f,i}(k) \quad (15)$$

It is assumed that $x(k)$ is not available and the estimated state $\hat{x}(k)$ should be used instead. Consequently, an *LPV state observer* is used to provide such state estimation:

$$\hat{x}(k+1) = A(\vartheta_k)\hat{x}(k) + Bu(k) + L(\vartheta_k)\left(y_c(k) - C\hat{x}(k)\right) \quad (16)$$

where $\hat{x}(k) \in \Re^{n_x}$ is the estimated state. The matrix $L(\vartheta_k) \in \Re^{n_x \times n_y}$ is the gain of the LPV state observer.

[1]Notice that the matrix C^* does not depend on γ_k because the matrix $P(\gamma_k)$ eliminates the effects of partial faults.

III. FTC Strategy using Virtual Sensor

A. LPV Controller

The LPV system (1)-(2) is controlled by a state feedback control with tracking reference input as proposed in [9]. The feedback control law is based on the classical state feedback:

$$u(k) = K(\vartheta_k)x(k) \tag{17}$$

while a nonlinear function N_{ux} is added to the state feedback control law (17). The basic idea in determining the function N_{ux} is that it should transform the reference input $r(k)$, expressed as either a desired value or trajectory of the system outputs, to a state reference $x_r(k)$ and a feedforward control $u_r(k)$ that correspond to an equilibrium point for this $r(k)$. Thus, the control law can be expressed as follows:

$$u(k) = u_r(k) + K(\vartheta_k)(x(k) - x_r(k)) \tag{18}$$

As in the case of the fault estimator, it is assumed that $x(k)$ is not available and the estimated state $\hat{x}(k)$ provided by the LPV state observer (16) should be used.

B. Reconfiguration Analysis

To analyze the reconfigured system, the *reconfiguration model* is considered. This augmented model includes the faulty plant (1), the LPV virtual sensor (9) and the state LPV observer (16) as follows[2]:

$$\begin{bmatrix} x_f(k+1) \\ x_v(k+1) \\ \hat{x}(k+1) \end{bmatrix} = \begin{bmatrix} A & 0 & 0 \\ L_vPC_f & A - L_vC^* & 0 \\ LPC_f & LC - LPC_f & A - LC \end{bmatrix} \begin{bmatrix} x_f(k) \\ x_v(k) \\ \hat{x}(k) \end{bmatrix} + \begin{bmatrix} B \\ B \\ B \end{bmatrix} u(k) \tag{19}$$

A state transformation is performed in order to introduce the virtual sensor error $x_\Delta(k) = x_v(k) - x_f(k)$ and the observation error $x_\delta(k) = \hat{x}(k) - x_v(k)$. By introducing the control law (18) and taking into account (8), the model of the reconfigured closed-loop behavior of the system (19) can be reshaped as:

$$\begin{bmatrix} x_f(k+1) \\ x_\Delta(k+1) \\ x_\delta(k+1) \end{bmatrix} = \begin{bmatrix} A - BK & -BK & -BK \\ 0 & A - L_vC^* & 0 \\ 0 & L_vC^* - LPC_f & A - LC \end{bmatrix} \begin{bmatrix} x_f(k) \\ x_\Delta(k) \\ x_\delta(k) \end{bmatrix} + \begin{bmatrix} -BK \\ 0 \\ 0 \end{bmatrix} x_r(k) + \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix} u_r(k) \tag{20}$$

Notice that looking at this model, the *separation principle* can be applied. The matrix $K(\vartheta_k)$ influences the behavior of the process state $x_f(k)$ through the submatrix $A(\vartheta_k) - BK(\vartheta_k)$ (nominal LPV controller). $L_v(\vartheta_k)$ affects the behavior of the virtual sensor error $x_\Delta(k)$ through the submatrix $A(\vartheta_k) - L_v(\vartheta_k)C^*$ (LPV virtual sensor). Finally, the observer error $x_\delta(k)$ is affected by the matrix $L(\vartheta_k)$ through the submatrix $A(\vartheta_k) - L(\vartheta_k)C$ (LPV state observer). The set σ of eigenvalues of the closed-loop system (20) consists of the set of eigenvalues of the nominal closed-loop system plus the LPV virtual sensor and the LPV state observer:

$$\sigma = \sigma\{A - BK\} \cup \sigma\{A - L_vC^*\} \cup \sigma\{A - LC\} \tag{21}$$

Thus, the nominal LPV controller, the LPV virtual sensor and the LPV state observer can be designed independently.

[2]To simplify the equations, the following expressions consider that $A = A(\vartheta_k)$, $L_v = L_v(\vartheta_k)$ and so on.

IV. Polytopic Approximation of a LPV System

A. Polytopic Approximation

In this paper, the kind of LPV systems considered are those whose time-varying parameter vector ϑ_k varies within a polytope Θ. Each polytope vertex corresponds to a particular value of scheduling variable ϑ_k. In other words,

$$A(\vartheta_k) \in Co\left(A_j, j = 1, \ldots, N\right) := \sum_{j=1}^{N} \alpha_k^j(\vartheta_k)A_j \tag{22}$$

with $\alpha_k^j(\vartheta_k) \geq 0$ and $\sum_{j=1}^{N} \alpha_k^j(\vartheta_k) = 1$. Because of this property, this type of LPV systems is referred as polytopic [10]. A common approach to design a controller and an observer for an LPV system is to approximate it by a polytopic LPV system [10]. The simplest polytopic approximation relies on bounding each LPV parameter by an interval (*bounding box* approach). Such a box can be found by considering every possible combination of minima and maxima of the elements of ϑ_k over the allowed range of variation of the measured system variables p_k.

B. Polytopic LPV Equations

Using the previous approximation, state equations of the system (1), virtual sensor (9), observer (16) and the control law (18) can be expressed in a polytopic way as follows:

$$x(k+1) = \sum_{j=1}^{N} \alpha_k^j(\vartheta_k)A_jx(k) + Bu(k) \tag{23}$$

$$x_v(k+1) = \sum_{j=1}^{N} \alpha_k^j(\vartheta_k)\left[A_jx_v(k) + Bu(k) + L_{v,j}\left(P(\gamma_k)C_f(\gamma_k)x_f(k) - C^*x_v(k)\right)\right] \tag{24}$$

$$\hat{x}(k+1) = \sum_{j=1}^{N} \alpha_k^j(\vartheta_k)\left[A_j\hat{x}(k) + Bu(k) + L_j\left(y_c(k) - C\hat{x}(k)\right)\right] \tag{25}$$

$$u(k) = u_r(k) + \sum_{j=1}^{N} \alpha_k^j(\vartheta_k)K_j\left(\hat{x}(k) - x_r(k)\right) \tag{26}$$

Here $A_j \in \Re^{n_x \times n_x}$ are time-invariant state matrices defined for the j^{th} model, $L_{v,j} \in \Re^{n_x \times n_y}$ are the gains of the LPV virtual sensor for each j^{th} model, Analogously, $L_j \in \Re^{n_x \times n_y}$ and $K_j \in \Re^{n_u \times n_x}$ are the state observer gains and the controller gains for each j^{th} model. The polytopic equations are scheduled through functions designed as follows: $\alpha_k^j(\vartheta_k), \forall j \in [1, \ldots, N]$ that lie in a convex set:

$$\Omega = \left\{\alpha_k^j(\vartheta_k) \in \Re^N, \alpha_k(\vartheta_k) = \left[\alpha_k^1(\vartheta_k), \ldots, \alpha_k^N(\vartheta_k)\right]^T, \alpha_k^j(\vartheta_k) \geq 0, \forall j, \sum_{j=1}^{N} \alpha_k^j(\vartheta_k) = 1\right\}. \tag{27}$$

There are several ways for implementing (22) depending on how $\alpha_k^j(\vartheta_k)$ functions are defined [11]. Here, the function $\alpha_k^j(\vartheta_k)$ is calculated via barycentric combination of vertexes as suggested by [10].

V. LPV Virtual Sensor design using LMI Pole Placement

A. LMI Regions

An LMI approach for the design by pole placement constraints is described in [6]. The main motivation for seeking pole clustering in specific regions of the complex plane is that, by constraining λ to lie in a prescribed region, stability can be guaranteed and a satisfactory transient response can be ensured. A subset $\mathcal{D}$ of the complex plane is called an LMI region if there exist a symmetric matrix $\rho = [\rho_{kl}] \in R^{m\times m}$ and a matrix $\beta = [\beta_{kl}] \in R^{m\times m}$ such that

$$\mathcal{D} = \{z \in C : f_{\mathcal{D}}(z) < 0\} \tag{28}$$

where the characteristic function $f_{\mathcal{D}}(z)$ is given by:

$$f_{\mathcal{D}}(z) = [\rho_{kl} + \beta_{kl} z + \beta_{kl}\bar{z}]_{l\leq k,\ l\leq m} \tag{29}$$

$f_{\mathcal{D}}(z)$ is valued in the space of $m \times m$ Hermitian matrices.

Using Gutman's theorem for LMI regions [12], pole location in a given LMI region can be characterized in terms of the $m \times m$ block matrix

$$\begin{aligned} M_D(A,X) &:= \rho \otimes X + \beta \otimes (AX) + \beta^T \otimes (AX)^T \\ &= \left[\rho_{kl} X + \beta_{kl} AX + \beta_{lk} XA^T\right]_{1\leqslant k,l\leqslant m} \end{aligned} \tag{30}$$

In [6] a polytopic state-space model as (23) and a state feedback control law as (26) is considered. The problem to be solved consists in finding a set of state-feedback gains K_j that places the closed-loop poles in some LMI stability region $\mathcal{D}$ with a characteristic function (29). The pole-placement constraint is satisfied if there exist a set of state-feedback gains K_j and a single Lyapunov matrix $X > 0$ such that:

$$\left[\rho_{kl} X + \beta_{kl}\left(A_j - BK_j\right)X + \beta_{lk} X\left(A_j - BK_j\right)^T\right]_{1\leqslant k,l\leqslant m} < 0 \quad \text{for all } i = 1,\dots,N \tag{31}$$

Using an auxiliary variable $W_j := K_j X$ the matrix inequalities (31) become the LMIs:

$$\left[\rho_{kl} X + \beta_{kl}\left(A_j X - BW_j\right) + \beta_{lk}\left(A_j X - BW_j\right)^T\right]_{1\leqslant k,l\leqslant m} < 0 \quad \text{for all } j = 1,\dots,N \tag{32}$$

that can be solved through convex optimization.

B. LPV Virtual Sensor design using LMI Pole Placement

The main objective of this section is to summarize the design procedure of the *LPV virtual sensor* (9) for the faulty LPV system (1)-(2). The design of LPV virtual sensor implies selecting:

- matrices K_j of (26) in order to guarantee closed-loop stability of the original system assuming that the pair $(A(\vartheta_k), B)$ is stabilizable for all $\vartheta_k \in \Theta$,
- matrices $L_{v,j}$ (see (24)) in order to correct the LPV virtual sensor error of the reconfigured system assuming that the pair $(A(\vartheta_k), C^*)$ is detectable for all $\vartheta_k \in \Theta$. This problem must be solved for each possible C^*,
- matrices L_j of (25) in order to correct the LPV state observer error of the reconfigured system assuming that the pair $(A(\vartheta_k), C)$ is detectable for all $\vartheta_k \in \Theta$,

Under these assumptions and taking advantage of the separation principle, it is possible to design the matrices K_j, $L_{v,j}$ and L_j using LMI techniques.

The *polytopic LPV controller* is designed with LMIs (31) assuming that eigenvalues are placed in an LMI region. The LMIs (31) are particularized for an LMI region consisting of the intersection between the unit circle and a vertical strip defined by the extreme values S_{min} and S_{max}:

$$\begin{cases} diag(Q_1, Q_2, Q_3) < 0 \\ X > 0 \end{cases} \tag{33}$$

$$Q_1 = \begin{pmatrix} -X & A_j X - BW_j \\ XA_j^T - W_j^T B^T & -X \end{pmatrix} \tag{34}$$

$$Q_2 = S_{min} X - \frac{1}{2}\left(A_j X + XA_j^T - BW_j - W_j^T B^T\right) \tag{35}$$

$$Q_3 = -S_{max} X + \frac{1}{2}\left(A_j X + XA_j^T - BW_j - W_j^T B^T\right) \tag{36}$$

Once these LMIs are solved, the gains of the polytopic LPV controller (26) can be determined by $K_j = W_j X^{-1}$.

Note that the solutions of the LMIs (33) are computed off-line and the evaluation of (26) just requires the computation of $\alpha_k^j(p_k)$ that is calculated via barycentric combination of vertexes as suggested by [10].

Similarly, the *LPV virtual sensor* (24) is designed with LMIs assuming that eigenvalues are placed in an LMI region. The design implies solving the LMIs (33) substituting A_j by A_j^T and B by C^*. Then, the set of gains $L_{v,j}$ can be calculated as $L_{v,j} = W_j X^{-1}$.

Finally, the design of the *LPV state observer* (25) implies solving the LMIs (33) substituting A_j by A_j^T and B by C. Then, the set of gains L_j can be calculated as $L_j = W_j X^{-1}$.

VI. Application example

A. Description of Twin-Rotor MIMO System

The TRMS is a laboratory setup developed by Feedback Instruments Limited for control experiments. The system is perceived as a challenging 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 the tail, both 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 and it determines a stable equilibrium position. The beam can rotate freely both in the horizontal and vertical planes.

The system input vector is $u = [u_h,\ u_v]^T$ where u_h/v is the input voltage of the tail/main motor. The system states vector is $x = [\omega_h,\ \Omega_h,\ \theta_h,\ \omega_v,\ \Omega_v,\ \theta_v]^T$ where $\omega_{h/v}$ 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 yaw/pitch angle of beam.

B. The TRMS LPV model

The mathematical model of the TRMS is given by a set of non-linear differential equations that can be found in [13]. This model has been validated with the equipment of the TRMS and, following the approach described in [14], it has been reshaped and brought to an LPV representation in [15]:

$$\begin{bmatrix} \omega_h(k+1) \\ \Omega_h(k+1) \\ \theta_h(k+1) \\ \omega_v(k+1) \\ \Omega_v(k+1) \\ \theta_v(k+1) \end{bmatrix} = A(\vartheta_k) \begin{bmatrix} \omega_h(k) \\ \Omega_h(k) \\ \theta_h(k) \\ \omega_v(k) \\ \Omega_v(k) \\ \theta_v(k) \end{bmatrix} + B(p_k) \begin{bmatrix} u_h(k) \\ u_v(k) \end{bmatrix} \quad (37)$$

where

$$A(\vartheta_k) = \begin{bmatrix} a_{11}(p_k) & 0 & 0 & 0 & 0 & 0 \\ a_{21}(p_k) & a_{22}(p_k) & a_{23}(p_k) & a_{24}(p_k) & a_{25}(p_k) & a_{26}(p_k) \\ 0 & T_s & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{44}(p_k) & 0 & 0 \\ a_{51}(p_k) & a_{52}(p_k) & 0 & a_{54}(p_k) & a_{55} & a_{56}(p_k) \\ 0 & 0 & 0 & 0 & T_s & 1 \end{bmatrix}$$

$$B(p_k) = \begin{bmatrix} b_{11} & 0 \\ 0 & b_{22}(p_k) \\ 0 & 0 \\ 0 & b_{44} \\ b_{51} & 0 \\ 0 & 0 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

where $\vartheta_k = [\vartheta_k^1, \ldots \vartheta_k^{12}]^T = [a_{11}(p_k), a_{21}(p_k), a_{22}(p_k), a_{23}(p_k), a_{24}(p_k), a_{25}(p_k), a_{26}(p_k), a_{44}(p_k), a_{51}(p_k), a_{52}(p_k), a_{54}(p_k), a_{56}(p_k)]^T$.

The model is scheduled with $p_k = [\hat{\omega}_h(k), \hat{\Omega}_h(k), \theta_h(k), \hat{\omega}_v(k), \theta_v(k)]^T$, using the available measurements outputs $\theta_h(k)$ and $\theta_v(k)$. The rest of states needed for control are estimated using a state observer. Since the scheduling vector p_k contains some state variables of the system, the model is said to be *quasi-LPV*[3].

To implement the control and virtual sensor approach for the TRMS it is necessary to obtain the polytopic representation (23) of the system (1). This is done by means of the *bounding box approach*.

The *LPV controller* is designed with LMIs (33) assuming that the eigenvalues are in LMI region with $S_{min} = 0.95$ and $S_{max} = 1$. Analogously, the *LPV state observer* is designed assuming that the eigenvalues are in LMI region with $S_{min} = 0.7$ and $S_{max} = 0.9$. Finally, the *LPV virtual sensors* are designed assuming that the eigenvalues are in LMI region with $S_{min} = 0$ and $S_{max} = 0.95$.[4]

The fault scenarios present the closed-loop behavior when the following change of set-point is introduced:

$$\theta_h^{ref} = \begin{cases} 1, & t < 50s \\ 1.4, & t \geq 50s \end{cases}, \theta_v^{ref} = \begin{cases} 0.1, & t < 50s \\ 0.2, & t \geq 50s \end{cases} \quad (38)$$

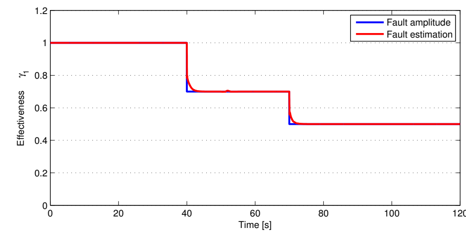
[3]Notice that b_{22} in (37) depends on p_k. However, it has been seen that its variations are small enough to let it be approximated by its mean value, leading the state equation to the form (1).

[4]In this application, the rank condition (7) is not satisfied in case of total loss of one sensor. Hence, 4 virtual sensors have to be designed, one for each sensor loss. In this case, matrices C^* are obtained from the matrix C by eliminating the row corresponding to the lost sensor.

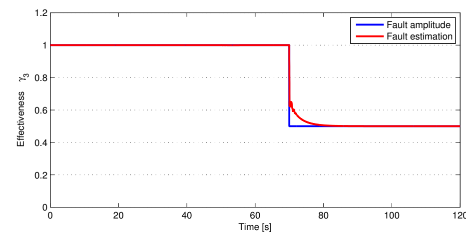
C. Fault scenarios

1) Fault scenario 1: The loss of sensor effectiveness in fault scenario 1 is defined as:

$$[\gamma_1, \gamma_2, \gamma_3, \gamma_4] = \begin{cases} [1,1,1,1], & \text{for } t < 40s \\ [0.7,1,1,1], & \text{for } 40s \leq t < 70s \\ [0.5,1,0.5,1], & \text{for } t \geq 70s \end{cases}. \quad (39)$$



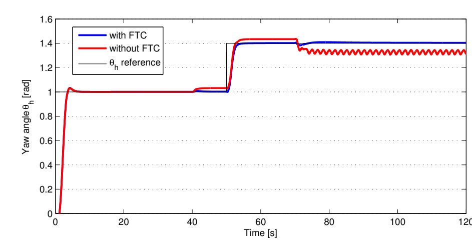
(a) Effectiveness of the tail rotor sensor ω_h



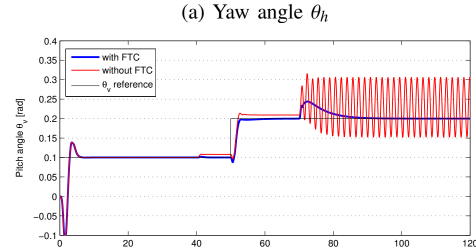
(b) Effectiveness of the main rotor sensor ω_v

Fig. 1: Sensors effectiveness in fault scenario 1

Fig. 1(a) and Fig. 1(b) present the effectiveness of the tail rotor sensor ω_h and the main rotor sensor ω_v in the first fault scenario. Both the real and the estimated values are shown, demonstrating that the faults are correctly estimated.



(a) Yaw angle θ_h



(b) Pitch angle θ_v

Fig. 2: TRMS angles in fault scenario 1

Fig. 2(a) and Fig. 2(b) present the yaw angle θ_h and the pitch angle θ_v, respectively. In both cases, the angles have perturbations at time $t = 40s$ and $t = 70s$ due to faults occurrences. However the overall loop is able to adapt in such a way that the fault effect is asymptotically masked .

2) Fault scenario 2: The loss of sensor effectiveness in fault scenario 2 is defined as:

$$[\gamma_1,\gamma_2,\gamma_3,\gamma_4] = \begin{cases} [1,1,1,1], & \text{for } t < 40s \\ [1,1,1,0.5], & \text{for } 40s \le t < 70s \\ [1,1,1,0], & \text{for } t \ge 70s \end{cases} . \quad (40)$$

Fig. 3(a) shows that the effectiveness in the pitch angle sensor θ_v is correctly estimated.

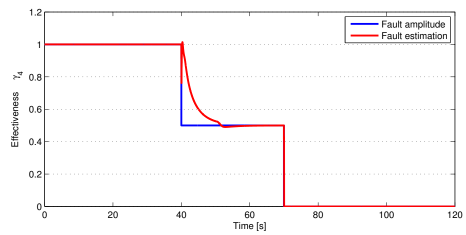
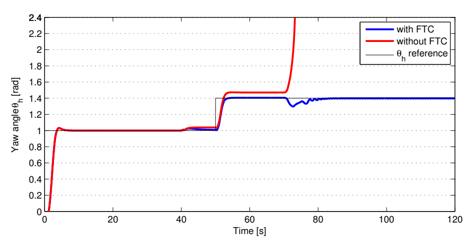
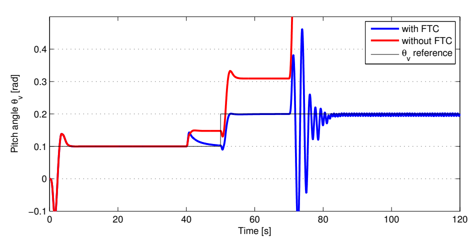


Fig. 3: Sensors effectiveness of the pitch angle sensor in fault scenario 2

Fig. 4(a) and Fig. 4(b) present the yaw angle θ_h and the pitch angle θ_v and their reference, respectively. In this case the fault occurrence provokes a loss of stability, while the proposed FTC strategy avoids such a loss and, moreover, the set-points are correctly followed.



(a) Yaw angle θ_h



(b) Pitch angle θ_v

Fig. 4: TRMS angles in fault scenario 2

VII. Conclusions

This paper has proposed an FTC strategy using an *LPV virtual sensor* for non-linear systems that can be approximated by an LPV model. This FTC method adapts the faulty plant to the nominal LPV controller instead of adapting the LPV controller to the faulty plant. In this way, the faulty plant together with the LPV virtual sensor block allows the LPV controller to see the same plant as before the fault. It has been demonstrated that an *LPV controller* can stabilize the faulty plant without having to redesign it at fault time.

The suggested approach implements an *FTC LPV virtual sensor* on-line by means of varying parameters that can change with the operating point or/and a fault. In particular case, the output sensor fault has been presented. Additionally the fault estimation is evaluated by the use of a block/batch least square approach where multiplicative and additive faults are present.

The implementation of the LPV controller requires a *state LPV observer*, that estimates the states with the information provided by the LPV virtual sensor. The LPV virtual sensor and observer are designed using polytopic LPV techniques and pole placement in LMI regions. This formulation allows a more versatile, straightforward and systematic characterization of the specifications.

The potential and performance of the approach have been demonstrated in an illustrative application to a two degree of freedom helicopter simulator.

References

[1] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*. Springer-Verlag Berlin Heidelberg, 2006.

[2] Y. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," *Annual Reviews in Control*, vol. 32, no. 2, pp. 229–252, December 2008.

[3] R. Hallouzi, V. Verdult, R. Babuska, and M. Verhaegen, "Fault Detection and Identification of Actuator Faults using Linear Parameter Varying Models," *16th IFAC Triennial World Congress, Prague, Czech Republic.*, pp. 119–124, 2005.

[4] L. R. D. Henrion, J. Bernussou, and F. Vary, "LPV Modeling of a Turbofan Engine," *In Preprints of the 16th World Congress of the International Federation of Automatic Control*, 2005.

[5] M. Rodrigues, D. Theilliol, S. Aberkane, and D. Sauter, "Fault Tolerant Control Design for Polytopic LPV Systems," *International Journal of Applied Mathematics and Computer Science*, vol. 17, no. 1, pp. 27–37, 2007.

[6] M. Chilali and P. Gahinet, "H_∞ Design with Pole Placement Constraints: An LMI Approach," *IEEE Transactions on Automatic Control*, vol. 41, no. 3, pp. 358–367, 1996.

[7] J. Jiang and Y. Zhang, "A Revisit to Block and Recursive Least Squares for Parameter Estimation," *International Journal of Computers and Electrical Engineering*, vol. 30, no. 5, pp. 403–416, 2004.

[8] ——, "A Novel Variable-Length Sliding Window Blockwise Least-Squares Algorithm for On-Line Estimation of Time-Varying Parameters," *International Journal of Adaptive Control and Signal Processing*, vol. 18, no. 6, pp. 505–521, 2004.

[9] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Addison Wesley Longman, 1997.

[10] P. Apkarian, P. Gahinet, and G. Becker, "Self-scheduled H_∞ Control of Linear Parameter-Varying Systems: A Design Example," *Automatica*, vol. 31, no. 9, pp. 1251 – 1261, 1995.

[11] R. Murray-Smith and T. A. Johansen, *Multiple Model Approaches to Modelling and Control*. Taylor and Francis, 1997.

[12] E. I. J. S. Gutman, "A general theory for matrix root clustering in subregions of the complex plane," *IEEE Transactions on Automatic Control*, vol. AC-26, pp. 853–863, 1981.

[13] A. Rahideh and M. H. Shaheed, "Mathematical Dynamic Modelling of a Twin-Rotor Multiple Input-Multiple Output System," *IMechE Journal of Systems and Control Engineering*, vol. 221, no. 1, pp. 89–101, 2007.

[14] A. Kwiatkowski, M. T. Boll, and H. Werner, "Automated Generation and Assessment of Affine LPV Models," *Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, CA, USA*, pp. 6690–6695, 2006.

[15] F. Nejjari, D. Rotondo, V. Puig, and M. Innocenti, "LPV Modelling and Control of a Twin Rotor MIMO System ," in *19th IEEE Mediterranean Conference on Control and Automation*, 2011.