

# A Fault-Hiding Approach for the Switching Quasi-LPV Fault Tolerant Control of a Four Wheeled Omnidirectional Mobile Robot

Damiano Rotondo, *Student Member, IEEE*, Vicenç Puig, Fatiha Nejari, and Juli Romera

**Abstract**—This paper proposes a reference model approach for the trajectory tracking of a four wheeled omnidirectional mobile robot. In particular, the error model is brought to a quasi-Linear Parameter Varying (LPV) form suitable for designing an error-feedback controller. It is shown that, if polytopic techniques are used to reduce the number of constraints from infinite to finite, a solution within the standard LPV framework could not exist due to a singularity that appears in the possible values of the input matrix. Adding a switching component to the controller allows to solve this problem. Moreover, a switching LPV virtual actuator is added to the control loop in order to obtain fault tolerance within the fault-hiding paradigm, keeping the stability and some desired performances under the effect of actuator faults without the need of retuning the nominal controller. The effectiveness of the proposed approach is shown and proved through simulation and experimental results.

**Index Terms**—Mobile robots, linear parameter varying (LPV) systems, fault tolerant control (FTC), virtual actuators, model reference, switched systems, tracking, identification

## I. INTRODUCTION

OMNIDIRECTIONAL mobile robots are gaining popularity due to their enhanced mobility with respect to traditional robots [1, 2, 3]. In order to achieve a good control and enhance the tracking performance of omnidirectional mobile robots, precise dynamical modeling is needed [4]. Hence, in recent years, there have been some efforts in developing a dynamic model [5], [6] and techniques for estimating the unknown model parameters [7], [8] for this type of robots.

Different techniques have been applied to solve the control problem for omnidirectional mobile robots, e.g. [9], [10]. However, when the mobile robots are intended to be used in hazardous environments or for long-time operations, it is needed to increase their robustness against possible failures [11]. In particular, four wheeled omnidirectional mobile robots have the relevant characteristic that they can still operate with three wheels in case some malfunctioning in one wheel

has been detected [12]. This makes them good setups for testing techniques that provide fault tolerance against actuator faults. The objective of a Fault Tolerant Control (FTC) system [13, 14, 15, 16] is to maintain current performances close to desirable ones and preserve stability conditions in the presence of faults. The existing design techniques mainly include the passive and the active approach [17], [18]. The *passive FTC techniques* take into account the fault as a system perturbation, such that the control law is designed to have inherent fault tolerance capabilities. On the other hand, the *active FTC techniques* try to satisfy the control objectives with minimum performance degradation either by selecting a precalculated control law or by synthesizing online a new control strategy. Examples of successful FTC strategies can be multiple-model-based [19], learning-based [20] and adaptive backstepping-based [21].

In recent years, the *fault-hiding* paradigm has been proposed as an active strategy to obtain fault tolerance [22]. In this paradigm, the faulty plant is reconfigured instead of the controller/observer by inserting a reconfiguration block when a fault occurs. The reconfiguration block is chosen so as to hide the fault from the controller point of view, allowing it to see the same plant as before the fault. In case of actuator faults, the reconfiguration block is named *virtual actuator*. This active FTC strategy has been extended successfully to many classes of systems, e.g. Linear Particular Varying (LPV) [23], Takagi-Sugeno [24] and piecewise affine [25]. An input-output formulation has been proposed recently in [26].

In the last decades, the Linear Parameter Varying (LPV) paradigm has become a standard formalism in systems and control, for analysis, controller synthesis and system identification [27, 28, 29]. This class of systems is important because, by embedding the system nonlinearities in the varying parameters, gain scheduling control of nonlinear systems can be performed using an extension of linear techniques. When there are both continuous valued and discrete valued varying parameters, the resulting system is referred to as *switching LPV* [30].

In this paper, a solution for the trajectory tracking problem in the inertial fixed coordinate system is proposed for a four wheeled omnidirectional robot. This solution relies on the use of a reference model that describes the desired trajectory, an idea that is well-established in the LTI framework [31], and has recently been extended to cope with the control of LPV systems [32]. The resulting nonlinear error model is brought to a quasi-LPV form suitable for designing an LPV controller solving a system of Linear Matrix Inequalities (LMIs), a

This work has been funded by the Spanish MINECO through the project CYCYT SHERECS (ref. DPI2011-26243), by the European Commission through contract i-Sense (ref. FP7-ICT-2009-6-270428), by AGAUR through the contracts FI-DGR 2013 (ref. 2013FIB00218) and FI-DGR 2014 (ref. 2014FLB1 00172) and by the DGR of Generalitat de Catalunya (SAC group Ref. 2014/SGR/374).

The authors are with the Research Center for Supervision, Safety and Automatic Control (CS2AC), Universitat Politècnica de Catalunya (UPC), Edifici TR11, Rambla Sant Nebridi 10, 08222 Terrassa, Spain (e-mails: name.surname@upc.edu). Vicenç Puig is also with the Institute of Robotics and Industrial Informatics (IRI), UPC-CSIC, Carrer de Llorens i Artigas, 4-6, 08028 Barcelona, Spain.

problem for which efficient solvers are available [33], [34]. In particular, it is shown that, if polytopic techniques are used to reduce the number of LMI constraints from infinite to finite, there could not exist a solution within the standard LPV framework. Hence, the switching LPV framework is considered. The values of the parameters in the model must be known in order to compute the controller gains, and this leads to the necessity of performing system identification. The procedure used to this end resembles the one used in [35]. Finally, the virtual actuators technique is extended to switching LPV systems subject to actuator faults in such a way that, once the controller has been designed, fault tolerance can be added to the control loop without need of redesigning the obtained controller. The effectiveness of the proposed approach is shown through experimental results obtained with a real testbed.

The paper is structured as follows: Section II describes the four wheeled omnidirectional robot. Section III introduces the dynamic model of the four wheeled omnidirectional robot and the identification approach used to estimate the unknown parameters. The error-feedback controller design using switching LPV techniques is presented in Section IV. Section V presents the proposed FTC strategy using virtual actuators in the context of switching LPV systems. Some further details about the application of the theory presented in the paper to the four wheeled omnidirectional robot are presented in Section VI. The simulation and experimental results are shown and commented in Section VII. Finally, the main conclusions are stated in Section VIII.

## II. SYSTEM DESCRIPTION

The robotic platform case study used in this paper is a four wheeled omnidirectional robot developed by ST Microelectronics under the i-Sense European project<sup>1</sup> (see Fig. 1). This platform is a holonomic robot presenting a great manoeuvrability and effectiveness. The omnidirectional feature is reached thanks to the characteristics of the wheels which roll forward like normal wheels but can slide sideways at the same time, allowing almost independent tangential, normal and angular velocities (holonomic property). The robot is composed of several parts: a netbook PC for control, electronic boards (1 STM32F4Discovery board, 2 motor driver boards and 1 DC-DC converter), sensors (4 encoders: one for each wheel, 8 ultra-sonic rangefinders and the iNemo Sensor Platform that includes a gyroscope, accelerometers and a magnetometer), 4 DC motors, 4 omnidirectional wheels plus the mechanical items.

The STM32F4 microcontroller, a Cortex-M4 with CPU working at 168MHz (210 DMIPS) with 192KB of RAM, is in charge of the low-level control of the robot. It takes care of all the low-level tasks such as ultrasound sensors acquisition and motor control. Moreover, it offers a high-level API to control the robot from the network via an USB port.

The high-level robot control algorithm is running in Matlab under Linux in the netbook fixed on top of the robot. The communications between the low-level hardware of the

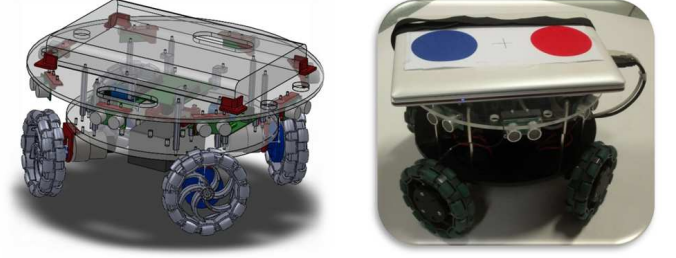


Fig. 1. i-Sense robotic platform

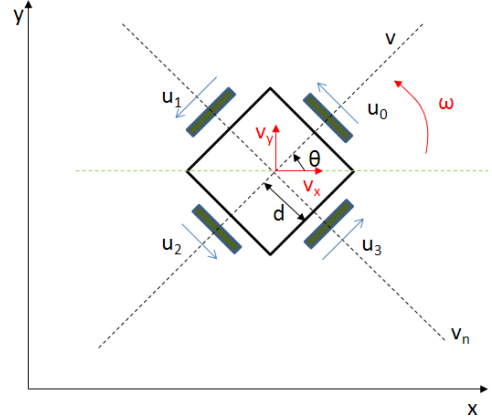


Fig. 2. Four wheeled omnidirectional mobile robot.

robot netbook is through the USB port connected to the STM32F4Discovery board. The control algorithm reads the sensor data (encoders), computes the control actions as a PWM duty cycle, and sends them back to the motors at 25Hz.

The i-Sense robotic platform has been released as an open source project on the SourceForge website<sup>2</sup>. The goal of this platform in the i-Sense project was to allow testing advanced FTC strategies by directly implementing them in MATLAB.

## III. FOUR WHEELED OMNIDIRECTIONAL MOBILE ROBOT MODELING AND IDENTIFICATION

### A. Nonlinear Dynamic Model

The dynamic model of the four wheeled omnidirectional robot (see Fig. 2) relates the wheel inputs and robot velocities with the corresponding accelerations, taking into account the traction, viscous friction and Coulomb friction forces. It is given by the following set of differential equations, obtained from the ones presented in [1] by considering the linear velocities on the static axis instead of the ones on the robot's axis:

$$\dot{x} = v_x \quad (1)$$

$$\begin{aligned} \dot{v}_x = & (A_{11}c_\theta^2 + A_{22}s_\theta^2)v_x + ((A_{11} - A_{22})s_\theta c_\theta - \omega)v_y \\ & + K_{11}c_\theta \text{sign}(v_x c_\theta + v_y s_\theta) - B_{21}s_\theta u_0 + B_{12}c_\theta u_1 \\ & - K_{22}s_\theta \text{sign}(-v_x s_\theta + v_y c_\theta) - B_{23}s_\theta u_2 + B_{14}c_\theta u_3 \end{aligned} \quad (2)$$

<sup>1</sup><http://www.i-sense.org/>

<sup>2</sup><https://sourceforge.net/projects/isenseroboticplatform/>

$$\dot{y} = v_y \quad (3)$$

$$\begin{aligned} \dot{v}_y = & ((A_{11} - A_{22})s_\theta c_\theta + \omega)v_x + (A_{11}s_\theta^2 + A_{22}c_\theta^2)v_y \\ & + K_{11}s_\theta \text{sign}(v_x c_\theta + v_y s_\theta) + B_{21}c_\theta u_0 + B_{12}s_\theta u_1 \\ & + K_{22}c_\theta \text{sign}(-v_x s_\theta + v_y c_\theta) + B_{23}c_\theta u_2 + B_{14}s_\theta u_3 \end{aligned} \quad (4)$$

$$\dot{\theta} = \omega \quad (5)$$

$$\dot{\omega} = A_{33}\omega + B_{31}u_0 + B_{32}u_1 + B_{33}u_2 + B_{34}u_3 + K_{33}\text{sign}(\omega) \quad (6)$$

where  $(x, y)$  is the robot position,  $\theta$  is the angle with respect to the defined front of robot ( $s_\theta \triangleq \sin \theta$  and  $c_\theta \triangleq \cos \theta$ ),  $v_x$ ,  $v_y$  and  $\omega$  are the corresponding linear/angular velocities, and  $u_0, u_1, u_2$  and  $u_3$  the motor voltage applied to the wheel 1, 2, 3 and 4, respectively. The coefficients  $A_{ii}$ ,  $B_{ij}$ ,  $K_{ii}$ ,  $i = 1, 2, 3$ ,  $j = 1, 2, 3, 4$ , are defined as follows:

$$\begin{aligned} A_{11} &= \frac{2K_t^2 l^2}{r^2 R M} - \frac{B_v}{M} & A_{22} &= \frac{2K_t^2 l^2}{r^2 R M} - \frac{B_{vn}}{M} & A_{33} &= -\frac{4d^2 K_t^2 l^2}{r^2 R J} - \frac{B_\omega}{J} \\ B_{12} = B_{23} &= -\frac{lK_t}{rRM} & B_{14} = B_{21} &= \frac{lK_t}{rRM} & B_{31} = B_{32} = B_{33} = B_{34} &= \frac{lK_t d}{rRJ} \\ K_{11} &= -\frac{C_v}{M} & K_{22} &= -\frac{C_{vn}}{M} & K_{33} &= -\frac{C_\omega}{J} \end{aligned}$$

where  $K_t$  is the motor torque constant,  $l$  the gearbox reduction,  $r$  the wheel radius,  $R$  the motor resistor,  $M$  the mass,  $d$  the distance between the wheels and the robot center,  $J$  the inertia moment,  $B_v$  the front viscous friction coefficient,  $B_{vn}$  the orthogonal viscous friction coefficient,  $B_\omega$  the angular viscous friction coefficient,  $C_v$  the front Coulomb friction coefficient,  $C_{vn}$  the orthogonal Coulomb friction coefficient and  $C_\omega$  the angular Coulomb friction coefficient.

The values of the parameters appearing in the equations (1)-(6) must be known in order to design a controller, and this leads to the necessity of performing system identification. At first, the unknown parameters of the angular subsystem (5)-(6) are identified. Later, those affecting the linear subsystem (1)-(4) are estimated. The spirit of the method is to compare, at each time step, the data obtained from the real setup with the data obtained by simulating part of the four wheeled omnidirectional robot continuous-time nonlinear model. This is included in an optimization problem to find the parameter values that give simulation results that better approximate the real system.

### B. Angular Subsystem Identification

The goal of this step is to identify the unknown parameters  $A_{33}$ ,  $B_3 \triangleq B_{31} = B_{32} = B_{33} = B_{34}$  and  $K_{33}$ . Due to the Coulomb friction force, represented by  $K_{33}$ , the angular subsystem will exhibit a dead-zone nonlinearity with respect to the sum of the inputs. In other words, applying an increasing sequence of input values  $u_0, u_1, u_2, u_3$ , the robot will not rotate until some critical value  $u_0^* + u_1^* + u_2^* + u_3^*$  is reached. This value, taking into account (6), leads to the following condition that can be used for further reducing the number of parameters to be identified through optimization:

$$-B_3(u_0^* + u_1^* + u_2^* + u_3^*) = K_{33} \quad (7)$$

Hence, the optimization procedure should identify some values for the unknown parameters  $A_{33}$ ,  $B_3$  in such a manner that the nonlinear model behavior resembles the real behavior of the angular subsystem. It is assumed to have at disposal  $N_\theta$  sets

of data  $\{u_0^i(k), u_1^i(k), u_2^i(k), u_3^i(k), \theta_i(k)\}$ , where  $i = 1, \dots, N_\theta$  and  $k = 1, \dots, K_\theta^i$  with  $K_\theta^i$  being the number of samples of the  $i$ th set of data.

The identification procedure finds the minimum of the following objective function over the unknown parameters:

$$\min_{A_{33}, B_3, K_{33}} J_\theta = \sum_{i=1}^{N_\theta} \sum_{k=1}^{K_\theta^i} (\theta_i(k) - \hat{\theta}_i(k))^2 \quad (8)$$

subject to (7), where  $\hat{\theta}_i(k)$  denotes the simulation provided by Eqs. (5)-(6).

However, as shown later in Section VII, by applying the proposed identification procedure to the available data sets, a dependence of the parameter  $A_{33}$  on  $\omega$  has been observed. This fact has not been considered by [1]. Hence, the nonlinear model used for control purposes is made up by (1)-(5) and by the following equation, obtained as a slight modification of (6):

$$\dot{\omega} = A_{33}(\omega)\omega + B_{31}u_0 + B_{32}u_1 + B_{33}u_2 + B_{34}u_3 + K_{33}\text{sign}(\omega) \quad (9)$$

### C. Linear Subsystem Identification

In this step, the unknown parameters  $A_{11}$ ,  $A_{22}$ ,  $B_l \triangleq B_{21} = -B_{12} = -B_{23} = B_{14}$ ,  $K_{11}$  and  $K_{22}$  should be identified. Due to the system symmetry, it is reasonable to assume also that  $A_{11} = A_{22} \triangleq A_l$  and  $K_{11} = K_{22} \triangleq K_l$  (this assumption, that is confirmed by the experimental data, is equivalent to assume that  $B_v = B_{vn}$  and  $C_v = C_{vn}$ ). A reasoning about the Coulomb friction force similar to the one made for the angular subsystem leads to the following condition:

$$K_l = -B_l(u_0^* - u_2^*) \quad (10)$$

Hence, the identification procedure should identify some values for the unknown parameters  $A_l$ ,  $B_l$  in such a way that the nonlinear model behavior resembles the real behavior of the linear subsystem. It is assumed to have at disposal  $N_l$  sets of data  $\{u_0^i(k), u_1^i(k), u_2^i(k), u_3^i(k), x_i(k), y_i(k)\}$ , where  $i = 1, \dots, N_l$  and  $k = 1, \dots, K_l^i$  with  $K_l^i$  being the number of samples of the  $i$ th set of data. The identification procedure finds the minimum of the following objective function over the unknown parameters:

$$\min_{A_l, B_l, K_l} J_l = \sum_{i=1}^{N_l} \sum_{k=1}^{K_l^i} [(x_i(k) - \hat{x}_i(k))^2 + (y_i(k) - \hat{y}_i(k))^2] \quad (11)$$

subject to (10), where  $\hat{x}_i(k)$  and  $\hat{y}_i(k)$  denote the simulation provided by Eqs. (1)-(4).

### D. Reference Model and Quasi-LPV Representation

Taking into account the fact that  $B_3 \triangleq B_{31} = B_{32} = B_{33} = B_{34}$ ,  $A_l \triangleq A_{11} = A_{22}$ ,  $B_l \triangleq B_{21} = -B_{12} = -B_{23} = B_{14}$  and  $K_l \triangleq K_{11} = K_{22}$ , let us introduce the following reference model, that will provide the state trajectory and the feedforward inputs:

$$\dot{x}_r = v_x^r \quad (12)$$

$$\dot{v}_x^r = A_l v_x^r - \omega v_y^r + B_l [c_\theta (u_3^r - u_1^r) - s_\theta (u_0^r - u_2^r)] + K_l [c_\theta \text{sign}(v_x c_\theta + v_y s_\theta) - s_\theta \text{sign}(-v_x s_\theta + v_y c_\theta)] \quad (13)$$

$$\dot{v}_r = v_y^r \quad (14)$$

$$\dot{v}_y^r = \omega v_x^r + A_l v_y^r + B_l [c_\theta (u_0^r - u_2^r) + s_\theta (u_3^r - u_1^r)] + K_l [s_\theta \text{sign}(v_x c_\theta + v_y s_\theta) + c_\theta \text{sign}(-v_x s_\theta + v_y c_\theta)] \quad (15)$$

$$\dot{\theta}_r = \omega_r \quad (16)$$

$$\dot{\omega}_r = A_{33}(\omega) \omega_r + B_3 (u_0^r + u_1^r + u_2^r + u_3^r) + K_{33} \text{sign}(\omega) \quad (17)$$

where  $(x_r, y_r)$  is the reference vehicle position,  $\theta_r$  is its angle,  $v_x^r, v_y^r$  and  $\omega_r$  are the corresponding linear/angular velocities, and  $u_0^r, u_1^r, u_2^r, u_3^r$  are the reference inputs (feedforward actions). Then, by defining the tracking errors  $e_1 \triangleq x_r - x$ ,  $e_2 \triangleq y_r - y$ ,  $e_3 \triangleq v_x^r - v_x$ ,  $e_4 \triangleq v_y^r - v_y$ ,  $e_5 \triangleq \theta_r - \theta$ ,  $e_6 \triangleq \omega_r - \omega$ , and the new inputs  $\Delta u_i \triangleq u_i^r - u_i$ ,  $i = 0, 1, 2, 3$ , and by using an Euler approximation with a sampling time  $T_s$  to allow a digital implementation of the controller, the error model for the four-wheeled omnidirectional mobile robot can be obtained from (1)-(5), (9) and (12)-(17).

By defining the vector of varying parameters as:

$$\vartheta_d(\theta(k), \omega(k)) = \begin{pmatrix} \vartheta_1^d(\omega(k)) \\ \vartheta_2^d(\omega(k)) \\ \vartheta_3^d(\theta(k)) \\ \vartheta_4^d(\theta(k)) \end{pmatrix} = \begin{pmatrix} \omega(k) T_s \\ 1 + A_{33}(\omega(k)) T_s \\ \sin(\theta(k)) T_s \\ \cos(\theta(k)) T_s \end{pmatrix}$$

the error model can be reshaped into the following quasi-LPV representation:

$$\begin{pmatrix} e_1(k+1) \\ e_2(k+1) \\ e_3(k+1) \\ e_4(k+1) \\ e_5(k+1) \\ e_6(k+1) \end{pmatrix} = \begin{pmatrix} 1 & T_s & 0 & 0 & 0 & 0 \\ 0 & A_l^d & 0 & -\vartheta_1^d & 0 & 0 \\ 0 & 0 & 1 & T_s & 0 & 0 \\ 0 & \vartheta_1^d & 0 & A_l^d & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 0 & 0 & \vartheta_2^d \end{pmatrix} \begin{pmatrix} e_1(k) \\ e_2(k) \\ e_3(k) \\ e_4(k) \\ e_5(k) \\ e_6(k) \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ -B_l \vartheta_3^d & -B_l \vartheta_4^d & B_l \vartheta_3^d & B_l \vartheta_4^d \\ 0 & 0 & 0 & 0 \\ B_l \vartheta_4^d & -B_l \vartheta_3^d & -B_l \vartheta_4^d & B_l \vartheta_3^d \\ 0 & 0 & 0 & 0 \\ B_3 T_s & B_3 T_s & B_3 T_s & B_3 T_s \end{pmatrix} \begin{pmatrix} \Delta u_0(k) \\ \Delta u_1(k) \\ \Delta u_2(k) \\ \Delta u_3(k) \end{pmatrix} \quad (18)$$

where  $A_l^d = 1 + A_l T_s$ .

#### IV. ERROR-FEEDBACK CONTROLLER DESIGN USING SWITCHING LPV TECHNIQUES

##### A. Controller Design for (quasi-)LPV Systems

Consider the following (quasi-)LPV error system:

$$e(k+1) = A(\vartheta(k)) e(k) + B(\vartheta(k)) \Delta u(k) \quad (19)$$

where  $e \in \mathbb{R}^{n_e}$  is the error vector,  $\Delta u \in \mathbb{R}^{n_u}$  is the input vector,  $A(\vartheta(k)), B(\vartheta(k))$  are varying matrices of appropriate dimensions and  $\vartheta \in \Theta \subset \mathbb{R}^{n_\vartheta}$  is the vector of varying parameters. The system is controlled through an error-feedback control law:

$$\Delta u(k) = K(\vartheta(k)) e(k) \quad (20)$$

and it is wished to solve the design problem of finding an error-feedback gain matrix  $K(\vartheta(k))$  such that the resulting closed-loop error system is stable with poles placed in some desired region of the complex plane.

In this paper, both stability and pole clustering are analyzed within the quadratic Lyapunov framework, where the specifications are assured by the use of a single quadratic Lyapunov function. Despite the introduction of conservativeness with respect to other existing approaches, where the Lyapunov function is allowed to be parameter varying, the quadratic approach has undeniable advantages in terms of computational complexity.

In particular, the (quasi-)LPV error system (19) with the error-feedback control law (20) is quadratically stable if and only if there exist  $X_s = X_s^T > 0$  and  $K(\vartheta(k))$  such that:

$$\begin{pmatrix} -X_s & (A(\vartheta) + B(\vartheta)K(\vartheta))X_s \\ X_s(A(\vartheta) + B(\vartheta)K(\vartheta))^T & -X_s \end{pmatrix} < 0 \quad (21)$$

$\forall \vartheta \in \Theta$ . On the other hand, pole clustering is based on the results obtained by [36], where subsets  $\mathcal{D}$  of the complex plane, referred to as *LMI regions*, are defined as:

$$\mathcal{D} = \{z \in \mathbb{C} : f_{\mathcal{D}}(z) < 0\} \quad (22)$$

where  $f_{\mathcal{D}}$  is the *characteristic function*, defined as:

$$f_{\mathcal{D}}(z) = \alpha + z\beta + \bar{z}\beta^T = [\alpha_{kl} + \beta_{kl}z + \beta_{lk}\bar{z}]_{1 \leq k, l \leq m} \quad (23)$$

with  $\alpha = \alpha^T \in \mathbb{R}^{m \times m}$  and  $\beta \in \mathbb{R}^{m \times m}$ . Hence, the (quasi-) LPV error system (19) with error-feedback control law (20) has poles<sup>3</sup> in  $\mathcal{D}$  if there exist  $X_{\mathcal{D}} = X_{\mathcal{D}}^T > 0$  and  $K(\vartheta(k))$  such that:

$$[\alpha_{kl}X_{\mathcal{D}} + \beta_{kl}(A(\vartheta) + B(\vartheta)K(\vartheta))X_{\mathcal{D}} + \beta_{lk}X_{\mathcal{D}}(A(\vartheta) + B(\vartheta)K(\vartheta))^T]_{1 \leq k, l \leq m} < 0 \quad (24)$$

$\forall \vartheta \in \Theta$ . The main difficulty with using (21) and (24) is that they impose an infinite number of constraints. In order to reduce this number to finite, a polytopic approximation of (19)-(20) is considered, as follows:

$$A(\vartheta(k)) = \sum_{i=1}^N \gamma_i(\vartheta(k)) A_i \quad \gamma_i(\vartheta) \geq 0, \quad \sum_{i=1}^N \gamma_i(\vartheta) = 1 \quad \forall \vartheta \in \Theta \quad (25)$$

$$B(\vartheta(k)) = \sum_{w=1}^W \delta_w(\vartheta(k)) B_w \quad \delta_w(\vartheta) \geq 0, \quad \sum_{w=1}^W \delta_w(\vartheta) = 1 \quad \forall \vartheta \in \Theta \quad (26)$$

$$K(\vartheta(k)) = \sum_{i=1}^N \gamma_i(\vartheta(k)) K_i \quad (27)$$

where each combination  $(A_i, B_w)$ ,  $i = 1, \dots, N$ ,  $w = 1, \dots, W$  is called *vertex system* and is controlled through the vertex controller  $K_i$ . Then, quadratic stability and pole clustering can be assessed through the following Linear Matrix Inequalities (LMIs), obtained from (21)-(24) using a common Lyapunov matrix  $X = X_s = X_{\mathcal{D}} > 0$  and the change of variables  $\Gamma_i \triangleq K_i X$ :

$$\begin{pmatrix} -X & A_i X + B_w \Gamma_i \\ X A_i^T + \Gamma_i^T B_w^T & -X \end{pmatrix} < 0 \quad (28)$$

<sup>3</sup>According to [37], and with a little abuse of language, the poles of an LPV system are defined as the set of all the poles of the LTI systems obtained by freezing  $\vartheta(k)$  to all its possible values  $\vartheta^* \in \Theta$ . It has been reported that the idea of poles, as introduced, has a connection with the dynamical behavior of the system, justifying, from the engineering point of view, the abuse of language.

$$\left[ \alpha_{kl}X + \beta_{kl}(A_iX + B_w\Gamma_i) + \beta_{lk}(A_iX + B_w\Gamma_i)^T \right]_{1 \leq k, l \leq m} < 0 \quad (29)$$

with  $i = 1, \dots, N$  and  $w = 1, \dots, W$ , that can be solved using available software, e.g. the YALMIP toolbox [33] with SeDuMi solver [34].

### B. The Switching (quasi-)LPV Controller Design

For some particular systems, due to the convexity of the polytopic approximation of  $A(\vartheta(k))$  and  $B(\vartheta(k))$ , some values that do not correspond to possible operating conditions, and for which the controllability is lost, could be considered. This fact causes the infeasibility of (28)-(29), that could be avoided by searching the solution to the design problem within the switching LPV framework, where the overall system behavior is given by an interaction between different LPV systems through discrete switching events, which can depend on states or time. Similarly, the overall controller is obtained from different LPV controllers that are switched when discrete switching events occur.

More specifically, it is assumed that the (quasi-)LPV error system (19)-(20) is modified including a switching part, as follows:

$$e(k+1) = A_\sigma(\vartheta(k))e(k) + B_\sigma(\vartheta(k))\Delta u(k) \quad (30)$$

$$\Delta u(k) = K_\sigma(\vartheta(k))e(k) \quad (31)$$

with:

$$A_\sigma(\vartheta(k)) = \begin{cases} \sum_{i=1}^{N_1} \gamma_i^{(1)}(\vartheta(k))A_i^{(1)}, & \gamma_i^{(1)}(\vartheta) \geq 0, \sum_{i=1}^{N_1} \gamma_i^{(1)}(\vartheta) = 1 \forall \vartheta \in \Theta_1 \\ \vdots \\ \sum_{i=1}^{N_\zeta} \gamma_i^{(\zeta)}(\vartheta(k))A_i^{(\zeta)}, & \gamma_i^{(\zeta)}(\vartheta) \geq 0, \sum_{i=1}^{N_\zeta} \gamma_i^{(\zeta)}(\vartheta) = 1 \forall \vartheta \in \Theta_\zeta \\ \vdots \\ \sum_{i=1}^{N_Z} \gamma_i^{(Z)}(\vartheta(k))A_i^{(Z)}, & \gamma_i^{(Z)}(\vartheta) \geq 0, \sum_{i=1}^{N_Z} \gamma_i^{(Z)}(\vartheta) = 1 \forall \vartheta \in \Theta_Z \end{cases} \quad (32)$$

$$B_\sigma(\vartheta(k)) = \begin{cases} \sum_{w=1}^{W_1} \delta_w^{(1)}(\vartheta(k))B_w^{(1)}, & \delta_w^{(1)}(\vartheta) \geq 0, \sum_{w=1}^{W_1} \delta_w^{(1)}(\vartheta) = 1 \forall \vartheta \in \Theta_1 \\ \vdots \\ \sum_{w=1}^{W_\zeta} \delta_w^{(\zeta)}(\vartheta(k))B_w^{(\zeta)}, & \delta_w^{(\zeta)}(\vartheta) \geq 0, \sum_{w=1}^{W_\zeta} \delta_w^{(\zeta)}(\vartheta) = 1 \forall \vartheta \in \Theta_\zeta \\ \vdots \\ \sum_{w=1}^{W_Z} \delta_w^{(Z)}(\vartheta(k))B_w^{(Z)}, & \delta_w^{(Z)}(\vartheta) \geq 0, \sum_{w=1}^{W_Z} \delta_w^{(Z)}(\vartheta) = 1 \forall \vartheta \in \Theta_Z \end{cases} \quad (33)$$

$$K_\sigma(\vartheta(k)) = \begin{cases} \sum_{i=1}^{N_1} \gamma_i^{(1)}(\vartheta(k))K_i^{(1)}, & \forall \vartheta \in \Theta_1 \\ \vdots \\ \sum_{i=1}^{N_\zeta} \gamma_i^{(\zeta)}(\vartheta(k))K_i^{(\zeta)}, & \forall \vartheta \in \Theta_\zeta \\ \vdots \\ \sum_{i=1}^{N_Z} \gamma_i^{(Z)}(\vartheta(k))K_i^{(Z)}, & \forall \vartheta \in \Theta_Z \end{cases} \quad (34)$$

where  $\Theta_1, \dots, \Theta_Z$  are subsets of the varying parameter space  $\Theta$ , such that  $\Theta = \Theta_1 \cup \dots \cup \Theta_\zeta \cup \dots \cup \Theta_Z$ . In each subset  $\Theta_\zeta$ ,  $\zeta = 1, \dots, Z$ , the system is described by a polytopic combination of vertex systems. The controller (31) with gain

(34) assures that the error system (30) with state and input matrix as in (32) and (33), respectively, is quadratically stable and has poles in  $\mathcal{D}$  if there exist  $X = X^T > 0$  and  $\Gamma_i^{(\zeta)} \triangleq K_i^{(\zeta)}X$ ,  $i = 1, \dots, N_Z$ ,  $\zeta = 1, \dots, Z$ , such that:

$$\begin{pmatrix} -X & A_i^{(\zeta)}X + B_w^{(\zeta)}\Gamma_i^{(\zeta)} \\ (A_i^{(\zeta)}X + B_w^{(\zeta)}\Gamma_i^{(\zeta)})^T & -X \end{pmatrix} < 0 \quad (35)$$

$$\left[ \alpha_{kl}X + \beta_{kl}(A_i^{(\zeta)}X + B_w^{(\zeta)}\Gamma_i^{(\zeta)}) + \beta_{lk}(A_i^{(\zeta)}X + B_w^{(\zeta)}\Gamma_i^{(\zeta)})^T \right]_{1 \leq k, l \leq m} < 0 \quad (36)$$

with  $w = 1, \dots, W_Z$ .

**Remark 1:** This result is a particular case of the one obtained in [30], where a common parameter-dependent Lyapunov function has been used for control design of switched LPV systems. In this paper, a common fixed Lyapunov function is used instead, since it has proved to be enough for stabilizing the four wheeled omnidirectional mobile robot and placing its poles in the desired LMI region  $\mathcal{D}$ .

**Remark 2:** In the case of (switching) quasi-LPV systems obtained from a nonlinear system, the closed-loop system could be unstable for some operating conditions despite the feasibility of the design conditions. A rigorous analysis of the stability should also take into account the region of attraction estimates as in [38].

## V. FAULT TOLERANT CONTROL USING SWITCHING LPV VIRTUAL ACTUATORS

### A. Fault Definition

Let us consider the switching (quasi-)LPV error system obtained from (30) including actuator faults as follows:

$$e(k+1) = A_\sigma(\vartheta(k))e(k) + B_{\sigma,f}(\vartheta(k), \phi(k))\Delta u(k) \quad (37)$$

where the multiplicative actuator faults are embedded in the matrix  $B_{\sigma,f}(\vartheta(k), \phi(k))$ , as follows:

$$B_{\sigma,f}(\vartheta(k), \phi(k)) = B_\sigma(\vartheta(k)) \text{diag}(\phi_1(k), \dots, \phi_{n_u}(k)) \quad (38)$$

where  $B_\sigma(\vartheta(k))$  denotes the faultless input matrix, and  $\phi_i(k)$  represents the effectiveness of the  $i$ th actuator, such that the extreme values  $\phi_i = 0$  and  $\phi_i = 1$  represent a total failure of the  $i$ th actuator and the healthy  $i$ th actuator, respectively.

### B. The Switching LPV Virtual Actuator

In this paper, the concept of virtual actuator introduced in [22] is extended to switching LPV systems. The virtual actuator can be either a static or a dynamic block, depending on the satisfaction of the following rank condition:

$$\text{rank}(B_{\sigma,f}(\vartheta(k), \phi(k))) = \text{rank}(B_\sigma(\vartheta(k))) \quad (39)$$

If (39) holds, e.g. in the case of multiplicative actuator faults, the reconfiguration structure is static and can be expressed as:

$$\Delta u(k) = N_{\sigma,v}(\vartheta(k), \phi(k))\Delta u_c(k) \quad (40)$$

where  $\Delta u_c(k)$  is the controller output and:

$$N_{\sigma,v}(\vartheta(k), \phi(k)) = B_{\sigma,f}^\dagger(\vartheta(k), \phi(k))B_\sigma(\vartheta(k)) \quad (41)$$

Cases where (39) is not satisfied should be described through values of the matrix  $B_\sigma^*(\vartheta(k))$  such that the following condition holds<sup>4</sup>:

$$B_\sigma^*(\vartheta(k)) = B_{\sigma,f}(\vartheta(k), \phi(k)) N_{\sigma,v}(\vartheta(k), \phi(k)) \quad (42)$$

In such cases, the reconfiguration structure is expressed by:

$$\Delta u(k) = N_{\sigma,v}(\vartheta(k), \phi(k)) (\Delta u_c(k) - M_{\sigma,v}(\vartheta(k)) x_v(k)) \quad (43)$$

where  $M_{\sigma,v}(\vartheta(k))$  is the gain of the switching LPV virtual actuator, while the virtual actuator state  $x_v(k)$  is calculated as:

$$x_v(k+1) = (A_\sigma(\vartheta(k)) + B_\sigma^*(\vartheta(k)) M_{\sigma,v}(\vartheta(k))) x_v(k) + (B_\sigma(\vartheta(k)) - B_\sigma^*(\vartheta(k))) \Delta u_c(k) \quad (44)$$

with  $x_v(0) = 0$ . When the actuator faults appear, the switching LPV virtual actuator reconstructs the vector  $\Delta u(k)$  from the output of the nominal controller  $\Delta u_c(k)$ , taking into account the fault occurrence. The faulty plant and the switching LPV virtual actuator are called the *reconfigured switching LPV plant*, which is connected to the nominal switching LPV controller. If the reconfigured switching LPV plant behaves like the nominal plant, the loop consisting of the reconfigured plant and the switching LPV controller behaves like the nominal closed-loop system.

When there is no fault,  $\Delta u(k) = \Delta u_c(k)$  and the switching LPV system is directly controlled by the switching LPV error-feedback controller (31). On the other hand, under faulty conditions, the signal entering into the controller is slightly modified, such that the output of the controller (31) becomes as follows:

$$\Delta u_c(k) = K_\sigma(\vartheta(k)) (e(k) + x_v(k)) \quad (45)$$

### C. Reconfiguration Analysis

In the following, it is shown that thanks to this modification and to the introduction of the virtual actuator block, the augmented system can be brought to a block triangular form.

*Theorem 1:* Consider the augmented system made up by the faulty system (37), the virtual actuator (43)-(44) and the control law (45)<sup>5</sup>:

$$\begin{pmatrix} e(k+1) \\ x_v(k+1) \end{pmatrix} = \begin{pmatrix} A_\sigma + B_\sigma^* K_\sigma & B_\sigma^* (K_\sigma - M_{\sigma,v}) \\ (B_\sigma - B_\sigma^*) K_\sigma & A_\sigma + B_\sigma^* M_{\sigma,v} + (B_\sigma - B_\sigma^*) K_\sigma \end{pmatrix} \begin{pmatrix} e(k) \\ x_v(k) \end{pmatrix} \quad (46)$$

Then, there exists a similarity transformation such that the state matrix of the augmented system in the new state variables is block triangular, as follows:

$$A_{aug}(\vartheta(k)) = \begin{pmatrix} A_\sigma + B_\sigma K_\sigma & 0 \\ (B_\sigma - B_\sigma^*) K_\sigma & A_\sigma + B_\sigma^* M_{\sigma,v} \end{pmatrix} \quad (47)$$

<sup>4</sup>Notice that the matrix  $B_\sigma^*(\vartheta(k))$  does not depend on  $\phi(k)$  because the matrix  $N_{\sigma,v}(\vartheta(k), \phi(k))$  eliminates the effects of actuator partial faults. A mathematical proof of this fact relies on the properties of the pseudo-inverse [39].

<sup>5</sup>The dependence of the matrices  $A_\sigma$ ,  $B_\sigma$ ,  $B_\sigma^*$ ,  $K_\sigma$  and  $M_{\sigma,v}$  on the varying parameter vector  $\vartheta(k)$  has been omitted for lack of space.

*Proof:* The proof is straightforward, and comes from introducing the new state variable  $x_1(k) \triangleq e(k) + x_v(k)$  and considering the state  $\begin{pmatrix} x_1(k) & x_v(k) \end{pmatrix}^T$ . ■

Looking at (47), it can be seen that the state  $x_1(k)$  is affected by  $K_\sigma(\vartheta(k))$  through the matrix  $A_\sigma(\vartheta(k)) + B_\sigma(\vartheta(k)) K_\sigma(\vartheta(k))$ , while the state  $x_v(k)$  is affected by  $M_{\sigma,v}(\vartheta(k))$  through the matrix  $A_\sigma(\vartheta(k)) + B_\sigma^*(\vartheta(k)) M_{\sigma,v}(\vartheta(k))$ . Hence, the switching LPV controller and the switching LPV virtual actuator can be designed independently.

The design conditions presented in Section IV can be applied to the case of virtual actuator design by making the changes  $B_\sigma(\vartheta(t)) \rightarrow B_\sigma^*(\vartheta(t))$  and  $K_\sigma(\vartheta(t)) \rightarrow M_{\sigma,v}(\vartheta(t))$  and considering a polytopic approximation of  $B_\sigma^*(\vartheta(t))$  and  $M_{\sigma,v}(\vartheta(t))$  similar to (33)-(34).

## VI. APPLICATION TO THE ROBOT

### A. Reference Inputs Calculation for Trajectory Tracking

To make the robot track a desired trajectory, proper values of  $u_0^r$ ,  $u_1^r$ ,  $u_2^r$ ,  $u_3^r$  should be fed to the reference model, such that its state equals the one corresponding to the desired trajectory. In this paper, a circular trajectory is chosen and defined as follows:

$$x_r(t) = \rho \cos(\theta_r(t)) \quad (48)$$

$$y_r(t) = \rho \sin(\theta_r(t)) \quad (49)$$

$$\theta_r(t) = \frac{2\pi t}{T} \quad (50)$$

where  $\rho$  is the circle radius and  $T$  is the desired revolution period around the circle center. Taking the derivatives and second derivatives of (48)-(50), taking into account (12), (14) and (16), and replacing into (12)-(17), the following is obtained:

$$A_{ref}(t) \begin{pmatrix} u_0^r(t) \\ u_1^r(t) \\ u_2^r(t) \\ u_3^r(t) \end{pmatrix} = B_{ref}(t) \quad (51)$$

with:

$$A_{ref}(t) = \begin{pmatrix} -B_l s_\theta & -B_l c_\theta & B_l s_\theta & B_l c_\theta \\ B_l c_\theta & -B_l s_\theta & -B_l c_\theta & B_l s_\theta \\ B_3 & B_3 & B_3 & B_3 \end{pmatrix} \quad (52)$$

$$B_{ref}(t) = \begin{pmatrix} \beta_{ref1}(t) & \beta_{ref2}(t) & \beta_{ref3}(t) \end{pmatrix}^T \quad (53)$$

$$\beta_{ref1}(t) = \rho \frac{2\pi}{T} \left( A_l \sin \frac{2\pi t}{T} + \cos \frac{2\pi t}{T} \left( \omega(t) - \frac{2\pi}{T} \right) \right) - K_l [c_\theta \text{sign}(v_x c_\theta + v_y s_\theta) - s_\theta \text{sign}(v_y c_\theta - v_x s_\theta)]$$

$$\beta_{ref2}(t) = \rho \frac{2\pi}{T} \left( \sin \frac{2\pi t}{T} \left( \omega(t) - \frac{2\pi}{T} \right) - A_l \cos \frac{2\pi t}{T} \right) - K_l [s_\theta \text{sign}(v_x c_\theta + v_y s_\theta) + c_\theta \text{sign}(v_y c_\theta - v_x s_\theta)]$$

$$\beta_{ref3}(t) = -A_{33}(\omega(t)) \frac{2\pi}{T} - K_{33} \text{sign}(\omega(t))$$

Finally, the reference model inputs  $u_i^{ref}(t)$ ,  $i = 0, 1, 2, 3$ , are obtained as:

$$\begin{pmatrix} u_0^r(t) \\ u_1^r(t) \\ u_2^r(t) \\ u_3^r(t) \end{pmatrix} = A_{ref}^\dagger(t) B_{ref}(t) \quad (54)$$

where  $A_{ref}^\dagger(t)$  is the pseudoinverse of  $A_{ref}(t)$ .

**Remark 3:** The obtained values  $u_i^r(t)$ ,  $i = 0, 1, 2, 3$  depend on the specifications, defined by the radius  $\rho$  and revolution period  $T$  of the desired circular trajectory (48)-(50). Special care should be taken in choosing  $\rho$  and  $T$ , such that the resulting reference inputs do not cause the motors to work near/in their saturation region.

**Remark 4:** The reference input calculation presented in this section can be applied to obtain the tracking of a wider class of trajectories. In particular, if  $x_r(t)$ ,  $y_r(t)$ ,  $\theta_r(t) \in \mathcal{C}^2$  in some time interval  $t \in [t_0, t_f]$ , then  $\beta_{ref1}(t)$ ,  $\beta_{ref2}(t)$  and  $\beta_{ref3}(t)$  in (53) take the following form for  $t \in [t_0, t_f]$ :

$$\begin{aligned}\beta_{ref1}(t) &= \ddot{x}_r(t) - A_l \dot{x}_r(t) + \omega \dot{y}_r(t) \\ &\quad - K_l [c_\theta \text{sign}(v_x c_\theta + v_y s_\theta) - s_\theta \text{sign}(-v_x s_\theta + v_y c_\theta)] \\ \beta_{ref2}(t) &= \ddot{y}_r(t) - A_l \dot{y}_r(t) - \omega \dot{x}_r(t) \\ &\quad - K_l [s_\theta \text{sign}(v_x c_\theta + v_y s_\theta) + c_\theta \text{sign}(-v_x s_\theta + v_y c_\theta)] \\ \beta_{ref3}(t) &= \ddot{\theta}_r(t) - A_{33}(\omega(t)) \dot{\theta}_r(t) - K_{33} \text{sign}(\omega(t))\end{aligned}$$

In this way, most of the trajectories that are of interest in mobile robot applications can be obtained, e.g. polynomials, conic and polygonal trajectories.

### B. Faulty Error Model of the Four Wheeled Robot

By including faults that cause a change of effectiveness in the actuators, the dynamic model of the four-wheeled omnidirectional robot is slightly changed. In particular, in Eqs. (2), (4) and (9),  $u_0$ ,  $u_1$ ,  $u_2$  and  $u_3$  are replaced with  $\phi_0 u_0$ ,  $\phi_1 u_1$ ,  $\phi_2 u_2$  and  $\phi_3 u_3$ , respectively, where  $\phi_i$ ,  $i = 0, 1, 2, 3$  denotes the multiplicative fault in the  $i$ th wheel.

By modifying Eqs. (13), (15) and (17), replacing  $u_0^r$ ,  $u_1^r$ ,  $u_2^r$  and  $u_3^r$  with  $\hat{\phi}_0 u_0^r$ ,  $\hat{\phi}_1 u_1^r$ ,  $\hat{\phi}_2 u_2^r$  and  $\hat{\phi}_3 u_3^r$ , where  $\hat{\phi}_i$ ,  $i = 0, 1, 2, 3$  is the multiplicative fault estimation, then, under the assumption that  $\hat{\phi}_i \approx \phi_i$ ,  $i = 0, 1, 2, 3$ , the discrete-time quasi-LPV representation of the faulty four wheeled omnidirectional mobile robot error model is the following:

$$\begin{aligned}\begin{pmatrix} e_1(k+1) \\ e_2(k+1) \\ e_3(k+1) \\ e_4(k+1) \\ e_5(k+1) \\ e_6(k+1) \end{pmatrix} &= \begin{pmatrix} 1 & T_s & 0 & 0 & 0 & 0 \\ 0 & A_1^d & 0 & -\vartheta_1^d & 0 & 0 \\ 0 & 0 & 1 & T_s & 0 & 0 \\ 0 & \vartheta_1^d & 0 & A_1^d & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 0 & 0 & \vartheta_2^d \end{pmatrix} \begin{pmatrix} e_1(k) \\ e_2(k) \\ e_3(k) \\ e_4(k) \\ e_5(k) \\ e_6(k) \end{pmatrix} \\ &+ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -B_1 \vartheta_3^d & -B_1 \vartheta_4^d & B_1 \vartheta_3^d & B_1 \vartheta_4^d & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ B_1 \vartheta_4^d & -B_1 \vartheta_3^d & -B_1 \vartheta_4^d & B_1 \vartheta_3^d & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ B_3 T_s & B_3 T_s & B_3 T_s & B_3 T_s & 0 & 0 \end{pmatrix} \dots \\ &\dots \begin{pmatrix} \phi_0 & 0 & 0 & 0 \\ 0 & \phi_1 & 0 & 0 \\ 0 & 0 & \phi_2 & 0 \\ 0 & 0 & 0 & \phi_3 \end{pmatrix} \begin{pmatrix} \Delta u_0(k) \\ \Delta u_1(k) \\ \Delta u_2(k) \\ \Delta u_3(k) \end{pmatrix}\end{aligned}\quad (55)$$

that is in the form (37)-(38). Hence, the virtual actuator technique can be applied to the faulty error model (55).

Adapting the reference input calculation presented in Section VI-A to the faulty case, the matrix  $A_{ref}(t)$  becomes:

$$A_{ref}(t) = \begin{pmatrix} -B_1 s_\theta \hat{\phi}_0 & -B_1 c_\theta \hat{\phi}_1 & B_1 s_\theta \hat{\phi}_2 & B_1 c_\theta \hat{\phi}_3 \\ B_1 c_\theta \hat{\phi}_0 & -B_1 s_\theta \hat{\phi}_1 & -B_1 c_\theta \hat{\phi}_2 & B_1 s_\theta \hat{\phi}_3 \\ B_3 \hat{\phi}_0 & B_3 \hat{\phi}_1 & B_3 \hat{\phi}_2 & B_3 \hat{\phi}_3 \end{pmatrix} \quad (56)$$

### C. Fault Estimation for the Four Wheeled Robot

In order to apply the proposed strategy, a fault estimation is needed. Hereafter, the fault estimation is formulated as a parameter estimation problem in such a way that any parameter estimation algorithm, such as least squares, could be used. In general, least squares algorithms can be formulated either in block or in recursive online form [40]. Once the equation is put in regressor form, the recursive formulation [41] and the block formulation [42] are interchangeable.

In order to obtain the regressors for estimating the multiplicative faults  $\hat{\phi}_0$ ,  $\hat{\phi}_1$ ,  $\hat{\phi}_2$  and  $\hat{\phi}_3$ , the discrete-time faulty versions of (13), (15) and (17), where  $u_0$ ,  $u_1$ ,  $u_2$ ,  $u_3$  are replaced with  $\phi_0 u_0$ ,  $\phi_1 u_1$ ,  $\phi_2 u_2$ ,  $\phi_3 u_3$  are considered. Then, using basic algebraic manipulations, it is possible to obtain:

$$z_x(k) = \mu_x^T(k) \phi(k) \quad (57)$$

$$z_y(k) = \mu_y^T(k) \phi(k) \quad (58)$$

$$z_\theta(k) = \mu_\theta^T(k) \phi(k) \quad (59)$$

where:

$$\phi(k) = [\phi_0(k-1) \quad \phi_1(k-1) \quad \phi_2(k-1) \quad \phi_3(k-1)]^T \quad (60)$$

$$\begin{aligned}z_x(k) &= v_x(k) - v_x(k-1) - (A_{11} c_\theta^2 + A_{22} s_\theta^2) T_s v_x(k-1) \\ &\quad - ((A_{11} - A_{22}) s_\theta c_\theta - \omega(k-1)) T_s v_y(k-1) \\ &\quad - K_{11} c_\theta T_s \text{sign}(v_x(k-1) c_\theta + v_y(k-1) s_\theta) \\ &\quad + K_{22} s_\theta T_s \text{sign}(v_y(k-1) c_\theta - v_x(k-1) s_\theta)\end{aligned} \quad (61)$$

$$\mu_x^T(k) = [\begin{matrix} -B_{21} s_\theta T_s u_0(k-1) & B_{12} c_\theta T_s u_1(k-1) & \dots \\ \dots & -B_{23} s_\theta T_s u_2(k-1) & B_{14} c_\theta T_s u_3(k-1) \end{matrix}] \quad (62)$$

$$\begin{aligned}z_y(k) &= v_y(k) - v_y(k-1) - (A_{11} s_\theta^2 + A_{22} c_\theta^2) T_s v_y(k-1) \\ &\quad - ((A_{11} - A_{22}) s_\theta c_\theta + \omega(k-1)) T_s v_x(k-1) \\ &\quad - K_{11} s_\theta T_s \text{sign}(v_x(k-1) c_\theta + v_y(k-1) s_\theta) \\ &\quad - K_{22} c_\theta T_s \text{sign}(v_y(k-1) c_\theta - v_x(k-1) s_\theta)\end{aligned} \quad (63)$$

$$\mu_y^T(k) = [\begin{matrix} B_{21} c_\theta T_s u_0(k-1) & B_{12} s_\theta T_s u_1(k-1) & \dots \\ \dots & B_{23} c_\theta T_s u_2(k-1) & B_{14} s_\theta T_s u_3(k-1) \end{matrix}] \quad (64)$$

$$\begin{aligned}z_\theta(k) &= \omega(k) - \omega(k-1) - K_{33} T_s \text{sign}(\omega(k-1)) \\ &\quad - A_{33}(\omega(k-1)) T_s \omega(k-1)\end{aligned} \quad (65)$$

$$\mu_\theta^T(k) = [\begin{matrix} B_{31} T_s u_0(k-1) & B_{32} T_s u_1(k-1) & \dots \\ \dots & B_{33} T_s u_2(k-1) & B_{34} T_s u_3(k-1) \end{matrix}] \quad (66)$$

with  $c_\theta = \cos \theta(k-1)$  and  $s_\theta = \sin \theta(k-1)$ . Then, if a block formulation with time window  $N_{LS}$  is used, the least squares fault estimation is obtained as:

$$\hat{\phi}(k) = M(k)^\dagger \xi(k) \quad (67)$$

$$\xi(k) = \begin{bmatrix} z_x(k) \\ z_y(k) \\ z_\theta(k) \\ \vdots \\ z_x(k-N_{LS}+1) \\ z_y(k-N_{LS}+1) \\ z_\theta(k-N_{LS}+1) \end{bmatrix} \quad M(k) = \begin{bmatrix} \mu_x^T(k) \\ \mu_y^T(k) \\ \mu_\theta^T(k) \\ \vdots \\ \mu_x^T(k-N_{LS}+1) \\ \mu_y^T(k-N_{LS}+1) \\ \mu_\theta^T(k-N_{LS}+1) \end{bmatrix} \quad (68)$$

with  $M(k)^\dagger$  denoting the pseudoinverse of  $M(k)$ .



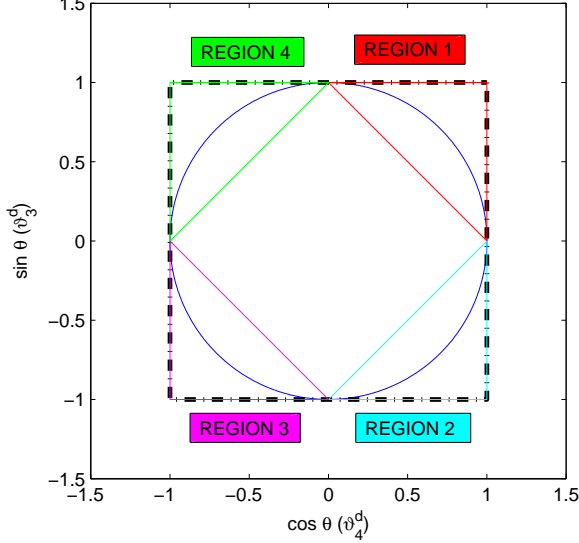


Fig. 3. Polytopic LPV and polytopic switching LPV approximations of the scheduling variables  $\vartheta_3^d(k)$  and  $\vartheta_4^d(k)$ .

#### D. Switching LPV Polytopic Model of the Four Wheeled Robot

When the polytopic LPV conditions (28)-(29) are applied to some polytopic approximation of the four wheeled omnidirectional mobile robot quasi-LPV model (18), a solution could not exist due to the loss of controllability occurring for  $\vartheta_3^d = \vartheta_4^d = 0$ , values for which the input matrix becomes:

$$B_{\vartheta_3^d=\vartheta_4^d=0} = \begin{pmatrix} 0_{5 \times 1} & 0_{5 \times 1} & 0_{5 \times 1} & 0_{5 \times 1} \\ B_3 & B_3 & B_3 & B_3 \end{pmatrix} \quad (69)$$

Due to the fact that the set described by the polytopic approximation (26) is convex, it is straightforward that any polytopic approximation of the admissible values for  $\vartheta_3^d(k) = \sin(\theta(k))T_s$  and  $\vartheta_4^d(k) = \cos(\theta(k))T_s$  will contain the origin, that is, the singularity (69) of the input matrix (see the dash-dotted black line in Fig. 3).

However, this problem can be avoided using a switching LPV controller, as described in Section IV-B, by splitting the subset of the parameter space generated by  $\vartheta_3^d$  and  $\vartheta_4^d$  in more regions, such that in each region the resulting polytopic approximation does not include the origin. In particular, in this work, the quadrants have been considered as regions, with  $\theta = k\pi/2$ ,  $k \in \mathbb{Z}$  being the switching condition (see Fig. 3), such that:

$$\sigma = \begin{cases} 1 & \text{if } \cos \theta \geq 0 \text{ AND } \sin \theta \geq 0 \\ 2 & \text{if } \cos \theta \geq 0 \text{ AND } \sin \theta < 0 \\ 3 & \text{if } \cos \theta < 0 \text{ AND } \sin \theta < 0 \\ 4 & \text{if } \cos \theta < 0 \text{ AND } \sin \theta \geq 0 \end{cases}$$

A triangular approximation has been used, in each region, for the pair  $\{\vartheta_3^d, \vartheta_4^d\}$ , with the following structure:

$$\begin{pmatrix} \vartheta_3^d \\ \vartheta_4^d \end{pmatrix} \in Co \left\{ \begin{pmatrix} \pm T_s \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \pm T_s \end{pmatrix}, \begin{pmatrix} \pm T_s \\ \pm T_s \end{pmatrix} \right\}$$

where  $Co$  denotes the convex set, and whether  $\pm$  is  $+$  or  $-$  depends, for each varying parameter, on the region that is

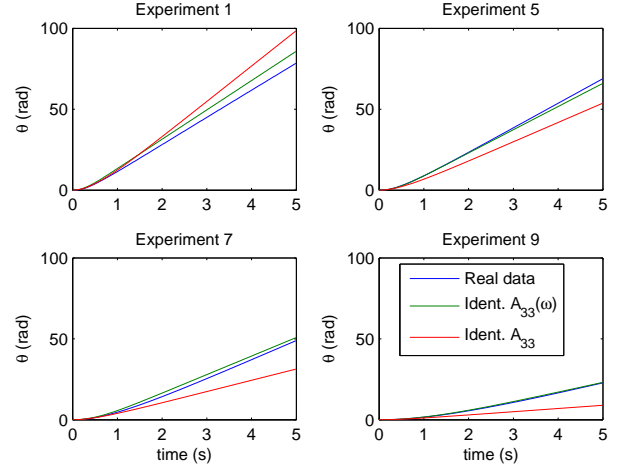


Fig. 4. Angular subsystem identification.

being considered. In particular, the polytopic approximations for  $\sigma = 1$ ,  $\sigma = 2$ ,  $\sigma = 3$  and  $\sigma = 4$  are given by the red, the cyan, the magenta and the green triangle in Fig. 3, respectively.

## VII. RESULTS

### A. Identification

At first, the identification procedure has been applied to sets of data that have been obtained as the angular response of the robot to input voltages of the same value ( $u_0 = u_1 = u_2 = u_3$ ). In these experiments, the inputs caused a rotation of the robot that was clockwise for positive values and anticlockwise for negative values. The critical value  $u^*$  such that (7) with  $u_0^* = u_1^* = u_2^* = u_3^*$  holds has been identified as  $u^* = 30$ . Hence, using  $N_\theta = 20$  sets of data which contained the results of experiments that lasted 5 s, with input values ranging from  $-250$  to  $250$ , the objective function (8) subject to (7) has been minimized, obtaining  $A_{33} = -2.009$  and  $B_3 = 0.05$ , that corresponds to  $K_{33} = -6$ . However, the simulations obtained with these parameters did not fit well the data, as shown in Fig. 4, when comparing the real data (blue lines) and the simulated data (red lines).

As already remarked in Section III-B, by applying the identification procedure to each set of data separately, a dependence of  $A_{33}$  on  $\omega$  has been observed that could be explained by a second-order polynomial, as follows:

$$A_{33}(\omega) = -0.0062\omega^2 + 0.0028\omega - 0.4406 \quad (70)$$

The improvement brought by considering  $A_{33}$  as a function of  $\omega$  is visible in Fig. 4 when the green line, representing the simulations obtained with the identified  $A_{33}(\omega)$ , is compared to the blue and red ones, representing the real data and the simulations obtained with a constant  $A_{33}$ , respectively.

Later, the identification procedure has been applied to sets of data that have been obtained as the linear response of the robot to input voltages that were pairwise of the same value ( $u_0 = u_2$  and  $u_1 = u_3$ ). By minimizing the objective function (11) subject to (10) with  $u^* = u_0^* = u_2^* = 30$ ,  $A_I = -1.4904$  and  $B_I = 0.0089$ , that correspond to  $K_I = -0.5340$ , were obtained.



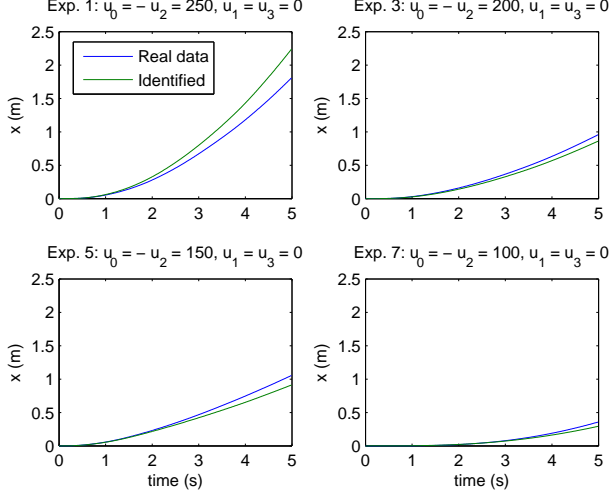


Fig. 5. Linear subsystem identification:  $x$ .

The comparison between the real data (blue lines) and the simulated data (green lines) in Fig. 5 shows that the model fits well the real behavior of the robot.

### B. Design

The polytopic approximation (32)-(33) of the four-wheeled omnidirectional mobile robot error model (18) has been obtained by considering  $T_s = 0.04$  s and  $\omega \in [-2.5 \text{ rad/s}, 2.5 \text{ rad/s}]$ .

The controller and the virtual actuators (one for each wheel) have been designed using (35)-(36), to assure stability and pole clustering in:

$$\mathcal{D} = \{z \in \mathbb{C} : \text{Re}(z) > 0.40, \text{Re}(z)^2 + \text{Im}(z)^2 < 0.9997^2\} \quad (71)$$

The 48 vertex controller and virtual actuator matrices (12 for each region) are stored in the Matlab workspace. When the robot is operating, at each time sample, the active region  $\zeta$  is selected taking into account the value of  $\theta$ . Then, the value of the vector of varying parameters  $\vartheta_d(\theta(k), \omega(k))$  is calculated and used for obtaining the 12 coefficients  $\gamma_i^{(\zeta)}$  of the polytopic decomposition (32). Finally, the controller gain, and the virtual actuator gain when the system is faulty, are obtained as a linear combination of the vertex controller/virtual actuator matrices using the coefficients  $\gamma_i^{(\zeta)}$ , and the feedback input  $\Delta u(k)$  can be computed. It is worth highlighting that only a small fraction of the overall computational cost of the proposed strategy is performed online (calculating the coefficients  $\gamma_i^{(\zeta)}$ , performing a linear combination of vertex matrices, and calculating the reference inputs), since solving the LMIs is a task performed offline.

Three control experiments have been considered, where the robot started from different initial states:

- Experiment 1

$$(x(0), v_x(0), y(0), v_y(0), \theta(0), \omega(0))^T = (1.5 \quad 0_{1 \times 5})^T$$

TABLE I  
MEAN SQUARED ERRORS WITHOUT AND WITH FTC (SIMULATION)

|                   | $e_1^2$ | $e_2^2$ | $e_3^2$ | $e_4^2$ | $e_5^2$ | $e_6^2$ |
|-------------------|---------|---------|---------|---------|---------|---------|
| Sim.1 without FTC | 0.024   | 0.004   | 0.022   | 0.003   | 1.438   | 0.025   |
| Sim.1 with FTC    | 0.007   | 0.001   | 0.002   | 0.001   | 0.297   | 0.016   |
| Sim.2 without FTC | 0.018   | 0.003   | 0.022   | 0.003   | 1.440   | 0.029   |
| Sim.2 with FTC    | 0.001   | 0.000   | 0.001   | 0.001   | 0.291   | 0.020   |
| Sim.3 without FTC | 0.037   | 0.007   | 0.020   | 0.003   | 1.440   | 0.027   |
| Sim.3 with FTC    | 0.021   | 0.004   | 0.002   | 0.001   | 0.295   | 0.023   |

- Experiment 2

$$(x(0), v_x(0), y(0), v_y(0), \theta(0), \omega(0))^T = (1 \quad 0_{1 \times 5})^T$$

- Experiment 3

$$(x(0), v_x(0), y(0), v_y(0), \theta(0), \omega(0))^T = 0_{6 \times 1}$$

and tracked the desired trajectory, a circle centered in the origin of the  $(x-y)$  plane with a radius of 1 m and a revolution period of 20 s, generated from the initial reference state:

$$(x_r(0), v_x^r(0), y_r(0), v_y^r(0), \theta_r(0), \omega_r(0))^T = (1, 0, 0, \pi/10, 0, \pi/10)^T$$

The reference input calculation described in Section VI-A has been applied before the fault appears and in the experiment without FTC after the fault appears. On the other hand, the trajectory is generated using the reference inputs as calculated in Section VI-B in the experiment with FTC after the fault appears.

The fault scenario considered in this paper is a total loss of the first wheel motor starting from time  $t = 20$  s:

$$\phi_0(t) = \begin{cases} 1 & \text{if } t < 20 \text{ s} \\ 0 & \text{if } t \geq 20 \text{ s} \end{cases} \quad (72)$$

### C. Simulation Results

In the following, simulation results are shown for Experiment 1, while Table I resumes the mean squared errors for the trajectory tracking in all the three considered experiments. The improvement brought by the proposed FTC strategy on the tracking performance can be seen clearly in all the considered experiments.

Fig. 6 shows the tracking of the desired circular trajectory in the  $(x-y)$  plane. It can be seen that, in the case where the proposed FTC technique is not applied, the robot trajectory (red line) deviates from the reference trajectory (black dots) after the fault appears. On the other hand, adding the virtual actuator to the control loop increases the tracking performance of the robot (blue line).

Fig. 7 shows the fault estimation obtained with the approach described in Section VI-C and  $N_{LS} = 50$ . It can be seen that, after a short transient, the fault is correctly estimated. Also, the results demonstrate an intrinsic robustness of the proposed FTC strategy against errors in the fault estimation.

### D. Experimental Results

In the following, experimental results are shown for Experiment 1, while Table II resumes the mean squared errors for the trajectory tracking in all the three considered experiments. It must be remarked that the odometry of omnidirectional mobile

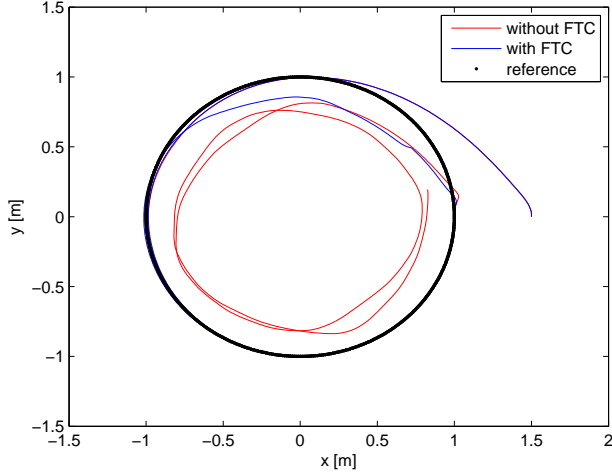


Fig. 6. Tracking of the desired circular trajectory:  $(x-y)$  plane (Sim. 1).

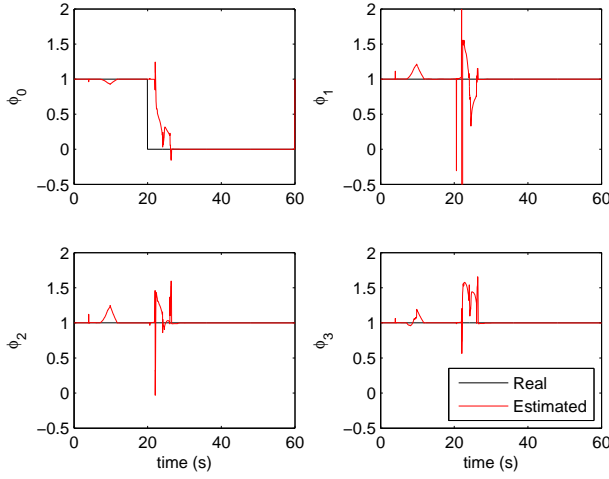


Fig. 7. Fault estimation with  $N_{LS} = 50$  (Sim. 1).

robots puts a real challenge in estimating the faults appearing in the motors using data coming from the available sensors. Also, another problem arises from the fact that in the real robot, under a change of effectiveness in a motor due to a fault, the corresponding wheel will keep rotating not only according to the motor torque, but also according to the robot movement due to the remaining wheels. These effects made the proposed fault estimation approach described in Section VI-C not as good as in simulation when applied to the real robot. However, the results presented hereafter, obtained under the assumption of perfect fault estimation, i.e.  $\hat{\phi}_i = \phi_i, i = 0, 1, 2, 3$  demonstrate that the omnidirectional mobile robot is able to operate under a severe fault occurrence, e.g. the total loss of one motor, if an appropriate fault-hiding strategy is implemented.

Fig. 8 and Fig. 9 show the tracking of the desired circular trajectory in the  $(x-y)$  plane and a comparison between the system states and the reference ones. When the proposed FTC strategy is applied, all the system states go to the reference ones, i.e. the tracking errors go to zero, except for a steady-

TABLE II  
MEAN SQUARED ERRORS WITHOUT AND WITH FTC (EXPERIMENTAL)

|                   | $e_1^2$ | $e_2^2$ | $e_3^2$ | $e_4^2$ | $e_5^2$ | $e_6^2$ |
|-------------------|---------|---------|---------|---------|---------|---------|
| Exp.1 without FTC | 0.110   | 0.017   | 0.081   | 0.016   | 7.284   | 0.259   |
| Exp.1 with FTC    | 0.009   | 0.001   | 0.002   | 0.001   | 2.023   | 0.014   |
| Exp.2 without FTC | 0.048   | 0.006   | 0.038   | 0.010   | 1.814   | 0.158   |
| Exp.2 with FTC    | 0.006   | 0.001   | 0.004   | 0.002   | 3.630   | 0.024   |
| Exp.3 without FTC | 0.085   | 0.015   | 0.051   | 0.012   | 1.757   | 0.153   |
| Exp.3 with FTC    | 0.024   | 0.004   | 0.003   | 0.002   | 3.417   | 0.026   |

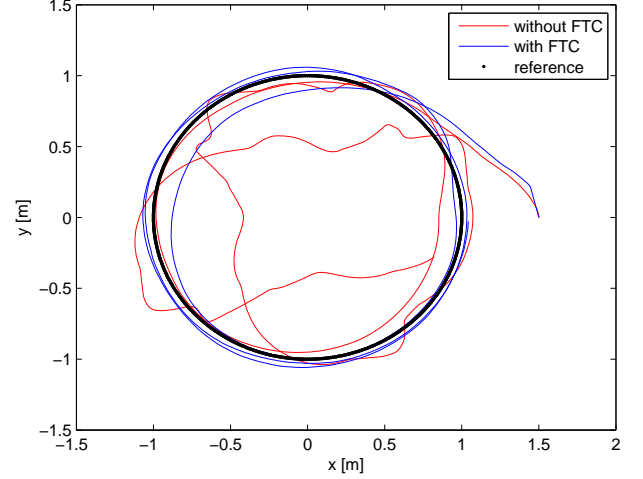


Fig. 8. Tracking of the desired circular trajectory:  $(x-y)$  plane (Exp. 1).

state error in the angle  $\theta$ . The addition of an integral action could eliminate such error, even though at the expense of a probable decrease in the system performance, as well as the appearance of the need to introduce anti-windup mechanisms to avoid undesired effects due to the motor saturation nonlinearities. Finally, in Fig. 10, the control inputs are presented. It can be seen that the control inputs are such that all the motors are working in their linear region. Moreover, under fault effect, the effect of the first wheel on the system is redistributed over the remaining wheels to achieve fault tolerance.

## VIII. CONCLUSIONS

This paper has presented the modeling, identification and control of a four wheeled omnidirectional robot including some fault-hiding mechanisms to achieve fault tolerance. The unknown parameters of the four wheeled robot nonlinear model have been identified by means of nonlinear least squares identification using data collected from the real robot. The problem of controlling the robot such that it tracks a desired trajectory has been solved. The proposed solution relies on the use of a reference model that describes the desired trajectory. The resulting nonlinear error model is brought to a quasi-LPV form that is used for designing a switching LPV controller using LMI-based techniques. A switching LPV virtual actuator that adapts the faulty plant to the nominal switching LPV controller instead of adapting the switching LPV controller to the faulty plant has also been added. In this way, the faulty plant together with the switching LPV virtual actuator block allows the switching LPV controller to see the same plant

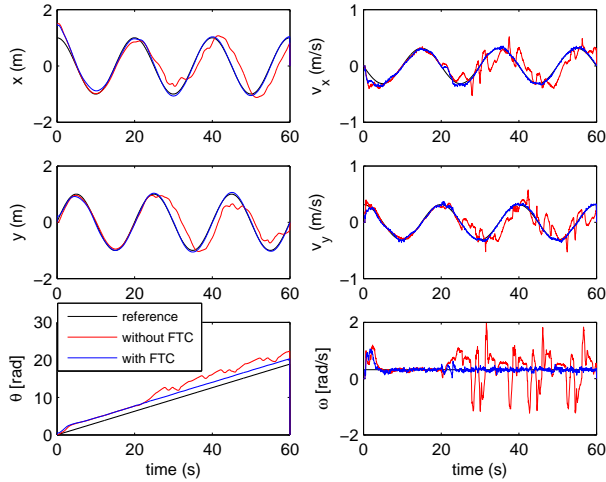


Fig. 9. Tracking of the desired circular trajectory: states (Experiment 1).

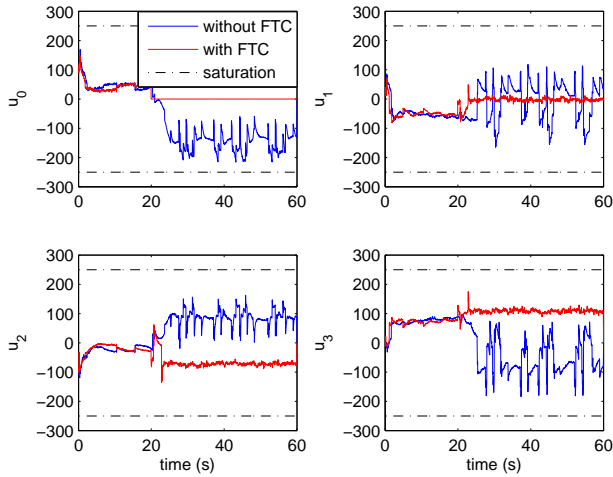


Fig. 10. Tracking of the desired circular trajectory: inputs (Experiment 1).

as before the fault. The addition of the virtual actuator block keeps the stability and some desired performances under fault occurrence. The overall loop consists of the nominal switching LPV controller and the switching LPV virtual actuators. Both are designed using polytopic techniques within the quadratic Lyapunov framework, solving a system of LMIs to achieve stability and pole clustering in a desired region through the use of a single quadratic Lyapunov function.

The simulation and experimental results have shown the potential and performances of the proposed approach using a scenario where the first wheel motor was subject to an actuator fault, i.e. a total loss of the motor. The proposed FTC strategy was able to recover the tracking of the desired trajectory despite the fault. However, the proposed least-squares-based fault estimation approach did not provide satisfactory enough results when applied to the real robot, due to the poor odometry and the presence of couplings between wheels, which effects were not taken into account by the model. Hence, some important issues for future research are the improvement of

the modeling of omnidirectional robots, and the increasing of the FTC strategy robustness against the different sources of uncertainty that affect the control system.

## REFERENCES

- [1] H. P. Oliveira, A. J. Sousa, A. P. Moreira, and P. J. Costa, "Modeling and assessing of omni-directional robots with three and four wheels," in *Contemporary robotics - challenges and solutions*, A. D. Rodić, Ed. InTech, 2009.
- [2] C.-C. Tsai, H.-C. Huang, and S.-C. Lin, "FPGA-Based Parallel DNA Algorithm for Optimal Configurations of an Omnidirectional Mobile Service Robot Performing Fire Extinguishment," *IEEE Trans. on Ind. Electron.*, vol. 58, no. 3, pp. 1016–1026, 2011.
- [3] H. Kim and B. K. Kim, "Online minimum-energy trajectory planning and control on a straight-line path for three-wheeled omnidirectional mobile robots," *IEEE Trans. Ind. Electron.*, vol. 61, no. 9, pp. 4771–4779, 2014.
- [4] H. P. Oliveira, A. J. Sousa, A. P. Moreira, and P. J. Costa, "Precise modeling of a four wheeled omni-directional robot," in *Proc. of the 8th conference on autonomous robot systems and competitions*, 2008, pp. 57–62.
- [5] G. Campion, G. Bastin, and B. Dandrea-Novet, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Trans. Robot. Automat.*, vol. 12, no. 1, pp. 47–62, 1996.
- [6] R. Williams, B. Carter, P. Gallina, and G. Rosati, "Dynamic model with slip for wheeled omnidirectional robots," *IEEE Trans. Robot. Automat.*, vol. 18, no. 3, pp. 285–293, 2002.
- [7] M. M. Olsen and H. G. Petersen, "A new method for estimating parameters of a dynamic robot model," *IEEE Trans. Robot. Automat.*, vol. 17, no. 1, pp. 95–100, 2001.
- [8] A. Conceicao, A. Moreira, and P. Costa, "Model identification of a four wheeled omni-directional mobile robot," in *Proc. of the 7th Portuguese conference on automatic control*, 2006.
- [9] O. Purwin and R. D'Andrea, "Trajectory generation and control for four wheeled omnidirectional vehicles," *Robot. Auton. Syst.*, vol. 54, pp. 13–22, 2006.
- [10] G. Indiveri, "Swedish wheeled omnidirectional mobile robots: kinematic analysis and control," *IEEE Trans. Robot.*, vol. 25, no. 1, pp. 164–171, 2009.
- [11] M.-J. Jung and J.-H. Kim, "Development of a Fault-Tolerant Omnidirectional Wheeled Mobile Robot Using Nonholonomic Constraints," *Int. J. Robot. Res.*, vol. 21, no. 5-6, pp. 527–539, 2002.
- [12] A. Glove, F. Wiesel, O. Tenchio, M. Simon, and R. Rojas, "Robot Heal Thyself: Precise and Fault-Tolerant Control of Imprecise and Malfuncting Robots," in *RoboCup 2005 International Symposium*, 2005.
- [13] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*. Springer-Verlag Berlin Heidelberg, 2006.
- [14] H. Noura, D. Theilliol, J.-C. Ponsart, and A. Chamseddine, *Fault-tolerant control systems: Design and Practical Applications*. Springer-Verlag London Ltd., 2009.
- [15] S. Yin, H. Luo, and S. X. Ding, "Real-time implementation of fault-tolerant control systems with performance optimization," *IEEE Trans. Ind. Electron.*, vol. 61, no. 5, pp. 2402–2411, 2014.
- [16] B. Mirafzal, "Survey of fault-tolerance techniques for three-phase voltage source inverters," *IEEE Trans. Ind. Electron.*, vol. 61, no. 10, pp. 5192–5202, 2014.
- [17] J. Jiang and X. Yu, "Fault-tolerant control systems: a comparative study between active and passive approaches," *Annu. Rev. Control*, vol. 36, no. 1, pp. 60–72, 2012.
- [18] Y. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," *Annu. Rev. Control*, vol. 32, pp. 229–252, 2008.
- [19] G. G. Yen and L.-W. Ho, "Online multiple-model-based fault diagnosis and accommodation," *IEEE Trans. Ind. Electron.*, vol. 50, no. 2, pp. 296–312, 2003.
- [20] M. M. Polycarpou, "Fault accommodation of a class of multivariable nonlinear dynamical systems using a learning approach," *IEEE Trans. Automat. Contr.*, vol. 46, no. 5, pp. 736–742, 2001.
- [21] X. Zhang, T. Parisini, and M. M. Polycarpou, "Adaptive fault-tolerant control of nonlinear uncertain systems: an information-based diagnostic approach," *IEEE Trans. Automat. Contr.*, vol. 49, no. 8, pp. 1259–1274, 2004.
- [22] J. Lunze and T. Steffen, "Control reconfiguration after actuator failures using disturbance decoupling methods," *IEEE Trans. Automat. Contr.*, vol. 51, no. 10, pp. 1590–1601, 2006.

- [23] D. Rotondo, F. Nejari, and V. Puig, "A virtual actuator and sensor approach for fault tolerant control of LPV systems," *J. Process Contr.*, vol. 24, pp. 203–222, 2014.
- [24] L. Dziekan, M. Witczak, and J. Korbicz, "Active fault-tolerant control design for Takagi-Sugeno fuzzy systems," *B. Pol. Acad. Sci.*, vol. 59, no. 1, pp. 93–102, 2011.
- [25] J. H. Richter, W. P. M. H. Heemels, N. van de Wouw, and J. Lunze, "Reconfigurable control of piecewise affine systems with actuator and sensor faults: stability and tracking," *Automatica*, vol. 47, pp. 678–691, 2011.
- [26] J. Blesa, D. Rotondo, V. Puig, and F. Nejari, "FDI and FTC of wind turbines using the interval observer approach and virtual actuators/sensors," *Control Eng. Pract.*, vol. 24, pp. 138–155, 2014.
- [27] A. Forrai, T. Ueda, and T. Yumura, "Electromagnetic Actuator Control: A Linear Parameter-Varying (LPV) Approach," *IEEE Trans. Ind. Electron.*, vol. 54, no. 3, pp. 1430–1441, 2007.
- [28] J. S. Shamma, "An overview of LPV systems," in *Control of Linear Parameter Varying Systems with Applications*, J. Mohammadpour and C. Scherer, Eds. Springer, 2012, pp. 3–26.
- [29] K. Zavari, G. Pipeleers, and J. Swevers, "Gain-scheduled controller design: illustration to an overhead crane," *IEEE Trans. Ind. Electron.*, vol. 61, no. 7, pp. 3713–3718, 2014.
- [30] X. He, G. M. Dimirovski, and J. Zhao, "Control of switched LPV systems using common Lyapunov function method and F-16 aircraft application," in *Proc. of the IEEE International Conference on Systems, Man and Cybernetics*, 2010, pp. 386–392.
- [31] Y. D. Landau, *Adaptive control - the model reference approach*. New York: Marcel Dekker, 1979.
- [32] A. Abdullah and M. Zribi, "Model reference control of LPV systems," *J. Franklin I.*, vol. 346, pp. 854–871, 2009.
- [33] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. of the CACSD Conference*, 2004.
- [34] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optim. Method. Softw.*, vol. 11-12, pp. 625–653, 1999.
- [35] D. Rotondo, F. Nejari, and V. Puig, "Quasi-LPV modeling, identification and control of a twin rotor MIMO system," *Control Eng. Pract.*, vol. 21, pp. 829–846, 2013.
- [36] M. Chilali and P. Gahinet, " $H_\infty$  Design with Pole Placement Constraints: An LMI Approach," *IEEE Trans. Automat. Contr.*, vol. 41, no. 3, pp. 358–367, 1996.
- [37] A. Ghersin and R. Sanchez-Peña, "LPV control of a 6-DOF vehicle," *IEEE Trans. Trans. Contr. Syst. Technol.*, vol. 10, no. 6, pp. 883–887, 2002.
- [38] F. Bruzelius, S. Pettersson, and C. Breitholtz, "Region of attraction estimates for LPV-gain scheduled control systems," in *Proc. of the 7th European Control Conference*, 2003.
- [39] S. Barnett, *Matrices: Methods and applications*. Oxford Applied Mathematics and Computing Science Series, 1990.
- [40] J. Jiang and Y. Zhang, "A revisit to block and recursive least squares for parameter estimation," *Comput. Electr. Eng.*, vol. 30, no. 5, pp. 403–416, 2004.
- [41] L. Ljung, *System Identification: Theory for the user*. Prentice Hall, Englewood Cliffs, NJ, USA, 1987.
- [42] J. Jiang and Y. Zhang, "A novel variable-length sliding window block-wise least-squares algorithm for on-line estimation of time-varying parameters," *Int. J. Adapt. Control*, vol. 18, no. 6, pp. 505–521, 2004.



Cyprus from January to March 2013, and the Research Center for Automatic Control of Nancy (CRAN) at University of Lorraine, Nancy, France, from March 2014 to July 2014. His main research interests include gain-scheduled control systems, Fault Detection and Isolation (FDI) and Fault Tolerant Control (FTC) of dynamic systems. He has published several papers in international conference proceedings and scientific journals.



control of dynamic systems. He has been involved in several European projects and networks and has published many papers in international conference proceedings and scientific journals.



and isolation and fault tolerant control of dynamic systems. She has published several papers in journals and international conferences and participated in several European projects and networks related with these topics.



dynamic systems field. He has published several papers in journals and international conferences and participated in several European projects and networks related with these topics.

**Damiano Rotondo** was born in Francavilla Fontana, Italy, in 1987. He received the B.S. degree (with honors) from the Second University of Naples, Italy, and the M.S. degree (with honors) from the University of Pisa, Italy, in 2008 and 2011, respectively. Since November 2011, he has been a Ph.D. student at the Research Center for Supervision, Safety and Automatic Control (CS2AC), Technical University of Catalonia (UPC), Barcelona, Spain. He has visited the KIOS Research Center for Intelligent Systems and Networks, University of Cyprus, Nicosia,

**Vicenç Puig** was born in Girona, Spain, in 6 November 1969. He has received the Ph.D. Degree in Control Engineering in 1999 and the Telecommunications Engineering Degree in 1993, both from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain. He is currently an Associate Professor of Automatic Control and leader of the Advanced Control Systems (SAC) research group of the Research Center for Supervision, Safety and Automatic Control (CS2AC) at UPC. His main research interests are fault detection, isolation of fault-tolerant

**Fatiha Nejari** received the M.Sc. in physics from Hassan II University, Casablanca, Morocco, in 1993 and the Ph.D. in automatic control from Cadi Ayyad University, Marrakech, Morocco, in 1997. She is currently an associate professor with the Department of Automatic Control, Universitat Politècnica de Catalunya (UPC). She is also a member of the Advanced Control Systems (SAC) research group of the Research Center for Supervision, Safety and Automatic Control (CS2AC) at UPC. Her main research areas include LPV control, fault detection

**Juli Romera** was born in Barcelona, Spain, in 1972. He received the Mechanics Engineering degree in 1996 and Automatic and Electronic Engineering degree in 2000 both at the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain. He worked at different engineering companies up to 2008. Since 2008 he is member of the Advanced Control Systems (SAC) at the same University and since 2013 member of the Research Center for Supervision, Safety and Automatic Control (CS2AC). His main research interests are in the optimal control on