INTECH
open science | open minds

OPEN ACCESS
ARTICLE

International Journal of Advanced Robotic Systems

Flexible-Link Robot Control Using a Linear Parameter Varying Systems Methodology

Regular Paper

Houssem Halalchi[1,2,*], Edouard Laroche[2] and G. Iuliana Bara[2]

[1]ICube, INSA de Strasbourg, CNRS. 24 Boulevard de la Victoire, F-67084 Strasbourg, France
[2]ICube, Université de Strasbourg, CNRS. Pôle API, 300 Boulevard Sébastien Brant, BP 10413, F-67412 Illkirch, France
* Corresponding author E-mail: houssem.halalchi@unistra.fr

Received D M 2012; Accepted D M 2012

DOI: 10.5772/chapter.doi

© 2012 FIRST AUTHOR; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract This paper addresses the issues of the Linear Parameter Varying (LPV) modeling and control of flexible-link robot manipulators. The LPV formalism allows the synthesis of nonlinear control laws and the assessment of their closed-loop stability and performances in a simple and effective manner, based on the use of Linear Matrix Inequalities (LMI). Following the quasi-LPV modeling approach, an LPV model of a flexible manipulator is obtained, starting from the nonlinear dynamic model stemming from Euler-Lagrange equations. Based on this LPV model, which has a rational dependence in terms of the varying parameters, two different methods for the synthesis of LPV controllers are explored. They guarantee the asymptotic stability and some level of closed-loop $\mathcal{L}_2$-gain performance on a bounded parametric set. The first method exploits a descriptor representation that simplifies the rational dependence of the LPV model, whereas the second one manages the troublesome rational dependence by using dilated LMI conditions and taking the particular structure of the model into account. The resulting controllers involve the measured state variables only, namely the joint positions and velocities. Simulation results are presented that illustrate the validity of the proposed control methodology. Comparisons with an inversion-based nonlinear control method are performed in the presence of velocity measurement noise, high-frequency inputs and model uncertainties.

Keywords Flexible robots, LPV modeling, LPV control, LMI conditions.

1. Introduction

The control of robot manipulators is a challenging research area that has attracted the attention of scientists and engineers for several decades (see e.g. the reference book [1]). In the most basic approaches, the mechanical structure of such articulated systems is assumed to be completely rigid and the control laws are developed based on this hypothesis. The rigidity of the robot can be reinforced by appropriately choosing the building materials, or by treating a posteriori the existing structure. However, due to safety and power consumption constraints, it is often important to keep a light weight, so as the structure strengthening is avoided and the flexibility effects remain significant. Lightweight robots that are used in aerospace [2] and medical [3] applications typically exhibit deformable structures. The distributed deformation is referred to as link-flexibility in the robotics field.

The control of multiple-links flexible manipulators is a difficult problem for several reasons, such as the significance of the nonlinear effects in the dynamic model, the presence of lightly damped oscillatory modes and the underactuated nature of the system (the deformation variables that describe flexibility are neither actuated nor measured). In this context, the end-effector trajectory tracking problem is definitely the hardest. Several works on the subject consider a linearization of the model around a nominal operating condition, or some a priori knowledge of the reference trajectory (see [4], [5] and references therein). Herein, the proposed methodology allows us to deal with the most significant nonlinear effects: the variations of the inertia matrix with respect to the robot configuration and the Coriolis and centripetal torques. The use of deformation sensors such as strain gauges, accelerometers and optical sensors is avoided for the sake of generality of the proposed approach that tackles an output feedback, i. e. a limited information, control problem, in which some of the state variables are not measured.

Our goal, in this paper, is to show some of the potentialities offered by the Linear Parameter Varying (LPV) systems methodology for the control of flexible-link manipulators. Some of the advantages of this methodology are its validity for systems undergoing unstable zero dynamics and its ability to guarantee performance and robustness properties, due to the tuning in the frequency domain. Therefore, LPV methods have become increasingly popular in robotic applications, starting from the results presented in [6], where the Coriolis and centripetal nonlinearities were not considered. More recently, the work in [7] used a conservative affine LPV model which is reduced using parameter set mapping. The work presented in [8] used an identified polynomial LPV model and carried out robust performance analysis using matrix sum of squares relaxations. Herein, an LPV model with rational parametric dependence is considered. This rational dependence is caused by the nonsingular descriptor structure of mechanical systems.

As a first step towards the design parameter-dependent control laws, an appropriate LPV model of the system must be derived. The LPV modeling procedure can be carried out in two ways. 1) A direct identification from experimental data, knowing the structure and the type of the parametric dependence of the LPV model. 2) A reformulation of the nonlinear dynamic model obtained from multibody dynamics. The work presented in this paper follows the second approach. Among the various methods of LPV modeling of physical systems [9], we have adopted the *virtual scheduling* change-of-variable one. It consists in directly reformulating some nonlinear terms that involve the measured states as varying parameters. This is consistent with our application because the involved models have a relatively small number of different types of nonlinearity.

The type of parametric dependence of the state matrices in terms of the varying parameters play a key role in the numerical tractability of the controller synthesis problem. For instance, the rational dependence may lead to an infinite dimensional Linear Matrix Inequality (LMI, see [10]) optimization problem that is difficult to handle numerically. Moreover, considering that the state variables that describe the deformations are unmeasured prevents the direct use of a state feedback controller. In such a situation, one may consider an observer-based control scheme or a dynamic output feedback (DOF) one. The latter has been adopted in our work because in this case, the performance requirements are easier to manage at the controller synthesis stage.

In order to address the above mentioned control design issues and to synthesize LPV-DOF controllers that ensure the internal stability and some performance level over a wide operating range, we propose the two following different approaches:

1. The use of an equivalent affine LPV descriptor model that represents the original rational LPV one.
2. The use of some dilated LMI conditions that exploit the nonsingular descriptor structure of the LPV model.

Both methods finally allow us to simplify the rational dependence of the LMI synthesis conditions and to obtain affine parameter-dependent LMIs (PDLMIs) that can be solved on the vertices of the parametric space.

The paper is organized as follows. The second section introduces some preleminaries and states the addressed problem. Section 3 deals with the nonlinear and the LPV modeling of flexible manipulators. The proposed control strategies are presented in Section 4. Section 5 is devoted to the implementation of the control laws on a system taken from the literature and presents some simulation results. In the sixth section, comparisons with an optimized inversion-based control law are conducted and Section 7 concludes the paper.

Notations - If A and B are symmetric matrices $A > 0$ (resp. ≥ 0) means that A is positive definite (resp. positive semi-definite), A^T is the transpose of A, $\mathrm{He}\{A\} = A + A^T$, $\mathrm{diag}(A, B) = \begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix}$ and $(\star)$ represents the blocks that are induced by symmetry. Matrix dimensions will be omitted if the context allows that.

2. Preliminaries and problem statement

First of all, let us define the following variables:

$\theta(t)$: vector of measured angular positions (rigid coordinates),
$\theta^\star(t)$: vector of joint reference trajectories,
$\delta(t)$: vector of unmeasured deflection variables (flexible coordinates),
$F(t) = [X(t) \quad Y(t)]^T$: vector of end-effector position variables in the plane (Cartesian coordinates),
$F^\star(t)$: vector of end-effector reference positions,
$\mathcal{S}(\theta, \delta, \dot{\theta}, \dot{\delta})$: nonlinear dynamic model of the robot that is established according to the adressed control issue,
$\mathcal{K}(\theta, \dot{\theta})$: nonlinear dynamic controller that is designed in order to achieve the specified control objective,
$\Lambda^{-1}(\theta)$: inverse kinematics of the robot, i. e. the relation between the configuration variables θ and the operational variables F.

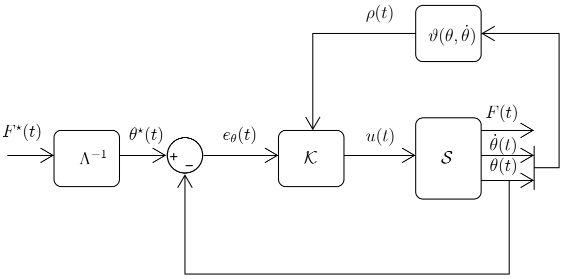

Figure 1. Joint space control scheme

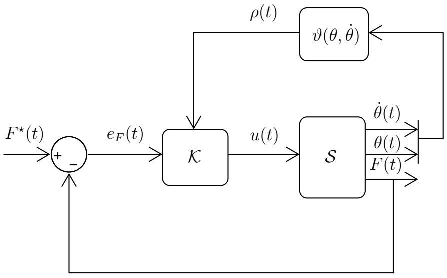

Figure 2. Operational space control scheme

Robot manipulators are generally used in order to perform a task that is often expressed in the operational space (i.e. the end-effector space). As the actuation takes place at the joint level via the use of motors, the task to be performed needs to be translated into the joint space by using the inverse kinematics of the robot. The computation of the joints reference trajectories $\theta^{\star}$ can be performed off-line if the task is known in advance or is repetitive. Otherwise, it must be updated in real-time. In our work, we suppose that the measured variables are the joint angular positions $\theta(t)$ and velocities $\dot{\theta}(t)$, as well as the end-effector position $F(t)$ given by an exteroceptive sensor such as a video camera [11]. Therefore, these variables are available for the computation of the control law. In contrast, the deflection variables $\delta(t)$ and $\dot{\delta}(t)$ are not measured.

According to the previous hypotheses, Figs. 1 and 2 represent the joint space and the operational space control schemes in the case of flexible manipulators. The quasi-LPV controller $\mathcal{K}(\rho)$, where $\rho = \vartheta(\theta,\dot{\theta})$, $\vartheta(.)$ being a bounded multivariate function, actually implements a nonlinear control law. The control input $u(t)$ may be the torques $\tau(t)$ provided by the joint motors, or the joint-velocity reference trajectories $\dot{\theta}^{\star}(t)$ if a low-level velocity controller is already implemented. This is often the case in practice in order to simplify the control issue and reduce the effects of strong nonlinearities such as Coulomb friction. The joint and the end-effector tracking errors are $e_\theta(t) = \theta^{\star}(t) - \theta(t)$ and $e_F(t) = F^{\star}(t) - F(t)$.

The methodology presented in this paper focuses on the control scheme of Fig. 1 while managing the most significant nonlinear effects such as the Coriolis and centripetal effects. Let us note, however, that this methodology can also be applied for direct operational space control, but an increase of the number of state variables and varying parameters is then expected.

3. Modeling of flexible manipulators

Robot manipulators can undergo two main classes of flexibilities [5]: joint flexibility that is concentrated at the joints of the manipulator and may be caused by the use of compliant transmission elements, and link flexibility that is distributed along the mechanical structure and may be caused by the use of lightweight materials or the excitation by high-bandwidth input torques. Our paper focuses on the link flexibility case which is the most challenging one. A robotic manipulator with two joints and a flexible forearm (second link) is considered as a case study.

3.1. *Dynamic modeling*

3.1.1. *Modeling of the bending deflection*

The assumed modes technique is a common approach for the modeling of the deformation occurring in a flexible link. Under the hypothesis of small deformations of pure bending nature, the deformation varies with respect to the position $x_k \in [0,\ l_k]$ in the k^{th} link according to the formula:

$$w_k(x_k,t) = \sum_{i=1}^{N} \phi_{ki}(x_k)\delta_{ki}(t) \tag{1}$$

where l_k is the length of the link, $\phi_{ki}(x_k)$ the shape of its i^{th} flexible mode, $\delta_{ki}(t)$ its time-varying amplitude and N is the number of considered flexible modes per link. The most common expressions of mode shapes are polynomial [12] and trigonometric functions [13]. Their coefficients are determined by imposing some boundary conditions on the deformation. For instance, in the *clamped-free* boundary conditions, the first extremity of the link is fixed the base with a zero tangent slope whereas this slope is nonzero in the *pinned-free* ones. The second extremity of the link is not constrained in both cases.

3.1.2. *Nonlinear dynamic model*

The equations of motion are obtained following the usual Euler-Lagrange approach, leading to the dynamic model:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + Kq + D\dot{q} + g(q) = G\tau \tag{2}$$

This model exhibits a second-order behaviour in terms of the vector of generalized coordinates $q = [\theta^T \ \ \delta^T]^T$. The inertia, stiffness and damping matrices are partitioned following the rigid-body and flexible variables as: $M(q) = \begin{bmatrix} M_{\theta\theta}(q) & M_{\theta\delta}(q) \\ M_{\theta\delta}^T(q) & M_{\delta\delta} \end{bmatrix}$, $K = \begin{bmatrix} 0 & 0 \\ 0 & K_\delta \end{bmatrix}$ and $D = \begin{bmatrix} 0 & 0 \\ 0 & D_\delta \end{bmatrix}$. The Coriolis and the gravitational vectors are partitioned as: $C(q,\dot{q})\dot{q} = \begin{bmatrix} c_\theta(q,\dot{q}) \\ c_\delta(q,\dot{q}) \end{bmatrix}$ and $g(q) = \begin{bmatrix} g_\theta(q) \\ g_\delta(q) \end{bmatrix}$. G is the input matrix that distributes the control torque τ among the generalized coordinates. The state vector of the system can be taken as $x(t) = [q^T(t) \ \ \dot{q}^T(t)]^T$. As $M(q)$ is always invertible (symmetric positive-definite matrix), the nonlinear state-space equations of motion are of the form: $\dot{x} = A(q,\dot{q})x + B(q)\tau$. An alternative modeling method that uses the virtual work principle together with graph theory has been proposed in [12] and implemented in MAPLE [14]. In contrast with the Euler-Lagrange method, the inertia matrix of a model that is obtained by this method is not necessarily symmetric and the bloc $M_{\delta\delta}$ is not necessarily constant.

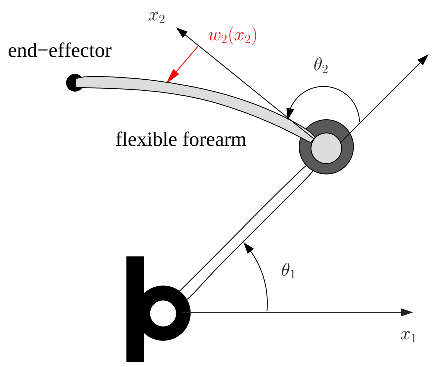

Figure 3. Schematic view of the flexible manipulator

3.2. Case study

The case study discussed in our paper is the model of the FLEXARM robot taken from [15]. A schematic view of this flexible manipulator is displayed in Fig. 3.

3.2.1. Nonlinear model

The considered robot is a planar 2-links manipulator with a flexible forearm. The lengths of its links are: $l_1 = 0.3\,\text{m}$ and $l_2 = 0.7\,\text{m}$. Its nonlinear dynamic model is given by (2), where $q(t) = [\theta_1(t)\,\theta_2(t)\,\delta_{21}(t)\,\delta_{22}(t)]^T$, $\theta_{1,2}(t)$ being the joint angular positions and $\delta_{2j}(t)$, $j = 1, 2$, the amplitudes of the two considered flexible modes with trigonometric shaping functions and clamped-free boundary conditions. The inertia matrix has the following expression:

$$M(q) = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{12} & m_{22} & 0 & 0 \\ m_{13} & 0 & 1 & 0 \\ m_{14} & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where:

$$\begin{aligned} m_{11} &= J_{1t} + J_{2t} + 2h_3 \cos\theta_2 - 2(h_1\delta_{21} + h_2\delta_{22}) \sin\theta_2, \\ m_{12} &= J_{2t} + h_3 \cos\theta_2 - (h_1\delta_{21} + h_2\delta_{22}) \sin\theta_2, \\ m_{13} &= h_1 \cos\theta_2, m_{14} = h_2 \cos\theta_2 \text{ and } m_{22} = J_{2t}. \end{aligned}$$

J_{1t}, J_{2t}, h_1 h_2 and h_3 are the contributions of the different elements (rigid bodies, rigid-flexible couplings) to the moments of inertia.

The Coriolis and centripetal torques are quadratic terms of the time-derivatives of the generalized coordinates. Their expressions can be developed from the inertia matrix $M(q)$ using the Christoffel symbols [16]. The components of the vector $C(q, \dot{q})\dot{q}$ are:

$$\begin{aligned} c_{[1]} &= -(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2)(h_3 \sin\theta_2 + (h_1\delta_{21} + h_2\delta_{22}) \cos\theta_2) \\ &\quad -2(\dot{\theta}_1 + \dot{\theta}_2)(h_1\dot{\delta}_{21} + h_2\dot{\delta}_{21}) \sin\theta_2, \\ c_{[2]} &= \dot{\theta}_1^2(h_3 \sin\theta_2 + (h_1\delta_{21} + h_2\delta_{22}) \cos\theta_2), \\ c_{[3]} &= \dot{\theta}_1^2 h_1 \sin\theta_2, \\ c_{[4]} &= \dot{\theta}_1^2 h_2 \sin\theta_2. \end{aligned}$$

The input matrix is: $G = \begin{bmatrix} I_2 \\ G_\delta \end{bmatrix}$ with $G_\delta = \begin{bmatrix} 0 & \phi'_{21_0} \\ 0 & \phi'_{22_0} \end{bmatrix}$. The damping and stiffness matrices D and K have diagonal structures: $D = \text{diag}(0, 0, d_1, d_2)$ and $K = \text{diag}(0, 0, k_1, k_2)$ and the gravitational torques are zero due to the horizontal workspace of the robot. Moreover, the deflection of the end-effector is given by:

$$y_t(t) = (\frac{\phi_{21_e}}{l_2} - \phi'_{21_0})\delta_{21} + (\frac{\phi_{22_e}}{l_2} - \phi'_{22_0})\delta_{22} \tag{4}$$

where ϕ_{21_e}, ϕ_{22_e}, ϕ'_{21_0} and ϕ'_{22_0} are boundary conditions of the shaping functions $\phi_{21}(x_2)$ and $\phi_{22}(x_2)$ of the flexible forearm. The numerical values of the physical parameters J_{1t}, J_{2t}, h_1, h_2, h_3, ϕ_{21_e}, ϕ_{22_e}, ϕ'_{21_0}, ϕ'_{22_0}, d_1, d_2, k_1 and k_2 are given in [15].

3.2.2. LPV model

In order to obtain a relevant LPV model for system (2), the terms involving the deformation variables δ_{2j}, $j = 1, 2$, are neglected in the inertia matrix. Indeed, their effect is insignificant when they are added to the rigid parts of the moments of inertia. It is not the case in the Coriolis vector where they must be considered. Then, it appears that $M(q)$ and $c(q, \dot{q})$ can be written as a linear combination of the following varying parameters, that involve the measured state variables of the system:

$$\begin{matrix} \rho_1 = \cos(\theta_2), & \rho_2 = \dot{\theta}_2 \sin(\theta_2), & \rho_3 = \dot{\theta}_1 \sin(\theta_2), \\ \rho_4 = \dot{\theta}_2^2 \cos(\theta_2), & \rho_5 = \dot{\theta}_1\dot{\theta}_2 \cos(\theta_2), & \rho_6 = \dot{\theta}_1^2 \cos(\theta_2). \end{matrix} \tag{5}$$

Each parameter ranges between known extremal values, as well as its time-derivative: $\rho_i(t) \in [\underline{\rho}_i,\ \overline{\rho}_i]$ and $\dot{\rho}_i(t) \in [\underline{\omega}_i,\ \overline{\omega}_i]$. These constraints generate the following bounded convex admissible sets for $\rho(t)$ and $\dot{\rho}(t)$:

$$\mathcal{S}_\rho = \{\rho = [\rho_1 \,...\, \rho_6]^T \in \mathbb{R}^6,\ \underline{\rho}_i \le \rho_i \le \overline{\rho}_i,\ i = 1, .., 6\} \tag{6}$$

$$\mathcal{S}_{\dot{\rho}} = \{\dot{\rho} = [\dot{\rho}_1 \,...\, \dot{\rho}_6]^T \in \mathbb{R}^6,\ \underline{\omega}_i \le \dot{\rho}_i \le \overline{\omega_i},\ i = 1, .., 6\} \tag{7}$$

The boundaries of these admissible sets are respectively delimited by the following vertices sets:

$$\mathcal{S}_\rho^\nu = \{\rho = [\rho_1 \,...\, \rho_6]^T \in \mathbb{R}^6,\ \rho_i \in \{\underline{\rho}_i, \overline{\rho}_i\},\ i = 1, .., 6\} \tag{8}$$

$$\mathcal{S}_{\dot{\rho}}^\nu = \{\dot{\rho} = [\dot{\rho}_1 \,...\, \dot{\rho}_6]^T \in \mathbb{R}^6,\ \dot{\rho}_i \in \{\underline{\omega}_i, \overline{\omega}_i\},\ i = 1, .., 6\} \tag{9}$$

Taking the regulated output of the system as the joint angular positions yields the following descriptor model :

$$G(s, \rho) : \begin{cases} M_1(\rho)\dot{x}(t) = A_1(\rho)x(t) + B_1(\rho)u(t) \\ y(t) = \theta(t) = Cx(t) \end{cases} \tag{10}$$

where: $M_1(\rho) = \text{diag}(I_4, M(\rho))$, $C = [I_2\ 0_{2\times6}]$, $B_1(\rho) = [0_{2\times4}\ I_2\ G_\delta^T]^T$ and $A_1(\rho) = \begin{bmatrix} 0_4 & I_4 \\ 0_{4\times2} & S(\rho) \end{bmatrix}$, with

$$S^T(\rho) = \begin{bmatrix} -h_1(2\rho_5 + \rho_4) & h_1\rho_6 & -k_1 & 0 \\ -h_2(2\rho_5 + \rho_4) & h_2\rho_6 & 0 & -k_2 \\ -2h_3\rho_2 & h_3\rho_3 & h_1\rho_3 & h_2\rho_3 \\ -h_3\rho_2 & 0 & 0 & 0 \\ -2h_1(\rho_2 + \rho_3) & 0 & -d_1 & 0 \\ -2h_2(\rho_2 + \rho_3) & 0 & 0 & -d_2 \end{bmatrix}.$$

The parameter-dependent entries of the matrix $S(\rho)$ are simply obtained by factoring the components of the Coriolis vector. For instance, $S_{[3,3]}(\rho) = h_1\rho_3$ is obtained by recognizing that $c_{[3]} = \dot{\theta}_1^2 h_1 \sin\theta_2 = h_1\rho_3\dot{\theta}_1$.

By passing $M_1(\rho)$ through the right-hand side, the model in (10) can be written in the usual state-space form:

$$G(s,\rho) : \begin{cases} \dot{x}(t) = A(\rho)x(t) + B(\rho)u(t) \\ y(t) = Cx(t) \end{cases} \tag{11}$$

where $A(\rho) = M_1^{-1}(\rho)A_1(\rho)$ and $B(\rho) = M_1^{-1}(\rho)B_1(\rho)$.

Due to the inversion of the inertia matrix $M(\rho)$ in the state matrices $A(\rho)$ and $B(\rho)$, the LPV system (11) is rational with a particular structure. Indeed, the state matrices are already factored in a $Denominator^{-1} \times Numerator$ form in a matrix sense.

4. LPV control strategies

4.1. Controller synthesis problem

Let us provide the LPV model $G(s,\rho)$ with a performance channel whose input is $w(t) \in \mathbb{R}^{n_w}$ and output is $z(t) \in \mathbb{R}^{n_z}$ (see e.g. the reference book [17]):

$$G_s(s,\rho) : \begin{cases} \dot{x}(t) = A(\rho)x(t) + B_w w(t) + B(\rho)u(t) \\ z(t) = C_z x(t) + D_{zw} w(t) \\ y(t) = Cx(t) \end{cases} \tag{12}$$

The LPV-DOF control problem consists in synthesizing a controller:

$$K(s,\rho) : \begin{cases} \dot{x}_K(t) = A_K(\rho)x_K(t) + B_K(\rho)y(t) \\ u(t) = C_K(\rho)x_K(t) \end{cases} \tag{13}$$

such that the closed-loop system defined by:

$$G_{Cl}(s,\rho) : \begin{cases} \dot{x}_{Cl}(t) = A_{Cl}(\rho)x_{Cl}(t) + B_{Cl}w(t) \\ z(t) = C_{Cl}x_{Cl}(t) + D_{Cl}w(t) \end{cases} \tag{14}$$

where $x_{Cl} = [x^T \ x_K^T]^T$, is asymptotically stable and satisfies a level of $\mathcal{L}_2$-gain performance over the parametric set $\mathcal{S}_\rho$.

The $\mathcal{L}_2$-gain of the closed-loop system is defined as:

$$\|G_{Cl}(s,\rho)\|_{\mathcal{L}_2} = \sup_{w \neq 0, \rho \in \mathcal{S}_\rho} \frac{\left(\int_0^{+\infty} \|z(t)\|^2 \, dt\right)^{\frac{1}{2}}}{\left(\int_0^{+\infty} \|w(t)\|^2 \, dt\right)^{\frac{1}{2}}} \tag{15}$$

4.2. Equivalent affine LPV descriptor representation

It has been stated in [18] that any rational LPV system can be expressed in a descriptor form with affine state matrices. Let us summarize, hereafter, some technical aspects of the method.

4.2.1. Modeling

In the general case, a rational LPV sytem is described by the state-space equation given in (11), in which the state matrices $A(\rho)$, $B(\rho)$ and potentially the output matrix $C(\rho)$ are rational functions of the parameter vector $\rho(t)$. Transforming model (11) into an affine descriptor model can be performed either by making some changes of variables involving the parameter-dependent entries of the state matrices (*ad-hoc* methods), or systematically by using a Linear Fractional Representation (LFR) modeling of the system. The LFR model itself can be obtained by manual calculations or by using available numerical software [19]. It is characterized by the feedback interconnection of a linear time-invariant (LTI) system and a parameter-dependent matrix $\Delta(\rho)$:

$$\begin{bmatrix} \dot{x}(t) \\ z(t) \\ y(t) \\ z_d(t) \end{bmatrix} = \begin{bmatrix} N_{11} & N_{12} & N_{13} & N_{14} \\ N_{21} & N_{22} & N_{23} & N_{24} \\ N_{31} & N_{32} & N_{33} & N_{34} \\ N_{41} & N_{42} & N_{43} & N_{44} \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \\ u(t) \\ v(t) \end{bmatrix} \quad \text{and} \quad v(t) = \Delta(\rho)z_d(t) \tag{16}$$

where $x(t) \in \mathbb{R}^r$ is the state, $z_d(t) \in \mathbb{R}^{n_{z_d}}$ and $v(t) \in \mathbb{R}^{n_v}$ are the input and the output signals of the parametric block $\Delta(\rho)$. N_{ij}, $i,j = 1..3$ are constant matrices of appropriate dimensions. In our application, the matrices: N_{14}, N_{23}, N_{24}, N_{32}, N_{33} and N_{34} are zero. Taking the augmented state vector as $x_d(t) = [x(t)^T \ v(t)^T]^T$ yields the affine LPV descriptor (singular) model [20]:

$$G_{ds}(s,\rho) : \begin{cases} E\dot{x}_d(t) = A_d(\rho)x_d(t) + B_w w(t) + B_d u(t) \\ z(t) = C_z x_d(t) + D_{zw} w(t) \\ y(t) = C_d x_d(t) \end{cases} \tag{17}$$

where:
$E = \text{diag}(I_r, 0_{n_v})$, $A_d(\rho) = \begin{bmatrix} N_{11} & 0 \\ \Delta(\rho)N_{41} & \Delta(\rho)N_{44} - I_{n_v} \end{bmatrix}$,
$B_w(\rho) = \begin{bmatrix} N_{12} \\ \Delta(\rho)N_{42} \end{bmatrix}$, $B_d(\rho) = \begin{bmatrix} N_{13} \\ \Delta(\rho)N_{43} \end{bmatrix}$, $C_z = [N_{21} \quad 0]$, $C_d = [N_{31} \quad 0]$ and $D_{zw} = N_{22}$. Note that the alternative choice: $x_d(t) = [x^T(t) \ v^T(t) \ u^T(t)]^T$ for the generalized state vector yields a constant B_d matrix.

4.2.2. $\mathcal{L}_2$-gain control

The following theorem gives some controller synthesis conditions for the affine LPV descriptor model (17).

Theorem 1. *[18] The closed-loop system is stable and the $\mathcal{L}_2$-gain from $w(t)$ to $z(t)$ is less than $\gamma > 0$ if the LMIs* (18)-(19) *have a solution $p = \{X, Y, F, G, H\}$, $\forall \rho \in \mathcal{S}_\rho^\nu$:*

$$\begin{bmatrix} YE^T & E \\ E^T & E^T X \end{bmatrix} = \begin{bmatrix} YE^T & E \\ E^T & E^T X \end{bmatrix}^T \geq 0 \tag{18}$$

$$\begin{bmatrix} M_A(\rho) + M_A^T(\rho) & M_B & M_C^T \\ M_B^T & -\gamma I & D_{zw}^T \\ M_C & D_{zw} & -\gamma I \end{bmatrix} < 0 \tag{19}$$

where:
$M_A(\rho) = \begin{bmatrix} A_d(\rho)Y^T + BF^T & A_d(\rho) \\ H^T & X^T A_d(\rho) + G^T C \end{bmatrix}$,
$M_B = \begin{bmatrix} B_w \\ X^T B_w \end{bmatrix}$ *and* $M_C = \begin{bmatrix} C_z Y^T & C_z \end{bmatrix}$.

The LPV controller is directly obtained in the readily implementale regular state-space form (13) (see [18]). The synthesis conditions of Theorem 1 are obtained based on a generalized version of the bounded real lemma theorem [10] to descriptor systems [21].

4.3. Dilated LMI conditions and structure constraints

The dilated LMI conditions involve, besides the usual Lyapunov, controller and plant matrices, some additional matrix variables referred to as slack variables. These variables offer additional degrees of freedom for the analysis and the synthesis problems. For instance, parameter-dependent Lyapunov functions (PDLF) can be considered, which results in a reduction of the conservatism caused by the use of a single Lyapunov function, as explained in [22].

The following $\mathcal{L}_2$-gain analysis conditions, given in [23] are based on the dilated LMI conditions proposed in [24].

Theorem 2. *The closed-loop system* (14) *is stable and the $\mathcal{L}_2$-gain from $w(t)$ to $z(t)$ is less than $\gamma > 0$ if $\exists P(\rho) = P^T(\rho)$ (Lyapunov matrix) and $\exists V$ (slack variable) solutions of the LMI* (20), $\forall(\rho, \dot{\rho}) \in \mathcal{S}_\rho \times \mathcal{S}_{\dot{\rho}}$:

$$\begin{bmatrix} -(V+V^T) & (\star) & (\star) & (\star) & (\star) \\ A_{Cl}(\rho)V + P(\rho) & -P(\rho) + \dot{P}(\rho) & (\star) & (\star) & (\star) \\ C_{Cl}V(\rho) & 0 & -\gamma I & (\star) & (\star) \\ 0 & B_{Cl}^T & D_{Cl}^T & -\gamma I & (\star) \\ V & 0 & 0 & 0 & -P(\rho) \end{bmatrix} < 0 \qquad (20)$$

We are now ready to introduce our main synhesis result, that is stated in the following theorem.

Theorem 3. *There exists a controller* (13) *that stabilizes the closed-loop system* (14) *and achieves an $\mathcal{L}_2$-gain less than $\gamma > 0$ if the LMI* (21) *has a solution $\Phi = \{\hat{V}_{11}, \hat{V}_{12}$ $\hat{W}_{11}, \hat{A}_K, \hat{B}_K, \hat{C}_K, \hat{U}(\rho), \hat{P}(\rho)\}, \forall \rho \in \mathcal{S}_\rho$:*

$$\begin{bmatrix} -\mathrm{He}\{M_1(\rho)\hat{V}_{11}\} & (\star) & (\star) & (\star) & (\star) & (\star) \\ -(\hat{U}(\rho) + M_1(\rho)) & -\mathrm{He}\{M_1(\rho)\hat{W}_{11}\} & (\star) & (\star) & (\star) & (\star) \\ \multicolumn{2}{c}{\begin{bmatrix} \alpha_1(\rho) & \alpha_2(\rho) \\ \alpha_3(\rho) & \alpha_4(\rho) \end{bmatrix} + \hat{P}(\rho)} & -\hat{P}(\rho) & (\star) & (\star) & (\star) \\ C_z\hat{V}_{11} & C_z(I + \hat{V}_{12}) & 0 & -\gamma I & (\star) & (\star) \\ 0 & 0 & \beta(\rho) & D_{zw}^T & -\gamma I & (\star) \\ \begin{matrix} M_1(\rho)\hat{V}_{11} \\ \hat{U}(\rho) \end{matrix} & \begin{matrix} M_1(\rho) \\ \hat{W}_{11}^T M_1(\rho) \end{matrix} & 0 & 0 & 0 & -\hat{P}(\rho) \end{bmatrix} < 0 \qquad (21)$$

where: $\hat{P}(\rho) = \hat{P}^T(\rho)$, $\alpha_1(\rho) = A_1(\rho)\hat{V}_{11} + B_1\hat{C}_K M_1(\rho)$, $\alpha_2(\rho) = A_1(\rho)$, $\alpha_3(\rho) = \hat{A}_K M_1(\rho)$, $\alpha_4(\rho) = \hat{W}_{11}A_1(\rho) + \hat{B}_K C$ and $\beta(\rho) = B_w^T M_1(\rho) \left[I \ \hat{W}_{11}\right]$.

The state matrices of the controller are then given by: $B_K(\rho) = W_{21}^{-T}(\rho)\hat{B}_K$; $C_K(\rho) = \hat{C}_K V_{21}^{-1}(\rho)$; and $A_K(\rho) = W_{21}^{-T}(\rho)[\hat{A}_K - \hat{W}_{11}^T A_1(\rho)\hat{V}_{11}M_1^{-1}(\rho) - W_{21}^T B_K(\rho)C\hat{V}_{11}M_1^{-1}(\rho) - \hat{W}_{11}^T B_1 C_K(\rho)V_{21}(\rho)]V_{21}^{-1}(\rho)$.

Proof: Starting from the following partitioning of the slack variable and its inverse, according to the dimensions of the system and the controller:

$$V = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} \text{ and } W = V^{-1} = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix},$$

some linearizing projections are defined as:

$$\Pi_V = \begin{bmatrix} V_{11} & I \\ V_{21} & 0 \end{bmatrix} \text{ and } \Pi_W = \begin{bmatrix} I & W_{11} \\ 0 & W_{21} \end{bmatrix}.$$

Let us now introduce the following transformation matrix $T(\rho)$, where $M_2(\rho) = \mathrm{diag}(M_1(\rho), I_8)$:

$$T(\rho) = \begin{bmatrix} M_2(\rho)\Pi_W^T & 0 & 0 & 0 \\ 0 & M_2(\rho)\Pi_W^T & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & M_2(\rho)\Pi_W^T \end{bmatrix} \qquad (22)$$

Keep also in mind that the closed-loop state matrices in (14) are partitioned as: $A_{Cl}(\rho) = \begin{bmatrix} A(\rho) & B(\rho)C_K(\rho) \\ B_K(\rho)C & A_K(\rho) \end{bmatrix}$, $B_{Cl}(\rho) = \begin{bmatrix} B_w \\ 0 \end{bmatrix}$, $C_{Cl}(\rho) = \begin{bmatrix} C_z^T \\ 0 \end{bmatrix}^T$.

The synthesis conditions of Theorem 3 are obtained by left multiplying the analysis inequality (20) by $T(\rho)$ and right multiplying by its transpose (congruence transformation), then making the changes of variables: $\hat{B}_K = W_{21}^T B_K(\rho)$, $\hat{C}_K = C_K(\rho)V_{21}$, $\hat{A}_K(\rho) = [W_{11}^T A(\rho)V_{11} + W_{11}^T B(\rho)C_K(\rho)V_{21} + W_{21}^T B_K(\rho)CV_{11} + W_{21}^T A_K(\rho)V_{21}]M_1(\rho)$, $\hat{V}_{11} = V_{11}M_1(\rho)$, $\hat{V}_{21} = V_{12}W_{21}$, $\hat{U} = M_1(\rho)U$ and $\hat{P} = M_2(\rho)\Pi_W^T P \Pi_W M_2(\rho)$, with $U = W_{11}^T V_{11} + W_{21}^T V_{21}$ and P is the closed-loop Lyapunov matrix. The problem is linear in terms of the decision variables due to the relations: $W\Pi_V = \Pi_W$ and $V\Pi_W = \Pi_V$.

Remark 1. From the solution of the LMI (21), the matrices W_{21} and V_{21} used in the controller formula are obtained by performing a factorization of the right-hand side of $W_{21}^T V_{21} = M_1^{-1}(\rho)\hat{U}(\rho) - \hat{W}_{11}^T M_1(\rho)\hat{V}_{11}M_1^{-1}(\rho)$.

Remark 2. The analysis conditions of Theorem 2 are independent from the type of the parametric dependence of the system $G_{Cl}(s, \rho)$ in (14). If the state matrix $A_{Cl}(\rho)$ is affine, it is equivalent to solve the PDLMI (20) on the set of vertices $\mathcal{S}_\rho^v \times \mathcal{S}_{\dot{\rho}}^v$.

Remark 3. In order to facilitate the numerical tractability of the controller synthesis conditions of Theorem 3, the parameter-dependent matrices $\hat{U}(\rho)$ and $\hat{P}(\rho)$ are set as affine. Therefore, the PDLMI (21) is solved on the set of vertices $\mathcal{S}_\rho^v$.

5. Implementation

5.1. Control issue

The descriptor and the dilated LMI controller synthesis methods have been implemented in simulation for the control of the FLEXARM robot. In particular, the control scheme presented in Fig. 4, that contains a 2-blocks performance channel, has been adopted. The vector of external inputs consists of the joint references $w(t) = \theta^\star(t)$. The vector of controlled outputs, $z(t) = [z_1^T(t) \ z_2(t)]^T$, contains the weighted tracking errors $e_\theta^T(t)$ and tip deflection $y_t(t)$ signals. Static or dynamic weighting functions $W_k(s)$ are used to tune the behavior of the closed-loop system but are not included in the control law.

5.2. Simulation results

Time simulations have been carried out in order to evaluate the proposed control methods. Let us define the following limits that generate the admissible set $\mathcal{S}_\rho$: $|\rho_1| \leq 1$ (by definition of the cosine function), $|\rho_2|, |\rho_3| \leq$

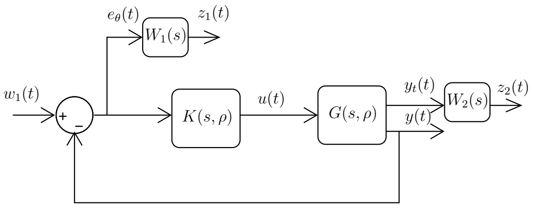

Figure 4. H_∞ control scheme

0.3 rad/s, and $|\rho_4|$, $|\rho_5|$, $|\rho_6| \leq 0.045\,(\mathrm{rad/s})^2$. These bounds are set so as to guarantee some relatively high performance levels. While the stability of the closed-loop system is ensured over the whole admissible set, the performance indices presented in the paper are guaranteed for $|\rho_1| \leq 0.5$. Let us point out that the LPV controllers can operate outside the admissible parametric set, but a degradation of the performance is exepcted in this case. In order to provide a tighter estimate of the actual operating range, it is usual to perform an a posteriory performance analysis, by fixing the controller matrices and computing the performance level for different parametric sets.

The control problem of Fig. 4 has been addressed using both proposed LPV methods. The LMI conditions (18)-(19) and (21) have been solved on the vertices set $\mathcal{S}_\rho^\nu$ using the solver SeDuMi [25] with the YALMIP interface [26]. When considering the whole performance channel with the descriptor method, the obtained performance index $\gamma = 17$ is relatively high. However, constraining the upper block of the performance channel $S_{11}(s) = T_{w_1 \to z_1}$ alone gives satisfactory simulation results. This fact can be explained by the generality and the conservatism of the descriptor method that can handle any rational LPV system regardless of its structure.

In the sequel, we present some simulation results using the weighting functions $W_1(s) = 1$ and $W_2(s) = 0.1$. Static functions are employed herein so as to avoid an increase of the order of the system. The obtained performance indices are $\gamma = 2.37$ and $\gamma = 2.10$ for the descriptor and the dilated LMI synthesis conditions, which guarantees a modulus margin of $\frac{1}{\gamma} = 0.422$ and 0.476 respectively. The number of decision variables of each LMI problem is the following: 253 variables (156 Lyapunov-type + 96 controller-type + 1 linear objective) for the descriptor method and 473 (120 Lyapunov + 96 controller + 256 slack variable + 1 linear objective) for the dialted LMI method. Figs. 5 and 6 display the realized frequency transfers $S_{11}(s)$ and $S_{12}(s) = T_{w_1 \to z_2}$ in each input-output block of the performance channel. Each transfer is compared to the relevant frequency template $\gamma / W_k(s), k = 1, 2$. Notice that, even if the weighting functions are constant, the sensitivity function $S_{11}(s)$ exhibits the suitable high-pass behaviour. Four frozen values of the parameters vector, selected among the $2^6 = 64$ vertices of the admissible set, are considered: $\rho_{\{1\}} = [\underline{\rho}_1 \ldots \underline{\rho}_6]^T$, $\rho_{\{32\}} = [\underline{\rho}_1\, \overline{\rho}_1 \ldots \overline{\rho}_6]^T$, $\rho_{\{33\}} = [\overline{\rho}_1\, \underline{\rho}_1 \ldots \underline{\rho}_6]^T$ and $\rho_{\{64\}} = [\overline{\rho}_1 \ldots \overline{\rho}_6]^T$.

The joint reference signals $\theta_1^\star(t)$ and $\theta_2^\star(t)$ are designed so as to assess the accuracy of the tracking and the decoupling of the two joints over the whole operating range. Smooth signals are involved in order to meet

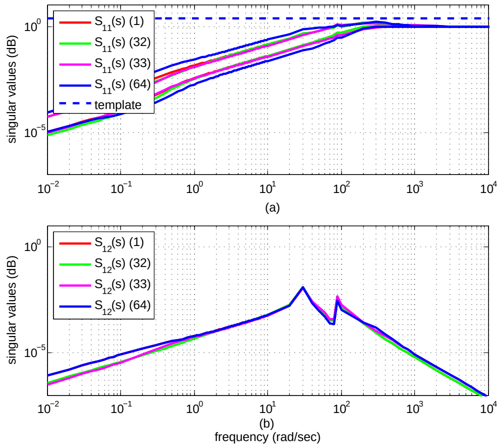

Figure 5. Realized frequency transfers (descriptor method)

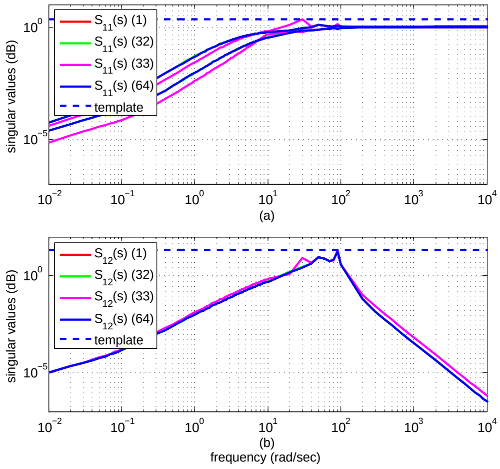

Figure 6. Realized frequency transfers (dilated LMI method)

the limitations on the parameters. Figs. 7-(a) and 8-(a) show the time-responses of the closed-loop system obtained with the original nonlinear model and the LPV controller. Figs. 7-(b) and 8-(b) display the corresponding trajectory tracking errors. Figs. 9 and 10 present the time evolution of the varying parameters for both methods. The presented figures demonstrate the effectiveness of the LPV methods in preserving the closed-loop system stability and performances over a wide operating range.

In addition to the previous control theory-oriented test, a practice-oriented simulation test has been made in order to assess the trajectory tracking in the Cartesian space. The end-effector of the robot has to follow a rectangular trajectory $F^\star(t) = (X^\star(t), Y^\star(t))$, starting from the position $(X, Y) = (0.2121\,\mathrm{m}, 0.9121\,\mathrm{m})$ that corresponds to the unstable equilibrium position $(\theta_1, \theta_2) = (45°, 45°)$. Figs. 11-(a) and 11-(b) show the time tracking of the X and Y coordinates respectively. Fig. 11-(c) shows the tracking in the Cartesian plane (X, Y). The subscripts 1 and 2 in the legends indicate the descriptor and the dilated LMI method respectively.

Remark 4. When a min-max description of the parametric space is used, as in (6)-(7), the number of LMI constraints that need to be solved increases exponentially with the

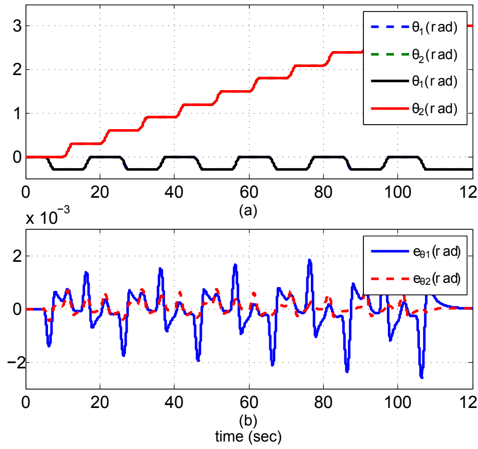

Figure 7. Tracking of the reference trajectories (descriptor method)

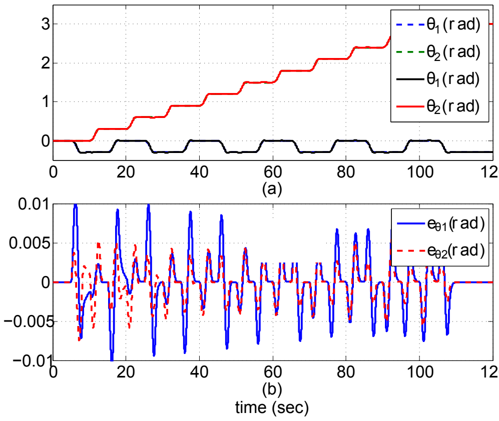

Figure 8. Tracking of the reference trajectories (dilated LMI method)

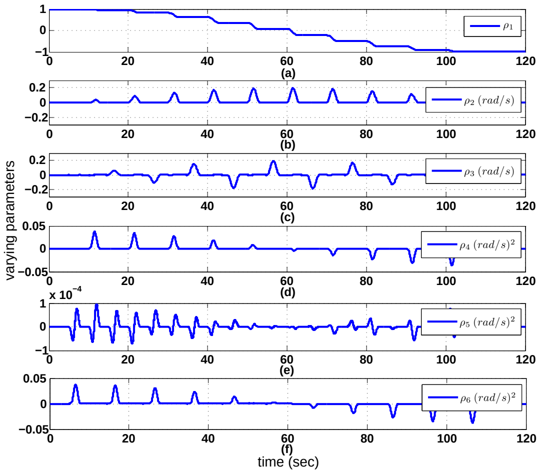

Figure 9. Evolution of the parameters (descriptor method)

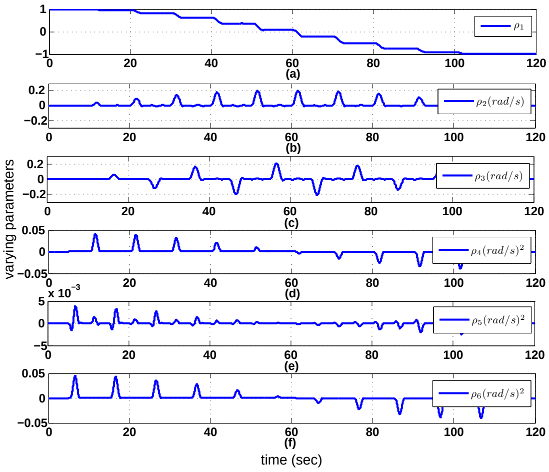

Figure 10. Evolution of the parameters (dilated LMI method)

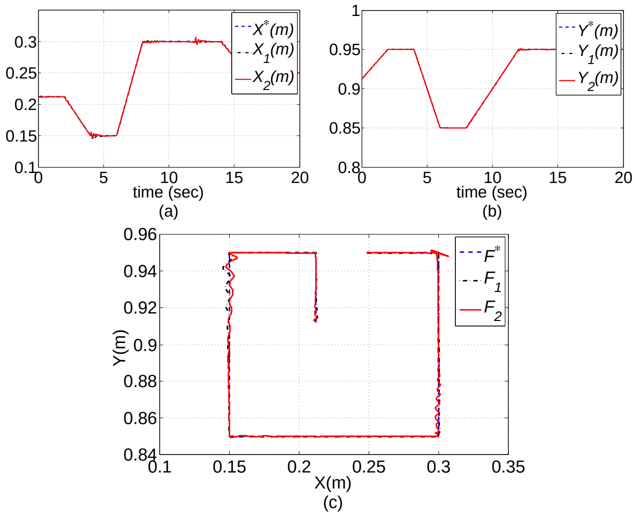

Figure 11. Trajectory tracking in the Cartesian space - (a): X coordinate, (b): Y coordinate, (c): (X, Y)-plane

number of varying parameters. This fact may constitute a limitation for systems with a large number of parameters. Nevertheless, this issue can be addressed by following an alternative representation of the parametric space, such as the use of semialgebraic sets together with a generalized version of the S-procedure theorem [8]. This approach leads to a single parameter-independent LMI of larger size.

6. Comparison with a nonlinear inversion-based control approach

The nonlinear inversion-based control method is very popular in robotics. In the following, we present a comparison of this method with the proposed LPV dilated LMI method.

6.1. Basic knowledge on inverse dynamics control

The principle of the method is to apply a nonlinear state feedback $\tau = \psi(q, \dot{q})$ in equation (2) in order to obtain a linear and decoupled tracking error dynamics (see e.g. [16]). The control input may be taken as:

$$\tau = M(q)\nu + c(q, \dot{q}) + Kq + D\dot{q} \tag{23}$$

A simple choice for the auxiliary control input ν is:

$$\nu = \ddot{q}^\star + K_p(q^\star - q) + K_d(\dot{q}^\star - \dot{q}) \tag{24}$$

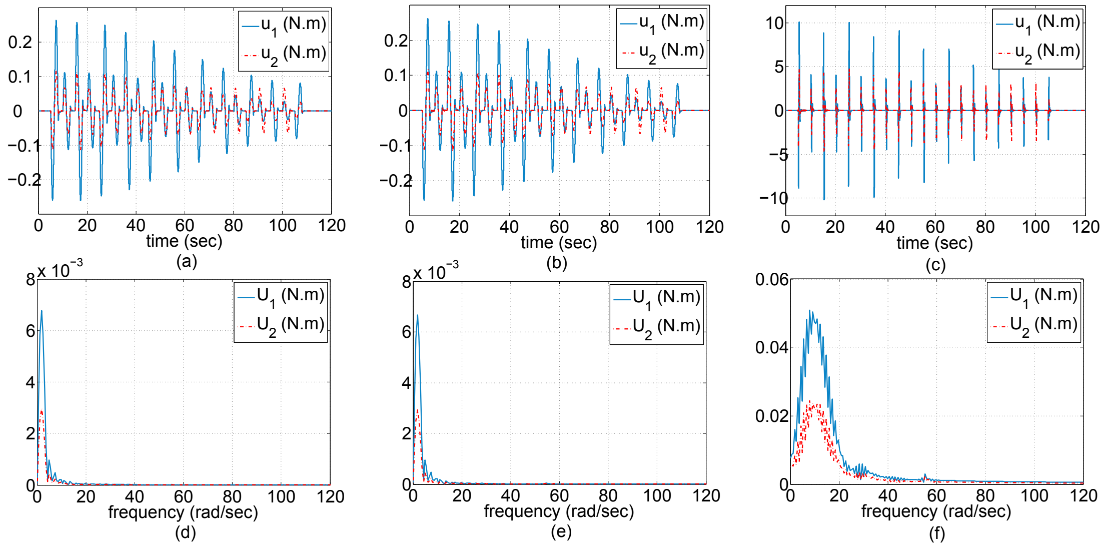

Figure 12. Robustness evaluation of the LPV control method: (a) & (d): velocity noise, (b) & (e): model uncertainty, (c) & (f): high-frequency inputs

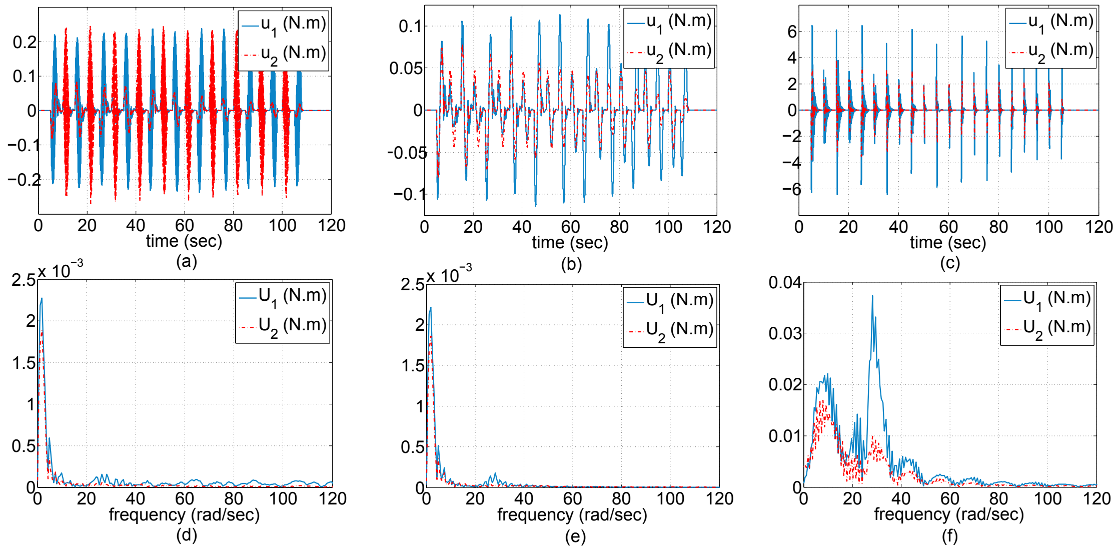

Figure 13. Robustness evaluation of the inversion-based control method

where $q^\star$ is the vector of reference trajectories of the generalized coordinates. Following (23)-(24), the dynamics of the tracking error $e_q = q^\star - q$ is:

$$\ddot{e}_q + K_p e_q + K_d \dot{e}_q = 0 \tag{25}$$

The diagonal structure of matrices K_p and K_d results in a decoupling of the error dynamics and their positive definiteness ensures the asymptotic cancellation of e_q. Robustness with respect to model uncertainties has been addressed in [16, 27].

6.2. Extension to flexible manipulators

The inversion-based control scheme described in (23)-(24) is built on the assumption that the state vector $x = [q^T \quad \dot{q}^T]^T$ is entirely measurable. However, in the case of flexible manipulators, such a control structure is not exploitable because the deformation variables δ and $\dot{\delta}$ are not available for feedback. This issue has been addressed in [13] where a feedforward-feedback control structure has been presented.

The feedforward part τ_N of the control is the torque needed to exactly reproduce the joint reference trajectory $\theta^\star(t)$ based on a perfect knowledge of the model of the robot. The feedback part τ_{PD}, e.g. of proportional and derivative (PD) nature as in (24), is used to provide some robustness against model uncertainties. The overall control law is then:

$$\tau = \tau_N + \tau_{PD} \tag{26}$$

Let us describe more precisely this control law. The presented developments are borrowed from [13]. Setting $g(q) = 0$ and $M_{\delta\delta} = I$ in (2), the deformation variable equation becomes:

$$\ddot{\delta} = G_\delta \tau - n_\delta - M_{\theta\delta}^T \ddot{\theta} \tag{27}$$

where $n_\delta = c_\delta + K_\delta \delta + D_\delta \dot{\delta}$.

Replacing this expression of $\ddot{\delta}$ in the rigid part of equation (2) gives:

$$\tau = H^{-1}[(M_{\theta\theta} - M_{\theta\delta} M_{\theta\delta}^T)\ddot{\theta} + c_\theta - M_{\theta\theta} n_\delta] \tag{28}$$

where $H = I - M_{\theta\delta}G_{\delta}$.

Let us now follow a model inversion approach and replace $\ddot{\theta}$ in (28) by the auxiliary input:

$$\nu = \ddot{\theta}^{\star} + K_p(\theta^{\star} - \theta) + K_d(\dot{\theta}^{\star} - \dot{\theta}) \tag{29}$$

Therefore, the control torque in (28) takes the composite form in (26), where:

$$\tau_N = H^{-1}[(M_{\theta\theta} - M_{\theta\delta}M_{\theta\delta}^T)\ddot{\theta}^{\star} + c_{\theta} - M_{\theta\theta}n_{\delta}]\Big|_{\substack{\theta=\theta^{\star},\dot{\theta}=\dot{\theta}^{\star}\\ \delta=\delta^{\star},\dot{\delta}=\dot{\delta}^{\star}}} \tag{30}$$

$$\tau_{PD} = \mathcal{K}_p(t)(\theta^{\star} - \theta) + \mathcal{K}_p(t)(\dot{\theta}^{\star} - \dot{\theta}) \tag{31}$$

with:

$$\mathcal{K}_p(t) = H^{-1}(M_{\theta\theta} - M_{\theta\delta}M_{\theta\delta}^T)\Big|_{\substack{\theta=\theta^{\star}\\ \delta=\delta^{\star}}} K_p \tag{32}$$

$$\mathcal{K}_d(t) = H^{-1}(M_{\theta\theta} - M_{\theta\delta}M_{\theta\delta}^T)\Big|_{\substack{\theta=\theta^{\star}\\ \delta=\delta^{\star}}} K_d \tag{33}$$

The feedforward part τ_N of the control law depends only on the reference trajectories and the model of the system. The rigid-body reference signals $\theta^{\star}$, $\dot{\theta}^{\star}$ and $\ddot{\theta}^{\star}$ are set by the user. On the other hand, the reference values of the deformation variables $\delta^{\star}$, $\dot{\delta}^{\star}$ are obtained by numerically integrating equation (27) in perfect tracking conditions, i.e. $\theta(t) = \theta^{\star}(t)$. Due to the replacement of the unmeasured states $\delta(t)$ and $\dot{\delta}(t)$ by their nominal values $\delta^{\star}(t)$ and $\dot{\delta}^{\star}(t)$ in (32)-(33), this control method actually performs an approximate inversion of the model of the system. The remaining synthesis issue consists in finding appropriate gains K_p and K_d in (32)-(33).

6.3. Comparative study

It is difficult to make a fair comparison between two methods that are so different in nature. In the comparison presented in the following, strong efforts have been made in order to obtain the best results with the inversion-based method. Constant feedback gains K_p and K_d can be designed so as to achieve some performance properties. For instance, the pole placement technique is used to set the damping ratio and the natural frequency of the tracking error dynamics. For our application, this tuning yielded degraded performances compared to the LPV method. Better results were obtained using recently developed tools that are based on nonsmooth optimization [28]. They allow synthesizing structured and fixed-order controllers for linear systems. The main tool is the `hinfstruct` function of MATLAB [29], devoted to H_∞ synthesis. We have performed a synthesis using a linearized model of the system around the nominal value $x = 0$ of the state vector. We minimzed the H_∞ norm of the sensitivity function weighted by a filter that imposes the following performance features: a modulus margin of 0.8, a bandwidth of 1 rad/s and a static position error of 5%. The obtained performance index is $\gamma_{\text{NS}} = 1.48$ and the feedback gains are: $K_p = \text{diag}(18.75, 19.55)$ and $K_d = \text{diag}(3.70, 4.60)$.

Both the LPV and the inversion-based controllers exhibit a satisfactory behaviour in nominal operating conditions (as described in Section 5.2). However, the LPV controller has proven to be more robust in challenging ones. Among the various comparative tests conducted, the results obtained in the three following situations allowed some discrimination.

1. A uniformly distributed pseudo-random noise on the joint velocity measurement whose amplitude is ±25% of the nominal value.
2. Faster steps for the reference inputs $\theta^{\star}$: settling time of 0.5 s instead of 3 s.
3. A time-varying uncertainty on the moments of inertia of 15% of their nominal value. This uncertainty can for instance represent an unknown time-varying load. It is implemented in simulation as a pseudo-random noise.

The columns of Figs. 12-13 gather the results obtained in these three operating conditions respectively. Their rows represent the time and frequency-domain (Fast Fourier Transform) descriptions of the control input $u(t) = \tau(t)$. While the tracking of the reference trajectories is accurate with both control methods, the figures show that the control input delivered by the LPV controller is much smoother in all cases, whereas a significant oscillating behaviour is observed with the inversion-based controller. Fig. 13-e shows that the application of a high-frequency input with this controller excites the resonances of the flexible modes, unlike the LPV controller (Fig. 12-e) that significantly attenuates them.

Further analysis results have been obtained for the three tests. In Table 1, the $\mathcal{L}_2$-norms of the joint tracking errors $e_{\theta_1}(t) = \theta_1^{\star}(t) - \theta_1(t)$ and $e_{\theta_2}(t) = \theta_2^{\star}(t) - \theta_2(t)$ are compared for the LPV and the nonlinear inversion-based methods. The values are also compared with those obtained in nominal operating conditions. In the first column, we notice that the nonlinear controller provides an accurate trajectory tracking for the two joints. Indeed, the inversion control method aims at exactly replicating the reference signals by the output of the system. However, we observe from the three last columns that the accuracy of the tracking is much more sensitive to the deterioration of the operating conditions than with the LPV controller. In particular, we note from the second and the third columns that the $\mathcal{L}_2$-norms of the tracking errors of the LPV controller do not vary in the presence of velocity noise and model uncertainty, whereas they increase significantly with the nonlinear controller. The last column contains a close value for the first joint with both methods and a lower one for the second joint with the nonlinear controller. Nevertheless, as pointed out in the previous paragraph, the achievement of this performance by the nonlinear controller involves an oscillatory control torque that is not suitable in practice. Our interpretation of these results is that the LPV controller achieves a compromise between the performance (in particular the accuracy of the tracking) and the robustness (sensitivity to the operating conditions), whereas the inversion-based controller focuses on the peformance only. All simulation results presented in the paper have been carried out using a fixed sampling time of $T_s = 1$ ms.

7. Conclusion

In this paper, we propose a methodology for the modeling and the control of flexible robot manipulators using

Control method	Nominal conditions	Velocity noise	Model uncertainty	High-freq. inputs
LPV	0.9807	0.9807	0.9370	3.1947
	0.8105	0.8105	0.7820	3.1028
NL	0.3426	1.3756	1.4452	3.2267
	0.1611	0.2686	0.2806	0.9918

Table 1. Comparison of the $\mathcal{L}_2$-norms of the tracking errors

the Linear Parameter Varying (LPV) systems framework. Smooth nonlinear effects of the dynamic model are handled following a quasi-LPV modeling approach. Two synthesis methods that deal with the rational parametric dependence of the system are investigated, namely the descriptor systems method and the dilated LMI method. Comparisons with a nonlinear inversion-based control technique are provided in the presence of measurement noise, high-frequency inputs and model uncertainty. They illustrate the effectiveness and the robustness of the proposed LPV control methodology.

8. References

[1] B. Siciliano and O. Khatib. *Springer Handbook of Robotics*. Springer-Verlag, Berlin, 2008.

[2] G. Hirzinger, N. Sporer, M. Schedl, J. Butterfass, and M. Grebenstein. Torque-controlled lightweight arms and articulated hands: Do we reach the technological limits now ? *Int. J. Robotics Research*, 23(4–5):331–340, 2004.

[3] U. Hagn, M. Nickl, S. Jörg, G. Passig, T. Bahls, A. Nothhelfer, F. Hacker, L. Le-Tien, A. Albu-Schäffer, R. Konietschke, M. Grebenstein, R. Warpup, R. Haslinger, M. Frommberger, and G. Hirzinger. The DLR MIRO: a versatile lightweight robot for surgical applications. *Industrial Robot*, 35(4):324–336, 2008.

[4] C. Canudas de Wit, B. Siciliano, and G. Bastin. *Theory of Robot Control*. Springer, London, 1996.

[5] A. De Luca and W. Book. *Robots with Flexible Elements*, chapter B.13, pages 287–319. Springer-Verlag, Berlin, 2008. in B. Siciliano and O. Khatib, *Springer Handbook of Robotics*.

[6] R.J. Adams, P. Apkarian, and J.P. Chrétien. Robust control approaches for a two-link flexible manipulator. In *Int. Conf. Dynamics Contr. Structures in Space*, Cranfield, UK, 1996.

[7] S. M. Hashemi, H. S. Abbas, and Werner H. Low-complexity linear parameter-varying modeling and control of a robotic manipulator. *Contr. Engineering Practice*, 20(3):248–257, 2012.

[8] H. Halalchi, E. Laroche, and G. I. Bara. A polynomial LPV approach for flexible robot end-effector position controller analysis. In *Proc. IEEE Conf. Decision Control*, Orlando, FL, USA, 2011.

[9] R. Tóth. *Modeling and Identification of Linear Parameter-Varying Systems*. Springer, Berlin, 2010.

[10] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM Studies in Applied Mathematics, Philadelphia, 1994.

[11] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. Robot. Autom.*, 8(3):313–326, 1992.

[12] P. Shi, J. McPhee, and G.R Heppler. A deformation field for Euler-Bernoulli beams with applications to flexible multibody dynamics. *Multibody System Dynamics*, 5:79–104, 2001.

[13] A. De Luca and B. Siciliano. Closed-form dynamic model of planar multilink lightweight robots. *IEEE Trans. Syst., Man, Cybern.*, 21(4):826–839, 1993.

[14] P. Shi, J. McPhee, and G.R Heppler. *DynaFlex User's Guide*. Systems Design Engineering, University of Waterloo, 2002.

[15] A. De Luca, L. Lanari, P. Lucibello, S. Panzieri, and G. Ulivi. Control experiments on a two-link robot with a flexible forearm. In *Proc. IEEE Conf. Decision Control*, Honolulu, HI, USA, 1990.

[16] M. W. Spong, Hutchinson S., and M. Vidyasagar. *Robot Modeling and Control*. Wiley and Sons, 2005.

[17] S. Skogestad and I. Postlethwaite. *Multivariable feedback control - Analysis and desing*. Wiley and Sons, 2005.

[18] I. Masubuchi, J. Kato, M. Saeki, and A. Ohara. Gain-scheduled controller design based on descriptor representation of LPV systems: Application to flight vehicle control. In *Proc. IEEE Conf. Decision Control*, Atlantis, Bahamas, 2004.

[19] J.-F. Magni. User manual of the linear fractional representation toolbox (version 2.0). Technical report, ONERA, Toulouse, France, 2006.

[20] L. Dai. *Singular Control Systems*. Springer-Verlag, New York, 1989.

[21] I. Masubuchi, T. Akiyama, and M. Saeki. Synthesis of output feedback gain scheduling controllers based on descriptor LPV system representation. In *Proc. IEEE Conf. Decision Control*, Maui, HI, USA, 2003.

[22] Y. Ebihara and T. Hagiwara. A dilated LMI approach to robust performance analysis of linear time-invariant uncertain systems. *Automatica*, 41(11):1933–1941, 2005.

[23] G. I. Bara and J. Daafouz. Parameter-dependent control with γ-performance for affine LPV systems. In *Proc. IEEE Conf. Decision Control*, Orlando, FL, USA, 2001.

[24] P. Apkarian, H.D. Tuan, and J. Bernussou. Continuous-time analysis, eigenstructure assignment and H_2 synthesis with enhanced LMI characterizations. *IEEE Trans. Autom. Control*, 46(12):1941–1946, 2001.

[25] J.F. Sturm. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optim. Methods and Software*, 11–12:625–653, 1999.

[26] J. Löfberg. YALMIP: a toolbox for modeling and optimization in Matlab. In *in Proc IEEE Conf. Computer-aided Contr. Systems Design*, Taipei, Taiwan, 2004.

[27] R. Bascetta and P. Rocco. Revisiting the robust-control design for rigid robot manipulators. *IEEE Trans. Robot.*, 26(1):180–187, 2010.

[28] P. Apkarian and D. Noll. Nonsmooth H_∞ synthesis. *IEEE Trans. Autom. Control*, 51(1):71–86, 2006.

[29] P. Gahinet and P. Apkarian. Structured H_∞ synthesis in Matlab. In *Proc. IFAC World Congress*, Milan, Italy, 2011.