

LPV Control of a 2-DOF Robot Using Parameter Reduction

Andreas Kwiatkowski and Herbert Werner

Abstract—This paper demonstrates the benefit of a new method for reducing the number of scheduling parameters of LPV controllers, by applying it to a robot manipulator. A full LPV representation of a planar two-link robot with ten scheduling parameters is considered. The number of parameters can be reduced by applying principal components analysis to typical scheduling trajectories. The proposed method enables a systematic trade-off between the number of reduced parameters and the desired accuracy of the model. The practicality of the approach is illustrated with a design example: an LPV controller, designed using a parameter reduced model, is shown in simulation studies to outperform a robust, fixed-gain controller designed to minimize the worst-case H_2 norm .

I. INTRODUCTION

In the last decade, linear parameter-varying (LPV) gain scheduling techniques have been developed into powerful tools for nonlinear controller design. Their strength lies in the extension of well known methods for linear optimal control, including the use of Linear Matrix Inequalities (LMIs), to the solution of nonlinear control problems. Many nonlinear systems of practical interest can be represented as LPV or quasi-LPV systems, where the scheduling parameters depend on external signals or measured system outputs respectively. However, as pointed out in [1], the number of papers on methods for LPV controller synthesis is still far from being matched by the number of reported practical applications. For controller design based on polytopic LPV models and models based on Linear Fractional Transformation (LFT), there are two major obstacles.

Number of scheduling parameters: For polytopic LPV models, the number N of LMIs to be solved for the standard H_∞ controller synthesis [2] grows exponentially with the number l of parameters

$$N = 2^{l+1} + 1.$$

With that, even simple problems become numerically challenging if l exceeds a number of $3 \sim 4$. If one uses methods that avoid the conservatism due to a fixed Lyapunov function by using parameter-dependent Lyapunov functions [3], the computational burden increases even further. For LFT representations, the dimension of the parameter matrix Θ increases with the number of parameters subject to the

structure of the given nonlinear or parameter dependent system. In some publications that deal with the application of LPV techniques to robot manipulators, e.g. [2], [4], these problems are addressed by introducing auxiliary controllers, by neglecting dynamic coupling and by fixing parameter dependent values in the LPV model. This paper will not discuss these simplifications but propose and illustrate a systematic way to simplify LPV models.

Conservatism due to overbounding: For quasi-LPV models usually the parameter range is a superset of the range that is spanned by the real parameter, if the same scheduling signals appear in more than one parameter function. Therefore, the parameter set includes vertices that cannot be attained by the real plant, resulting in conservatism. One way to reduce this conservatism for polytopic models, is to determine operational trajectories of the real plant and to 'reshape' the parameter region such that it matches the given operating points as closely as possible [5], [6]. Unfortunately, this method often results in an increasing number of vertices of the polytope and with that, in increased computational burden. The same holds true for LPV systems, if some combinations of extreme values of the external scheduling signals do not occur in real operation.

In [1] it has been shown how principal components analysis (PCA) of typical operating trajectories can be used to obtain models with fewer parameters and less overbounding. An advantage of this method is the possibility to trade modelling accuracy against the number of parameters. This paper presents the application of PCA-based parameter reduction to the design of an LPV controller for a robot manipulator. It is shown that good control performance can be achieved with a design based on a reduced model.

The paper is organized as follows: Section 2 briefly reviews the method of parameter reduction proposed in [1]. Section 3 gives the nonlinear model of a planar two-link manipulator, derives a full LPV model and describes the parameter reduction for the given control task. The controller synthesis and a comparison with a robust H_2 controller design are described in Section 4. Section 5 examines the performance of the approximation, while the last section gives a brief outlook on future work.

II. PARAMETER REDUCTION USING PCA

This section briefly reviews the method of parameter reduction proposed in [1]. The main idea is

A. Kwiatkowski and H. Werner are with the Institute of Control Engineering, Hamburg University of Technology, Eissendorfer Str. 40, 21073 Hamburg, Germany {kwiatkowski,h.werner}@tu-harburg.de

- 1) To generate typical operating trajectories for the plant by measurement or simulation. These trajectories should roughly span the expected range of operation of the controlled plant and generate typical scheduling signal trajectories. A model derived this way reflects only the plant behavior in the given operating range; this can be the entire operating region but can also be used to restrict the full operating region, for example to design a controller for a particular control task.
- 2) To determine the parameter trajectories for these scheduling signal trajectories and normalize this data to form a matrix of normalized parameter data.
- 3) To use the method of principal components analysis to determine a set of fewer principal components that can be used to approximate the data.
- 4) To apply the same PCA reduction to the scheduling parameters of an LPV model to obtain an approximated LPV model that has fewer scheduling parameters.

A. Problem formulation

Suppose we are given the LPV or quasi-LPV system

$$\begin{aligned} \dot{x}(t) &= A(\theta(t))x(t) + B(\theta(t))u(t), \\ y(t) &= C(\theta(t))x(t) + D(\theta(t))u(t), \end{aligned} \quad (1)$$

where $\theta(t) \in \mathbb{R}^l$ represents a time-varying parameter vector, and the mappings $A(\cdot)$, $B(\cdot)$, $C(\cdot)$ and $D(\cdot)$ are continuous functions of θ . The parameter vector $\theta(t)$ depends on a vector of measurable signals $v(t)$ referred to as *scheduling signals*, according to

$$\theta(t) = f(v(t)), \quad (2)$$

where $f: \mathbb{R}^k \rightarrow \mathbb{R}^l$ is a continuous mapping. Here it is assumed that $v(t)$ contains internal signals $v_i(t)$ and external signals $v_e(t)$, $v(t) = [v_i(t) \ v_e(t)]^T \in \mathbb{R}^k$. The problem considered in this section is the following. Find a mapping $g: \mathbb{R}^k \rightarrow \mathbb{R}^m$ such that $m \leq l$, and

$$\phi(t) = g(v(t)), \quad (3)$$

yields a model

$$\begin{aligned} \dot{x}(t) &= \hat{A}(\phi(t))x(t) + \hat{B}(\phi(t))u(t), \\ y(t) &= \hat{C}(\phi(t))x(t) + \hat{D}(\phi(t))u(t), \end{aligned} \quad (4)$$

that provides a satisfactory approximation of the system (1). Moreover, find the smallest integer m for which a satisfactory approximation is possible.

B. Parameter Reduction Algorithm

Assume that the scheduled signals $v(t)$ have been sampled at time instants $t = kT$, $k = 0, 1, \dots, N$, where $N \gg l$. This data set can be used to generate the $l \times N$ data matrix

$$\begin{aligned} \Theta &= [\theta(0) \ \theta(T) \ \dots \ \theta(NT)] \\ &= [f(v(0)) \ f(v(T)) \ \dots \ f(v(NT))]. \end{aligned} \quad (5)$$

The rows Θ_i of the data matrix need to be normalized by an operation \mathcal{N}_i to achieve zero mean data with unity deviation,

$$\Theta_i^n = \mathcal{N}_i(\Theta_i), \quad \Theta_i = \mathcal{N}_i^{-1}(\Theta_i^n). \quad (6)$$

Row-wise normalization of all rows is denoted by \mathcal{N} and results in a normalized data matrix $\Theta^n = \mathcal{N}(\Theta)$. Now, one applies the Principle Components Analysis (PCA) [7] to the normalized data. Introduce a singular value decomposition of Θ^n

$$\Theta^n = [U_s \ U_n] \begin{bmatrix} \Sigma_s & 0 & 0 \\ 0 & \Sigma_n & 0 \end{bmatrix} \begin{bmatrix} V_s^T \\ V_n^T \end{bmatrix}, \quad (7)$$

with $\Sigma_s = \text{diag}(\sigma_1 \cdots \sigma_m)$, $\Sigma_n = \text{diag}(\sigma_{m+1} \cdots \sigma_l)$, $U_s \in \mathbb{R}^{l \times m}$, $V_s \in \mathbb{R}^{N \times m}$, $U_n \in \mathbb{R}^{l \times (l-m)}$ and $V_n \in \mathbb{R}^{N \times (N-m)}$, and assume that U_s , Σ_s and V_s correspond to the m significant singular values, so that

$$\hat{\Theta}^n = U_s \Sigma_s V_s^T \approx \Theta^n, \quad (8)$$

is a reasonable approximation of the given data. The comparison of $\hat{\Theta}^n$ and Θ^n allows to assess the quality of the approximation, and - by choosing the number m of scheduling variables - to trade accuracy of the model against complexity. The matrix U_s represents a basis of the significant column space of the data matrix Θ^n , and can be used to obtain a reduced mapping g from $v(t)$ to $\phi(t)$ by computing

$$\phi(t) = g(v(t)) = U_s^T \mathcal{N}(f(v(t))), \quad (9)$$

i.e. by applying the transformation U_s and the scaling (6) to the mapping f in (2). Note that the rows of $\Sigma_s V_s^T$ represent the principal components of the data matrix Θ^n , and that the approximate mappings $\hat{A}(\cdot)$, $\hat{B}(\cdot)$, $\hat{C}(\cdot)$, $\hat{D}(\cdot)$ in (4) are related to (1) by

$$\begin{bmatrix} \hat{A}(\phi(t)) & \hat{B}(\phi(t)) \\ \hat{C}(\phi(t)) & \hat{D}(\phi(t)) \end{bmatrix} = \begin{bmatrix} A(\hat{\theta}(t)) & B(\hat{\theta}(t)) \\ C(\hat{\theta}(t)) & D(\hat{\theta}(t)) \end{bmatrix}, \quad (10)$$

where

$$\hat{\theta}(t) = \mathcal{N}^{-1}(U_s \phi(t)) = \mathcal{N}^{-1}(U_s U_s^T \mathcal{N}(\theta(t))), \quad (11)$$

and \mathcal{N}^{-1} denotes row-wise re-scaling. At any given time, (9) can be used to compute the reduced parameter vector $\phi(t)$, while (10) together with (11) can be used to generate the approximate LPV model.

III. EXAMPLE: ROBOT MANIPULATOR

A two link planar robot manipulator is shown in Fig. 1. The lower part is referred to as shoulder, the upper one

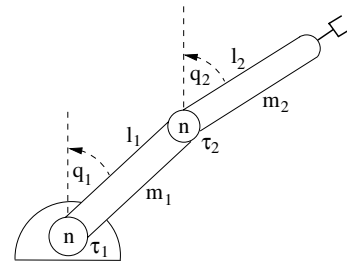


Fig. 1. Robot manipulator

as arm. The arm tip holds the end effector and the links are connected via joints that are equipped with gear servo

motors. LPV gain scheduling control of manipulators has attracted significant attention recently, see e.g. [4], [8]. This is because the dynamics of the plant depend on the actual state of the manipulator, i.e. the position of the joints and the angular velocities, and are highly nonlinear. To reduce the number of scheduling parameters of an LPV model, usually the dynamic couplings between links are neglected and some terms in the model are set to zero or fixed values. The example in this section shows how the number of parameters can be reduced considerably in a systematic way by using principal components analysis. The control task - trajectory tracking - is illustrated in Fig. 2. The end effector moves

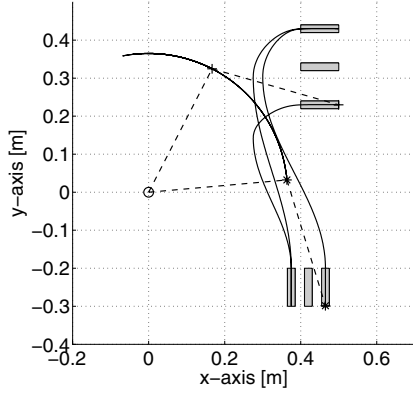


Fig. 2. Control task: reference trajectory

objects between three boxes at the bottom and three boxes at the top. A possible trajectory is shown, starting from the bottom right box and ending at the lower box at the top. Also indicated by dotted lines are the start and end position of the manipulator links.

A. Derivation of the LPV model

The robot can be described by the equation of motion $M(q)\ddot{q} + C(q, \dot{q}) + g(q) = \tau$ with the angles $q = [q_1 \ q_2]^T$, the torques $\tau = [\tau_1 \ \tau_2]^T$ and the matrices

$$M = \begin{bmatrix} a & b \cos \Delta \\ b \cos \Delta & c \end{bmatrix}, \quad g = \begin{bmatrix} -d \sin(q_1) \\ -e \sin(q_2) \end{bmatrix},$$

$$C = \begin{bmatrix} b \sin \Delta \dot{q}_2^2 + (R + nR_m)\dot{q}_1 \\ -b \sin \Delta \dot{q}_1^2 + (R + nR_m)(\dot{q}_2 - \dot{q}_1) \end{bmatrix},$$

with $\cos \Delta = \cos(q_1 - q_2)$, $\sin \Delta = \sin(q_1 - q_2)$. The variables

$$a = m_1(l_{c1}^2 + l_1^2/12) + m_2l_1^2 + I_m n^2, \quad b = m_2l_1l_{c2},$$

$$c = m_2(l_{c2}^2 + l_2^2/12) + I_m n^2, \quad d = g(m_1l_{c1} + m_2l_1),$$

$$e = gm_2l_{c2},$$

have been introduced, where l_i , l_{ci} , m_i denote the length of the links, the center of gravity and the masses for $i = 1, 2$, respectively, and g , R , R_m , I_m and n are the gravity constant, friction in the joints and motors, the inertia of the motors and the gear ratio. The following LPV model (1) with state vector $x^T = [q_1 \ q_2 \ \dot{q}_1 \ \dot{q}_2]$ can be derived, following the

ideas in [4], where k_m denotes a motor constant that links the currents to the torques.

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ cd\theta_3 & -be\theta_4 & \theta_5 & b\theta_6 \\ -bd\theta_7 & ae\theta_8 & \theta_9 & \theta_{10} \end{bmatrix}, \quad C = [I \ O],$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ cnk_m\theta_1 & -bnk_m\theta_2 \\ -bnk_m\theta_2 & an k_m\theta_1 \end{bmatrix}, \quad D = O,$$

$$h = ac - b^2 \cos^2 \Delta, \quad \theta_1 = 1/h, \quad \theta_2 = \cos \Delta / h,$$

$$\theta_3 = \text{si}(q_1)/h, \quad \theta_4 = \cos \Delta \text{ si}(q_2)/h,$$

$$\theta_5 = (-b^2 \sin \Delta \cos \Delta \dot{q}_1 - (c + b \cos \Delta)(R + nR_m))/h,$$

$$\theta_6 = (-c \sin \Delta \dot{q}_2 + \cos \Delta (R + R_m n))/h,$$

$$\theta_7 = \cos \Delta \text{ si}(q_1)/h, \quad \theta_8 = \text{si}(q_2)/h,$$

$$\theta_9 = (ab \sin \Delta \dot{q}_1 + (R + R_m n)(a + b \cos \Delta))/h,$$

$$\theta_{10} = (b^2 \sin \Delta \cos \Delta \dot{q}_2 - a(R + R_m n))/h.$$

Here $\text{si}(q_i) = \sin(q_i)/q_i$, and I , O denote identity and zero matrices of appropriate dimensions. This LPV model has 10 parameters which depend on $q_i(t)$ and $\dot{q}_i(t)$ and result in a number of $N=2049$ LMIs to be simultaneously solved for controller design.

B. Parameter reduction

To generate typical data for the scheduling signals, it is assumed that the controlled manipulator follows the reference trajectory sufficiently well, such that the scheduling trajectories can be constructed from the reference trajectories, shown in Fig. 2. The corresponding angles and angular velocities are displayed in Fig. 3.

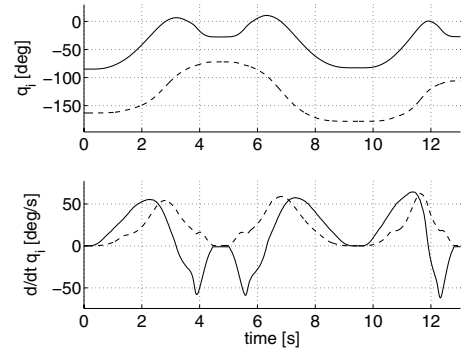


Fig. 3. Scheduling signals v : q_1 , \dot{q}_1 (solid) and q_2 , \dot{q}_2 (dashed)

The parameter vector θ in (2) depends on the angles q_i and the angular velocities \dot{q}_i , this would require $v = v_i = x$. However, it turns out that the dependence on the angular velocities can be neglected. This facilitates design and implementation of a controller, because the number of parameters is smaller and because only the angles are measured, i.e. the available scheduling signal vector is defined as $v^0 = [q_1 \ q_2 \ 0 \ 0]^T$. To justify the simplification, PCA is applied

to data generated from v and v^0 . First, the trajectories shown in Fig. 3 are sampled and evaluated with (2) to determine the data matrix Θ in (5). After normalization, the singular value decomposition in (7) is applied. To determine the number of required principal components one can use the fraction of total variation Σ_v , where σ_i denote the singular values in (7). Fig. 4 compares Σ_v versus the number of parameters m for both cases.

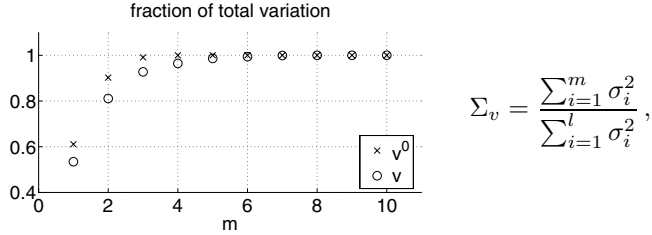


Fig. 4. Fraction of total variation Σ_v vs. m

For both data sets around 60% of the information is displayed by the first principal component. It turns out that around 99% of the information of Θ^n should be displayed to ensure a good approximation of the model, requiring $m = 5$ for the full data set and $m^0 = 3$ for the set v^0 where angular velocities are neglected. Fig. 5 shows some typical parameter trajectories for both data sets.

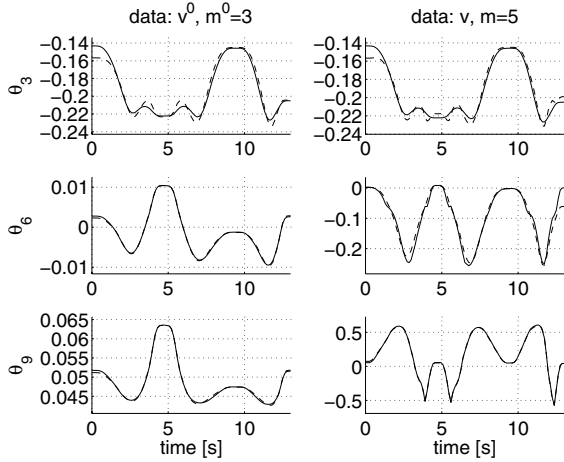


Fig. 5. Parameter θ_i (solid) and approximation $\hat{\theta}_i$ (dashed)

Comparing the parameter trajectories θ_i and their approximations $\hat{\theta}_i$, one can say that the trajectories are matched well so that the reduction to 3 and 5 parameters, respectively, is an adequate choice. On the other hand, comparing a given parameter θ_i generated from v and v^0 shows that the parameter trajectories of θ_6 and θ_9 differ considerably. That the data set generated from v^0 results in a good approximation of the full LPV model, even though the parameter trajectories are different, is demonstrated by the following comparison of properties of the parameter-reduced LPV model. Although it is well known [9], that the properties of the parameterized (frozen) models do usually not allow to infer properties

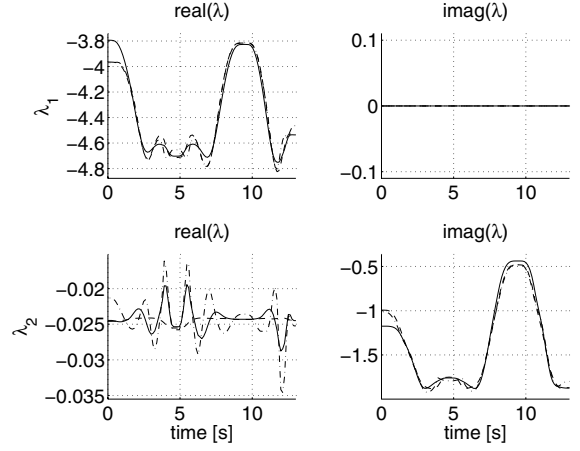


Fig. 6. Eigenvalue trajectories of λ_i (solid), $\hat{\lambda}_i^0$ (dashed), $\hat{\lambda}_i$ (dash-dotted)

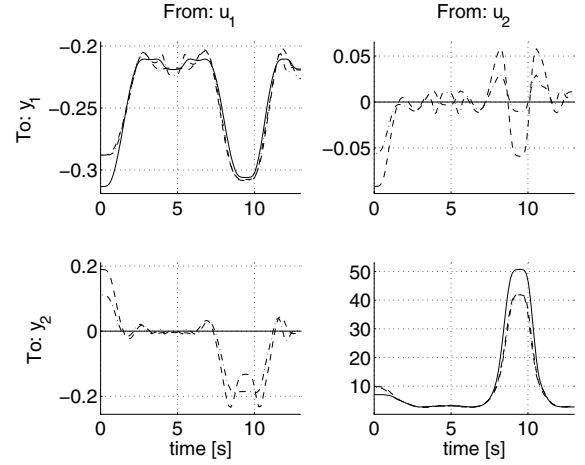


Fig. 7. Static gain of K (solid), \hat{K}^0 (dashed), \hat{K} (dash-dotted)

of the LPV model, we consider the eigenvalues and static gains of local models parameterized along the trajectories to investigate the quality of the approximation:

$$\begin{aligned} \lambda_i &= \lambda(A(\theta(v))), \quad K(v) = -CA(\theta(v))^{-1}B(\theta(v)), \\ \hat{\lambda}_i &= \lambda(A(\hat{\theta}(v))), \quad \hat{K}(v) = -CA(\hat{\theta}(v))^{-1}B(\hat{\theta}(v)), \\ \hat{\lambda}_i^0 &= \lambda(A(\hat{\theta}(v^0))), \quad \hat{K}^0(v) = -CA(\hat{\theta}(v^0))^{-1}B(\hat{\theta}(v^0)). \end{aligned}$$

The trajectories of the eigenvalues are shown in Fig. 6, where it turns out that $\lambda_4 = -\lambda_1$ and $\lambda_3 = \lambda_2^*$. Because the static gains from the inputs to the velocities are zero, Fig. 7 presents the static gain to the angles only. For both plots, a good match can be observed between the models generated with and without measurements of angular velocities. Therefore, the data set generated from v^0 is chosen for constructing an approximated model, and the full model is reduced by 7 parameters to $m = 3$. Note that this reduces the number of vertices of the polytopic LPV model and with that the computational effort by a factor of 128.

IV. CONTROLLER DESIGN

This section presents the design of a polytopic LPV controller using a mixed sensitivity approach, and compares its performance with that achieved with fixed-gain controller designed using a robust H_2 approach. The control structure for the LPV controller is shown in Fig. 8. The nonlinear plant

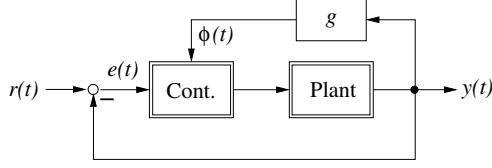


Fig. 8. Structure of LPV gain scheduling control

is controlled by a LPV controller that is scheduled by the reduced parameter vector $\phi(t)$ and is driven by the control error vector $e(t)$. The design of the LPV controller is based on the mixed-sensitivity loop shaping criterion

$$\left\| \begin{array}{c} W_S S \\ W_T T \end{array} \right\|_{\infty} < \gamma \quad (12)$$

with S and T denoting the sensitivity and complementary sensitivity, respectively. Because plant and controller are time-varying systems, the H_{∞} norm must be understood here in terms of the induced L_2 gain. For a design based on a polytopic LPV model, the model must not have a parameter dependent input matrix $B(\theta)$. To overcome this difficulty, the plant is augmented by a first order low-pass filter with sufficiently large bandwidth $V(s) = \text{diag}(1/(0.1s+1), 1/(0.1s+1))$. This removes the parameter dependent input matrix to the augmented system matrix. A full order controller

$$\begin{aligned} \dot{x}_k(t) &= A_k(\phi)x_k(t) + B_k(\phi)e(t) \\ u(t) &= C_k(\phi)x_k(t) + D_k(\phi)e(t), \end{aligned} \quad (13)$$

is then computed using a slightly modified version of the LMI toolbox command `hinfgs`, that enforces a constraint on the spectral radius of the matrix variables Y_{opt} and X_{opt} . This modification was necessary to prevent ill-conditioned Ljapunov matrices. Although the performance is satisfactory, this controller was hard to simulate due to a spectral radius of the controller matrix A_k greater than 10^6 . To reduce the spectral radius, the controller design was carried out using a P-K iteration approach [10] as follows: after initially calculating the Lyapunov matrix P from Y_{opt} and X_{opt} , one alternately searches a controller with P fixed, while minimizing the spectral radius $\rho(A_k)$, then searches for a new matrix P with the controller fixed. It was possible to reduce the spectral radius to $\rho(A_k) < 5 \cdot 10^4$ in only four iterations. To compare the performance of the LPV controller with that achievable with a fixed-gain controller, a robust controller was designed using a robust H_2 approach and K-S iteration [11] with quadratic S-scaling. The nominal und uncertainty models have been derived from a set of linearized models along the reference trajectory. The uncertainty representation was determined by singular value decomposition [12]. With both controllers, the trajectory shown in Fig. 2 and 3 was

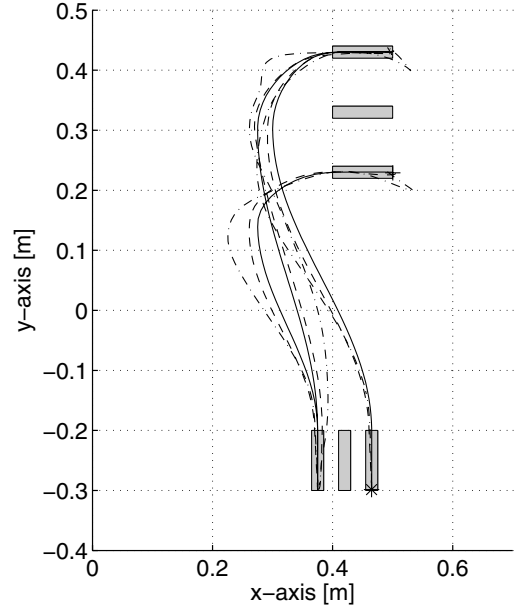


Fig. 9. Comparison of robot trajectories: ref (solid), LPV (dashed), robust (dash-dotted)

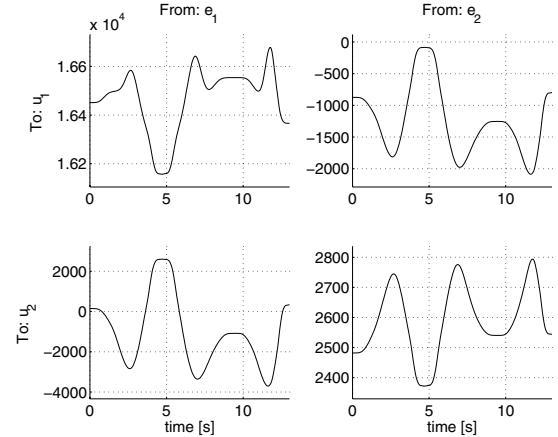


Fig. 10. Static gain of parameterized controller (frozen along trajectory)

simulated using the nonlinear model of the robot. Comparing the magnitude of the control errors shows that the maximum error is about four times larger with the robust controller. The trajectories obtained with both controllers are shown in Fig. 9. Clearly the robust controller is outperformed by the gain-scheduled LPV controller, in particular when approaching the boxes at the top. To illustrate that the LPV gain scheduling controller is really adapting its parameters to the changing operation conditions, the controller is parameterized with the outputs $v^T = [q_1(t) \ q_2(t) \ 0 \ 0]$, determined in the closed-loop simulation. While the eigenvalues of $A_k(g(v))$ show little variation, the (frozen) gains of the controller

$$K(v) = -C_k(v)A_k(v)^{-1}B_k(v) + D_k(v) \quad (14)$$

vary considerably, as shown in Fig. 10.

The main couplings display only small effects of the scheduling, whereas the cross couplings are clearly adapted. The gain $K_{1,2}(v)$ is scaled by a factor of 25, the other cross coupling deviates with twice the magnitude of $K_{2,2}(v)$ and changes its sign.

V. APPROXIMATION ERROR

This section further investigates the quality of the model approximation and its dependence on the parameter range and the number of reduced parameters. For this purpose, the reduction is applied to a set of square trajectories, shown in Fig. 11. The outer trajectory is t_1 , the inner ones are t_2 to t_5 . The maximum velocity in the x - y plane is equal for all trajectories, resulting in times of 10s for t_1 to 2s for t_5 for completing the trajectory. To evaluate the model accuracy,

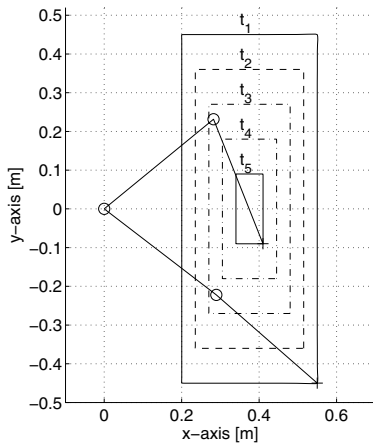


Fig. 11. Square trajectories

the performance measure

$$J_{m,i} = \frac{\int_{t=0}^T (\theta(t) - \hat{\theta}(t))^T (\theta(t) - \hat{\theta}(t)) dt}{\int_{t=0}^T \theta(t)^T \theta(t) dt}$$

is introduced, where the subscript i denotes trajectory t_i and m is the number of parameters used. The performance measure $J_{i,m}$ is shown in Fig. 12. Note that the performance is plotted in decibel, and that smaller values of i correspond to larger parameter ranges. As expected, the quality of the approximation decreases with the size of the parameter range, and increases with the number of parameters m .

VI. CONCLUSIONS AND FUTURE WORK

This paper illustrates how a novel approach for reducing the number of scheduling parameters for LPV and quasi LPV systems can be efficiently used for the design of gain-scheduled controllers. The approach was illustrated by applying it to the design of a gain-scheduled controller for a two-link manipulator. While a full model for a polytopic approach has 10 scheduling parameters, it can be reduced to 3 parameters for the given control task in a systematic way, thus enabling an LMI-based design. Simulation

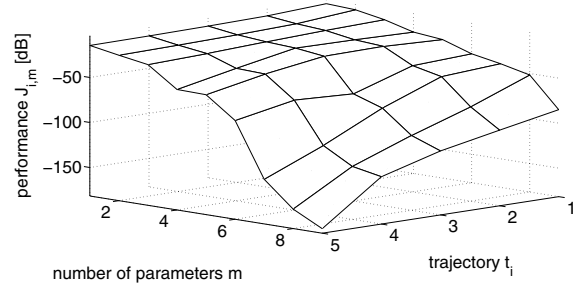


Fig. 12. Accuracy of approximation depending on m and trajectories t_i

results with the nonlinear robot model shows that the LPV controller clearly outperforms a robust fixed-gain controller. The dependence of the modelling error incurred by the parameter reduction on the size of the parameter range and the number of parameters has been investigated. An issue under investigation is to exploit the fact that the LPV representation of a nonlinear system is not unique, and to determine the LPV model that leads to the best control performance.

REFERENCES

- [1] A. Kwiatkowski and H. Werner, "Parameter Reduction for LPV Systems via Principle Components Analysis," in *Proc. 16th IFAC World Congress*. IFAC, 2005.
- [2] H. Kajiwara, P. Apkarian, and P. Gahinet, "LPV techniques for control of an inverted pendulum," *IEEE Control Systems Magazine*, vol. 19, no. 1, pp. 44–54, 1999.
- [3] P. Apkarian and H. Tuan, "Parameterized LMIs in control theory," *SIAM J. Control and Optimization*, vol. 38, pp. 1241–1264, 2000.
- [4] Z. Yu, H. Chen, and P. Woo, "Gain scheduled LPV H_∞ control based on LMI approach for a robotic manipulator," *Journal of Robotic Systems*, no. 12, pp. 585–593, 2002.
- [5] T. Azuma, R. Watanabe, K. Uchida, and M. Fujita, "A new LMI approach to analysis of linear systems depending on scheduling parameters in polynomial forms," *automatisierungstechnik*, vol. 48, no. 4, pp. 119–204, 2000.
- [6] F. Bruzelius, C. Breitholtz, and S. Pettersson, "LPV-based gain scheduling technique applied to a turbo fan engine model," in *Proc. IEEE International Conference on Control Applications, 2002*, 2002.
- [7] J. Jackson, *A user's guide to principle components*, ser. Wiley series in probability and mathematical statistics. John Wiley & Sons. inc, 1991.
- [8] Z. Yu, H. Chen, and P. Woo, "Polytopic gain scheduled H_∞ control for robotic manipulators," *Robotica*, vol. 21, pp. 495–504, 2003.
- [9] W. Rugh, *Linear System Theory*. Prentice Hall, 1993.
- [10] L. E. Ghaoui and V. Balakrishnan, "Synthesis of fixed structure controllers via numerical optimization," in *Proc. IEEE Conference on Decision and Control*, 1994.
- [11] A. Farag, H. Werner, and M. Aten, "Robust H_2 controller design and tuning for the ACC benchmark problem and a real-time application," in *Proceedings 15th IFAC World Congress*, 2002.
- [12] H. Werner, P. Korba, and T. Yang, "Robust tuning of power system stabilizers using LMI techniques," *IEEE Transactions on Control Systems Technology*, vol. 1, no. 11, pp. 147–152, 2003.