On the Discretization of LMI-Synthesized Linear Parameter-Varying Controllers

Pierre Apkarian *

Key Words - Discretization; Linear Parameter-Varying systems; Gain-scheduling control; LMI-based synthesis; LMI optimization.

Abstract

Reliable and efficient techniques are now available to synthesize gain-scheduled controllers. Such techniques take advantage of Linear Matrix Inequality (LMI) characterizations of gain-scheduled controllers and provide a global and systematic treatment of the gain-scheduling problem. This paper examines how such controller structures, designed in continuous time, can be discretized and implemented in practical applications. A special emphasis is placed on reducing the computational cost of discrete implementations. The validity of the proposed concepts and schemes are illustrated through a two-link flexible manipulator design example.

1 Introduction

Gain-scheduling techniques are commonly used in industrial applications whenever there is a need for real-time control adjustment to achieve adequate stability and performance in the closed-loop system. Such techniques are known to provide improved performance and stability as compared to fixed controllers, disregarding informations on the plant's operating conditions. Until recently, there was few systematic techniques to address gain-scheduling problems. In most applications, the design is accomplished pointwise, that is, by designing individual controllers associated with a set of operating conditions. Major weaknesses of such approaches are

- the ignorance of the inherent time-varying nature of the plant,
- the lack of supporting theory giving indisputable validity to such synthesis procedures.

Over the recent years, new techniques bypassing such difficulties have been developed (Packard 1994; Packard and Becker 1992; Apkarian and Gahinet 95; Becker 1995; Apkarian, Gahinet, and Becker 1995). The main thrust of this work is to provide systematic and fully-automated synthesis methodologies to solve the gain-scheduling problem as a whole entity, without the need of separated designs and repeated simulations. A number of techniques targeting this problem are now available from simple ones (Packard 1994; Apkarian and Gahinet 95; Apkarian, Gahinet, and Becker 1995) to more refined versions (Wu, Yang, Packard, and Becker 1995; Becker 1996; Scherer 1995; Apkarian and Adams 95). A key ingredient in the construction of such controllers is the formulation of the solvability conditions in terms of Linear Matrix Inequalities (LMIs). These constitute a set of convex constraints from which one can extract a particular solution using efficient solvers in convex semi-definite programming (Gahinet, Nemirovski, Laub, and Chilali).

Although some of these have been developed in discrete time (Packard 1994; Apkarian, Gahinet, and Becker 1995), some practical reasons still advocate the use of continuous time synthesis techniques. Firstly, since one is dealing with a continuous plant, stability and performance requirements a more comfortably expressed in continuous time. This in turn may facilitate the selection of adequate criteria along with corresponding weighting functions. Another possible reason is that the sampling questions are reflected on the

*CERT, Dept. of Control Engineering, 2 Av Ed. Belin, 31055 Toulouse, France.
Phone: +33 62 25 27 84 - Fax: +33 62 25 25 64 - Email: **apkarian@cert.fr**
Work supported by D.R.E.T

controller and not on the plant. Therefore, the sampling frequency can be tuned freely in the very last phase of the synthesis task, thus bypassing possible controller redesigns.

A great deal of research has been dedicated to the discretization of Linear Time-Invariant (LTI) controllers. Available techniques can be separated into two distinct classes (Hanselmann 87): closed-loop type techniques accounting for the fact that the discrete controller will be connected to the plant and will operate in closed loop (Singh, Kuo, and Yackel 81; Kuo and Peterson 73); and "isolated" techniques disregarding connections of the discrete controller with the plant. See (Katz 1981; Franklin and Powell 1980; Forsyhthe 83; Middleton and Goodwin 1990), to cite a few. Unfortunately, no such work is directly applicable to gain-scheduled controllers due to their inherent time-varying nature. The primary goal of this paper consists in developing simple methods which extend to Linear Parameter-Varying (LPV) systems and are amenable to real-time implementations. A special attention is paid to the Tustin's or trapezoidal approximation scheme, a prominent and most widely used method in classical discretization theory. Apart from the discretization task, we examine flop-saving strategies for controller implementation. An immediate continuity result shows that the controller dynamics need not be updated or refreshed at each sampling instant. This practically means that one can freeze the controller dynamics on some time intervals and refresh them when significant evolution of the scheduled variable is observed.

The organization of the paper has been chosen according to the complexity of LPV controllers. Some prerequisites and notations are introduced in Section 2, while a brief and general description of the problem is given in Section 3. In section 4, we examine the trapezoidal discretization of LPV controllers together with some simpler methods. An immediate application of the trapezoidal technique to Linear Fractional Transformation (LFT) controllers is presented in Section 5. In Section 6, the discretization of a class of more complicated LPV controllers is investigated. An alternative time-saving formulation of the discretized dynamics is also derived. Aiming at further reducing the necessary computational power for suitable implementation of LPV controllers, Section 7 discusses a simplistic refreshment procedure of the LPV dynamics. Finally, illustrations of concepts and techniques are presented in Section 8, for the control of a two-link flexible manipulator.

2 Notations

The notation used in the remainder of the paper is fairly conventional. We shall need to refer to continuous time signals and to sampled values of this signal. The value of continuous time signal $y(.)$ at time t will be denoted $y(t)$. The corresponding sampled values are simply denoted

$$y(kT), \qquad k = 0, 1, 2, \ldots$$

where T is the sampling period of the continuous time signal. For notational simplicity, we will also drop the explicit dependence of $y(kT)$ on T and denote

$$y_k := y(kT), \qquad k = 0, 1, 2, \ldots$$

We shall also make use of the star product notation. For appropriately dimensioned matrices $N = \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix}$ and $M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$ and assuming the inverses exist, the star product $N \star M$ is defined as

$$N \star M := \begin{bmatrix} F_l(N, M_{11}) & N_{12}(I - M_{11}N_{22})^{-1}M_{12} \\ M_{21}(I - N_{22}M_{11})^{-1}N_{21} & F_u(M, N_{22}) \end{bmatrix}, \tag{1}$$

where

$$F_l(N, V) = N_{11} + N_{12}V(I - N_{22}V)^{-1}N_{21}. \tag{2}$$

defines the lower Linear Fractional Transformation (LFT) map $F_l(N, .)$ and similarly

$$F_u(M, W) = M_{22} + M_{21}W(I - M_{11}W)^{-1}M_{12}, \tag{3}$$

defines the upper LFT map $F_u(M,.)$.

The star product notation can be slightly extended to the cases where one of the matrices N or M have only one block by adopting the conventions

$$N \star M_{11} \;:= F_l(N, M_{11}); \quad N_{22} \star M \;:= F_u(M, N_{22})\,.$$

With these definitions, it is readily verified that the star product is associative.

For real symmetric matrices M, $M > 0$ stands for "positive definite" and means that all the eigenvalues of M are positive. Similarly, $M < 0$ means "negative definite" (all the eigenvalues of M are negative) and $M \geq 0$ stands for "nonnegative definite" (the smallest eigenvalue of M is nonnegative). In large symmetric matrix expressions, terms denoted $\star$ will be induced by symmetry. For instance,

$$\begin{bmatrix} M + N + (\star) & \star \\ Q & P \end{bmatrix} := \begin{bmatrix} M + M^T + N + N^T & Q^T \\ Q & P \end{bmatrix}.$$

For ease of manipulations, given a set of matrices $X_1, \ldots, X_N$, the notation $\mathbf{diag}_{i=1}^N X_i$ designates the block-diagonal matrix with the X_i's on its main diagonal.

3 Problem Statement

Throughout the paper, we examine the discretization of LPV controllers which are described in state-space form by

$$\begin{array}{rcl} \dot{x} & = & A(\theta)x + B(\theta)y\ , \\ u & = & C(\theta)x + D(\theta)y. \end{array} \tag{4}$$

where $\theta := [\theta_1, \cdots, \theta_L]^T$ is a time-varying parameter valued in a compact set Θ of $\mathbb{R}^L$, the signal y designates the plant's measured outputs and u is the control signal delivered to the plant.

Note that the state-space representation (4) can be interpreted as a controller structure either scheduled on an external parameter θ, indirectly correlated to the plant dynamics, or scheduled on the plant's measured output y. In the latter case, the controller dynamics become nonlinear and can be written in the form

$$\begin{array}{rcl} \dot{x} & = & A(y)x + B(y)y\ , \\ u & = & C(y)x + D(y)y. \end{array} \tag{5}$$

Although we shall mostly refer to (4), both controller structures (4) and, possibly conservatively, (5) will be concerned in the rest of the paper.

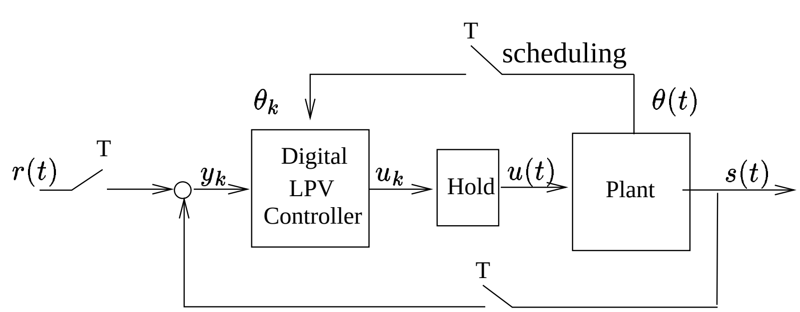

Figure 1: Discretized Implementation of LPV Controllers

The basic idea behind the discretization of LPV controllers is illustrated in Figure 1. There is an hold device at the input of the plant for the reconstruction of the continuous time control signal. The samplers as well as the hold device are assumed synchronized with sampling period T. With all these elements in place, the discretization task consists of finding a discrete time LPV system whose response u_k approximates the sampled response of the continuous time controller.

4 Trapezoidal Discretization of a LPV Controller

This section gives the counterpart of the Tustin or bilinear transformation for LPV systems. Note first that a natural method for deriving a discrete model approximating the continuous controller (4) is to exploit explicit expressions of its state trajectories. Assuming that the controller is in state $x(kT) = x_k$ at the time kT and that the scheduled parameter and the measurements on the time interval $[kT, (k+1)T)$ can be approximated by their values at time kT, that is

$$\theta(t) \approx \theta_k; \quad y(t) \approx y_k \quad \text{for } kT \le t < (k+1)T ,$$

then the state trajectories for $t \ge (k+1)T$ are readily evaluated with

$$x(t) = e^{A(\theta_k)(t-kT)} x_k + \left[\int_{kT}^{t} e^{A(\theta_k)(t-\tau)} B(\theta_k) d\tau \right] y_k .$$

The validity of such an approximation is also motivated in Section 7, Proposition 7.1. Therefore, the discrete model evaluating the controller state at time $(k+1)T$ is simply obtained as

$$x_{k+1} = e^{A(\theta_k)T} x_k + \left[\int_{0}^{T} e^{A(\theta_k)(T-\tau)} B(\theta_k) d\tau \right] y_k . \tag{6}$$

Despite successful applications of this simple technique to different LPV controllers, the number of computations required to evaluate the matrix exponential and the matrix integral in (6) is often prohibitive for real-time implementations. We are thus led to examine alternative methods bearing comparison with respect to accuracy with the above technique. The trapezoidal method offers one such possible alternative. It is formalized in the following theorem.

Theorem 4.1 (Trapezoidal Approximation) *Consider the LPV controller governed by (4) and assume the sampling period is T. A trapezoidal approximation of the sampled dynamics of the system can be described in state space by the following discrete time LPV system*

$$z_{k+1} = (I - \frac{T}{2}A(\theta_k))^{-1}(I + \frac{T}{2}A(\theta_k))\, z_k + \sqrt{T}(I - \frac{T}{2}A(\theta_k))^{-1}B(\theta_k)\, y_k , \tag{7}$$

$$u_k = \sqrt{T}C(\theta_k)(I - \frac{T}{2}A(\theta_k))^{-1}\, z_k + (\frac{T}{2}C(\theta_k)(I - \frac{T}{2}A(\theta_k))^{-1}B(\theta_k) + D(\theta_k))\, y_k. \tag{8}$$

Proof: Considering the sampling interval $[kT, (k+1)T]$ and assuming $x(kT) = x_k$ is known, the state vector at time $(k+1)T$, is given as

$$x_{k+1} = x_k + \int_{kT}^{(k+1)T} (A(\theta(\tau))x(\tau) + B(\theta(\tau))y(\tau))\, d\tau . \tag{9}$$

Using now the trapezoidal rule to estimate the integral in (9) leads to the approximation

$$x_{k+1} \approx x_k + \frac{T}{2}\left(A(\theta_k)x_k + B(\theta_k)y_k + A(\theta_{k+1})x_{k+1} + B(\theta_{k+1})y_{k+1}\right) .$$

Isolating the term x_{k+1} in the left-hand side yields the recursive formula

$$x_{k+1} = (I - \frac{T}{2}A(\theta_{k+1}))^{-1}(I + \frac{T}{2}A(\theta_k))x_k + \frac{T}{2}(I - \frac{T}{2}A(\theta_{k+1}))^{-1}(B(\theta_k)y_k + B(\theta_{k+1})y_{k+1}) . \tag{10}$$

Besides, the sampled version of the measurement equation is

$$u_k = C(\theta_k)x_k + D(\theta_k)y_k \,. \tag{11}$$

Note that (10) does not define a standard discrete time system, since the right-hand side involves variables and data at time $(k+1)T$. A more classical input-output equivalent representation can be derived using the following change of variables

$$z_{k+1} := T^{-\frac{1}{2}}(I - \frac{T}{2}A(\theta_{k+1}))x_{k+1} - \frac{T^{\frac{1}{2}}}{2}B(\theta_{k+1})y_{k+1} \tag{12}$$

where it is implicitly assumed that the inverse exists. Exploiting this change of variables, straightforward matrix manipulations lead to (7). The same substitution in the measurement equation (11) yields (8). Finally, (7)-(8) is an equivalent representation of the trapezoidal approximation (10)-(11), provided that the matrix $(I - \frac{T}{2}A(\theta_k))$ is invertible at all times. ■

4.1 Alternative formulations and validity questions

The state space data in (7)-(8) can be formulated as simple LFTs in the sampled data of the original continuous time LPV system. It is readily verified that

$$\begin{bmatrix} z_{k+1} \\ u_k \end{bmatrix} = \mathcal{B} \star \begin{bmatrix} A(\theta_k) & B(\theta_k) \\ C(\theta_k) & D(\theta_k) \end{bmatrix} \begin{bmatrix} z_k \\ y_k \end{bmatrix}, \tag{13}$$

where matrix $\mathcal{B}$ completely defines the bilinear transformation

$$\mathcal{B} := \begin{bmatrix} I & \sqrt{T}I \\ \sqrt{T}I & \frac{T}{2}I \end{bmatrix}. \tag{14}$$

Note that (7)-(8) or (13)-(14) reduce to the standard Tustin transformation whenever the controller is LTI, that is, does not depend on θ. For the validity of the above technique, the following conditions are implicitly assumed.

- The matrix $(I - \frac{T}{2}A(\theta_k))$ is invertible at each sampling instant kT. This is guaranteed whenever the sampling frequency $f := \frac{1}{T}$ satisfies

  $$f > \frac{\overline{\lambda}(A(\theta))}{2}, \quad \forall \theta \in \Theta,$$

  where $\overline{\lambda}(.)$ stands for the spectral radius of a matrix.

- The sampling frequency f is selected in such a way that there is negligible loss of information in the sampling process of the scheduled variable $\theta(t)$.

The first condition refers to the frozen properties of the LPV system. That is, with the parameter θ maintained fixed, the sampling frequency must be consistent with the dynamics of the underlying linear time-invariant system. The second condition concerns the time-varying properties of the LPV system and thus the representation of its trajectories $\theta(t)$ by the sampled values θ_k. This is an additional condition as compared to customary LTI systems.

With these precautions in mind, LPV controllers can be formally discretized as customary LTI controllers except that a bilinear transformation is required at each sample of time kT. Due to the matrix inversion in (7)-(8) such an approach is generally time-consuming and may require a high speed processor. A less costly representation can be derived by using a rectangular approximation of the integral in (9). We obtain the following discrete time LPV controller

$$x_{k+1} = (I + TA(\theta_k))\, x_k + TB(\theta_k)\, y_k \,, \tag{15}$$
$$u_k = C(\theta_k)\, x_k + D(\theta_k)\, y_k. \tag{16}$$

Equations (15)-(16) can also be viewed as resulting from the application of Euler's method to the continuous time system. Therefore, as is well known in numerical analysis, it will require a very short step length to ensure reasonable accuracy. An alternative approach (see (Hanselmann 87) and references therein) is to use a second-order in T approximation of the state-space representation in (7)-(8). This yields the discrete time controller

$$z_{k+1} = (I + (I + \frac{T}{2}A(\theta_k))TA(\theta_k))\, z_k + (I + \frac{T}{2}A(\theta_k))TB(\theta_k)\, y_k \,, \quad (17)$$

$$u_k = C(\theta_k)(I + (I + \frac{T}{2}A(\theta_k))\frac{T}{2}A(\theta_k))\, z_k + (C(\theta_k)(I + \frac{T}{2}A(\theta_k))\frac{T}{2}B(\theta_k) + D(\theta_k))\, y_k . \quad (18)$$

This form of implementation is close to using a second-order Runge-Kutta discretization of the continuous differential equations. It is inversion free, but may suffer from the same disadvantages as Euler's approximation. See the numerical examples Section 8.

5 LFT Controllers

LFT controllers, issued for example from the synthesis technique in (Apkarian and Gahinet 95), are described by the state space equations

$$\begin{bmatrix} \dot{x} \\ u \end{bmatrix} = \left(\begin{bmatrix} A & B_y & B_\theta \\ C_u & D_{uy} & D_{u\theta} \\ C_\theta & D_{\theta y} & D_{\theta\theta} \end{bmatrix} \star \Delta(t) \right) \begin{bmatrix} x \\ y \end{bmatrix} \quad (19)$$

where $\Delta(t)$ is a time-varying matrix with block-diagonal structure

$$\Delta(t) := \mathbf{diag}_{i=1}^{L}(\theta_i(t) I_{r_i}) \,.$$

Owing to their specific structure, the real-time implementation of such controllers can be greatly simplified. Using the bilinear transformation (13), the discretized dynamics of such controllers can be computed as

$$\mathcal{B} \star \left(\begin{bmatrix} A & B_y & B_\theta \\ C_u & D_{uy} & D_{u\theta} \\ C_\theta & D_{\theta y} & D_{\theta\theta} \end{bmatrix} \star \Delta_k \right) ,$$

which is in turn equivalent to

$$\left(\mathcal{B} \star \begin{bmatrix} A & B_y & B_\theta \\ C_u & D_{uy} & D_{u\theta} \\ C_\theta & D_{\theta y} & D_{\theta\theta} \end{bmatrix} \right) \star \Delta_k \,, \quad (20)$$

in virtue of the associativity of the star product (1).

In contrast to the general formulas (7)-(8), the trapezoidal approximation of a continuous time LFT controller can be computed *off-line* by simply performing the first star-product in (20). The dynamics are updated at each sample kT with the second star-product. This offers obvious advantages for real-time implementations when there are limits on the computational capacity.

6 Advanced Gain-Scheduled Controllers

In this section, we examine the implementation and discretization of a more general class of gain-scheduled controllers. Such controllers are constructed on the basis of refined gain-scheduling synthesis techniques as discussed in references (Wu, Yang, Packard, and Becker 1995; Scherer 1995; Apkarian and Adams 95). Most of this work was directed at developing less conservative techniques explicitly taking into account the rates of variation of the scheduled variable, a key point of the gain-scheduling problem. This aspect was recognized to be a weakness of the techniques of the first generation. Another important feature of this work is to impose almost no restriction on the functional dependence of the LPV plant in θ, thus resulting in a wider scope of application. For future use, we introduce some essential elements of the theory.

The problem addressed by the theory is the following. Consider the LPV plant

$$\begin{array}{rcl} \dot{x} & = & A(\theta)x + B_1(\theta)w + B_2(\theta)u \\ z & = & C_1(\theta)x + D_{11}(\theta)w + D_{12}(\theta)u \\ y & = & C_2(\theta)x + D_{21}(\theta)w \, , \end{array} \tag{21}$$

where $A(.)$, $B_1(.)$,... are continuous and matrix valued functions of $\theta(t)$, a continuously differentiable functions of time. The gain-scheduling problem with guaranteed H_∞ performance consists of finding a LPV controller, with state-space equations

$$\begin{array}{rcl} \dot{x}_K & = & A_K(\theta,\dot{\theta})x_K + B_K(\theta,\dot{\theta})y \\ u & = & C_K(\theta,\dot{\theta})x_K + D_K(\theta,\dot{\theta})y \, , \end{array} \tag{22}$$

which ensures internal stability and a minimal L_2-gain bound γ for the closed-loop operator (21)-(22) from the disturbance signal w to the error signal z. Therefore, the problem has the same statement as the customary H_∞ control problem, except that the controller is allowed to depend on $(\theta,\dot{\theta})$ and stability and performance must be maintained for all trajectories $\theta(t) := [\theta_1(t),\ldots,\theta_L(t)]^T$ described by

(a) the parameter θ ranges in a compact set Θ of $\mathbb{R}^L$,

(b) the rate of variation $\dot{\theta}$ is assumed well-defined at all times and is valued in a hypercube Θ^d of $\mathbb{R}^L$ defined as

$$\dot{\theta}_i \in [\underline{\nu}_i, \bar{\nu}_i], \qquad \underline{\nu}_i \le 0 \le \bar{\nu}_i \, , \tag{23}$$

where $\underline{\nu}_i$ and $\bar{\nu}_i$ are known lower bounds and upper bounds on $\dot{\theta}_i(t)$.

Note that the above description of the LPV plant and of its trajectories is fairly general and encompasses most practical situations.

The construction of LPV controllers (22) relies on the existence of parameter-dependent symmetric matrices $X(\theta)$, $Y(\theta)$ and a parameter-dependent quadruple of state-space data $\widehat{A}_K(\theta)$, $\widehat{B}_K(\theta)$, $\widehat{C}_K(\theta)$ and $D_K(\theta)$ such that the following LMI system holds for all pairs $(\theta,\dot{\theta})$ in $\Theta \times \Theta^d$.

$$\begin{bmatrix} \dot{X} + XA + \widehat{B}_K C_2 + (\star) & \star & \star & \star \\ \widehat{A}_K^T + A + B_2 D_K C_2 & -\dot{Y} + AY + B_2\widehat{C}_K + (\star) & \star & \star \\ (XB_1 + \widehat{B}_K D_{21})^T & (B_1 + B_2 D_K D_{21})^T & -\gamma I & \star \\ C_1 + D_{12} D_K C_2 & C_1 Y + D_{12}\widehat{C}_K & D_{11} + D_{12} D_K D_{21} & -\gamma I \end{bmatrix} < 0 \tag{24}$$

$$\begin{bmatrix} X & I \\ I & Y \end{bmatrix} > 0. \tag{25}$$

In such case, the controller matrices in (22) are obtained as

$$\begin{array}{rcl} A_K(\theta,\dot{\theta}) & = & (I - XY)^{-1}(X\dot{Y} + \widehat{A}_K - X(A - B_2 D_K C_2)Y \\ & & -\widehat{B}_K C_2 Y - XB_2\widehat{C}_K) \qquad (26) \\ B_K(\theta) & = & (I - XY)^{-1}(\widehat{B}_K - XB_2 D_K) \qquad (27) \\ C_K(\theta) & = & (\widehat{C}_K - D_K C_2 Y)\, , \qquad (28) \end{array}$$

where the dependence of some data and variables on the scheduled variables θ and $\dot{\theta}$ has been dropped for simplicity. If we further add the (conservative) condition

$$\frac{d}{d\theta} Y(\theta) = 0 \, ,$$

that is, Y does not depend on θ, then the controller matrices do not longer depend on $\dot{\theta}$ and the equations (26)-(28) characterize a practically implementable gain-scheduled controller. See (Becker 1996; Apkarian and Adams 95) for a complete discussion.

With the help of these formulas, the controller dynamics can be updated in real time according to the evolution of θ. Discretizing such controllers with the trapezoidal method theoretically requires at each sample of time

1. the inversion of $(I - X(\theta_k)Y(\theta_k))$ to compute the continuous time data (26)-(28),
2. computing the inverse in (7)-(8) to get the discrete time data.

Careful examination of the formulas shows that these operations can actually be reduced to only one inversion per sample. Defining $\Pi_A(\theta_k)$ and $\Pi_B(\theta_k)$ with

$$A_K(\theta_k) = (I - X(\theta_k)Y(\theta_k))^{-1}\Pi_A(\theta_k); \quad B_K(\theta_k) = (I - X(\theta_k)Y(\theta_k))^{-1}\Pi_B(\theta_k)\,, \tag{29}$$

and computing the discrete time controller according to (7)-(8) leads to the θ_k-dependent representation

$$\begin{aligned} z_{k+1} &= (I - XY - \frac{T}{2}\Pi_A)^{-1}(I - XY + \frac{T}{2}\Pi_A)\, z_k + \sqrt{T}(I - XY - \frac{T}{2}\Pi_A)^{-1}\Pi_B\, y_k\,, \\ u_k &= \sqrt{T}C_K(I - XY - \frac{T}{2}\Pi_A)^{-1}(I - XY)\, z_k + (\frac{T}{2}C_K(I - XY - \frac{T}{2}\Pi_A)^{-1}\Pi_B + D_K)\, y_k, \end{aligned} \tag{30}$$

which only requires inverting $(I - X(\theta_k)Y(\theta_k) - \frac{T}{2}\Pi_A(\theta_k))$ to update the controller dynamics and is with no computational overhead in regard to the continuous version (26)-(28). This result is also valid for the dual case $\frac{d}{d\theta}X = 0$ with minor modifications.

7 Refreshing LPV controllers

For both the LFT gain-scheduling technique developed in (Apkarian and Gahinet 95) or advanced gain-scheduling techniques outlined in the previous section, the LPV controller dynamics must be refreshed at each sampling instant. See equations (20) and (30). This might be a major impediment to a broader use of such controller structures when there is a lack of computing speed in critical applications. It is however of interest to point out that in most applications, the plant's data depend continuously on the scheduled variable θ. In such case, associated gain-scheduled controllers are continuous functions of the same variable as well. Since we are using strict LMIs to characterize such controllers, it turns out that the LTI controller issued from the LPV controller by freezing $\theta(t) := \theta_0$ maintains stability and performance for plant trajectories in a sufficiently small neighborhood of θ_0. As a result, there is no need to refresh the LPV controller at each sampling instant. With computational benefits, the refreshment can be performed at certain multiples of the sampling period. This amounts to "blocking" the discrete LPV controller. This intuitive idea, also invoked in (Shamma and Athans 1991), can be justified by the following proposition.

Proposition 7.1 *Consider the LPV plant governed by (21) and assume further that the state-space data $A(\theta)$, $B_1(\theta), \ldots$ are continuous functions of θ. Assume the existence of continuous matrix functions ($\widehat{A}_K(\theta)$, $\widehat{B}_K(\theta)$, $\widehat{C}_K(\theta)$ and $D_K(\theta)$) and continuously differentiable matrix functions $X(\theta)$, $Y(\theta)$ such that (24)-(25) hold. Let θ_0 denote any point in the range of values Θ of θ. Then there exist scalars α and β such that the linear time-invariant controller obtained by blocking the LPV controller $A_K(\theta), B_K(\theta), C_K(\theta), D_K(\theta)$ at $\theta(t) = \theta_0$ ensures both stability and an L_2-gain performance bound γ for trajectories $\theta(t)$ characterized by $(\theta(t), \dot{\theta}(t)) \in \Theta \times \Theta^d$, and*

$$\|\theta(t) - \theta_0\| < \alpha \text{ and } \|\dot{\theta}(t)\| < \beta, \quad t \geq 0\,.$$

Proof: As developed in (Apkarian and Adams 95), the solvability of (24)-(25) guarantees the existence of a LPV controller determined by $A_K(\theta)$, $B_K(\theta)$, $C_K(\theta)$, $D_K(\theta)$ and of a parameter-dependent positive-definite matrix

$$P(\theta) := \begin{bmatrix} X(\theta) & I \\ I - Y(\theta)X(\theta) & 0 \end{bmatrix} \begin{bmatrix} I & Y(\theta) \\ 0 & I \end{bmatrix}^{-1} \tag{31}$$

enforcing both stability and an L_2-gain performance γ, for the closed-loop system (21)-(22). More formally, this was expressed through the Bounded Real Lemma Inequalities

$$\begin{bmatrix} \frac{d}{dt}P + P\mathcal{A} + P\mathcal{B}_2\Omega\mathcal{C}_2 + (\star) & \star & \star \\ (\mathcal{B}_1 + \mathcal{B}_2\Omega\mathcal{D}_{21})^T P & -\gamma I & \star \\ \mathcal{C}_1 + \mathcal{D}_{12}\Omega\mathcal{C}_2 & D_{11} + \mathcal{D}_{12}\Omega\mathcal{D}_{21} & -\gamma I \end{bmatrix} < 0 \tag{32}$$

$$P > 0, \tag{33}$$

for all pairs $(\theta, \dot{\theta})$ in $\Theta \times \Theta^d$ and using the notations

$$\begin{bmatrix} \mathcal{A}(\theta) & \mathcal{B}_1(\theta) & \mathcal{B}_2(\theta) \\ \mathcal{C}_1(\theta) & D_{11}(\theta) & \mathcal{D}_{12}(\theta) \\ \mathcal{C}_2(\theta) & \mathcal{D}_{21}(\theta) & 0 \end{bmatrix} := \left[\begin{array}{cc|c|cc} A(\theta) & 0 & B_1(\theta) & 0 & B_2(\theta) \\ 0 & 0 & 0 & I & 0 \\ \hline C_1(\theta) & 0 & D_{11}(\theta) & 0 & D_{12}(\theta) \\ \hline 0 & I & 0 & 0 & 0 \\ C_2(\theta) & 0 & D_{21}(\theta) & 0 & 0 \end{array}\right]$$

and

$$\Omega(\theta) := \begin{bmatrix} A_K(\theta) & B_K(\theta) \\ C_K(\theta) & D_K(\theta) \end{bmatrix}.$$

Now consider the function $F_{\theta_0}(.,.)$ defined as

$$F_{\theta_0}(\theta, \dot{\theta}) := \begin{bmatrix} \frac{d}{dt}P + P\mathcal{A} + P\mathcal{B}_2\Omega(\theta_0)\mathcal{C}_2 + (\star) & \star & \star \\ (\mathcal{B}_1 + \mathcal{B}_2\Omega(\theta_0)\mathcal{D}_{21})^T P & -\gamma I & \star \\ \mathcal{C}_1 + \mathcal{D}_{12}\Omega(\theta_0)\mathcal{C}_2 & D_{11} + \mathcal{D}_{12}\Omega(\theta_0)\mathcal{D}_{21} & -\gamma I \end{bmatrix}.$$

This is a continuous function of $(\theta, \dot{\theta})$ by virtue of the continuity of P and $\frac{d}{dt}P$, see (31), and by assumption on the variables and data $X(\theta)$, $Y(\theta)$, $A(\theta)$, $B_1(\theta)$,... Moreover, since $\Omega(\theta)$ solves the LMI system (32) for $\theta = \theta_0$ and $\dot{\theta} = 0$, it follows that $F_{\theta_0}(\theta_0, 0) < 0$. By a continuity argument, it is easily inferred that for sufficiently small scalars α and β we have

$$F_{\theta_0}(\theta, \dot{\theta}) < 0, \qquad \forall (\theta, \dot{\theta}) \in \Theta \times \Theta^d \text{ with } \|\theta - \theta_0\| < \alpha \text{ and } \|\dot{\theta}\| < \beta,$$

This result together with the positivity condition (33) simply says that the linear time-invariant controller $\Omega(\theta_0)$ maintains stability and performance for sufficiently slow trajectories in the vicinity of θ_0. This establishes the proof of the proposition. ∎

Remark: This continuity result does give some confidence in the validity of the refreshment process. Unfortunately, Proposition 7.1 is not practically constructive. However, values of α and β can be estimated, with unknown theoretical conservatism, either by

- performing Bounded Real Lemma tests of the form $F_{\theta_0}(\theta, \dot{\theta}) < 0$, around each point θ_0 in a predefined grid of Θ,
- or by direct verification of time-domain simulations for physically motivated trajectories $\theta(t)$.

Such estimations may be useful in a practical perspective in providing a general rule for refreshing the LPV controller when the value of θ at the current sample "substantially deviates" from its value at the last refreshment. The first approach has been used successfully in (Feron, Apkarian, and Gahinet 96; Becker, Bendotti, Gahinet, and Falinower) in the context of robustness analysis. Both approaches, however, are computationally intensive, the latter being less conservative since only meaningful trajectories are investigated.

8 Numerical Examples

In this section, the techniques described previously are applied to an output gain-scheduled controller for a two-link flexible manipulator, Figure 2. The manipulator equations of motion can be written as a "quasi-linear" system in θ_2

$$\dot{x}(t) = \begin{bmatrix} 0 & I \\ -M(\theta_2)^{-1}K & -M(\theta_2)^{-1}D \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ M(\theta_2)^{-1}F \end{bmatrix} u(t)\,, \tag{34}$$

where $x := [\theta_1, \theta_2, \theta_3, \theta_4, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4]^T$, and θ_1, θ_2 are the shoulder and elbow joint angles, respectively. As usually denoted, the matrices M, D and K designate the inertia matrix, the damping matrix and the stiffness matrix, respectively. The control vector u is composed of two torques $u = [\tau_1, \tau_2]^T$ and the measurement vector is $y = [\theta_1, \theta_2]^T$. A detailed description of this model is given in (Adams, Apkarian, and Chretien 1996).

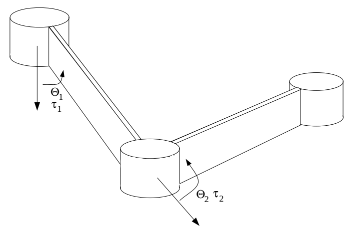

Figure 2: Two-Link Flexible Manipulator

Due to the variable geometry of the manipulator, the inertia matrix M is a function of the second joint angle θ_2, in the form

$$M(\theta_2) = M(\pi/2) + cos(\theta_2)[M(\pi/2) - M(\pi)]\,.$$

In order to meet adequate requirements of robustness and performance over the full range of manipulator dynamics ($\theta_2(t) \in [0, \pi]$), a gain-scheduled controller was designed in continuous time with the refined technique outlined in Section 6. See (Apkarian and Adams 95) for a complete derivation. In state-space form, the differential equations of the controller are given as

$$\begin{aligned} \dot{x_K} &= A_K(\cos(\theta_2))x_K + B_K(\cos(\theta_2))y\,, \\ u &= C_K(\cos(\theta_2))x_K + D_K(\cos(\theta_2))y, \end{aligned} \tag{35}$$

which is therefore a (nonlinear) gain-scheduled on the plant's output controller. It has been designed to provide robustness to uncertainty in the high frequency range, flexible modes attenuation and rapid non-overshooting response to position commands.

8.1 Minimum sampling frequency

The controller is discretized with the three different techniques developed in Sections 4 and 6. In the simulations, the reference commands are chosen to take the manipulator through its entire range of dynamics ($\theta_2 \in [0, \pi]$) as quickly as possible. They consist of simple step commands of 180 degrees in both angles θ_1 and θ_2. For the assessment of proposed discretization techniques, we roughly estimated the minimum tolerable sampling frequency until simulations explode or are no longer satisfying in regard to the specifications. This quantity is indicative of the accuracy of the discretization technique to represent the continuous time LPV dynamics. For instance, a smaller sampling frequency reveals a more accurate discretization technique. Results are given in Table 1.

	Euler	Second-order	Trapezoidal
minimum tolerable sampling frequency f (Hz.)	> 2000	> 1000	45

Table 1: Minimum tolerable sampling frequency

Since they require very large sampling frequencies, Euler's and the second-order approximation schemes must be ruled out in this application. By contrast, the trapezoidal approximation requires a reasonable sampling rate and has been retained in the subsequent simulations. Our conclusion at this stage is that similarly to experienced results for LTI controllers, the trapezoidal technique appears the most appropriate answer for the real-time implementation of LPV controllers.

8.2 Refreshment technique

In this section, we carry on the investigation on how the computational burden can be further alleviated by using the refreshment scheme proposed in Section 7. The gain-scheduled controller is discretized with the trapezoidal method as described in equations (30), with a fixed sampling rate of 50 Hz. An absolute error test is used to activate the refreshment phases. Namely, the discrete controller is refreshed when the current scheduled variable exhibits substantial deviation with respect to its last update $\theta_2((k-j)T)$, for some j, $0 < j \leq k$. The nonlinear simulations are performed with the same input commands covering the entire range of the manipulator dynamics.

	each sample	30 deg.	50 deg.	300 deg.
# of refreshments	599	5	3	0

Table 2: Number of controller refreshments in nonlinear simulations.

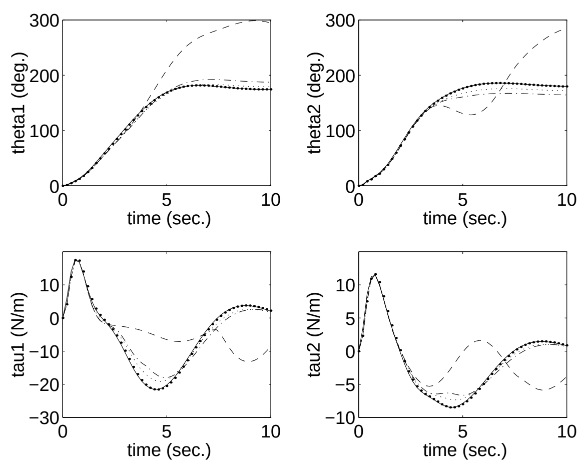

Figure 3: Nonlinear time responses with different refreshment strategies

• continuous controller: solid line
• discrete controller refreshments:
each sample: point every 30 deg.: dotted
every 50 deg.: dashdot; no refreshment: dashed

The simulations corresponding to different refreshment strategies, Table 2, are compared in Figure 3 with the fully-continuous simulations, that are obtained using the original continuous gain-scheduled controller. It can be seen that when the discrete controller is refreshed at each sample, the time responses are indistinguishable from the fully-continuous ones. Therefore, there is essentially no loss of performance in using the trapezoidal method with a sampling frequency of 50 Hz. When the discrete controller is refreshed every 30 degrees of observed deviation the total amount of flops is roughly divided by 120 (Table 2) without significant performance degradation. With fewer refreshments (every 50 degrees), alterations of the performance are more visible in the transient and steady-state zones of the responses. Finally, increasing the refreshing criterion (300 deg.), the controller is no longer gain-scheduled since the full range is $[0, 180]$ deg. The closed-loop system has no longer satisfactory stability and performance in that case. As a consequence, the second option (30 deg.) provides a practically acceptable compromise between computational efficiency and adequate performance.

9 Conclusions

This paper investigated methods for the discretization of LPV controllers issued from recently developed LMI-based synthesis techniques. It has been shown that suitable formulations and extensions of the classical Tustin transformation provides an efficient and accurate technique for numerical implementation of such controllers. When necessary, the flop cost of real-time implementations can be reduced by exploiting the special structure and also refreshment strategies of the LPV dynamics. Such strategies, however, are to be determined by the designer through potentially heavy computations. As demonstrated with a two-link manipulator control problem, the simple techniques in this paper are expected to give additional means and encouragements for practical use of modern gain-scheduling methods.

References

Adams, R. J., P. Apkarian, and J. P. Chretien (1996, May). Robust Control Approaches for a Two-Link Flexible Manipulator. In *3rd International Conference on Dynamics and Control of Structures in Space*, Cranfield, U.K.

Apkarian, P. and R. J. Adams (95). Advanced Gain-Scheduling Techniques for Uncertain Systems. *Submitted to IEEE Trans. on Control System Technology.*

Apkarian, P. and P. Gahinet (95, May). A Convex Characterization of Gain-Scheduled H_∞ Controllers. *IEEE Trans. Automat. Contr. 40*(5), 853–864. See also p. 1681.

Apkarian, P., P. Gahinet, and G. Becker (1995, Sept.). Self-Scheduled H_∞ Control of Linear Parameter-Varying Systems: A Design Example. *Automatica 31*(9), 1251–1261.

Becker, G. (1995). Parameter-Dependent Control of an Under-Actuated Mechanical System. In *Proc. IEEE Conf. Decision Contr.*, LA.

Becker, G. (1996). Additional Results on Parameter-Dependent Controllers for LPV Systems. In *IFAC World Congress*, San Fransisco, USA.

Becker, G., P. Bendotti, P. Gahinet, and C.-M. Falinower. Analysis and Controller Synthesis for a Pressurized Water Reactor Using Linear Parameter Varying Systems . submitted to the 1996 CDC.

Feron, E., P. Apkarian, and P. Gahinet (96, July). Analysis and Synthesis of Robust Control Systems via Parameter-Dependent Lyapunov Functions. *IEEE Trans. Automat. Contr. 41*(7), 1041–1046.

Forsyhthe, W. (83). Algorithms for Digital Control. *Trans. of the Institute of Meas. and Control 5*(3), 123–135.

Franklin, G. F. and J. D. Powell (1980). *Digital Control of Dynamic Systems* . Addison-Wasley.

Gahinet, P., A. Nemirovski, A. J. Laub, and M. Chilali. *LMI Control Toolbox* . The MathWorks Inc., 1994.

Hanselmann, H. (87). Implementation of Digital Controllers - A Survey. *Automatica 23*(1), 7–32.

Katz, P. (1981). *Digital Control Using Microprocessors.* Prentice-Hall;Englewood Cliffs, NJ, USA.

Kuo, B. C. and D. W. Peterson (73). Optimal Discretization of Continuous-Data Control System. *Automatica 9*(1), 125–129.

Middleton, R. H. and G. C. Goodwin (1990). *Digital Control and Estimation - A Unified Approach* . Printice-Hall.

Packard, A. (1994, october). Gain Scheduling via Linear Fractional Transformations. *Syst. Contr. Lett. 22*(2), 79–92.

Packard, A. and G. Becker (1992, november). Quadratic Stabilization of Parametrically-Dependent Linear Systems using Parametrically-Dependent Linear, Dynamic Feedback. *Advances in Robust and Nonlinear Control Systems DSC-Vol. 43*, 29–36.

Scherer, C. (1995). Mixed H_2/H_∞ Control for Linear Parametrically Varying Systems. In *Proc. IEEE Conf. Decision Contr.*, New Orleans, LA, pp. 3182–3187.

Shamma, J. F. and M. Athans (1991, march). Guaranteed Properties of Gain Scheduled Control for Linear Parameter-varying Plants. *Automatica 27*(3), 559–564.

Singh, G., B. C. Kuo, and R. A. Yackel (81). Digital approximation by point-by-point state matching with higher-order holds. *Int. J. Contr. 20*, 81–90.

Wu, F., X. Yang, A. Packard, and G. Becker (1995). Induced L_2-Norm Control for LPV System with Bounded Parameter Variations Rates. In *Proc. Amer. Contr. Conf.*, Seattle, Wa., pp. 2379–2383.