Gain-Scheduling Through Continuation of Observer-Based Realizations - Applications to H_∞ and μ Controllers

Paulo C. Pellanda [1] Pierre Apkarian Daniel Alazard

ONERA-CERT, Control System Dept., 2 av. Edouard Belin, 31055 Toulouse, FRANCE
Email: pellanda(apkarian,alazard)@cert.fr - Fax: +33 5.62.25.25.64

Abstract

The dynamic behavior of gain-scheduling controllers is highly depending on the state-space representations adopted for the family of linear controllers designed on a set of operating conditions. In this paper, a technique for determining a set of consistent and physically motivated linear state-space transformations to be applied to the original set of linear controllers is proposed. After transformation, these controllers exhibit an observer-based structure and are therefore easily interpolated and implemented. This method is applicable to discrete- or continuous-time and full- or augmented-order compensators, particularly including H_∞ and μ controllers, which do not generally enjoy ease of implementation.

1 Introduction

Gain-scheduling constitutes a very effective way of controlling systems whose dynamics changes with operating conditions. Most of the recent gain-scheduling methods were developed based on a set of plant equilibrium conditions and a corresponding set of linear controllers [4, 6, 10, 12]. In this context, plant equilibrium conditions can be parameterized by some scheduling variables. Assuming that scheduling variables are measured during operation, the gain-scheduled controller is a non-stationary and/or nonlinear system obtained by simple interpolation or more sophisticated nonlinear realizations.

When the nominal plant models can be taken Linear Time-Invariant (LTI), modern optimal and robust control techniques are available to design linear controllers which provide reasonable compromise between performance and robustness around given operating conditions. In comparison, the available techniques to interpolate them are relatively immature. Some of the most used strategies in classical gain-scheduling approach have practical appeal and involve either linear interpolation of gains, poles and zeros of transfer functions [8] or of state-space matrices or state-feedback gains [3, 5].

In classical gain-scheduling context of this paper, if linear controllers change significantly for one operating condition to the next, fast rate of variations of the closed-loop dynamics are artificially introduced and this may result in a destabilizing effect or a loss of performance. Then, to obtain desirable interpolation behaviors, the linear controllers to be scheduled should have consistent structures. This becomes critical when state-space data are utilized in the interpolation scheme, because dynamic behaviors of gain-scheduling controllers can be strongly dependent on the adopted realizations. See [13, 14] for illuminating demonstrations of this fact.

In [13, 14], Stilwell and Rugh provide a theoretically justified sufficient conditions on the "placement" of LTI controllers such that a stability preserving interpolated controller always exists. An upper bound on the rate of variation of the scheduling variable can also be determined to assess time-varying stability. Their results clearly show that satisfactory transitions depend not only on the "distance" between operating points but also on the "proximity" between the respective LTI controller coefficients. This is strengthened by our results. Unfortunately, in the state-space interpolation context, their methods are restricted to full-order controllers. In addition, when the set of operational points is appropriately chosen, generating the corresponding set of state-space realizations that are amenable to interpolation is a delicate issue which requires insight and remains open in classical gain-scheduling. Yet, in the particular case of the observer-based control structure, the gains are not the only variables to be scheduled. The set of controller coefficients also depend on the state-space system data of the plant and must evolve consistently with the plant dynamics. That is, significant variations or nonlinearities of the plant must be accordingly compensated by adequate adjustments in the controller.

In this paper, we propose a method to derive a set of state-space linear transformations applied to an original family of linear controllers such that the dynamic discrepancies between controllers in the transformed

[1] Supported by Brazilian Ministry of Defense at ONERA.

family are minimized and the dynamic behavior of the physical plant is respected. Consequently, the nonlinear gain-scheduled controller has similar dynamics to the linear family and this lead to weaker restrictions on the scheduling variable rate-bound that guarantees stability. Additionally, our methodology generates a set of stable Youla parameters having a particular structure which tolerates a simple linear interpolation. This generalizes for augmented-order controllers the stability preserving interpolation methods proposed in [13] and [14].

2 Observer-based structures

Observer-based structures present important features which become particularly interesting in the interpolation step of the gain-scheduling problem for realistic applications. Modern controller design techniques as H_∞ and μ synthesis and their variants generate high-order controllers whose dynamics remain obscure to the designer and may change significantly with plant operating conditions. The interpolation of general state-space representations to these controllers is highly questionable from an implementation viewpoint and in many cases will lead to an insuperable computational effort, particularly, for problems necessitating fast real-time adjustment of the controller data. In opposite, for observer-based controllers, assuming that the linear plant model is available in real-time, it is only required the storage of two static gains and one Youla parameter to update the controller dynamics at each sampling instant. Furthermore, in practice, scheduling variable depends on the plant outputs and/or states being interesting to estimate efficiently the plant states for all scheduling space. Practical techniques to compute estimator/controller forms to arbitrary compensators have been investigated in [1, 2] and references therein. Nevertheless, these methods treat LTI controllers in a separated and disconnected way, leading to a set of state transformations which are not always adapted to the gain-scheduling context.

In this section, we recall a technique proposed in [1] to compute equivalent state estimator-state feedback representations of an arbitrary stabilizing compensator associated with a given plant. The techniques in [1] are general and encompass discrete- or continuous-time and reduced-, full- or augmented-order controllers. They are also applicable to proper or non-strictly proper controllers and plants. However, we are mainly concerned with compensators whose orders are greater or equal to the plant's order, and particularly H_∞ and μ controllers. Reduced-order controllers are not considered here. For simplicity of presentation, the non-strictly proper plants and discrete-time cases are omitted, but they are also encompassed by our method.

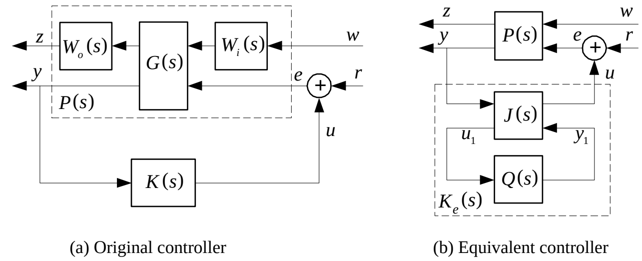

Figure 1: Closed-loop systems

Consider the closed-loop systems depicted in Figure 1, where

$$G(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix}, \qquad (1)$$

and

$$P(s) = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \qquad (2)$$

are the nominal and augmented system models, respectively. The signals w, e, z and y are the exogenous input, the control command, the controlled output and the measurement of $P(s)$, respectively. The plants $G_{22}(s)$ and $P_{22}(s)$, assumed strictly proper without loss of generality, are defined by the following stabilizable and detectable realizations:

$$G_{22}(s) = \left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right], \quad P_{22}(s) = \left[\begin{array}{c|c} A_p & B_p \\ \hline C_p & 0 \end{array}\right], \quad (3)$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$ and $spec(A_p) \supseteq spec(A)$. $P_{22}(s)$ incorporates some additional fictitious dynamics (stable weights and/or dynamic scalings $W_i(s)$ and $W_o(s)$). $G_{22}(s)$ corresponds to the physical plant's dynamics (eventually including actuators and sensors) whose state trajectories should be stabilized and estimated by an observer-based controller.

The problem can be stated as follows. *Given* the plants $G_{22}(s)$ and $P_{22}(s)$ and an original stabilizing controller (Figure 1-a)

$$K(s) = \left[\begin{array}{c|c} A_k & B_k \\ \hline C_k & D_k \end{array}\right], \qquad (4)$$

where $A_k \in \mathbb{R}^{n_k\times n_k}$, $n_k \geq n$, $D_k \in \mathbb{R}^{m\times p}$ and B_k, C_k are real with compatible dimensions, *find* an observer-based controller with an explicitly separated structure (Figure 1-b)

$$K_e(s) = \left[\begin{array}{c|c} T^{-1}A_kT & T^{-1}B_k \\ \hline C_kT & D_k \end{array}\right], \qquad (5)$$

where T is a similarity transformation, *such that* (5) is input-output equivalent to (4). $K_e(s)$ is a lower Linear

Fractional Transformation with respect to the Youla parameter $Q(s) \in \mathcal{RH}_\infty$.

We denote

$$Q(s) = \left[\begin{array}{c|c} A_q & B_q \\ \hline C_q & D_q \end{array}\right] \quad (6)$$

a minimal realization of Q, where $dim(A_q) = n_q = n_k - n$ and $B_q,\ C_q,\ D_q$ have compatible dimensions. The transfer function $J(s)$ between $[y;\ y_1]$ and $[u;\ u_1]$ is defined as an observer-based controller for $G_{22}(s)$ and is completely characterized by a state-feedback gain K_c and an observer gain K_f. Its state $\hat{x}$ is an (asymptotic) estimation of the state x of the plant $G_{22}(s)$. Then the state vector of the closed-loop system incorporates the plant states (or physical states), observer states and Youla parameter states: $[x;\ \hat{x};\ x_q]$.

From these representations, one can show that the separation principle appears clearly. The closed-loop eigenvalues can then be separated into n closed-loop state-feedback poles, n closed-loop state-estimator poles and n_q Youla parameter poles: $spec(A - BK_c)$, $spec(A - K_fC)$ and $spec(A_q)$, respectively. Yet, if we represent the closed-loop system in terms of the original controller (4) and if we adopt a suitable partition $T = [T_1 \quad T_2]$, the closed-loop state vector becomes $[x;\ T_1\hat{x};\ T_2x_q]$,

Equipped with these definitions and notations, one can derive:

$$\left[\begin{array}{cc} -T_1 & I \end{array}\right] \overbrace{\left[\begin{array}{cc} A + BD_kC & BC_k \\ B_kC & A_k \end{array}\right]}^{H = A_{cl}} \left[\begin{array}{c} I \\ T_1 \end{array}\right] = 0, \quad (7)$$

$$(A_k - T_1BC_k)T_2 = T_2A_q, \quad (8)$$

$$T^{-1}B_k = \left[\begin{array}{c} K_f + BD_q \\ B_q \end{array}\right], \quad (9)$$

$$C_kT = \left[\begin{array}{cc} -(K_c + D_qC) & C_q \end{array}\right] \quad (10)$$

and

$$D_k = D_q. \quad (11)$$

Therefore, the Hamiltonian matrix H associated with the generalized non-symmetric and rectangular Riccati equation (7) is nothing else than the dynamic closed-loop system matrix A_{cl}. The Riccati equation (7) can be solved in $T_1 \in \mathbb{R}^{n_k \times n}$ by standard invariant subspace techniques consisting in:

- Computing an invariant subspace associated with a set of n eigenvalues, $spec(\Lambda_n)$, chosen among $2n + n_q$ eigenvalues in $spec(A_{cl})$, that is,

$$H \left[\begin{array}{c} U_1 \\ U_2 \end{array}\right] = \left[\begin{array}{c} U_1 \\ U_2 \end{array}\right] \Lambda_n, \quad (12)$$

  where $U_1 \in \mathbb{R}^{n \times n}$ and $U_2 \in \mathbb{R}^{n_k \times n}$. Such subspaces are easily computed using a Schur factorizations of the matrix A_{cl}.

- Computing the solution

$$T_1 = U_2U_1^{-1}, \quad (13)$$

  whose existence is guaranteed whenever all closed-loop eigenvalues are distinct.

Using the later result and similarly, the Sylvester equation in $T_2 \in \mathbb{R}^{n_k \times n_q}$ (8) is reduced in computing an invariant subspace associated with a set of n_q eigenvalues, $spec(A_q)$, chosen among $n + n_q$ eigenvalues in $spec(A_k - T_1BC_k)$.

So, the problem is reduced in solving (7) and (8) in $T \in \mathbb{R}^{n_k \times n_k}$, and next in computing K_c, K_f, B_q, C_q and D_q using (9), (10) and (11).

We can also establish the following proposition:

Proposition 1 *[1] The n eigenvalues chosen for the computation of the solution T_1 in (7) using the Hamiltonian approach are the n eigenvalues of the closed-loop state feedback associated with the equivalent observer-based compensator, i.e., $spec(A - BK_c)$. Moreover, the remaining closed-loop dynamics (not chosen in the solution to (7), i.e., dynamics of $A_k - T_1BC_k$) contains the observation dynamics $(A - K_fC)$ increased by the Youla parameter dynamics (A_q).*

There is a combinatoric of solutions according to the choice of the partition of the closed-loop eigenvalues, first in the computation of T_1, and secondly, in the computation of T_2. In the full version of this paper [9] some additional considerations about the possible solutions to (7) and (8) are presented and discussed. An algorithm (a variant of the method presented in [1]) for the computation of input-output equivalent controllers which are more amenable to interpolation is also proposed.

3 A continuous interpolation method

In classical gain-scheduling methods, although the scheduling variable is a function of time in controller implementation, it is viewed as a parameter in the design process. Suppose the equilibrium manifold can be parameterized by a scheduling variable $\theta \in \mathbb{R}$ which evolves in a compact set $\Theta \subset \mathbb{R}$. We assume that $A(\theta)$,

$B(\theta)$, $C(\theta)$, $A_p(\theta)$, $B_p(\theta)$ and $C_p(\theta)$ in (3) are continuous functions on Θ. Suppose that $K(s,\theta_i)$ in (4) are stabilizing controllers designed on $\theta = \theta_i$, $i = 1, 2, \cdots, r$. The main role of a scheduling procedure is to provide a continuous transition law between operating points, θ_i and θ_{i+1}, $\forall\, i = 1, \cdots, r-1$, in order to preserve the performance obtained by the LTI controllers in their neighborhood.

Transition laws always introduce distortions in terms of stability/performance degradation in the intersample behavior. Distortions are kept within acceptable limits when controller coefficients evolve continuously and their ranges of variation are as small as possible. The next section discusses an effective method, based on observer-based representations, for attacking this problem.

More specifically, the problem addressed in this section is: *Given* the plants $G_{22}(s,\theta)$ and $P_{22}(s,\theta)$ (3), now considered parameter-dependent, and a set of stabilizing controllers $K(s,\theta_i)$ (4), *find* a set of equivalent observer-based controllers $K_e(s,\theta_i)$ (5), *such that* the underlying eigenstructure of the interpolated controller is connected continuously between operating conditions. This is realized by an homotopy or continuation technique of Euler-Newton type. This procedure allows to compute a dynamically compatible set of LTI equivalent controllers and ensures that there exists a continuous path connecting their observer-based realizations.

3.1 Continuation of the selected invariant subspaces

Let $\tilde\theta$ be the normalized parameter, $\tilde\theta := (\theta-\theta_i)/\|\theta_{i+1}-\theta_i\|$. Then for $\theta \in [\theta_i,\ \theta_{i+1}]$ we have $\tilde\theta \in [0,\ 1]$. A homotopy method to compute the eigenpairs of a given Hamiltonian matrix $H(\tilde\theta = 1) = H(1)$ is presented in [7]. From the eigenpairs of some real matrix $H(0)$, the eigenpairs of

$$H(\tilde\theta) := (1-\tilde\theta)H(0) + \tilde\theta H(1) \tag{14}$$

are followed separately and successively from $\tilde\theta = 0$ to $\tilde\theta = 1$ using continuation techniques. These techniques are well-suited for parallel computing and large sparse matrices. At $\tilde\theta = 1$, the corresponding eigenpairs of $H(1)$ are computed. The evolution of an eigenpair as a function of $\tilde\theta$ is called an eigenpath.

We propose the use of a similar homotopy method to obtain the corresponding set of eigenvalues of adjacent Hamiltonian matrices. Instead of following each eigenpair independently, the idea is to follow separately each selected invariant subspace corresponding to the chosen partition of the Hamiltonian spectrum. This is an indirect way to follow a set of eigenpairs simultaneously. In the control application context of this paper, this continuation method is then more reliable from a computational viewpoint since some bifurcation problems and ill conditioning due to nearly colinear eigenvectors can be bypassed.

Consider the Hamiltonian matrix

$$H := \begin{bmatrix} F & R \\ S & M \end{bmatrix}, \tag{15}$$

where $F \in \mathbb{R}^{n\times n}$, $M \in \mathbb{R}^{n_k\times n_k}$ and R, S are real with compatible dimensions.

Consider also the following sets of equations:

$$\begin{bmatrix} F & R \\ S & M \end{bmatrix}\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}\Lambda_n, \tag{16}$$

and

$$\begin{bmatrix} F & R \\ S & M \end{bmatrix}\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}\Lambda_k, \tag{17}$$

where $\Lambda_n \in \mathbb{R}^{n\times n}$ and $\Lambda_k \in \mathbb{R}^{n_k\times n_k}$ are real and block-diagonal as in (12), $U_1 \in \mathbb{R}^{n\times n}$ and $V_2 \in \mathbb{R}^{n_k\times n_k}$ are invertible, $U_2 \in \mathbb{R}^{n_k\times n}$ and $V_1 \in \mathbb{R}^{n\times n_k}$. Suppose that the columns of $\begin{bmatrix} U_1; & U_2 \end{bmatrix}$ and of $\begin{bmatrix} V_1; & V_2 \end{bmatrix}$ form a basis of a n-dimensional and a n_k-dimensional invariant subspace of H, respectively. Then, $T_1 = U_2U_1^{-1}$ ($\in \mathbb{R}^{n_k\times n}$) and $T_3 = V_1V_2^{-1}$ ($\in \mathbb{R}^{n\times n_k}$) are the solutions to the generalized non-symmetric and rectangular Riccati equations

$$T_1F - MT_1 + T_1RT_1 - S = 0 \tag{18}$$

and

$$T_3M - FT_3 + T_3ST_3 - R = 0, \tag{19}$$

respectively. Notice that the columns of $\begin{bmatrix} I; & T_1 \end{bmatrix}$ and of $\begin{bmatrix} T_3; & I \end{bmatrix}$ also span a n-dimensional and a n_k-dimensional invariant subspace of H, respectively.

If we have $H(\theta_i) = A_{cl}(\theta_i)$, $spec(\Lambda_n(\theta_i)) = spec(A(\theta_i) - B(\theta_i)K_c(\theta_i))$ and $spec(\Lambda_k(\theta_i)) = spec(A(\theta_i) - B(\theta_i)K_c(\theta_i)) \cup spec(A_q(\theta_i))$ at an operating point θ_i, then the continuation of $T_1(\theta_1)$ and $T_3(\theta_1)$ is sufficient to determine the corresponding dynamics at the adjacent operating point θ_{i+1}.

Let (14) be the homotopy associated to two adjacent operating points, $H(\tilde\theta = 0) = H(0)$ and $H(\tilde\theta = 1) = H(1)$. The Riccati equation

$$\mathcal{F}(\tilde T, \tilde\theta) = \tilde T(\tilde\theta)\tilde F(\tilde\theta) - \tilde M(\tilde\theta)\tilde T(\tilde\theta) + \tilde T(\tilde\theta)\tilde R(\tilde\theta)\tilde T(\tilde\theta) - \tilde S(\tilde\theta) = 0, \tag{20}$$

where $\tilde\theta \in\ [0,\ 1]$, corresponds to (18) if

$$\left\{\begin{array}{l} \tilde T(\tilde\theta) := T_1(\tilde\theta) \\ \tilde F(\tilde\theta) := (1-\tilde\theta)\tilde F(0) + \tilde\theta\tilde F(1) = F(\tilde\theta) := A(\tilde\theta) + B(\tilde\theta)D_k(\tilde\theta)C(\tilde\theta) \\ \tilde M(\tilde\theta) := (1-\tilde\theta)\tilde M(0) + \tilde\theta\tilde M(1) = M(\tilde\theta) := A_k(\tilde\theta) \\ \tilde R(\tilde\theta) := (1-\tilde\theta)\tilde R(0) + \tilde\theta\tilde R(1) = R(\tilde\theta) := B(\tilde\theta)C_k(\tilde\theta) \\ \tilde S(\tilde\theta) := (1-\tilde\theta)\tilde S(0) + \tilde\theta\tilde S(1) = S(\tilde\theta) := B_k(\tilde\theta)C(\tilde\theta) \end{array}\right. \tag{21}$$

and to (19) if

$$\begin{cases} \tilde{T}(\tilde{\theta}) := T_3(\tilde{\theta}) \\ \tilde{F}(\tilde{\theta}) := (1-\tilde{\theta})\tilde{F}(0) + \tilde{\theta}\tilde{F}(1) = M(\tilde{\theta}) := A_k(\tilde{\theta}) \\ \tilde{M}(\tilde{\theta}) := (1-\tilde{\theta})\tilde{M}(0) + \tilde{\theta}\tilde{M}(1) = F(\tilde{\theta}) := A(\tilde{\theta}) + B(\tilde{\theta})D_k(\tilde{\theta})C(\tilde{\theta}) \\ \tilde{R}(\tilde{\theta}) := (1-\tilde{\theta})\tilde{R}(0) + \tilde{\theta}\tilde{R}(1) = S(\tilde{\theta}) := B_k(\tilde{\theta})C(\tilde{\theta}) \\ \tilde{S}(\tilde{\theta}) := (1-\tilde{\theta})\tilde{S}(0) + \tilde{\theta}\tilde{S}(1) = R(\tilde{\theta}) := B(\tilde{\theta})C_k(\tilde{\theta}) \end{cases} . \quad (22)$$

To perform a continuation of $\tilde{T}$ on the interval $[\theta_i, \ \theta_{i+1}]$, it is necessary to subdivide it into subintervals of the form $0 = \tilde{\theta}_0 \leq \tilde{\theta}_1 \leq \tilde{\theta}_2 \leq \cdots \leq \tilde{\theta}_L = 1$. Considering that $\tilde{T}(\tilde{\theta}_l)$, $(l = 0, \cdots, L)$, is a known solution to $\mathcal{F}(\tilde{T}, \tilde{\theta}_l) = 0$ in (20), we perform an Euler-Newton continuation method to compute the solution $\tilde{T}(\tilde{\theta}_{l+1})$ to $\mathcal{F}(\tilde{T}, \tilde{\theta}_{l+1}) = 0$.

3.1.1 Euler approximation : To obtain the Riccati solution at a later interval $\tilde{\theta}_{l+1}$, we apply Newton's method to the equation $\mathcal{F}(\tilde{T}, \tilde{\theta}_{l+1}) = 0$ with initial guess $\tilde{T}(\tilde{\theta}_{l+1})^{(0)} = \tilde{T}(\tilde{\theta}_l) + (\tilde{\theta}_{l+1} - \tilde{\theta}_l)\dot{\tilde{T}}(\tilde{\theta}_l)$, where the dot denotes the parameter derivative. Then, differentiating (20) with respect to $\tilde{\theta}$ and evaluating at $\tilde{\theta}_l$, we obtain the following Sylvester equation:

$$\left[\tilde{T}(\tilde{\theta}_l)\tilde{R}(\tilde{\theta}_l) - \tilde{M}(\tilde{\theta}_l)\right]\dot{\tilde{T}}(\tilde{\theta}_l) + \dot{\tilde{T}}(\tilde{\theta}_l)\left[\tilde{F}(\tilde{\theta}_l) + \tilde{R}(\tilde{\theta}_l)\tilde{T}(\tilde{\theta}_l)\right] +$$

$$\left[\tilde{T}(\tilde{\theta}_l)\dot{\tilde{F}} - \dot{\tilde{M}}\tilde{T}(\tilde{\theta}_l) + \tilde{T}(\tilde{\theta}_l)\dot{\tilde{R}}\tilde{T}(\tilde{\theta}_l) - \dot{\tilde{S}}\right] = 0, \quad (23)$$

where

$$\begin{cases} \dot{\tilde{F}} = \tilde{F}(1) - \tilde{F}(0) \\ \dot{\tilde{M}} = \tilde{M}(1) - \tilde{M}(0) \\ \dot{\tilde{R}} = \tilde{R}(1) - \tilde{R}(0) \\ \dot{\tilde{S}} = \tilde{S}(1) - \tilde{S}(0) \end{cases} . \quad (24)$$

So, the first Newton iterate $\tilde{T}(\tilde{\theta}_{l+1})^{(0)}$ is obtained by solving (23) in $\dot{\tilde{T}}(\tilde{\theta}_l)$ and by computing the Euler step $(\tilde{\theta}_{l+1} - \tilde{\theta}_l)\dot{\tilde{T}}(\tilde{\theta}_l)$.

3.1.2 Newton's algorithm : Now, the objective is to solve iteratively the algebraic equation

$$\mathcal{F}(\tilde{T}, \tilde{\theta}_{l+1}) = \tilde{T}\tilde{F}(\tilde{\theta}_{l+1}) - \tilde{M}(\tilde{\theta}_{l+1})\tilde{T} + \tilde{T}\tilde{R}(\tilde{\theta}_{l+1})\tilde{T} - \tilde{S}(\tilde{\theta}_{l+1}) = 0 \quad (25)$$

starting with an initial estimate $\tilde{T} = \tilde{T}(\tilde{\theta}_{l+1})^{(0)}$. The equation (25) can be expanded using Taylor's theorem. Neglecting higher-order terms, the expanded form of (25) at the kth iteration becomes the following first-order approximation:

$$\mathcal{F}\left(\tilde{T}(\tilde{\theta}_{l+1})^{(k)} + \Delta\tilde{T}, \tilde{\theta}_{l+1}\right) \approx \left[\tilde{T}(\tilde{\theta}_{l+1})^{(k)}\tilde{R}(\tilde{\theta}_{l+1}) - \tilde{M}(\tilde{\theta}_{l+1})\right]\Delta\tilde{T} +$$

$$\Delta\tilde{T}\left[\tilde{F}(\tilde{\theta}_{l+1}) + \tilde{R}(\tilde{\theta}_{l+1})\tilde{T}(\tilde{\theta}_{l+1})^{(k)}\right] + \mathcal{F}(\tilde{T}(\tilde{\theta}_{l+1})^{(k)}, \tilde{\theta}_{l+1}) = 0 \quad (26)$$

If the estimate $\tilde{T}(\tilde{\theta}_{l+1})^{(k)}$ is exact, then $\mathcal{F}$ and $\Delta\tilde{T}$ are zero. However, since $\tilde{T}(\tilde{\theta}_{l+1})^{(k)}$ is only an estimate of $\tilde{T}(\tilde{\theta}_{l+1})$, the error $\mathcal{F}$ is finite. Update values are calculated from $\tilde{T}(\tilde{\theta}_{l+1})^{(k+1)} = \tilde{T}(\tilde{\theta}_{l+1})^{(k)} + \Delta\tilde{T}$, where $\Delta\tilde{T}$ is the solution to the Sylvester equation (26). The process is repeated until the error $\| \mathcal{F} \|$ is lower than a user-specified tolerance. Provided that $\tilde{\theta}_{l+1} - \tilde{\theta}_l$ is sufficiently small, the Newton iteration will converge quadratically to $\tilde{T}(\tilde{\theta}_{l+1})$. Moreover, when the stepsize is overly ambitious or the linear system involved in the solution of the Newton iterate (26) has a large condition number (some eigenvalues corresponding to different partitions are poorly separated at a given point of the continuation trajectory), the Newton iteration can not converge or converge to another solution, characterizing a path jumping. The latter situation is not likely to occur in practice. Even if one existed, it would be transparent to our continuation method because the correct path can be easily recovered by computing Λ_n (or Λ_k) and its complements in the course of a continuation.

3.1.3 Overall algorithm : A global procedure for the observer-based controller computation at the operating point θ_{i+1} starting from an adjacent point θ_i is as follows:

Step 1: Choose a partition of $spec(H(\theta_i))$, that is, define $\Lambda_n(\theta_i)$, $A_q(\theta_i)$ and $\Lambda_k(\theta_i)$, and compute an equivalent observer-based controller (Section 2) to the operating point θ_i.

Step 2: Compute the solutions $T_1(\theta_i)$ and $T_3(\theta_i)$ to the Riccati equations (18) and (19), according to the chosen partition of $spec(H(\theta_i))$.

Step 3: Perform an Euler-Newton continuation of $T_1(\theta_i)$ and $T_3(\theta_i)$ (Sections 3.1.1 and 3.1.2) to obtain the corresponding $T_1(\theta_{i+1})$ and $T_3(\theta_{i+1})$.

Step 4: Compute

$$\Lambda_n(\theta_{i+1}) = \begin{bmatrix} I \\ T_1(\theta_{i+1}) \end{bmatrix} H(\theta_{i+1}) \begin{bmatrix} I \\ T_1(\theta_{i+1}) \end{bmatrix}^{-1} \quad (27)$$

and

$$\Lambda_k(\theta_{i+1}) = \begin{bmatrix} T_3(\theta_{i+1}) \\ I \end{bmatrix} H(\theta_{i+1}) \begin{bmatrix} T_3(\theta_{i+1}) \\ I \end{bmatrix}^{-1} \quad (28)$$

and block-diagonalize them.

Step 5: Separate $A_q(\theta_{i+1})$ from $\Lambda_k(\theta_{i+1})$ by comparing the eigenvectors u_p and v_j associated to $\Lambda_n(\theta_{i+1})$ and $\Lambda_k(\theta_{i+1})$, respectively. That is, compute $\cos(\theta_{pj}) = \frac{|u_p^T v_j|}{\|u_p\| \, \|v_j\|}$ for all p, j ($p = 1, 2, \cdots, n$ and $j = 1, 2, \cdots, n_k$) to separate the v_j's which do not have a corresponding parallel u_p.

Step 6: Using $A_q(\theta_{i+1})$ and $\Lambda_k(\theta_{i+1})$, compute the equivalent observer-based controller to the operating point θ_1.

Note that here $T_1(\theta_{i+1})$ corresponds exactly to the first partition of $T(\theta_{i+1})$, while the second partition $T_2(\theta_{i+1})$ is determined by $A_q(\theta_{i+1})$. As the solution of a Riccati equation is independent of the eigenpair ordering, the gains $K_c(\theta_{i+1})$ and $K_f(\theta_{i+1})$ are independent of the ordering of $\Lambda_n(\theta_{i+1})$ and $\Lambda_k(\theta_{i+1})$. A possible change in eigenvalue ordering in the diagonalization process (Step 4) would just affect the ordering of the columns of $T_2(\theta_{i+1})$. However, the correct ordering is easily recovered by analyzing the proximity between the coefficients of $A_q(\theta_i)$ and $A_q(\theta_{i+1})$.

It is also worthwhile to note that if $n_k = n$ then $T_3(\theta) = T_1(\theta)^{-1}$, and it is sufficient to perform a continuation of $T_1(\theta)$. Yet, for a set of LTI controllers, once Step 1 is performed in the beginning of the process ($i = 1$, for instance), only Steps 2 and 3 are necessary to determine the entire family of linear state-transformation ($i = 2, \cdots, r$). This procedure allows to compute all the set of equivalent controllers from a unique choice of partition of close-loop eigenvalues and ensures that there is a continuous path connecting their observer-based realizations.

3.2 Interpolation

The proposed method generates an adequate set of state-space realizations for interpolation of gain-scheduled controllers. When the set of operational points is appropriately chosen, a good proximity between the corresponding controller coefficients is, in general, obtained comparatively to generic realizations. So, it can be hoped that a linear interpolation is enough to ensure local closed-loop stability for each intermediate value. However, there is no restriction to another interpolation strategy.

With regard to the Youla parameters, this methodology generates block-diagonal dynamic matrices A_q which are stable at each operating point. It is easy to show that, in our context, the linear interpolation of A_q is stable [9]. Since the observer-based structure $J_{11}(s)$ may be also considered a full-order stabilizing controller for a generic plant (A, B, C), this later result generalizes for augmented-order compensators the stability preserving interpolation methods proposed in [13] and [14].

4 Conclusions

This paper has considered the computation of a set of linear state-space transformations for a family of LTI controllers to be scheduled. The transformed set of controllers exhibit compatible observer-based structures having little dynamic discrepancy. This permits continuous interpolation between designs and leads to weaker restrictions on the parameter rate-bound that guarantees stability. This approach has a physical appeal and it proceeds in two stages. First, invariant subspaces associated with the closed-loop eigenvalues of a given operating point are selected to compute a linear transformation which separates the controller states into plant estimation and Youla parameter states. Secondly, a continuation of the selected invariant subspaces is performed to obtain a family of consistent linear state-space transformations. An algorithm based on an Euler-Newton continuation of two generalized nonsymmetric and rectangular Riccati equations has been devised.

A realistic missile pilot problem has been discussed in [9] to demonstrate the advantages of the method: by using a simple linear interpolation strategy we have obtained satisfactory transitions between controllers and good physical estimation. Since the placement of the LTI controllers and the scheduling strategy have been determined, these benefits are generally independent of the original controller state-space representations but can be highly dependent on other controller properties. For instance, a great number of closed-loop double modes resulting from inappropriate controllers can prejudice the performance of the approach.

References

[1] D. Alazard and P. Apkarian, *Exact Observer-Based Structures for Arbitrary Compensators*, Int. J. Robust Nonlinear Contr. **9** (1999), 101–118.

[2] D. J. Bender and R. A. Fowell, *Computing the Estimator-Controller form of a compensator*, Int. J. Contr. **41** (1985), no. 6, 1565–1575.

[3] R. A. Hyde and K. Glover, *The Application of Scheduled H_∞ Controllers to a VSTOL Aircraft*, IEEE Trans. Automat. Contr. **38** (1993), no. 7, 1021–39.

[4] I. Kaminer, A. M. Pascoal, P. P. Khargonekar, and E. E. Coleman, *A Velocity Algorithm for the Implementation of Gain-Scheduled Controllers*, Automatica **31** (1995), no. 8, 1185–1191.

[5] M. Kellet, *Continuous Scheduling of H_∞ Controllers for a MS760 Paris Aircraft, in Robust Control Systems Design Using H_∞ and Related Methods*, pp. 197–219, P. H. Hammond, London, 1991.

[6] D. A. Lawrence and W. J. Rugh, *Gain Scheduling Dynamic Linear Controllers for a Nonlinear Plant*, Automatica **31** (1995), no. 3, 381–390.

[7] S. H. Lui, H. B. Keller, and T. W. C. Kwok, *Homotopy Method for the Large, Sparse, Real Nonsymmetric Eigenvalue Problem*, SIAM, J. Matrix Anal. Appl. **18** (1997), no. 2, 312–333.

[8] R. A. Nichols, R. T. Reichert, and W. J. Rugh, *Gain Scheduling for H_∞ Controllers: a Flight Control Example*, IEEE Trans. Contr. Systems Technology **1** (1993), no. 2, 69–79.

[9] P. C. Pellanda and P. Apkarian, *Gain-Scheduling Through Continuation of Observer-Based Realizations - Applications to H_∞ and μ Controllers*, Submitted.

[10] W. J. Rugh, *Analytical Framework for Gain Scheduling*, IEEE Contr. Syst. Mag. **11** (1991), no. 2, 79–84.

[11] S. M. Shahruz and S. Behtash, *Design of Controllers for Linear Parameter Varying Systems by the Gain Scheduling Technique*, Journal of Mathematical Analysis and Applications **168** (1992), no. 1, 195–217.

[12] J. S. Shamma and M. Athans, *Analysis of Gain Scheduled Control for Nonlinear Plants*, IEEE Trans. Automat. Contr. **35** (1990), no. 8, 898–907.

[13] D. J. Stilwell and W. Rugh, *Interpolation of Observer State Feedback Controllers for Gain Scheduling*, IEEE Trans. Automat. Contr. **44** (June 1999), no. 6, 1225–1229.

[14] ———, *Stability Preserving Interpolation Methods for the Synthesis of Gain Scheduled Controllers*, Automatica **36** (May 2000), no. 5, 665–671.