JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 24, No. 3, May–June 2001

State-Space Interpolation for a Gain-Scheduled Autopilot

Daniel J. Stilwell
U.S. Naval Academy, Annapolis, Maryland 21402-5025

A procedure for interpolating state-space, linear time-invariant controllers for the synthesis of gain-scheduled controllers is introduced. The interpolation method is based on state-space Youla parameterization and generates a gain-scheduled controller that is locally stabilizing at every operating point of a nonlinear plant. The interpolation method also provides guidance on the selection of operating points at which linear time-invariant controllers should be designed. A gain-scheduled autopilot is designed to illustrate the interpolation method.

I. Introduction

GAIN-SCHEDULED controllers are perhaps the most common class of control systems for nonlinear plants, particularly in the aerospace community. In fact, there are few aerospace vehicles without a gain-scheduled controller. Despite gain scheduling's importance among practitioners, it is only within the last decade that it has been seriously addressed within the research community.[1,2] Yet the literature remains nearly silent on the interpolation problem that arises in the synthesis of gain-scheduled controllers. In this paper, we define the interpolation problem and provide a theoretically justified solution for state-space descriptions of linear time-invariant (LTI) controllers.

In a traditional gain-scheduled design, a nonlinear plant is linearized at a number of operating points at which LTI controllers are designed for the corresponding linearized plant. The gain-scheduled controller is implemented by interpolating (scheduling) the LTI controllers in real time as the plant moves among operating points. The interpolation of LTI controllers is addressed in this paper by providing a theoretically justified alternative for standard ad hoc methods.

To introduce the notation and describe the interpolation problem more precisely, we briefly outline the gain-scheduled controller design process. Consider a nonlinear plant

$$\dot{x}(t) = f(x(t), w(t), u(t)), \qquad t \geq 0$$

$$e(t) = h_e(x(t), w(t), u(t)), \qquad y(t) = h_y(x(t), w(t)) \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the control signal, $e(t) \in \mathbb{R}^{p_e}$ is the output, $w(t) \in \mathbb{R}^{m_w}$ is an exogenous signal, and $y(t) \in \mathbb{R}^p$ is the signal measured by the controller. Suppose there exists an equilibrium manifold that can be parameterized by a scheduling variable, $\rho \in \mathbb{R}^l$. That is, there exist continuous functions, $x^o: \mathbb{R}^l \mapsto \mathbb{R}^n$, $u^o: \mathbb{R}^l \mapsto \mathbb{R}^m$, and $w^o: \mathbb{R}^l \mapsto \mathbb{R}^{m_w}$ such that

$$0 = f(x^o(\rho), w^o(\rho), u^o(\rho))$$

for all $\rho \in \Gamma$ where $\Gamma \subset \mathbb{R}^l$ is a connected compact set. The scheduling variable ρ can be a function of the state, input, and an exogenous signal. Although the scheduling variable is time varying in the gain-scheduled controller implementation, it is viewed as a parameter in the design process.

For each ρ, the standard Jacobian linearization of the nonlinear plant is written as

$$\dot{x}_\delta(t) = F(\rho)x_\delta(t) + G_1(\rho)w_\delta(t) + G_2(\rho)u_\delta(t)$$

$$e_\delta(t) = H_1(\rho)x_\delta(t) + J_{11}(\rho)w_\delta(t) + J_{12}(\rho)u_\delta(t)$$

$$y_\delta(t) = H_2(\rho)x_\delta(t) + J_{21}(\rho)w_\delta(t) \tag{2}$$

where, for example,

$$F(\rho) = \frac{\partial f}{\partial x}(x^o(\rho), w^o(\rho), u^o(\rho)) \tag{3}$$

and $x_\delta(t) = x(t) - x^o(\rho)$. We assume that the matrix functions in Eq. (2) are continuous. Based on the plant equilibrium and linearization data, linear controllers are designed for selected values of ρ. The gain-scheduled controller is the linear parameter-varying (LPV) controller produced by interpolating these linear controllers.

In practice, a number of ad hoc methods are available to interpolate LTI controllers. Perhaps the most common is to linearly interpolate the coefficients of controller transfer functions. Other approaches have been reported, including linear interpolation of poles, zeros, and gains of controller transfer functions[3]; linear interpolation of observer and state feedback gains[4]; linear interpolation of matrix coefficients of balanced realization of LTI controllers[5]; and linear interpolation of the solutions of Riccati equations for $\mathcal{H}_\infty$ controllers.[6] Other more recent approaches include implementing LTI controllers in parallel and linearly interpolating their outputs.[7–9] This last method is called *controller blending.* All these approaches have an intuitive appeal and performed satisfactorily in the application considered. In other examples, standard ad hoc methods can generate interpolated controllers that are not locally stabilizing for certain values of the scheduling variable.[10]

In addition to ad hoc interpolation methods, theoretically justified methods have also been presented. An algorithm for the design of linearly interpolated state feedback controllers to solve a

Daniel J. Stilwell received a B.S. degree in computer engineering from the University of Massachusetts, Amherst, MA, in 1991, a M.S. degree in electrical engineering from Virginia Polytechnic Institute and State University, Blacksburg, VA, in 1993, and a Ph.D. degree, also in electrical engineering, from Johns Hopkins University, Baltimore, MD, in 1999. Dr. Stilwell is currently an Assistant Professor in the Systems Engineering Department at the United States Naval Academy, where he teaches courses on linear and nonlinear control systems. He is Founding Manager of the Naval Academy Unmanned Underwater Vehicles Laboratory. His research interests are in nonlinear control and decentralized control with applications in underwater vehicles.

Received 27 December 1999; revision received 14 June 2000; accepted for publication 30 August 2000.

pole-placement problem is presented in Ref. 11. A more recent approach to gain-scheduled control utilizes LPV controller design techniques to directly synthesize an LPV controller.[12,13] In this case, the interpolation problem is addressed implicitly in that the controllers are parameter varying with respect to the scheduling variable. LPV design techniques incorporate an upper bound on the rate of variation of the scheduling variable such that stability is guaranteed. Unfortunately, such an LPV controller may not exist for a given plant and scheduling variable set Γ. In this paper, we propose an interpolation method that can be applied to arbitrary state-space LTI controllers. A sufficient condition on the selection of operating points at which the LTI controllers are designed must be satisfied, and the controllers must all have the same state dimension. If the sufficient condition is not satisfied, then information generated while checking the sufficient condition yields guidance on the selection of operating points at which additional LTI controllers should be designed.

The theoretically justified interpolation method presented here belongs to the class of stability-preserving interpolation methods. We briefly outline the major features of stability-preserving interpolation and refer to Ref. 10 for further discussion. For a given linear parameter-varying plant $\Sigma(\rho)$, as in Eq. (2), with $\rho \in \Gamma$ and a collection of LTI controllers Λ_i, designed for the LTI plant $\Sigma(\rho_i)$, $i = 1, \ldots, q$, a stability-preserving interpolation is an LPV controller $\Lambda(\rho)$ that satisfies

1) $\Lambda(\rho_i) = \Lambda_i$.

2) The coefficients of $\Lambda(\rho)$ are continuous functions of the parameter ρ.

3) $\Lambda(\rho)$ stabilizes $\Sigma(\rho)$ for each fixed $\rho \in \Gamma$.

The first requirement assures that a stability-preserving interpolation must recover the original designed LTI controllers. The second requirement prevents discontinuities on the control signal due solely to the interpolation method. The third requirement states that the interpolated controller must be stabilizing for all values of the parameter ρ. The first two requirements are satisfied by most ad hoc interpolation methods, although the third requirement may not be satisfied.

A stability-preserving interpolation method for controllers with an observer-state feedback structure was reported in Ref. 14. For full-order controllers, that is, controllers whose state dimension equals that of the plant, a stability-preserving interpolation was presented in Ref. 10. A stability-preserving interpolation of transfer functions was also presented in Ref. 10, though it is primarily of theoretical interest because it suffers from the same implementation issues as controller blending.[9] The stability-preserving interpolation method presented here expands upon those methods previously reported by being suitable for LTI controllers of arbitrary state dimension, so long as the state dimension of each LTI controller is the same. For the design example in Sec. V, model reduction techniques are utilized to obtain LTI controllers that are not full-order and do not have an observer-state feedback structure. Thus, the previously reported stability-preserving interpolation methods cannot be applied.

Our interpolation method is based on state-space Youla parameterization, a common systems theory tool for parameterizing stabilizing controllers.[15] Youla parameterization is used to express each LTI controller as the interconnection of two dynamic systems, denoted $\mathcal{J}$ and $\mathcal{Q}$. Thus, we refer to the interpolation method as $\mathcal{J}$-$\mathcal{Q}$ interpolation. In this approach, the LTI controllers Λ_i, $i = 1, \ldots, q$ are parameterized by a common dynamic system $\mathcal{J}$ and distinct systems Q_i. Then a specialized stability-preserving interpolation is developed for the Q_i systems.

II. Notation and Assumptions

State-space descriptions of the LTI controllers to be interpolated are written

$$\Lambda_i \triangleq \begin{cases} \dot{z}(t) = A_i z(t) + B_i y(t) \\ u(t) = C_i z(t) + D_i y(t) \end{cases} \tag{4}$$

for $i = 1, \ldots, q$ and $A_i \in \mathbb{R}^{n_k \times n_k}$. Throughout, we refer to an exponentially stable system as simply a stable system. For example, if A_i in Eq. (4) has all eigenvalues of the open left-half plane, then Eq. (4) is stable.

The linear fractional transformation plays a key role in our work. The lower linear fractional transformation of the plant $\Sigma(\rho)$ in Eq. (2) and the controller Λ_i in Eq. (4) is denoted $\mathcal{F}_l(\Sigma(\rho), \Lambda_i)$ and written

$$\mathcal{F}_l(\Sigma(\rho), \Lambda_i) \triangleq \begin{cases} \dot{x}(t) = \hat{A}_i(\rho)x(t) + \hat{B}_i(\rho)u(t) \\ y(t) = \hat{C}_i(\rho)x(t) + \hat{D}_i(\rho)u(t) \end{cases}$$

where

$$\hat{A}_i(\rho) = \begin{bmatrix} F(\rho) + G_2(\rho)D_i H_2(\rho) & G_2(\rho)C_i \\ B_i H_2(\rho) & A_i \end{bmatrix}$$

$$\hat{B}_i(\rho) = \begin{bmatrix} G_1(\rho) + G_2(\rho)D_i J_{12}(\rho) \\ B_i J_{21}(\rho) \end{bmatrix}$$

$$\hat{C}_i(\rho) = [H_1(\rho) + J_{12}(\rho)D_i H_2(\rho) \quad J_{12}(\rho)C_i]$$

$$\hat{D}_i(\rho) = [J_{11}(\rho) + J_{12}(\rho)D_i J_{21}(\rho)] \tag{5}$$

Further information on the linear fractional transformation is found in Ref. 15. We require two lemmas. In both cases, $\Sigma(\rho)$ is considered an LTI plant with ρ a fixed parameter.

Lemma 2.1: Given the LTI controller Λ_i in Eq. (4) and the LPV system $\Sigma(\rho)$ in Eq. (2), with ρ a constant, suppose there exists K and L such that $F(\rho) + G_2(\rho)K$ and $F(\rho) + LH_2(\rho)$ are stable. Then Λ_i has the same transfer function as $\mathcal{F}_l(\mathcal{J}(\rho), \mathcal{Q}_i(\rho))$ where

$$\mathcal{J}(\rho) \triangleq \begin{cases} \dot{x}(t) = (F(\rho) + G_2(\rho)K + LH_2(\rho))x(t) \\ \qquad -Lw(t) + G_2(\rho)u(t) \\ e(t) = K(\rho)x(t) + u(t) \\ y(t) = H_2(\rho)x(t) - w(t) \end{cases} \tag{6}$$

$$\mathcal{Q}_i \triangleq \begin{cases} \dot{z}(t) = \begin{bmatrix} F(\rho) + G_2(\rho)D_i H_2(\rho) & G_2(\rho)C_i \\ B_i H_2(\rho) & A_i \end{bmatrix} z(t) \\ \qquad + \begin{bmatrix} L - G_2(\rho)D_i \\ -B_i \end{bmatrix} y(t) \\ u(t) = [-K + D_i H_2(\rho) \quad C_i] z(t) - D_i y(t) \end{cases}$$

Lemma 2.2: Given the LPV systems $\Sigma(\rho)$ in Eq. (2) with ρ a constant, suppose there exist K and L such that $F(\rho) + G_2(\rho)K$ and $F(\rho) + LH_2(\rho)$ are stable. Then with $\mathcal{J}$ defined as in Eq. (6), the LPV system $\mathcal{F}_l(\mathcal{J}(\rho), \mathcal{Q})$ stabilizes Σ where $\mathcal{Q}$ is any LTI system that is stable and has compatible input and output dimensions.

The major assumption concerning the LTI controllers we interpolate is a stability-covering condition. In general terms, this condition requires that for every fixed $\rho \in \Gamma$, the LTI plant $\Sigma(\rho)$ is stabilized by at least one LTI controller.

Definition 2.3 (stability-covering condition): Given a parameter-varying linear plant $\Sigma(\rho)$, $\rho \in \Gamma$, suppose a finite set of controllers $\Lambda_1, \Lambda_2, \ldots, \Lambda_q$ has been designed at operating points $\rho_1, \rho_2, \ldots, \rho_q$, such that Λ_i stabilizes $\Sigma(\rho_i)$. Let U_i be an open set such that Λ_i stabilizes $\Sigma(\rho)$ for each fixed $\rho \in U_i$, $i = 1, \ldots, q$. If $\Gamma \subset \cup_{i=1}^{q} U_i$, then the controllers satisfy the stability-covering condition.

The stability-covering condition can be interpreted as a sufficient condition on the selection of operating points for which linear controllers are designed. If there exists a $\bar{\rho} \in \Gamma$ for which no LTI controller stabilizes $\Sigma(\bar{\rho})$, then a new LTI controller should be designed for $\Sigma(\bar{\rho})$. This process is repeated until the stability-covering condition is satisfied.

III. Interpolation

For brevity and notational convenience, we examine the scalar-scheduling variable case, $\Gamma \subset \mathbb{R}$, and describe $\mathcal{J}$-$\mathcal{Q}$ interpolation for the interpolation of two LTI controllers Λ_1 and Λ_2 over a single interpolation interval $\rho \in [a, b]$. The interpolation method presented here can be easily generalized to vector-scheduling variables and multiple interpolation regions, as discussed in Sec. IV.

Theorem 3.1: Given the LPV plant $\Sigma(\rho)$, $\rho \in [a, b]$, with state dimension n, suppose Λ_1 and Λ_2 are controllers, both with state dimension n_k, and there exists an interval $(c, d) \subset [a, b]$ such that Λ_1 stabilizes $\Sigma(\rho)$ for $\rho \in [a, d)$ and Λ_2 stabilizes $\Sigma(\rho)$ for $\rho \in (c, b]$ (the stability-covering condition is satisfied). Then there exists a parameter-varying controller $\Lambda(\rho)$, with state dimension $2n + n_k$ that stabilizes $\Sigma(\rho)$ for each $\rho \in [a, b]$ such that $\Lambda(a)$ and $\Lambda(b)$ have the same transfer functions as Λ_1 and Λ_2, respectively, and the matrix-valued coefficients of $\Lambda(\rho)$ are continuous functions of ρ.

Proof: For ease of notation, we take $c = 0$ and $d = 1$. Let $\Sigma(\rho)$ be as in Eq. (2), and let $K(\rho)$ and $L(\rho)$ be continuous matrix functions such that $F(\rho) + G_2(\rho)K(\rho)$ and $F(\rho) + L(\rho)H_2(\rho)$ are stable (eigenvalues in the open left-hand plane) for each $\rho \in [a, b]$. Existence of such matrices is guaranteed by the fact that $\Sigma(\rho)$ is stabilized by Λ_1 (or Λ_2). One choice of $K(\rho)$ and $L(\rho)$ is given in Remark 3.2. Define

$$\tilde{A}_i(\rho) = \begin{bmatrix} F(\rho) + G_2(\rho)D_i H_2(\rho) & G_2(\rho)C_i \\ B_i H_2(\rho) & A_i \end{bmatrix}$$

$$\tilde{B}_i(\rho) = \begin{bmatrix} L(\rho) - G_2(\rho)D_i \\ B_i \end{bmatrix}$$

$$\tilde{C}_i(\rho) = [-K(\rho) + D_i H_2(\rho) \quad C_i], \qquad \tilde{D}_i(\rho) = -D_i$$

and the systems

$$\mathcal{J}(\rho) \triangleq \begin{cases} \dot{x}(t) = (F(\rho) + G_2(\rho)K(\rho) + L(\rho)H_2(\rho))x(t) \\ \qquad -L(\rho)w(t) + G_2(\rho)u(t) \\ e(t) = K(\rho)x(t) + u(t) \\ y(t) = H_2(\rho)x(t) - w(t) \end{cases} \tag{7}$$

$$Q_i(\rho) \triangleq \begin{cases} \dot{z}(t) = \tilde{A}_i(\rho)z(t) + \tilde{B}_i(\rho)y(t) \\ u(t) = \tilde{C}_i(\rho)z(t) \end{cases} \tag{8}$$

Then by Lemma 2.1, the system $\mathcal{F}_l(\mathcal{J}(\rho), Q_1(\rho))$ has the same transfer function as Λ_1 for $\rho \in [a, 1)$, and $\mathcal{F}_l(\mathcal{J}(\rho), Q_2(\rho))$ has the same transfer function as Λ_2 for $\rho \in (0, b]$. We now describe stability-preserving interpolation of the matrix-valued functions $\tilde{A}_1(\rho)$ and $\tilde{A}_2(\rho)$ appearing in Eq. (8), both of which are stable for $\rho \in (0, 1)$. Let $W_1(\rho)$ and $W_2(\rho)$ be symmetric positive-definite matrices such that

$$\tilde{A}_1^T(\rho)W_1(\rho) + W_1(\rho)\tilde{A}_1(\rho) < -I \tag{9}$$

for $\rho \in [a, 1)$ and

$$\tilde{A}_2^T(\rho)W_2(\rho) + W_2(\rho)\tilde{A}_2(\rho) < -I \tag{10}$$

for $\rho \in (0, b]$. Though we pursue the general case where $W_1(\rho)$ and $W_2(\rho)$ are parameter varying, in practice it is desirable to restrict solutions of Eqs. (9) and (10) to be constant matrices. Scaling and summing the left-hand sides of Eqs. (9) and (10) yield

$$(1-\rho)\big(\tilde{A}_1^T(0)W_1(0) + W_1(0)\tilde{A}_1(0)\big) + \rho\big(\tilde{A}_2^T(1)W_2(1) + W_2(1)\tilde{A}_2(1)\big) < -I \tag{11}$$

for $\rho \in (0, 1)$. Define

$$\tilde{W}(\rho) = (1-\rho)W_1(0) + \rho W_2(1) \tag{12}$$

Then Eq. (11) can be written as

$$\big((1-\rho)\tilde{A}_1^T(0)W_1(0) + \rho\tilde{A}_2^T(1)W_2(1)\big)\tilde{W}^{-1}(\rho)\tilde{W}(\rho) + \tilde{W}(\rho)\tilde{W}^{-1}(\rho)\big((1-\rho)W_1(0)\tilde{A}_1(0) + \rho W_1(1)\tilde{A}_2(1)\big) < -I$$

which implies that

$$\tilde{A}_W(\rho) = \tilde{W}^{-1}(\rho)\big((1-\rho)W_1(0)\tilde{A}_1(0) + \rho W_2(1)\tilde{A}_2(1)\big)$$

is stable for each $\rho \in (0, 1)$. Define

$$\tilde{A}(\rho) = \begin{cases} \tilde{A}_1(\rho), & \rho \in [a, 0] \\ \tilde{A}_W(\rho), & \rho \in (0, 1) \\ \tilde{A}_2(\rho), & \rho \in [1, b] \end{cases} \tag{13}$$

$$\hat{W}(\rho) = \begin{cases} W_1(\rho), & \rho \in [a, 0] \\ \tilde{W}(\rho), & \rho \in (0, 1) \\ W_2(\rho), & \rho \in [1, b] \end{cases} \tag{14}$$

Then

$$\tilde{A}^T(\rho)\hat{W}(\rho) + \hat{W}(\rho)\tilde{A}(\rho) < -I \tag{15}$$

for $\rho \in [a, b]$. Let

$$\tilde{B}(\rho) = \begin{cases} \tilde{B}_1(\rho), & \rho \in [a, 0] \\ (1-\rho)\tilde{B}_1(0) + \rho\tilde{B}_2(1), & \rho \in (0, 1) \\ \tilde{B}_2(\rho), & \rho \in [1, b] \end{cases}$$

$$\tilde{C}(\rho) = \begin{cases} \tilde{C}_1(\rho), & \rho \in [a, 0] \\ (1-\rho)\tilde{C}_1(0) + \rho\tilde{C}_2(1), & \rho \in (0, 1) \\ \tilde{C}_2(\rho), & \rho \in [1, b] \end{cases}$$

$$\tilde{D}(\rho) = \begin{cases} \tilde{D}_1(\rho), & \rho \in [a, 0] \\ (1-\rho)\tilde{D}_1(0) + \rho\tilde{D}_2(1), & \rho \in (0, 1) \\ \tilde{D}_2(\rho), & \rho \in [1, b] \end{cases}$$

Then

$$Q(\rho) = \begin{cases} \dot{z}(t) = \tilde{A}(\rho)z(t) + \tilde{B}(\rho)y(t) \\ u(t) = \tilde{C}(\rho)z(t) + \tilde{D}(\rho)y(t) \end{cases} \tag{16}$$

is a stable system for each $\rho \in [a, b]$, and by Lemma 2.2

$$\Lambda(\rho) = \mathcal{F}_l(J(\rho), Q(\rho)) \tag{17}$$

stabilizes $\Sigma(\rho)$ for each $\rho \in [a, b]$. By construction, the matrix-valued coefficients of $\Lambda(\rho)$ are continuous functions of ρ. □

We now show the existence of K and L claimed in the proof of Theorem 3.1. Though there are a number of possibilities for computing K and L, we present a method that relies on the data required for $\mathcal{J}$-$\mathcal{Q}$ interpolation.

Remark 3.2 [existence of $K(\rho)$ and $L(\rho)$]: From Eqs. (9) and (10), let $W_1(\rho)$ and $W_2(\rho)$ (both in $\mathbb{R}^{(n+n_k)\times(n+n_k)}$) be partitioned

$$W_i(\rho) = \begin{bmatrix} S_i(\rho) & N_i(\rho) \\ N_i^T(\rho) & P_i(\rho) \end{bmatrix}, \qquad W_i^{-1}(\rho) = \begin{bmatrix} R_i(\rho) & M_i(\rho) \\ M_i^T(\rho) & Q_i(\rho) \end{bmatrix}$$

for $i = 1, 2$ with $S_i(\rho) \in \mathbb{R}^{n\times n}$ and $P_i(\rho) \in \mathbb{R}^{n_k}$ and define

$$L_i(\rho) = G_2(\rho)D_i - S_i^{-1}(\rho)N_i(\rho)^T B_i$$

$$K_i(\rho) = D_i H_2(\rho) + C_i M_i^T(\rho)R_i^{-1}(\rho)$$

Then Eqs. (9) and (10) together with Theorem 2.2 of Ref. 14 imply that

$$L(\rho) = \begin{cases} L_1(\rho), & \rho \in [a, 0) \\ S^{-1}(\rho)((1-\rho)S_1(\rho)L_1(\rho) + \rho S_2(\rho)L_2(\rho)), & \rho \in [0, 1] \\ L_2(\rho), & \rho \in (0, b] \end{cases}$$

$$K(\rho) = \begin{cases} K_1(\rho), & \rho \in [a, 0) \\ ((1-\rho)K_1(\rho)R_1(\rho) \\ \quad +\rho K_2(\rho)R_2(\rho))R(\rho)^{-1}(\rho), & \rho \in [0, 1] \\ K_2(\rho), & \rho \in (0, b] \end{cases}$$

are the required matrices where

$$S(\rho) = (1-\rho)S_1(\rho) + \rho S_2(\rho) \tag{18}$$

$$R(\rho) = (1-\rho)R_1(\rho) + \rho R_2(\rho) \tag{19}$$

Furthermore, it can be shown that Eqs. (9) and (10) imply

$$(F(\rho) + G_2(\rho)K(\rho))^T R^{-1}(\rho) + R^{-1}(\rho)(F(\rho) + G_2(\rho)K(\rho)) < -I \tag{20}$$

and

$$(F(\rho) + L(\rho)H_2(\rho))^T S(\rho) + S(\rho)(F(\rho) + L(\rho)H_2(\rho)) < -I \tag{21}$$

verifying that $F(\rho) + G_2(\rho)K(\rho)$ and $F(\rho) + L(\rho)H_2(\rho)$ are stable for $\rho \in [a, b]$. □

IV. Interpolation of More Than Two Controllers

For clarity, $\mathcal{J}$-$\mathcal{Q}$ interpolation is derived in Sec. III for two controllers over a single interpolation region. Extensions of $\mathcal{J}$-$\mathcal{Q}$ interpolation for the case that ρ is a scalar and there are multiple controllers, and thus multiple interpolations regions, are discussed in Ref. 10. Here, we briefly discuss the vector-scheduling variable case when $\rho \in \mathbb{R}^l$, $l = 2$. The case when $l > 2$ is completely analogous.

Suppose that $\rho = [\rho_1, \rho_2]^T$, the interpolation region is the square defined

$$\Phi = \{\rho \mid 0 \leq \rho_1 \leq 1, 0 \leq \rho_2 \leq 1\}$$

and that four LTI controllers, Λ_1, Λ_2, Λ_3, and Λ_4, have been designed for $\rho_1 = [0, 0]^T$, $\rho_2 = [1, 0]^T$, $\rho_3 = [1, 1]^T$, and $\rho_4 = [0, 1]^T$, respectively. We assume that the stability-covering condition is satisfied and search for solutions of the Lyapunov inequalities in Eqs. (9) and (10).

$$\tilde{A}_i^T(\rho)W_i(\rho) + W_i(\rho)\tilde{A}_i^T(\rho) < -I, \qquad i = 1, \ldots, 4 \tag{22}$$

where $\tilde{A}_i$ is composed of the matrix coefficients of the plant and LTI controller Λ_i. Then $\tilde{W}(\rho)$ in Eq. (12) is written

$$\tilde{W}(\rho) = (1-\rho_1)(1-\rho_2)W_1(\rho)(0,0) + \rho_1(1-\rho_2)W_2(1,0) + \rho_1\rho_2 W_3(1,1) + (1-\rho_1)\rho_2 W_4(0,1)$$

and $\tilde{A}_W(\rho)$ in Eq. (13) is written

$$\tilde{A}_W(\rho) = \tilde{W}^{-1}(\rho)((1-\rho_1)(1-\rho_2)W_1(\rho)(0,0)\tilde{A}_1(\rho) + \rho_1(1-\rho_2)W_2(1,0)\tilde{A}_2(\rho) + \rho_1\rho_2 W_3(1,1)\tilde{A}_3(\rho) + (1-\rho_1)\rho_2 W_4(0,1)\tilde{A}_4(\rho))$$

The matrix-valued functions $\tilde{B}(\rho)$, $\tilde{C}(\rho)$, and $\tilde{D}(\rho)$ are defined similarly.

V. Example

Features of $\mathcal{J}$-$\mathcal{Q}$ interpolation are illustrated with the design of a gain-scheduled autopilot for a pitch-axis missile. Autopilots for this missile model have also been presented in other works.[3,13,14] The missile equations of motion are

$$\dot{\alpha}(t) = K_\alpha M C_n(\alpha(t), \delta(t), M)\cos(\alpha(t)) + q(t)$$

$$\dot{q}(t) = K_q M^2 C_m(\alpha(t), \delta(t), M)$$

$$\eta(t) = K_z M^2 C_n[\alpha(t), \delta(t), M]$$

where α is angle of attack, q is pitch rate, δ is the tailfin deflection angle, and η is normal acceleration. We assume that Mach number M is fixed at Mach 3. The constants K_α, K_q, and K_z, along with the coefficients $C_n(\cdot,\cdot,\cdot)$ and $C_m(\cdot,\cdot,\cdot)$ are found in Ref. 3. A second-order unity-gain tailfin actuator with transfer function

$$\frac{22{,}500}{s^2 + 210s + 22{,}500} \tag{23}$$

is assumed. The transfer function in Eq. (23) relates tailfin angle commands δ_c generated by the autopilot to the actual tailfin angle δ.

The closed-loop system is shown in Fig. 1. The autopilot generates tailfin deflection commands so that the missile tracks normal acceleration commands. (Normal acceleration is the component of acceleration normal to the centerline of the missile). Normal acceleration error $(\eta - \eta_c)$ and pitch rate (q) are measured and fed back to the autopilot.

Equilibrium conditions of the nonlinear plant model are parameterized about the angle of attack, denoted α. Two LTI controllers, Λ_0 and Λ_{20}, are designed for the linearized plant models corresponding to $\alpha = 0$ and 20 deg, respectively. An LTI controller is not needed for $\alpha < 0$ because the missile model is symmetric about $\alpha = 0$ and the controllers can be scheduled on $|\alpha|$.

The LTI controllers are designed using the loop-shaping $\mathcal{H}_\infty$ method of McFarlane and Glover.[16] Because our emphasis is on the interpolation problem, we provide only a cursory overview of the LTI controller design process. The McFarlane–Glover algorithm is applied to the shaped plant $N(s)G(s)$, where $G(s)$ is the transfer function of the linearized plant cascaded with the tailfin actuator and $N(s)$ is the shaping filter

$$N(s) = \begin{bmatrix} q_2[(s+q_1)/s] & 0 \\ 0 & q_5[(s+q_3)/(s+q_4)] \end{bmatrix}$$

where the coefficients q_i, $i = 1, \ldots, 5$ are given in Table 1 for each of the two LTI controller designs.

Table 1 Poles, zeros, and gains of $N(s)$

Angle of attack	q_1	q_2	q_3	q_4	q_5
0	114	0.178	16.00	6.60	0.755
20	127	0.231	3.30	1.06	0.341

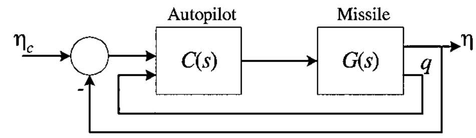

Fig. 1 Autopilot topology.

LTI controllers are obtained for $\alpha = 0$ and 20 deg and cascaded with the weighting filter $N(s)$ to form the autopilot $C(s)$. The LTI autopilot has state dimension 8. Model reduction techniques are used to obtain autopilots with state dimension 5 from the designed autopilots. Step responses of the closed-loop linear system, with reduced-order linear autopilots, are shown in Fig. 2.

By virtue of the McFarlane–Glover loop-shaping design method, the original autopilots are full-order and have an observer-state feedback structure, but both of these properties are lost as a result of the model reduction process. Thus, interpolation of observer and state feedback gains[4,14] and full-order state-space interpolation[10] cannot be utilized. Though our model reduction efforts may seem somewhat artificial in this context, we have purposefully generated an

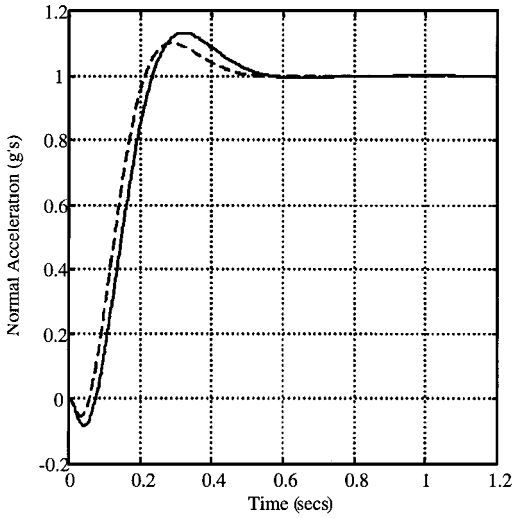

Fig. 2 Step response of linear closed-loop system: - - - -, α = 20, and ——, α = 0.

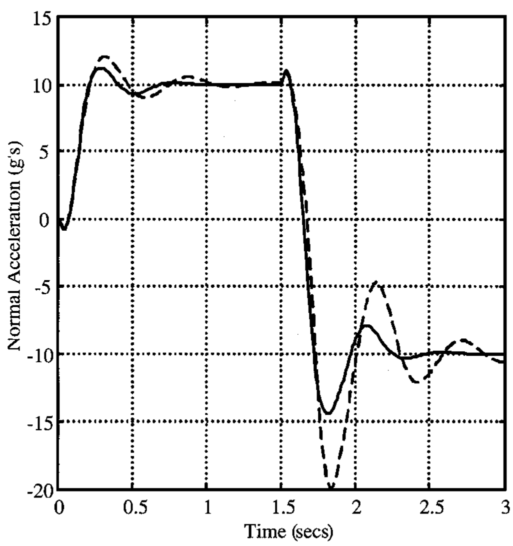

Fig. 3 Normal acceleration of closed-loop nonlinear system: - - - -, linear interpolation, and ——, $\mathcal{J}$-$\mathcal{Q}$ interpolation.

autopilot that is not full-order and does not have an observer-state feedback structure. Thus, previously reported stability-preserving interpolation methods[10,14] are not suitable.

$\mathcal{J}$-$\mathcal{Q}$ interpolation of the two LTI controllers proceeds as described in Sec. III. We first verify that the stability-covering condition is satisfied and find that Λ_0 and Λ_{20} stabilize the family of linear plants for $\alpha \in [0, 20]$ and [5, 20] deg, respectively. Symmetric positive-definite matrices W_0 and W_{20} are computed that satisfy Eq. (9) for $\alpha \in [0, 20]$ and Eq. (10) for $\alpha \in [5, 20]$. Obtaining constant solutions to Eqs. (9) and (10) simplifies the equations appearing in the proof of Theorem 3.1 and Remark 3.2. The two LTI autopilots are interpolated over the intersection of the stability regions, $\alpha \in [5, 20]$. For example, Eq. (13) in Sec. III is written

$$\tilde{A}(\rho) = \begin{cases} \tilde{A}_0(\rho), & |\alpha| \in [0, 5] \\ [(20 - |\alpha|)/15]\hat{A}_0 + [(|\alpha| - 5)/15]\hat{A}_{20}(\rho), & |\alpha| \in (5, 20) \\ \tilde{A}_{20}(\rho), & |\alpha| > 20 \end{cases}$$

where $\hat{A}_0$ and $\hat{A}_{20}$ are composed of the state-space matrices for Λ_0 and Λ_{20}, respectively, and the linearized plant family. Gain matrices L and K are computed as in Remark 3.2 and the gain-scheduled controller is implemented $\mathcal{F}(J(\rho), Q(\rho))$, where $J(\rho)$ and $Q(\rho)$ are defined in Eqs. (7) and (16), respectively.

For purposes of comparison, we also interpolate the two LTI autopilots by linearly interpolating matrix coefficients over the interval [0, 20] and simulate the nonlinear closed-loop system.

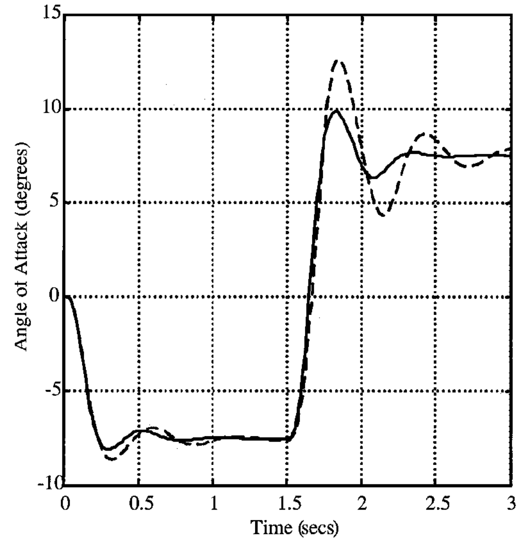

Fig. 4 Angle of attack of closed-loop nonlinear system: - - - -, linear interpolation, and ——, $\mathcal{J}$-$\mathcal{Q}$ interpolation.

Figure 3 shows the response of the nonlinear closed-loop system to a 10-g normal acceleration step command at 0 s followed by a -20-g step command at 1.5 s. The nonlinear system with $\mathcal{J}$-$\mathcal{Q}$-interpolated controller has significantly less overshoot and shorter settling time than that of the linearly interpolated controller. Figure 4 shows the angle of attack (scheduling variable) for the same step commands. Though the performance of the nonlinear system with $\mathcal{J}$-$\mathcal{Q}$-interpolated autopilot was superior to that of the nonlinear system with linearly interpolated autopilot, the $\mathcal{J}$-$\mathcal{Q}$-interpolated autopilot had larger state dimension. The $\mathcal{J}$-$\mathcal{Q}$-interpolated autopilot has 13 states, but the linearly interpolated autopilot has 5 states.

VI. Conclusions

A theoretically justified method for interpolating LTI controllers that is suitable for the synthesis of gain-scheduled controllers has been presented. The stability-covering condition, a sufficient condition for the selection of operating points at which the LTI controllers are designed, must be satisfied for the interpolation to be successful. As a design tool, the stability-covering condition helps the design engineer determine if additional LTI controllers are required and at which operating points they should be designed. From a practical standpoint, $\mathcal{J}$-$\mathcal{Q}$ interpolation requires that a parameter-varying Lyaponuv inequality is solved for each LTI controller. Choosing the sets over which the Lyapunov inequalities are solved requires a certain amount of engineering judgment and possible iteration. At present, this fact limits the applicability of $\mathcal{J}$-$\mathcal{Q}$ interpolation for systems that require many LTI controllers. Our current research is focused on resolving this limitation.

References

[1]Rugh, W. J., "Analytical Framework for Gain Scheduling, *IEEE Control Systems Magazine*, Vol. 11, No. 1, 1991, pp. 79–84.

[2]Shamma, J. S., and Athans, M., "Analysis of Gain Scheduled Control for Nonlinear Plants," *IEEE Transactions on Automatic Control*, Vol. 35, No. 8, 1990, pp. 898–907.

[3]Nichols, R. A., Reichert, R. T., and Rugh, W. J., "Gain Scheduling for H-Infinity Controllers: A Flight Control Example," *IEEE Transactions on Control Systems Technology*, Vol. 1, No. 2, 1993, pp. 69–79.

[4]Hyde, R. A., and Glover, K., "The Application of Scheduled H-Infinity Controllers to a VSTOL Aircraft," *IEEE Transactions on Automatic Control*, Vol. 38, No. 7, 1993, pp. 1021–1039.

[5]Kellet, M. G., "Continuous Scheduling of H-Infinity Controllers for a MS760 Paris Aircraft," *Robust Control System Design Using H-Infinity and Related Methods*, edited by P. H. Hammond, Inst. of Measurement and Control, London, 1991, pp. 197–219.

[6]Reichert, R. T., "Dynamic Scheduling of Modern-Robust-Control Autopilot Designs for Missiles," *IEEE Control Systems Magazine*, Vol. 12, No. 5, 1992, pp. 35–42.

[7]Buschek, H., "Robust Autopilot Design for Future Missile Systems," *AIAA Guidance, Navigation, and Control Conference*, AIAA, Washington, DC, 1997, pp. 1672–1681.

[8]Kelly, J. H., and Evers, J. H., "An Interpolation Strategy for Scheduling Dynamic Compensators," *AIAA Guidance, Navigation, and Control Conference*, AIAA, Washington, DC, 1997, pp. 1682–1690.

[9]Lawrence, D. A., Kelly, J. H., and Evers, J. H., "Gain Scheduled Missile Autopilot Design Using a Control Signal Interpolation," *AIAA Guidance, Navigation, and Control Conference*, AIAA, Washington, DC, 1998, pp. 1394–1402.

[10]Stilwell, D. J., and Rugh, W. J., "Interpolation Methods for the Synthesis of Gain Scheduled Controllers," *Automatica*, Vol. 36, No. 5, 2000, pp. 665–671.

[11]Shahruz, S. M., and Behtash, S., "Design of Controllers for Linear Parameter-Varying Systems by the Gain Scheduling Technique," *Journal of Mathematical Analysis and Applications*, Vol. 168, No. 1, 1992, pp. 125–217.

[12]Apkarian, P., Gahinet, P., and Becker, G., "Self-Scheduled $\mathcal{H}_\infty$ Control of Linear Parameter-Varying Systems: A Design Example," *Automatica*, Vol. 31, No. 9, 1995, pp. 1251–1261.

[13]Wu, F., Packard, A., and Balas, G., "LPV Control Design for Pitch-Axis Missile Autopilots," *Proceedings of the 34th IEEE Conference on Decision and Control*, Inst. of Electrical and Electronics Engineers, New York, 1995, pp. 188–193.

[14]Stilwell, D. J., and Rugh, W. J., "Interpolation of Observer State Feedback Controllers for Gain Scheduling," *IEEE Transactions on Automatic Control*, Vol. 44, No. 6, 1999, pp. 1225–1229.

[15]Zhou, K., Doyle, J. C., and Glover, K., *Robust and Optimal Control*, Prentice–Hall, Upper Saddle River, NJ, 1996.

[16]McFarlane, D. C., and Glover, K., *Robust Controller Design Using Normalized Coprime Factor Plant Descriptions*, Springer-Verlag, New York, 1989.