Missile autopilot design via a multi-channel LFT/LPV control method

Paulo C. Pellanda[*] *Pierre Apkarian*[‡] *Hoang Duong Tuan*[¶]

Abstract

In this paper, the missile pitch-axis autopilot design is revisited using a new and recently available Linear Parameter-Varying control (LPV) technique. The missile plant model is characterized by a Linear Fractional Transformation (LFT) representation. The synthesis task is conducted by exploiting new capabilities of the LPV method: firstly, a set of H_2/H_∞ criteria defined channel-wise are considered; secondly, different Lyapunov and scaling variables are used for each channel/specification which is known to reduce conservatism; and finally, the controller gain-scheduling function is constructed as affine matrix-valued function in the polytopic coordinates of the scheduled parameter. All these features are examined and evaluated in turn for the missile control problem. The method is shown to provide additional flexibility to tradeoff conflicting and demanding performance and robustness specifications for the missile while preserving the practical advantage of previous single-objective LPV methods. Finally, the method is shown to perform very satisfactorily for the missile autopilot design over a wide range of operating conditions.

Key words: missile autopilots, LPV synthesis, LFT, mixed H_2/H_∞, multi-channel control, gain scheduling.

1 Introduction

Gain-scheduling techniques and LPV control theory have been used extensively for the synthesis of non-linear controllers and especially in designing missile autopilots [14, 17, 20, 21, 5, 19, 22, 23]. Despite this popularity, missile autopilot design remains a challenging control problem since it operates over a wide range of flight conditions and tight design specifications are generally prescribed. A consequence is that fast on-line controller adjustment is necessary to achieve this goal.

Classical gain scheduling is typically based on a family of linear controllers designed on a set of plant equilibrium conditions [11, 15, 18]. Modern optimal and robust control techniques can be employed to locally satisfy several robustness and performance specifications. A transition law is then established to switch or interpolate the resulting set of Linear Time-Invariant (LTI) controllers according to the evolution of the (on-line available) scheduling variables. This approach often generates alterations of the LTI stability/performance properties when non-stationary maneuver are investigated. Unfortunately, the loss of performance cannot be measured at the synthesis level

[*]Supported by Brazilian Ministry of Defense at ONERA-CERT, Control System Dept., 2 av. Edouard Belin, 31055 Toulouse, FRANCE - Email : pellanda@cert.fr - Tel : +33 5.62.25.25.25 - Fax : +33 5.62.25.25.64

[‡]ONERA-CERT, Control System Dept., 2 av. Edouard Belin, 31055 Toulouse, FRANCE - Email : apkarian@cert.fr - Tel : +33 5.62.25.27.84 - Fax : +33 5.62.25.25.64

[¶]Dept. of Control and Information, Toyota Technological Institute, Hisakata 2-12-1, Tenpaku, Nagoya 468-8511 JAPAN - Email : tuan@toyota-ti.ac.jp - Tel: +81 52.809.1815 - Fax : +81 52.809.1721

and extensive simulations are required to evaluate the closed-loop behavior for a rich enough set of potential trajectories. Moreover, when scheduling variables have fast dynamics, the number of LTI controllers must be considerably increased to ensure satisfactory non-stationary performance, hence the practical restriction of the scheme.

In contrast to these techniques, the LPV framework provides elegant and solidly founded methodologies to meet design specifications over wide operating ranges. Control problems are formulated as Linear Matrix Inequalities (LMI) optimization problems [7, 10], which are then solved very efficiently using currently available semi-definite programming codes. Many techniques are now available to handle the gain-scheduling problem. Among these, some assume an LFT plant representation [13, 2], while others are much more general [4, 24, 1]. Several variants have been proposed to improve these original approaches. A current research direction consists of seeking methods which provide a reasonable compromise between the conservatism of LFT-based techniques and the more sophisticated general LPV approaches.

In [3] a technique for solving mixed H_2/H_∞ multi-channel LFT/LPV control problem in discrete-time has been derived. It can be viewed as an extension of LFT/LPV single-objective results in [2, 13] and of nominal multi-objective techniques in [9, 16]. A practically interesting capability of this method is to offer additional flexibility to tradeoff various performance and robustness specifications. Similarly to the LTI case, different Lyapunov and scaling variables are used for each channel/specification to reduce conservatism as compared to earlier methods. In this paper, we discuss its applicability to a realistic LPV missile autopilot design. We show how the method can be used for discrete- or continuous time LPV systems. We paid a special attention to the controller construction and implementation which are of prior importance in missile problems.

The remainder of the paper is organized as follows. The problem statement and a brief review of the results in [3] are presented in Section 2. In Section 3, useful and practical techniques for the discretization of continuous systems and for the computation of polytopic coordinates are developed. In Section 4, we investigate the application to the missile autopilot problem and discuss how the multi-objective/channel features of the LPV method are central to achieve good design properties. Concluding remarks are given in Section 5.

The notation used throughout the paper is standard. M^T is the transpose of the matrix M. For Hermitian or real symmetric matrices, $M > N$ means that $M - N$ is positive definite and $M \geq N$ means that $M - N$ is positive semi-definite. $x(k)$ is used to denote the signal x at (discrete) time k. For an appropriately dimensioned matrix $P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$ and assuming the inverses exist, $\mathcal{F}_u(P,\Delta) = P_{22} + P_{21}\Delta(I - P_{11}\Delta)^{-1}P_{12}$ defines the upper LFT operator and similarly $\mathcal{F}_l(P,\Delta) = P_{11} + P_{12}\Delta(I - P_{22}\Delta)^{-1}P_{21}$ defines the lower LFT.

2 Multi-objective LFT/LPV result

In this section, we state the multi-objective LPV control problem and give a brief overview of the synthesis method in [3].

Consider a discrete-time LPV plant with LFT structure

$$\begin{bmatrix} x(k+1) \\ z_\Delta(k) \\ z(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} A & B_\Delta & B_1 & B_2 \\ C_\Delta & D_{\Delta\Delta} & D_{\Delta 1} & D_{\Delta 2} \\ C_1 & D_{1\Delta} & D_{11} & D_{12} \\ C_2 & D_{2\Delta} & D_{21} & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ w_\Delta(k) \\ w(k) \\ u(k) \end{bmatrix}, \qquad (1)$$

$$w_\Delta(k) = \Delta(k)\, z_\Delta(k),$$

where $A \in \mathbf{R}^{n\times n}$, $\Delta(k) \in \mathbf{R}^{N\times N}$, $D_{12} \in \mathbf{R}^{p_1\times m_2}$ and $D_{21} \in \mathbf{R}^{p_2\times m_1}$ define the problem dimension. The notation for signals is standard:

- x for the state vector,
- w for exogenous inputs,
- z for controlled or performance variables,
- u for the control signal,
- y for the measurement signal.

$\Delta(k)$ is a time-varying matrix-valued parameter evolving in a polytopic set $\mathcal{P}_{\boldsymbol{\Delta}}$, with

$$\mathcal{P}_{\boldsymbol{\Delta}} := \mathrm{co}\,\{\Delta_1, \ldots, \Delta_i, \ldots, \Delta_L\} \ni 0\,, \tag{2}$$

where co stands for the convex hull and the Δ_i's denote the vertices of $\mathcal{P}_{\boldsymbol{\Delta}}$. That is,

$$\Delta := \sum_{i=1}^{L} \alpha_i \Delta_i, \qquad \sum_{i=1}^{L} \alpha_i = 1, \tag{3}$$

where $\alpha_i \geq 0$ are the polytopic coordinates of Δ. Polytopic coordinates are computed in real time as functions of the scheduling variables (Section 3) and can be exploited by the controller. According to our definitions, the pair (w_Δ, z_Δ) is the gain-scheduling channel.

For the LPV plant (1) the gain-scheduling control problem consists in seeking an LPV controller with LFT structure

$$\begin{aligned} \begin{bmatrix} x_K(k+1) \\ u(k) \\ z_K(k) \end{bmatrix} &= \begin{bmatrix} A_K & B_{K1} & B_{K\Delta} \\ C_{K1} & D_{K11} & D_{K1\Delta} \\ C_{K\Delta} & D_{K\Delta 1} & D_{K\Delta\Delta} \end{bmatrix} \begin{bmatrix} x_K(k) \\ y(k) \\ w_K(k) \end{bmatrix}, \qquad A_K \in \mathbf{R}^{n\times n} \\ w_K(k) &= \Delta_K(k) z_K(k), \qquad \Delta_K \in \mathbf{R}^{N\times N}, \end{aligned} \tag{4}$$

such that H_2 and/or H_∞ specifications are achieved for a family of channels (w_j, z_j), $j = 1, 2, \cdots$, where the w_j's and z_j's are sub-vectors of w and z, respectively (Figure 1). In other words, bounds ν_j on the variance of the outputs z_j and/or bounds γ_j on the L_2-induced gain of the operator mapping w_j into z_j are guaranteed for all parameter trajectories $\Delta(k) \in \mathcal{P}_{\boldsymbol{\Delta}}$. The notation Δ_K is used for the controller gain-scheduling function which is a function of the plant's parameter Δ, that is, $\Delta_K := \Delta_K(\Delta)$.

In this application, we consider the special situation in which Δ has a block-diagonal structure determined by a vector of parameters $\theta := (\theta_1, \cdots, \theta_r)^T$ with

$$\Delta = \mathrm{diag}(\theta_1 I_{s_1}, \cdots, \theta_r I_{s_r})\,. \tag{5}$$

Also, we assume that θ evolves in a box defined as

$$\theta_l \in [\underline{\theta}_l, \bar{\theta}_l], \quad \underline{\theta}_l < \bar{\theta}_l, \quad \forall t \geq 0. \tag{6}$$

The assumptions (5) and (6) mean that:

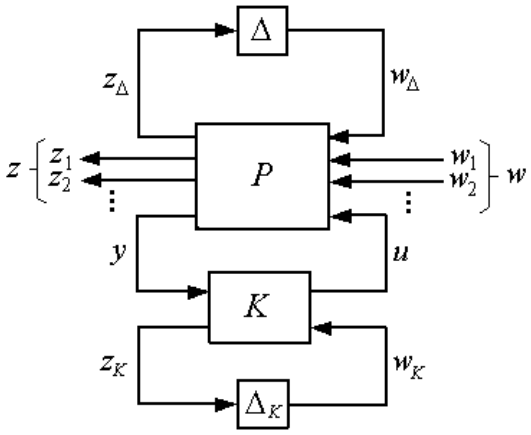


Figure 1: Mixed H_2/H_∞ multi-channel LPV interconnection

- the time-varying parameter θ is valued in a hyper-rectangle $\mathcal{P}_{\boldsymbol{\Theta}}$ of $\mathbf{R}^r$, with

$$\mathcal{P}_{\boldsymbol{\Theta}} := \mathrm{co}\,\{\Theta_1, \ldots, \Theta_L\}\ , \tag{7}$$

  where the Θ_i's are the vertices of $\mathcal{P}_{\boldsymbol{\Theta}}$;

- Δ and θ have the same polytopic coordinates $\{\alpha_i\}$;

- $L = 2^r$ and $N = \sum_{l=1}^{r} s_l$.

Hereafter, $i\ (= 1, \cdots, L)$ indexes the vertices Θ_i and Δ_i, $j\ (= 1, 2, \cdots)$ indexes the channels and specifications, and $l\ (= 1, \cdots, r)$ indexes the parameters.

It is shown in [3] that sufficient conditions for the existence of a solution to the multi-objective LPV control problem can be written as an LMI program. The general synthesis scheme is described below.

Algorithm 2.1 *Controller synthesis*

Step 1: *Define the following general non-symmetric decision variables which are common to all specifications and channels (Table 1):*

- *the set $\boldsymbol{S_v}$ of general slack variables; the set $\boldsymbol{K_v}$ of transformed controller variables, whose dimensions must be defined in according to the controller dimensions; and the set $\boldsymbol{\Delta_{Kv}}$ of scheduling function coefficients.*

Step 2: *For each H_2-channel, define the set $\boldsymbol{H_{2v}}$ of the following symmetric decision variables:*

- *Lyapunov variables ($\boldsymbol{X_{2j}}$ and $\boldsymbol{Z_j}$); scaling variables ($\boldsymbol{Q_{1j}}$, $\boldsymbol{Q_{2j}}$, $\boldsymbol{R_{1j}}$ and $\boldsymbol{R_{2j}}$); and a performance variable (ν_j).*

Step 3: *For each H_∞-channel, define the set $\boldsymbol{H_{\infty v}}$ of the following symmetric decision variables:*

- *a Lyapunov variable ($\boldsymbol{X_{\infty j}}$); scaling variables ($\boldsymbol{Q_{\infty j}}$ and $\boldsymbol{R_{\infty j}}$); and a performance variable (γ_j).*

Step 4: *For each channel/specification, construct the LMI constraint system derived in Appendix A of [3] and represented here by the simple notations below:*

Table 1: Decision variables

Set	Variables	Dimension	Number of scalar variables
$\boldsymbol{S_v}$	$\boldsymbol{U}$, V_{11}, W_{11}	$n \times n$	$3n^2$
	$\boldsymbol{M}$, $\boldsymbol{N}$, E_{11}, F_{11}, G_{11}, H_{11}	$N \times N$	$6N^2$
$\boldsymbol{K_v}$	$\boldsymbol{A_K}$, $\boldsymbol{B_{K1}}$, $\boldsymbol{B_{K\Delta}}$, $\boldsymbol{C_{K1}}$, $\boldsymbol{C_{K\Delta}}$, $\boldsymbol{D_{K11}}$, $\boldsymbol{D_{K1\Delta}}$, $\boldsymbol{D_{K\Delta 1}}$, $\boldsymbol{D_{K\Delta\Delta}}$	appropriate	$n^2 + N^2 + 2nN +$ $(n+N)(m_2+p_2) + m_2p_2$
$\boldsymbol{\Delta_{Kv}}$	$\boldsymbol{\Delta_{K,i}}$, $i = 1, \cdots, L$	$N \times N$	LN^2
$\boldsymbol{H_{2v}}$	$\boldsymbol{X_{2j}}$	$2n \times 2n$	$n(2n+1)$
	$\boldsymbol{Z_j}$	$p_{1j} \times p_{1j}$	$p_{1j}(p_{1j}+1)/2$
	$\boldsymbol{Q_{1j}}$, $\boldsymbol{Q_{2j}}$, $\boldsymbol{R_{1j}}$, $\boldsymbol{R_{2j}}$	$2N \times 2N$	$4N(2N+1)$
	ν_j	scalar	1
$\boldsymbol{H_{\infty v}}$	$\boldsymbol{X_{\infty j}}$	$2n \times 2n$	$n(2n+1)$
	$\boldsymbol{Q_{\infty j}}$, $\boldsymbol{R_{\infty j}}$	$2N \times 2N$	$2N(2N+1)$
	γ_j	scalar	1

- *H_2 performance:*

$$\mathcal{L}_{H_2}(\boldsymbol{S_v}, \boldsymbol{K_v}, \boldsymbol{\Delta_{Kv}}, \boldsymbol{H_{2v}}, \Delta_i, P_j) \; < \; 0 \tag{8}$$

- *H_∞ performance:*

$$\mathcal{L}_{H_\infty}(\boldsymbol{S_v}, \boldsymbol{K_v}, \boldsymbol{\Delta_{Kv}}, \boldsymbol{H_{\infty v}}, \Delta_i, P_j) \; < \; 0 \tag{9}$$

  where P_j is the set of state-space matrices representing the LPV plant (1) with only the channel/specification (w_j, z_j) under consideration.

Step 5: *(LMI optimization problem) - Minimize a specific performance variable γ_j or ν_j subject to the LMI constraints (8)-(9), fixing the remaining performance variables at some adequate set of values; or simply compute a feasible solution to the LMI constraints (8)-(9).*

Step 6: *As described in [3], compute the LPV controller data (4) as functions of the decision variables (Table 1) obtained in Step 5. Note that the set $\boldsymbol{K_v}$ (bold notation) does not represent the set of controller data. The controller gain-scheduling function is determined by*

$$\Delta_K(\Delta) := \sum_{i=1}^{L} \alpha_i \Phi_i, \tag{10}$$

where the Φ_i's can be computed off line as functions of the decision variables.

Remark: The introduction of general matrix variables ($\boldsymbol{S_v}$) and of linearizing transformations variables ($\boldsymbol{K_v}$) allows the use of multiple Lyapunov functions and scalings and leads to a full LMI characterization of the LPV control problem. The price to pay for these new capabilities comes in terms of an extra computational overhead in the synthesis step. The matrix variables whose dimensions depend on N increase considerably the dimension of the overall decision vector in the LMI problem (see Table 1) and appears as a limitation for problems of large dimensions with today semi-definite programming solvers.

3 Discretization and polytopic coordinates

In this section, techniques for the discretization of continuous-time systems and for the computation of polytopic coordinates are discussed. They will be used in the missile application (Section 4).

3.1 Discretization

While genuine extensions of the foregoing method to the continuous-time case remain challenging, it can be indirectly applied to continuous plants with the help of a formal bilinear transformation. Continuous-time controllers can be synthesized since H_2 problems are properly posed in continuous time.

The bilinear transformation between the s-domain and z-domain

$$\frac{1}{s} = \frac{z+1}{z-1} \tag{11}$$

can be written in the standard LFT form as

$$s^{-1}I \;=\; I + \sqrt{2}I \;\; z^{-1}I \;\; (I - z^{-1}I)^{-1} \;\; \sqrt{2}I \;\; := \;\; \mathcal{F}_u(\mathcal{B}, z^{-1}I),$$

where $\mathcal{F}_u$ is the customary notation for upper LFTs and $\mathcal{B} = \begin{bmatrix} I & \sqrt{2}I \\ \sqrt{2}I & I \end{bmatrix}$.

Now consider a continuous-time system

$$\begin{array}{rcl} \dot{x}(t) & = & \tilde{A}x(t) + \tilde{B}\xi(t) \\ \psi(t) & = & \tilde{C}x(t) + \tilde{D}\xi(t)\ , \end{array} \tag{12}$$

whose transfer matrix is

$$\tilde{P}(s) \;=\; \tilde{D} + \tilde{C}\left(sI - \tilde{A}\right)^{-1}\tilde{B} \;:=\; \mathcal{F}_u\left(\tilde{G}, s^{-1}I\right), \tag{13}$$

where $\tilde{G} := \begin{bmatrix} \tilde{A} & \tilde{B} \\ \tilde{C} & \tilde{D} \end{bmatrix}$. Then the corresponding discrete-time system is obtained as

$$P(z) \;=\; \mathcal{F}_u\left(\tilde{G}, \frac{z+1}{z-1}I\right) \;=\; \mathcal{F}_u\left(\tilde{G}, \mathcal{F}_u(\mathcal{B}, z^{-1}I)\right) \;=\; \mathcal{F}_u(G, z^{-1}I), \tag{14}$$

where

$$G := \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \mathcal{F}_u(\tilde{G}, \mathcal{B}) \tag{15}$$

represents a state-space realization of $P(z)$. A transformation from the z-domain to the s-domain can be obtained similarly. Hence, a discrete-time system represented by G has a corresponding continuous-time system represented by

$$\tilde{G} = \mathcal{F}_u(G, \tilde{\mathcal{B}}), \quad \tilde{\mathcal{B}} = \begin{bmatrix} -I & \sqrt{2}I \\ \sqrt{2}I & -I \end{bmatrix}. \tag{16}$$

Supposing now that $\xi(t) := [w_\Delta^T(t),\ w^T(t),\ u^T(t)]^T$ and $\psi(t) := [z_\Delta^T(t),\ z^T(t),\ y^T(t)]^T$ in (12), a corresponding discrete-time system in the form (1) is readily obtained by applying the bilinear transformation (15). Since H_2 problems are properly posed in continuous time, the methodology described in the previous section can be applied without restrictions to the transformed system. For

the H_2 performance index ν_j to be well defined in continuous time, the state-space data must be such that the closed-loop feedthrough term of the channel/specification j is zero. Without imposing restrictions to the controller, this is achieved with $\tilde{D}_{11j} = 0$ and either $\tilde{D}_{1\Delta j} = 0$ and $\tilde{D}_{12j} = 0$ or $\tilde{D}_{\Delta 1j} = 0$ and $\tilde{D}_{21j} = 0$.

It is well known that bilinear transformations applied to rational strictly proper transfer functions generate non strictly proper transfer functions. See the example of a pure integration in (11). However, there is no restrictions on the feedthrough matrices for defining H_2 specifications in discrete time and our method remains applicable.

Once the LFT discrete-time controller has been computed, one can use the transformation (16) to recover the corresponding continuous-time controller. It is worth mentioning that only the LTI components of the LFT plant and of the LFT controller are modified by bilinear transformations, whereas the Δ- and Δ_K-blocks remain unchanged.

3.2 Polytopic coordinates and hypercubes

Modern flight control systems undergo highly maneuverable trajectories which requires a fast controller update. Controllers designed through general LPV/gridding techniques [4, 24, 1] show little conservatism but require more complex on-line computations at the gain-scheduling level. Contrarily, LFT/LPV controllers are often more conservative but their favorable LFT structure offers obvious advantages in this respect. In comparison with the single-objective H_∞ LFT/LPV control methods [2, 13], the foregoing mixed H_2/H_∞ multi-objective approach allows to consider a richer class of scheduling functions (10), instead of replicating the parameter block of the plant ($\Delta_K := \Delta$). This is another factor which reduces conservatism and that is immediately penalized by an increase in complexity of on-line computations. Fast algorithms for the calculation of polytopic coordinates should therefore be utilized in order to overcome this difficulty.

For a parameter evolving in a hyper-rectangle, barycentric coordinates can be directly and quickly computed by ratio of hyper-volumes. Consider the two-dimensional example depicted in Figure 2, where $\theta_1 \in [-5, 3]$ and $\theta_2 \in [5, 10]$. The vertices of the parameter box are

$$\{\Theta_1, \cdots, \Theta_4\} = \{[-5, 5]^T, [3, 5]^T, [-5, 10]^T, [3, 10]^T\} .$$

Any point θ into this parameter domain can be used to define 4 sub-rectangles whose areas are V_l, $l = 1, \cdots, 4$. Note that V_l corresponds to the sub-rectangle opposite of Θ_l. Polytopic coordinates of θ can be computed in terms of the ratio between each area V_l and the total area $V = \sum_{l=1}^{4} V_l$. That is,

$$\alpha_l = \frac{V_l}{V} , \quad l = 1, \cdots, 4 , \quad \text{with} \quad \theta = \sum_{l=1}^{4} \alpha_l \Theta_l .$$

As an example, the point $\theta = [-3\,,\ 7]^T$ can be represented by its polytopic coordinates

$$\alpha = \left[\frac{18}{40}, \frac{6}{40}, \frac{12}{40}, \frac{4}{40}\right]^T = [0.45\,,\ 0.15\,,\ 0.3\,,\ 0.1]^T .$$

The following algorithm extends this procedure for general hyper-rectangles (6) with vertices in (7):

Algorithm 3.1 *Computation of polytopic coordinates*

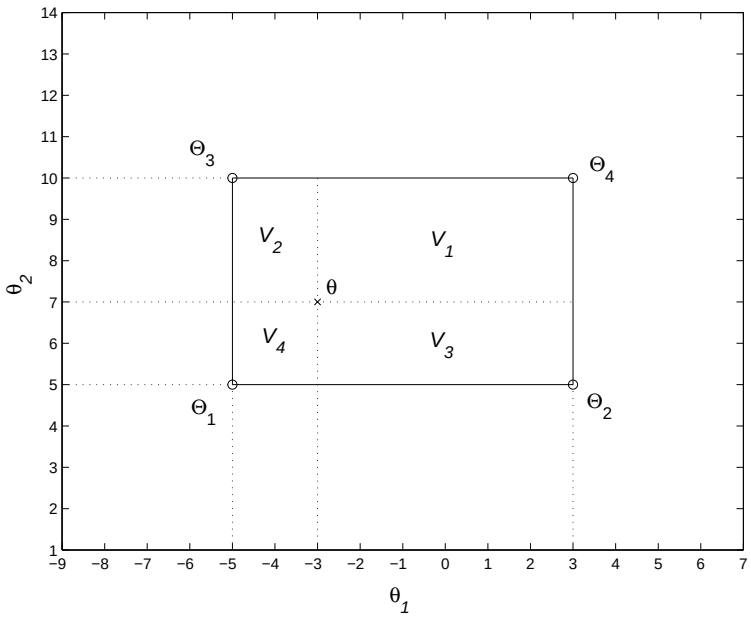


Figure 2: Parameter domain in the space $\mathbf{R}^2$

Step 1: *Given a parameter* $\theta := (\theta_1, \cdots, \theta_r)^T$, *compute its normalized coordinates*

$$\vartheta_l := \frac{\left(\bar{\theta}_l - \theta_l\right)}{\left(\bar{\theta}_l - \underline{\theta}_l\right)} , \quad l = 1, \cdots, r .$$

Step 2: *For each vertex* Θ_i, $i = 1, \cdots, L$, *compute the corresponding polytopic coordinates*

$$\alpha_i = \prod_{l=1}^{r} \tilde{\vartheta}_l , \quad \textit{where} \quad \tilde{\vartheta}_l = \begin{cases} \vartheta_l , & \textit{if} \quad \underline{\theta}_l \quad \textit{is a coordinate of} \quad \Theta_i ; \\ 1 - \vartheta_l , & \textit{if} \quad \bar{\theta}_l \quad \textit{is a coordinate of} \quad \Theta_i . \end{cases}$$

Then, computing polytopic coordinates from measured rectangular coordinates is not a costly procedure. It can be readily performed on line through simple operations basically consisting in (r) scalar normalizations and $(L\,r - L)$ scalar multiplications.

Since the matrix polytope $\mathcal{P}_{\mathbf{\Delta}}$ is considered as a hypercube, a practical interest exists in centralizing it at 0. It can be shown that solutions $\mathbf{\Delta_{K,i}}$ associated with opposite points of the polytope are also opposite of each other. As a result, the solution family $\{\mathbf{\Delta_{K,i}}\}_{i=1,\ldots,L}$ determine a hypercube with a similar arrangement as the original hypercube $\{\Delta_i\}_{i=1,\ldots,L}$. Two consequences of this result can be inferred: first, the number of inequalities in (8) and (9) can be reduced (see [3] for details); secondly, because the matrices Φ_i in (10) are linear in Δ_i and $\mathbf{\Delta_{K,i}}$, they enjoy the same properties as $\mathbf{\Delta_{K,i}}$ and half of them should be stored for gain-scheduling. We shall take advantage of this property in the missile autopilot problem.

Indeed, any hyper-rectangle can be transformed into a hypercube centered at 0 by translation and scaling. It is easily shown that for any parameter in (6) defined as

$$\theta'_l = (S_l)\theta_l + T_l , \tag{17}$$

with the scaling $S_l = \frac{\bar{\theta}'_l - \underline{\theta}'_l}{2}$ and the translation $T_l = \frac{\bar{\theta}'_l + \underline{\theta}'_l}{2}$, a corresponding parameter

$$\theta_l = \frac{\theta'_l - T_l}{S_l} \tag{18}$$

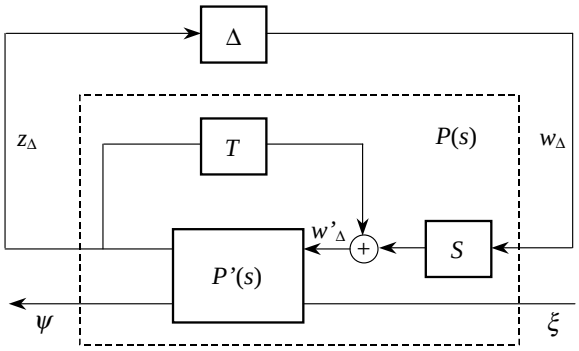


Figure 3: Translation and scaling interconnections

ranges between -1 and 1. Since Δ is diagonal and linear in θ_l, one can deduce from (17) and (18):

$$\Delta' z_\Delta = S\Delta z_\Delta + T z_\Delta \qquad \text{or} \qquad w'_\Delta = S w_\Delta + T z_\Delta \ , \tag{19}$$

where

$$\Delta' = S\Delta + T = \mathrm{diag}(\theta'_1 I_{s_1}, \cdots, \theta'_r I_{s_r}) \ ,$$
$$S = \mathrm{diag}(S_1 I_{s_1}, \cdots, S_r I_{s_r}) \quad \text{and} \quad T = \mathrm{diag}(T_1 I_{s_1}, \cdots, T_r I_{s_r}) \ .$$

Equation (19) is graphically represented by the interconnections depicted in Figure 3. The LFT $\mathcal{F}_u(P'(s), \Delta')$, where Δ' evolves in any hyper-rectangle, can be then transformed into the LFT $\mathcal{F}_u(P(s), \Delta)$, where all parameter trajectories lie in a normalized hypercube centered at 0. In fact, Δ depends on θ which has entries $\theta_l \in [-1, 1]$.

4 Missile control problem

In this section, we apply the technique presented in Sections 2 and 3 to a realistic missile gain-scheduling autopilot problem. This problem consists in controlling a missile to track commanded normal acceleration $\eta_c(t)$, by generating a commanded tail fin deflection $\delta_c(t)$. The nonlinear missile model and actuator dynamics are borrowed from [14, 12].

4.1 Missile model

The pitch-axis missile model involves the angle of attack $\alpha(t)$, the pitch-rate $q(t)$ and the tail deflection angle $\delta(t)$ and its derivative $\dot{\delta}(t)$. Normal acceleration $\eta(t)$ and pitch-rate are measured outputs. A quasi-LPV description of the missile and actuator models is given by:

$$\begin{aligned}
\begin{bmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{\delta} \\ \ddot{\delta} \end{bmatrix} &= \begin{bmatrix} Z_\alpha & 1 & Z_\delta & 0 \\ M_\alpha & 0 & M_\delta & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_a^2 & -2\zeta\omega_a \end{bmatrix} \begin{bmatrix} \alpha \\ q \\ \delta \\ \dot{\delta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \omega_a^2 \end{bmatrix} \delta_c \\
\begin{bmatrix} \eta \\ q \end{bmatrix} &= \begin{bmatrix} N_\alpha & 0 & N_\delta & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ q \\ \delta \\ \dot{\delta} \end{bmatrix} ,
\end{aligned} \tag{20}$$

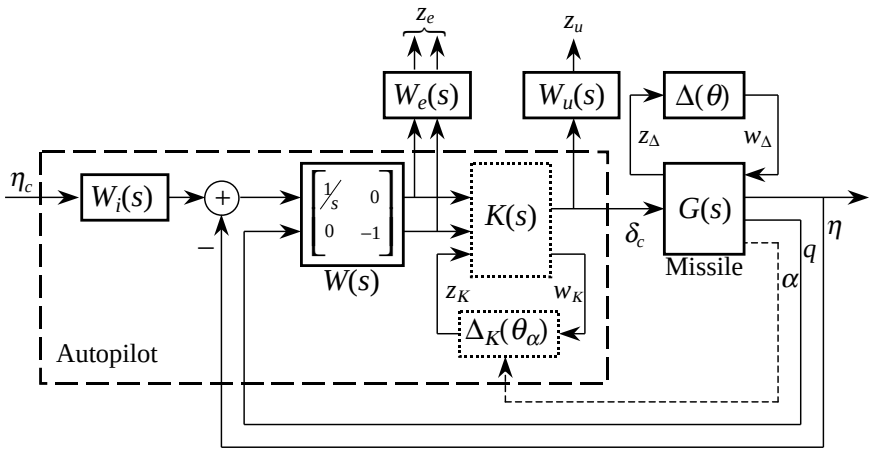


Figure 4: Control structure and synthesis interconnection

with:

$$\begin{aligned} Z_\alpha &= K_\alpha M \cos\alpha \left(a_n\alpha^2 + b_n|\alpha| + c_n(2 - M/3)\right) \\ Z_\delta &= K_\alpha M \cos\alpha\, d_n \\ M_\alpha &= K_q M^2 \left(a_m\alpha^2 + b_m|\alpha| + c_m(-7 + 8M/3)\right) \\ M_\delta &= K_q M^2 d_m \\ N_\alpha &= K_z M^2 \left(a_n\alpha^2 + b_n|\alpha| + c_n(2 - M/3)\right) \\ N_\delta &= K_z M^2 d_n \ . \end{aligned} \tag{21}$$

The above nonlinear description represents a missile flying at an altitude of 20000 ft. Numerical data and units of a_m, b_m, c_m, d_m, a_n, b_n, c_n, d_n, K_α, K_q, K_z, ω_a, and ζ are given in [14, 12]. The plant dynamics can be parameterized by $\theta(t) = [\alpha(t),\ M(t)]^T$, where the Mach number $M(t)$ is an exogenous variable which is treated here as an uncertainty. We consider that only the state variable $\alpha(t)$ is available for scheduling purposes. In fact, the parameter channel can be split into two channels by defining $z_\Delta := [z_\alpha^T,\ z_M^T]^T$ and $w_\Delta := [w_\alpha^T,\ w_M^T]^T$. Due to the missile symmetry about $\alpha = 0$, controllers are designed for $\alpha \geq 0$ and scheduled on $|\alpha|$.

4.2 Control and synthesis structures and performance objectives

The performance and robustness specifications for the closed-loop system are similar to those detailed in [23, 12]. Our aim is to maintain robust stability over the entire operating range, $\alpha \in [-30, 30]$ degrees and $M \in [2, 4]$, and to track step commands in η_c with time constant no more than 0.35 sec, maximum overshoot of 10%, steady-state error less than 1% and an adequate high-frequency roll-off for noise attenuation and withstand neglected high frequency dynamics and flexible modes. In order to avoid saturation of the actuator, the maximum tail deflection rate for $1g$ step command in η_c should not exceed 25 deg/sec.

We adopt the closed-loop control structure depicted in Figure 4. The LFT missile model $F_u(G(s), \Delta(\theta))$ is derived from (20) and (21) in Appendix A. In order to utilize the approach

discussed in this paper, we express the performance objectives by choosing appropriate weighting functions. The precompensator $W_i(s)$ is used to achieve the command shaping. The weighting functions $W(s)$ and $W_e(s) := \mathrm{diag}(W'_e(s), 0.01)$ penalize the tracking error and $W_u(s)$ incorporates bounds on the norm of unmodeled dynamics and also reflects magnitude restriction on the control signal.

Hence, the specifications above can be met by a controller $K(s)$ together with its scheduling function $\Delta_K(\theta_\alpha)$ which:

- minimize the L_2-induced gain γ_M of the operator mapping z_M into w_M,
- maintain the variance of z_e due to the disturbance η_c below an appropriate bound ν_e, and
- guarantee an upper bound γ_u on the L_2-induced gain of the operator mapping η_c into z_u,

for all trajectories $\alpha(t) \in [-30, 30]$ degrees.

Then, this problem can be solved by running Algorithm 2.1 and consists in finding an adequate compromise between three conflicting objectives over the entire operating range: one H_2 and two H_∞ specifications. Note that such a problem cannot be solved by earlier LPV methodologies for plants described by LFT representations.

The discrete-time synthesis plant $P(z)$ and the final continuous-time controller $K(s)$ are computed through bilinear transformations, respectively from $P(s)$ and the designed $K(z)$, as indicated in Section 3.1. The continuous-time synthesis plant $P(s)$ is readily obtained from the connections in Figure 4 and incorporates the missile model $G(s)$ and the weighting functions, $W_i(s)$, $W(s)$, $W_e(s)$, and $W_u(s)$. These frequency-dependent weights have been tuned by performing a few trials-and-errors of synthesis and simulations for the nominal plant. That is, an LTI plant model obtained from the linearization about the point $\theta = [0\,,\ 0]^T$, ($\alpha = 15,\ M = 3$), and an appropriate compromise between ν_e and γ_u has guided our weight selection. This has been carried out by using the same synthesis methodology described in Section 2 with $\Delta = 0$. Notice also that the adopted frequency shapes for the filters are fairly standard (Figure 5). Their state-space representations are given below:

$$
\begin{aligned}
W_i(s) \quad : \quad \begin{bmatrix} \dot{x}_i \\ y_i \end{bmatrix} &= \begin{bmatrix} -14.0 & 1.0 \\ 14.0 & 0.0 \end{bmatrix} \begin{bmatrix} x_i \\ \eta_c \end{bmatrix}, \\
W'_e(s) \quad : \quad \begin{bmatrix} \dot{x}_{e_1} \\ \dot{x}_{e_2} \\ z_{e_1} \end{bmatrix} &= \begin{bmatrix} -138.42 & -604.32 & 128.0 \\ 16.00 & 0.00 & 0.00 \\ 2.142 & 193.12 & 0.106 \end{bmatrix} \begin{bmatrix} x_{e_1} \\ x_{e_2} \\ u_{e_1} \end{bmatrix}, \qquad (22) \\
W_u(s) \quad : \quad \begin{bmatrix} \dot{x}_{u_1} \\ \dot{x}_{u_2} \\ z_u \end{bmatrix} &= \begin{bmatrix} -2.9132e+5 & -4.923e+6 & 2.0972e+6 \\ 8192.0 & 0.00 & 0.00 \\ -1.2742e+5 & -2.154e+6 & 9.1757e+5 \end{bmatrix} \begin{bmatrix} x_{u_1} \\ x_{u_2} \\ \delta_c \end{bmatrix}.
\end{aligned}
$$

4.3 Results and simulations

In order to put in light the potentials of our multi-channel LPV synthesis method and to allow comparisons, we have considered two designs. The first LPV controller, $K_1(s)$ and $\Delta_{K_1}(\theta_\alpha)$, has been synthesized considering M as a constant ($= 3$); the second one, $K_2(s)$ and $\Delta_{K_2}(\theta_\alpha)$, considers M as a bounded uncertain parameter ($M \in [2, 4]$). In short, we have used the following strategy to compute these controllers:

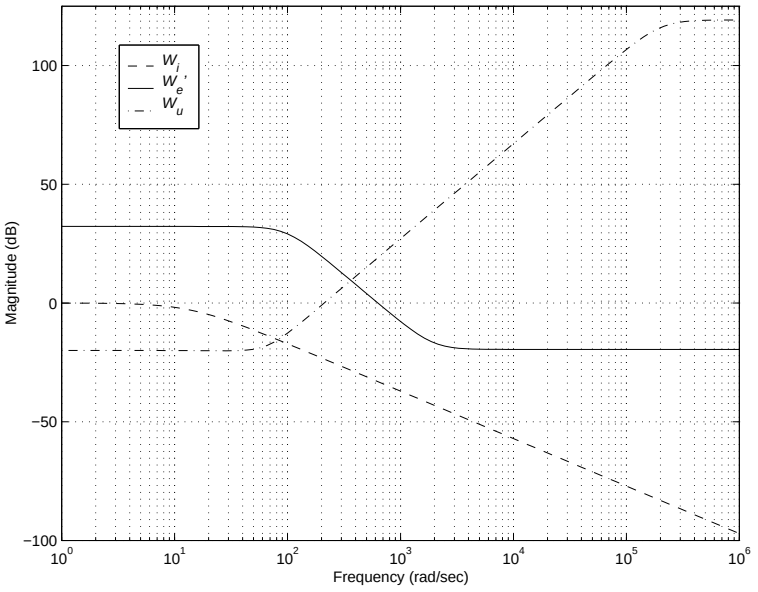


Figure 5: Frequency responses of the weighting functions

- $K_1(s)$ and $\Delta_{K_1}(\theta_\alpha)$:
  - Starting with a small value ν_e, synthesize controllers which minimize the H_∞ performance γ_u subject to a H_2 constraint $\sqrt{\nu_e}$.
  - Through successive relaxations in ν_e, find a reasonable compromise between these objectives. To check out when a good balance has been achieved, perform non-stationary ($\alpha(t)$) and nonlinear simulations for $M = 3$ and evaluate the closed-loop performance in the time domain.
- $K_2(s)$ and $\Delta_{K_2}(\theta_\alpha)$:
  - As mentioned in the previous subsection and analogously to $K_1(s)$, minimize γ_M subject to the constraints γ_u and ν_e.
  - Starting with the final values γ_u and ν_e obtained in designing $K_1(s)$, relax them alternately in order to find an adequate balance between the three objectives.

It must be emphasized that the LFT missile representation derived in Appendix A is not minimal with respect to the angle of attack α. In fact, the Δ-block can be reduced from order 12 to order 10, ($\Delta = diag(\theta_\alpha I_4, \theta_M I_6)$) without any loss of accuracy in the model. We have used numerical techniques based on controllability and observability conditions and on singular value decomposition proposed in [8, 6] to obtain a reduced LFT representation. Note that the synthesis plant $P(s)$ has order 10 and generates controllers $K(s)$ of the same order and autopilots of order 12. For this application, the LTI components of the overall LPV controller are stable but exhibit very fast modes which can be detrimental to on-line implementations. They can then be decomposed into the sum of a fast- and a slow-dynamic subsystems represented by the state-space matrices $(A_f, B_f, C_f, 0)$ and (A_s, B_s, C_s, D_s), respectively. The fast-dynamic part of the controller has been approximated by a static gain in the form of a feedthrough matrix ($D_f = -C_f(A_f)^{-1}B_f$). Therefore, final controllers $K(s)$ have order 7 and the state-space representation $(A_s, B_s, C_s, D_s - D_f)$. Numerical data for $K_2(s)$ and its scheduling function coefficients Φ_i's are provided in Appendix B.

Table 2: H_2 and H_∞ performances

Contr.	$\sqrt{\nu_e}$	γ_u	γ_M
K_1, Δ_{K_1}	6.0	0.85	–
K_2, Δ_{K_2}	15.0	3.0	6.56

Values of H_2 and H_∞ objectives for both LPV controllers are listed in Table 2. Nonlinear simulation results for fixed values of M are displayed on Figure 6. Figure 7 shows nonlinear simulations for time-varying $M(t)$. The input is a sequence of step commanded acceleration η_c whose amplitudes have been chosen such that the parameter α covers most of the scheduling range, thus inducing significant variations in the aerodynamic coefficients. As in [12, 23], the Mach number time trajectory has been generated by

$$\dot{M} = \frac{1}{v_s}\left[-|\eta|\sin(|\alpha|) + A_x M^2 \cos(\alpha)\right] \tag{23}$$

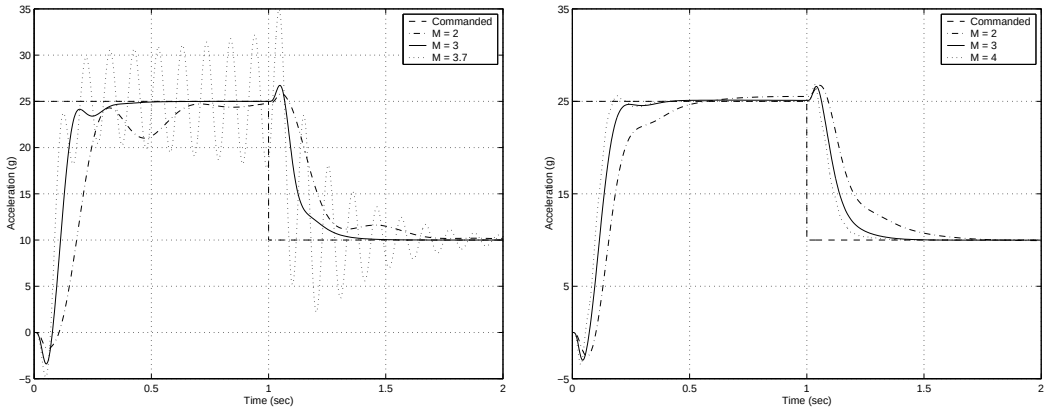
with $M(0) = 4$ as a realistic Mach profile. Numerical values and units of v_s and A_x are detailed in [14, 12, 23]. As theoretically expected, all performance objectives are met for all considered trajectories when (K_2, Δ_{K_2}) is employed for controlling the system. In contrast, the desired closed-loop behavior is satisfied only at the central point ($M = 3$) for (K_1, Δ_{K_1}). We recall that (K_2, Δ_{K_2}) has been computed in order to ensure robustness with respect to variations in the Mach number through an extra H_∞ constraint on the Mach channel M. From the later result, the advantages of using this multi-objective LPV synthesis method became evident.

5 Conclusions

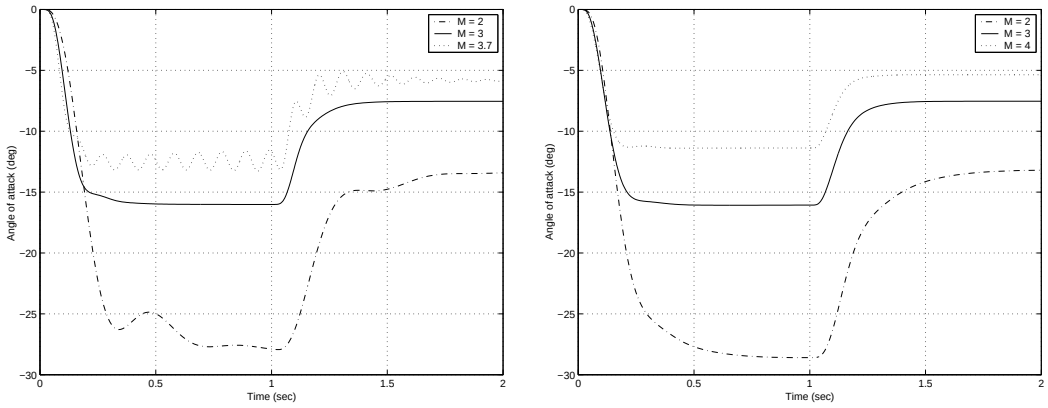
We have discussed a multi-objective/channel H_2/H_∞ LPV control technique for the design of a missile autopilot over a broad range of operating conditions in both the angle of attack and the Mach number. The proposed method provides additional flexibility to handle various and stringent specifications attached to the missile problem while maintaining the same operational simplicity as earlier single-objective LPV techniques:

- the missile nonlinearities are captured through the use of an LFT representation,
- different channels are defined to translate tracking performance, control limitation and robustness properties,
- balancing the different design requirements is carried out in a very natural way within the proposed design framework and conservatism is kept reasonable thanks to the use of different Lyapunov and scalings for each channel/specification,
- also, we describe simple schemes to construct the controller scheduling function and show how all these manipulations carry over the continuous-time case.

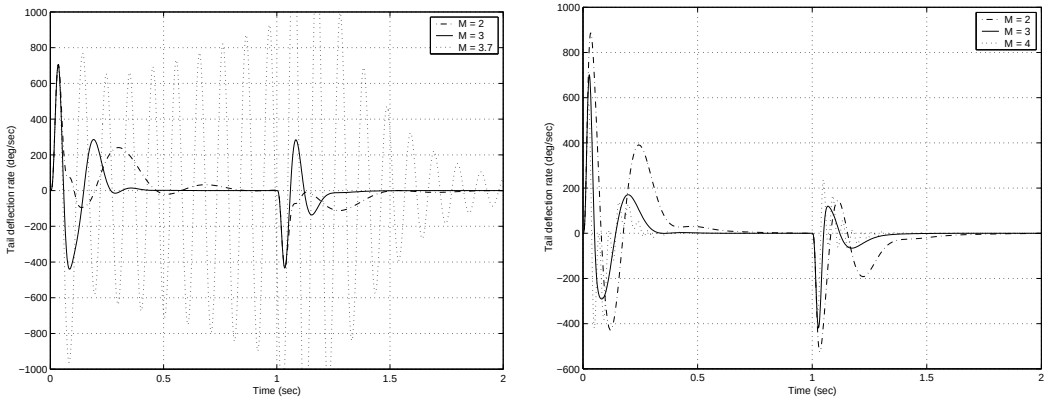
The determination of a full genuine continuous-time methodology remains, however, challenging and future research should be oriented in this direction. Also, we think that other practical reasons might dictate the use of observer-based LPV controllers. This is a delicate and seemingly untouched topic that will be considered in a future study.



(a) - Acceleration, $\eta(t)$

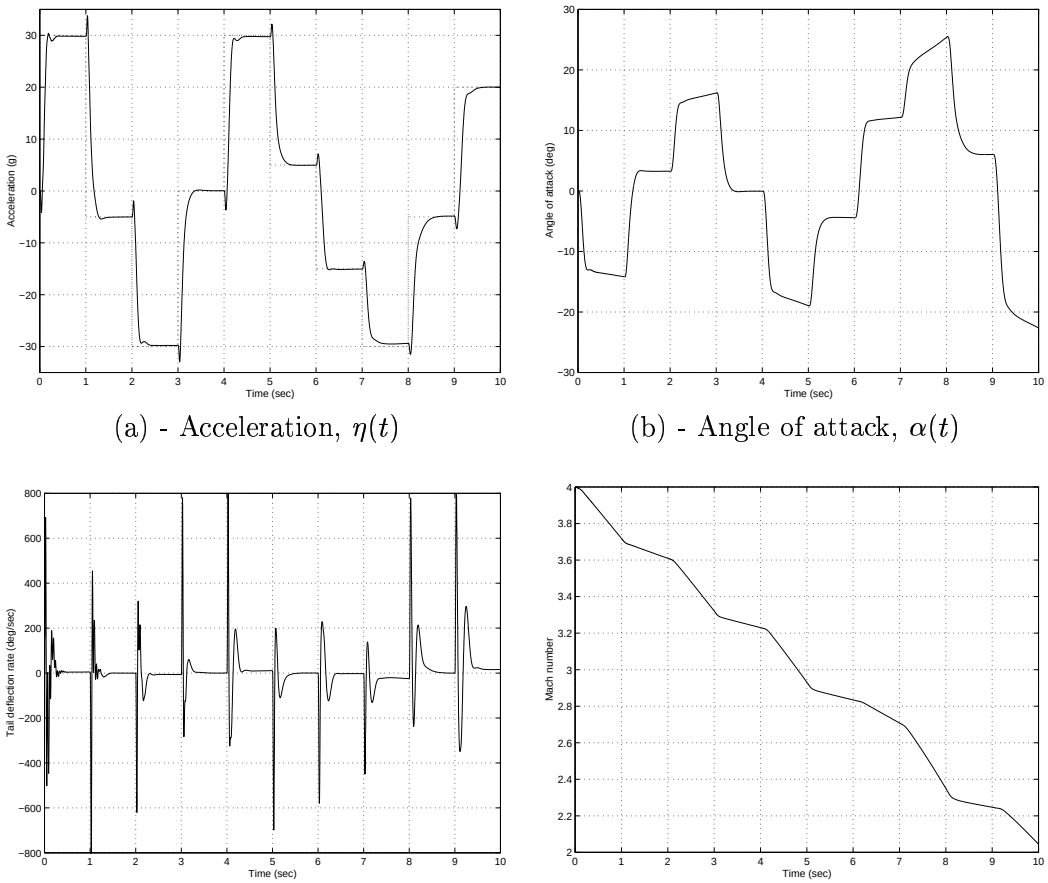


(b) - Angle of attack, $\alpha(t)$



(c) - Tail deflection rate, $\delta(t)$

Figure 6: Nonlinear closed-loop response for fixed M: (K_1, Δ_{K_1}) on the left column and (K_2, Δ_{K_2}) on the right column



(a) - Acceleration, $\eta(t)$

(b) - Angle of attack, $\alpha(t)$

(c) - Tail deflection rate, $\delta(t)$

(d) - Mach number, $M(t)$

Figure 7: Nonlinear closed-loop response using (K_2, Δ_{K_2}) for time-varying M

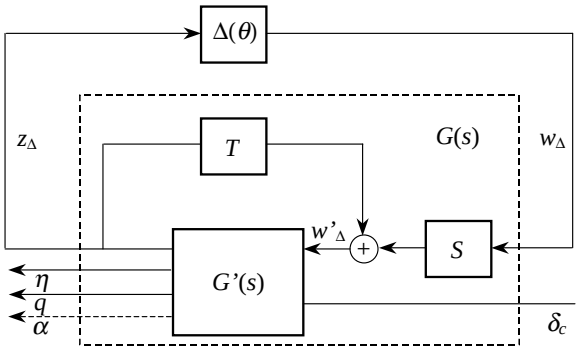


Figure 8: LFT missile representation

A Appendix - LFT parameter-dependent missile model

The missile plant in (20) is approximated by the upper LFT parameter dependence $F_u(G(s), \Delta(\theta))$ in Figure 8, where $G(s)$ is the LTI plant involving the missile rigid body $G'(s)$ and the matrices

$$T = \begin{bmatrix} T_\alpha I_6 & 0 \\ 0 & T_M I_6 \end{bmatrix} \qquad \text{and} \qquad S = \begin{bmatrix} S_\alpha I_6 & 0 \\ 0 & S_M I_6 \end{bmatrix}, \tag{A-1}$$

which correspond to a translation and a scaling. Then $\Delta(\theta(t))$ is a diagonal operator specifying how the normalized parameter θ enters the plant dynamics:

$$\Delta(\theta(t)) = \begin{bmatrix} \theta_\alpha(t) I_6 & 0 \\ 0 & \theta_M(t) I_6 \end{bmatrix}. \tag{A-2}$$

The following assumptions have been made to obtain the LFT plant model:

- angle of attack $|\alpha(t)| = S_\alpha \theta_\alpha(t) + T_\alpha$, where $|\theta_\alpha(t)| \leq 1$, $\forall t \geq 0$, and the constants $T_\alpha \geq 0$ and $S_\alpha > 0$ (in degrees) are used to appropriately restrict the parameter range as in (17);
- similarly, Mach number $M(t) = S_M \theta_M(t) + T_M$, with $|\theta_M(t)| \leq 1$, $\forall t \geq 0$, $T_M \geq 0$ and $S_M > 0$;
- as in (18), the normalized parameter is $\theta(t) = \left[\frac{|\alpha(t)| - T_\alpha}{S_\alpha} \ , \ \frac{M(t) - T_M}{S_M} \right]^T$;
- $\cos(\alpha)$ is approximated by $1 - \alpha^2/2$ with a maximum error of 0.36% over the range $|\alpha| \leq 30$ degrees.

With these simplifications, the state-space model of $G'(s)$ is completely described as:

$$G'(s) := \begin{bmatrix} C_\theta \\ C_g \end{bmatrix} (sI - A_g)^{-1} \begin{bmatrix} B_\theta & B_g \end{bmatrix} + \begin{bmatrix} D_{\theta\theta} & D_{\theta g} \\ D_{g\theta} & D_{gg} \end{bmatrix}, \tag{A-3}$$

with the state vector $x_g = [\alpha, \quad q, \quad \delta, \quad \dot{\delta}]^T$ and

$$A_g = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_a^2 & -2\zeta\omega_a \end{bmatrix},$$

$$B_\theta = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & K_\alpha & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & K_q & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad B_g = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \omega_a^2 \end{bmatrix},$$

$$C_\theta = \begin{bmatrix} \text{sign}(\alpha)b_n & 0 & 0 & 0 \\ a_n - 2c_nK_\pi & 0 & -d_nK_\pi & 0 \\ -\text{sign}(\alpha)b_nK_\pi & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ \text{sign}(\alpha)b_m & 0 & 0 & 0 \\ \text{sign}(\alpha)b_n & 0 & 0 & 0 \\ 2c_n & 0 & d_n & 0 \\ 1 & 0 & 0 & 0 \\ -7c_m & 0 & d_m & 0 \\ 0 & 0 & 0 & 0 \\ 2c_n & 0 & d_n & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$C_g = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix},$$

$$D_{\theta\theta} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & K_\pi c_n/3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -a_nK_\pi & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_m & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_n & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -c_n/3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 8c_m/3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -c_n/3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \qquad D_{\theta g} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$D_{g\theta} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & K_z \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad D_{gg} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

where $K_\pi = \frac{\pi^2}{2(180^2)}$. The plant $G(s)$ is readily determined by the connections and given data.

B Appendix - LFT controller data

The final controller (K_2, Δ_{K_2}) is described by the following state-space data and gain-scheduling function:

$$A_{K_2} = \begin{bmatrix} -35.369 & -164.39 & 0.3279 & -29.321 & -48.583 & -4.6766 & 0.6184 \\ 164.48 & -96.323 & 7.2159 & -37.636 & -136.86 & -10.861 & 1.3651 \\ 0.2290 & -7.1549 & -0.0112 & 0.4713 & 0.6386 & 0.0497 & -0.0040 \\ 29.701 & -39.039 & -0.5076 & -18.542 & -112.53 & -7.3457 & 0.9094 \\ -48.626 & 137.32 & 0.5342 & 113.98 & -307.74 & -54.307 & 7.4718 \\ -5.0085 & 11.760 & 0.0613 & 7.9718 & -58.257 & -99.534 & 67.391 \\ -1.7821 & 4.1619 & 0.0240 & 2.8059 & -21.310 & -87.785 & -49.757 \end{bmatrix},$$

$$B_{K_2} = \begin{bmatrix} 956.21 & -4.4202 & 0.0001 & 0.00003 & -0.0004 & 0.0062 \\ -1114.6 & 6.3520 & 0.0006 & 0.0007 & -0.0008 & 0.0164 \\ -4.9757 & 10.074 & 0.0031 & 0.0046 & 0.0028 & -0.0271 \\ -317.93 & -1.4891 & -0.0014 & -0.0017 & 0.0018 & -0.0392 \\ 697.47 & -3.5028 & -0.0002 & -0.0002 & 0.0003 & -0.0063 \\ 67.814 & -0.7953 & 0.0002 & 0.0002 & -0.0003 & 0.0067 \\ 24.080 & -1.4741 & -0.0000 & 0.0000 & 0.0000 & -0.0004 \end{bmatrix},$$

$$C_{K_2} = \begin{bmatrix} 17.078 & -39.016 & 0.4518 & -18.593 & 39.561 & 9.7018 & -2.9175 \\ -34.502 & -49.971 & -6.2874 & -50.521 & -25.923 & -6.3236 & 1.1467 \\ -108.53 & -141.61 & -4.6665 & -72.008 & -81.963 & 0.6421 & -11.448 \\ 944.96 & 1098.2 & -6.1213 & 304.84 & 687.50 & 65.171 & -11.355 \\ 90.247 & 111.18 & -5.2219 & 8.2498 & 69.762 & -14.754 & 17.669 \end{bmatrix},$$

$$D_{K_2} = \begin{bmatrix} 0.6781 & -0.0031 & -0.0000 & -0.0000 & 0.0000 & -0.0000 \\ -211.02 & 0.5030 & 0.0381 & -0.0221 & -0.0099 & -0.2526 \\ -167.65 & 0.6268 & 1.2180 & -0.0884 & 0.0027 & 0.0710 \\ -1067.8 & 1.4150 & -0.4489 & 0.1926 & -0.4524 & 11.267 \\ -179.32 & 0.0247 & -0.0355 & 0.0109 & 0.0103 & 0.0097 \end{bmatrix},$$

with

$$\Delta_{K_2}(\Delta) = \sum_{i=1}^{2} \alpha_i \Phi_i \ ,$$

where

$$\Phi_1 = -\Phi_2 = \begin{bmatrix} -0.8162 & -0.0536 & -0.0310 & -0.4511 \\ -0.0921 & -0.9155 & -0.0556 & -0.4774 \\ -0.0715 & 0.0268 & -0.8941 & -0.5415 \\ -0.0049 & 0.0017 & 0.0096 & -0.9759 \end{bmatrix}$$

and α_1 and α_2 are computed on line using Algorithm 3.1.

References

[1] P. Apkarian and R. Adams, *Advanced Gain-Scheduling Techniques for Uncertain Systems*, IEEE Trans. Contr. Syst. Technology **6** (1998), 21–32.

[2] P. Apkarian and P. Gahinet, *A Convex Characterization of Gain-Scheduled H_∞ Controllers*, IEEE Trans. Automat. Contr. **40** (1995), no. 5, 853–864. See also pp. 1681.

[3] P. Apkarian, P. C. Pellanda, and H. D. Tuan, *Mixed H_2/H_∞ Multi-Channel Linear Parameter-Varying Control in Discrete Time*, Syst. & Contr. Letters **41** (2000), 333–346.

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

[5] J.-M. Biannic and P. Apkarian, *Missile Autopilot Design via a modified LPV Synthesis Technique*, Aerospace Science and Technology (1999), no. 3, 153–160.

[6] G. E. Boukarim and J. H. Chow, *Modeling of Nonlinear System Uncertainties Using a Linear Fractional Transformation Approach*, in Proc. Amer. Contr. Conf. (1998).

[7] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in Systems and Control Theory*, vol. 15, SIAM Studies in Applied Mathematics, Philadelphia, 1994.

[8] R. D'Andrea and S. Khatri, *Kalman Decomposition of Linear Fractional Transformation Representations and Minimality*, in Proc. Amer. Contr. Conf. (1997), 3557–3561.

[9] M. C. de Oliveira, J. C. Geromel, and J. Bernussou, *An LMI Optimization Approach to Multiobjective Controller Design for Discrete-Time Systems*, in Proc. IEEE Conf. on Decision and Control, Phoenix, AZ (1999), 3611–3616.

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

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

[12] 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.

[13] A. Packard, *Gain-Scheduling via Linear Fractional Transformations*, System & Control Letters **22** (1994), 79–92.

[14] R. T. Reichert, *Dynamic Scheduling of Modern-Robust-Control Autopilot Designs for Missiles*, IEEE Contr. Syst. Magazine **12** (1992), no. 5, 35–42.

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

[16] C. Scherer, P. Gahinet, and M. Chilali, *Multi-Objective Output-Feedback Control via LMI Optimization*, IEEE Trans. Automat. Contr. **42** (1997), 896–911.

[17] C. Schumacher and P. P. Khargonekar, *Missile Autopilot Designs Using H_∞ Control with Gain Scheduling and Dynamic Inversion*, AIAA J. Guidance, Contr., and Dynamics **21** (1998), no. 2, 234–243.

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

[19] J. S. Shamma and J. Cloutier, *Gain-Scheduled Missile Autopilot Design Using Linear Parameter Varying Transformations*, AIAA J. Guidance, Contr., and Dynamics **16** (1993), no. 2, 256–263.

[20] D. J. Stilwell and W. Rugh, *Interpolation of Observer State Feedback Controllers for Gain Scheduling*, Johns Hopkins Univ., Department of Electrical and Computer Enginnering, Baltimore, Tech. Rep. JHU/ECE 97-09, 1997.

[21] ———, *Interpolation of Observer State Feedback Controllers for Gain Scheduling*, IEEE Trans. Automat. Contr. **44** (June 1999), no. 6, 1225–1229.

[22] W. Tan, A. Packard, and G. Balas, *Quasi-LPV Modeling and LPV Control of a Generic Missile*, in Proc. Amer. Contr. Conf., Chicago, Illinois (2000), 3692–3696.

[23] F. Wu, A. Packard, and G. Balas, *LPV Control Design for Pitch-Axis Missile Autopilots*, in Proc. IEEE Conf. Decision Contr., New Orleans, LA **vol. 1** (1995), 188–193.

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