

# Ad-hoc Control Strategies

## Overview

In this chapter two gain scheduling control strategies are applied to the autopilot problem of an Air-to-air missile. These strategies are the *controller blending* and *state feedback/observer*-based interpolation methods detailed in Chapter 1 of this thesis and they are used for the control of the pitch-axis nonlinear model of the Reichert Air-to-air missile. These methods present some advantages in terms of using powerful tools of the modern robust control theory to control a nonlinear system; even so, in terms of implementation they are not so realistic compared to the *gain blending* method detailed in the next chapter. For low dimension and/or not much nonlinear systems however they still remain attractive and rather intuitive. The scope of this chapter is mostly to insist on their characteristics, rather than perform an exhaustive simulation procedure, remaining though inside a practical context.

## Chapter contents

---

<b>5.1</b>	<b>Introduction</b>	<b>127</b>
<b>5.2</b>	<b>Related Work</b>	<b>127</b>
<b>5.3</b>	<b>Missile Control Objectives</b>	<b>131</b>
<b>5.4</b>	<b>Controller Blending</b>	<b>133</b>
5.4.1	LTI Controller Synthesis	134
5.4.2	Gain-scheduled Controller	141
5.4.2.1	<i>Practical Issues</i>	<i>141</i>
5.4.2.2	<i>Simulation Results</i>	<i>145</i>
5.4.2.3	<i>Discussion</i>	<i>145</i>
<b>5.5</b>	<b>Observer/State Feedback Interpolation</b>	<b>149</b>
5.5.1	LTI Synthesis	149
5.5.2	Gain-scheduled Controller	149
<b>5.6</b>	<b>Conclusions</b>	<b>151</b>

---

## 5.1 Introduction

In this chapter two major interpolation methods will be applied to the pitch axis autopilot problem of an Air-to-air missile. These methods are the *controller blending* and *observer/state feedback* interpolation methods detailed in Sections 1.3.2.1 and 1.3.2.6 respectively. These methods have been selected over others (e.g. *controller switching*, *state space matrix interpolation*) due to the advantages they present. The *controller blending* method for example demands less computational effort and presents a rather small interpolation difficulty due to the fact that only output signals are interpolated whereas the *observer/state feedback* method is rather straightforward since it gives an estimate of the plant state while being a true MIMO method. In addition, it rises as a good extension to the *controller blending* method since by Youla parameterization (see Section 3.2) the same LTI controllers may be used for both methods and thus a comparison between the methods is easier to perform.

Method  
overview

These methods are compared to each other and to the more systematic *gain blending* method (detailed in the next chapter) using a realistic scenario and extensive simulations. The advantages and disadvantages of each method are stressed out and potential improvements are proposed. This chapter comes as a natural extension of Chapter 4 where the first two steps of the linearization-based gain scheduling procedure (see Section 1.3.1) were applied to obtain the *trim control surface/operating point parametrization* and the corresponding *LTI/LPV models* for each value of the scheduling vector, both for the Reichert missile and the re-entry vehicle. In this chapter, the three remaining steps are detailed, namely: the *LTI controller computation*, *controller interpolation* and *controller implementation & validation*.

The chapter is organized in two main parts: the first one (see Section 5.3) presents the *controller blending* method whereas the second (see Section 5.4) presents the *observer/state feedback method*. Finally, the chapter ends with some conclusions.

## 5.2 Related Work

A full bibliographic study on the subject of missile control is beyond the scope of this work for two reasons: first, several methods have been proposed to cope with different problems and second, the missile autopilot design dates back to the 40's and a huge number (maybe hundreds) of references on the subject can be found; in this monograph only the most notable works since the 90's are cited.

The  
origins

Perhaps one of the first works published<sup>1</sup> is the one found in [16] exactly half a century ago. It concerns a simple angular rate/position feedback in order to stabilize the roll motion of an Air-to-air missile. Of course there was no question using even the simplest adaptive control scheme with the technology of the time.

<sup>1</sup>One can find even earlier research work on guided missiles; see for example [1].

An adaptive control approach using parameter estimation and gain scheduling was used in [76] to obtain an autopilot for a flexible Air-to-air missile. An ad-hoc control scheme is used and its gains are scheduled by identifying the fin position pitching coefficient for the entire flight envelope of the missile.

One of the first examples in gain scheduling applied on missile control is found in [112] in the early 90's. In this work modern  $\mathcal{H}_\infty$ - $\mu$  techniques are used to obtain LTI controllers which are put in observer/state feedback form. The solutions to the corresponding Riccati equations are then scheduled as a function of the angle of attack (AoA) and the Mach number in order to provide a time-varying gain-scheduled controller.

In [17], the pitch axis missile nonlinear model is linearized and put in LFT form with the uncertainty block  $\Delta$  using the AoA. Linear controllers are designed (being robust with respect to the AoA) and scheduled for different values of the Mach number. The results are promising, even though the simulations are not too exhaustive. Another similar approach using tools of the  $\mu$ -analysis to guarantee stability between the controller synthesis points is found in [37]. The controllers are optimal LQ regulators but the results were not too good.

An early attempt to use a  $q$ -LPV modeling of a missile's dynamics starting from a pure nonlinear model can be found in [123]. Then a particular type of trajectory scheduling is used in order to avoid the classical procedure of designing a finite number of LTI controllers for various design points. The controllers used are once again optimal LQ regulators, however the simulation results mostly demonstrate the feasibility of the approach.

The  
'Reichert'  
missile

A major advance in missile autopilot design was done by the classical work found in [103]. This paper presents a benchmark pitch axis model of an air-air missile (Reichert missile Benchmark or *R'm'B*) and uses classic  $\mathcal{H}_\infty$  control theory, along with a particular type of controller implementation found in more detail in [79], in order to remove the famous *hidden coupling terms*<sup>2</sup>. The controllers are scheduled using *directly* the output and not the state (as well as the Mach number), and ZPK interpolation (see Section 1.3.2.3) and the overall control scheme is tested using extensive simulations. Similar ideas, along with a very small reference on the missile's flight envelope can be found in [150] where instead of robust  $\mathcal{H}_\infty$ , reduced order controllers are considered. In the same framework as in the previous two references, an observer/state feedback form of the central  $\mathcal{H}_\infty$  controller obtained by appropriate frequency weighting is used in [55] in order to construct a gain-scheduled controller.

A modern approach using modern  $\mathcal{H}_\infty$  LPV control and LMI's is the famous paper [13] (see equally [10]). In this paper the controllers are scheduled using two nonlinear functions (depending on the Mach, altitude and AoA respectively) inside a polytope. The overall scheme gives all the stability guarantees of this class of control methods but suffers also of all their inherent drawbacks (conservativeness, complexity).

<sup>2</sup>For more details on this important notion see the discussion in 1.3.3.2.

Another application example using a  $q$ -LPV formulation (using state transformations) of a 6DoF bank-to-turn (BTT) missile is the one in [25]. Tools of the  $\mu$ -synthesis robust control theory are used and the resulting controllers for the pitch and roll/yaw channels are scheduled using the AoA and the roll rate (similar approaches may be found either in [124] or in [123]).

In [116] a pure nonlinear control strategy (dynamic inversion) is compared with  $\mathcal{H}_\infty$  gain scheduling using the so-called  $\mathcal{D}$  implementation for the autopilot of a bank-to-turn missile. The two methods were found to be equivalent in terms of performance and robustness thus clearly favoring the gain scheduling approach due to its simplicity.

The *controller blending* interpolation method, along with the ideas concerning controller realization treating the hidden coupling terms<sup>3</sup>, is tested using the Reichert missile benchmark in [81]. The methodology, even though it appears rather promising, is not validated by simulations in an appropriate manner.

A very good reference on the subject is the work found in [45]. A full autopilot is designed for an Aerospatiale missile with very lightly damped bending modes. The classic Glover&McFarlane  $\mathcal{H}_\infty$  loop shaping design procedure (LSDP) is used and robustness is verified using the  $\nu$ -tool and a complete simulation suite.

In [24] an autopilot for a 6DoF skid-to-turn missile is designed using  $\mu$ -analysis. The controllers are scheduled using the dynamic pressure and a signal conditioning/blending technique similar to *controller blending*. The effectiveness of the control loop is also validated by an interesting engagement scenario with the missile pursuing an aircraft sustaining 9g normal accelerations.

A self-scheduled nonlinear pitch-axis autopilot for a missile is designed in [63] using LPV synthesis tools coupled with  $\mathcal{H}_\infty$  loop shaping design criteria. The controller is scheduled using the Mach number, altitude and AoA over a wide flight envelope. The approach is systematic and stability preserving, offering good results in terms of performance and robustness.

An interesting theoretical work may be found in [47] where notions like incremental stability are used for the analysis of a PI controlled missile (namely the Reichert benchmark model). The analysis, even though it turns out to require the solution of LMI's is not put in the traditional autopilot performance/robustness framework and has not been used in other cases to date.

The ideas behind stability preserving interpolation using the stability covering condition analysis of 1.3.3.2 were applied as a benchmark to the autopilot synthesis of the Reichert benchmark model in [132] and [129]. The first paper concerns observer/state feedback interpolation whereas the second interpolation using the Youla parameter. Both papers give a theoretical flare to the application but unfortunately lack significantly on the controller validation part.

Another approach using a  $q$ -LPV formulation of a missile dynamics and modern LPV control (casts the problem as a generalized disturbance rejection one) gives some good results and is detailed in [136].

---

<sup>3</sup>See once again the classic paper [79] or Section 1.3.3.2 for further details.

A comparative paper demonstrating various control strategies for a skid-to-turn Aerospatiale missile is found in [33]. The control strategies include classic PI-like loops as well as nonlinear linearizing static and dynamic state feedbacks. The results clearly favor the nonlinear control methods; however it remains to be seen if they are on the one hand realistic for implementation and on the other hand, if being compared to more advanced gain-scheduled robust control schemes, retain their advantages.

A rather unusual missile control problem is outlined in [101]. The problem is the trajectory following of a particular 6DOF skid-to-turn missile. The controlled outputs are the AoA, sideslip and Euler roll angle and the control setup consists of two parts: first an open loop dynamic inverse of the plant puts it on the desired trajectory and second an eigenstructure assignment LTV (due to trajectory linearization) control law stabilizes the open loop dynamics. The method is promising and according to the authors presents advantages over the traditional gain scheduling design but robustness testing is yet required.

An alternative strategy for missile autopilot design based on velocity-based gain-scheduling is found in [90] and [89]. This class of methods has been analyzed in Chapter 2 and their inventors claim that they present superior features over the traditional gain scheduling ones, even though their merits are doubted by some (see for example [84]). In any case, the simulation results appearing in both seem nice and demonstrate the feasibility of the approach.

Some work on digitally implemented autopilot control laws in the  $\mathcal{H}_\infty$  loop shaping context can be found in [43] or equally in [42]. In the first case, reduced order discrete time dynamic controllers are considered and interpolated using the output (vertical acceleration) and the Mach number, whereas in the second case, multi-rate design is considered.

Another method based on multi-model eigenstructure assignment and  $\mu$ -iteration is presented in [36]. This work is notable since on the one hand, it presents very good simulation results on the well known Reichert benchmark missile and on the other hand, because several other classical works on gain scheduling control autopilots are compared (see for example Section 4.5 of the paper) to the proposed approach.

In [3], integral quadratic constraints (IQC's) and LPV modeling are used to analyze the stability robustness of an uncertain nonlinear missile control system based on dynamic inversion. The tuning of the controller was done using a genetic algorithm and then it was put (together with the plant) in LPV/LFT form in order to use the IQC analysis tools. Since the latter can be put into LMI form they are computationally tractable. Finally, the stability of the autopilot was proven under the appearance of time-varying parameters.

Some very recent work may be found in [141] where a fuzzy interpolation control is used for the missile autopilot. In [99] an LPV control scheme is compared to an eigenvalue assignment technique for the autopilot of a skid-to-turn missile. Finally a paper concerning control of a square cross section missile using classical  $\mathcal{H}_\infty$  control and nonlinear dynamic inversion is the one in [30].

## 5.3 Missile Control Objectives

The missile autopilot control goals-objectives will be detailed in this section. Recall from Chapter 4 that the pitch axis missile nonlinear parameter-dependent model  $\mathcal{S}_{pd}$ , is a SIMO system with its state being comprised by the angle of attack (AoA)  $\alpha$  (in rad) and the pitch rate  $q$  (in rad/s); its control input being the tail elevator deflection angle  $\delta$  (in rad) and the measure vector being comprised by the vertical acceleration  $\eta$  (in g's) and the pitch rate  $q$ . The nonlinear model is parameterized also by the Mach number  $M$  considered as an internal variable (see Fig. 5.1).

Missile  
model

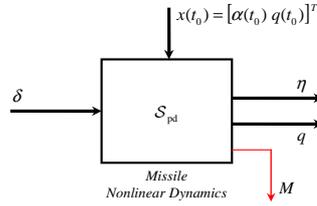


Figure 5.1: Missile block diagram.

The missile's rectangular flight envelope is formed due to the restrictions on the AoA and Mach number:  $-20^\circ \leq \alpha \leq 20^\circ$  and  $1.5 \leq M \leq 3$  respectively and linear models may be computed for every  $\alpha, M$  inside this envelope. Given though that the AoA *is not* available as a measurement, it *should not* be used directly when calculating trim conditions and linear models.

Flight  
envelope

The vertical acceleration  $\eta$  should be used in its stead, and thus the flight envelope can be re-parametrized as a function of  $\eta, M$ . The obtained flight envelope  $\mathbf{\Gamma}_{fe}^{[\eta, M]}$  has now a more complicated non-convex trapezoidal form (see Fig. 4.5) and thus, for every value of  $M$ , the maximum admissible value for  $\eta$  (due to the limits on  $\alpha$ ) is given by (see also Section 4.1.2.3)<sup>4</sup>:

$$\eta_{fe}(M) \simeq -0.454M^3 + 5.035M^2, \quad \alpha \leq 0. \quad (5.1)$$

As it has been already detailed, a linear approximation of the aforementioned equation yields a superset of  $\mathbf{\Gamma}_{fe}^{[\eta, M]}$ ; this new, slightly redundant<sup>5</sup>, flight envelope (denoted as  $\mathbf{\Gamma}_{fe,lin}^{[\eta, M]}$ ) has been used instead for simplicity, and it forms a trapezium whose four corners are the following pairs:

$$[\eta, M] : \quad [0, 1.5], [9.7969, 1.5], [0, 3], [0, 33.0559]. \quad (5.2)$$

The scheduling vector  $\varrho$  (being now formed by  $\eta, M$ ) takes values inside  $\mathbf{\Gamma}_{fe,lin}^{[\eta, M]}$  and parameterizes all the plants equilibrium points, trim control and linear models, as it has been detailed in the previous chapter.

<sup>4</sup>Of course since the AoA may also take positive values, the flight envelope is symmetric with respect to the Mach number and so do the linear models, trim control etc.

<sup>5</sup>Since this approximation adds up about 3.6% of surface.

**Control objectives**

The *control objective* for the autopilot is simple enough: track step reference signals  $\eta_r(t)$  of different amplitudes inside the flight envelope while the Mach varies with certain *performance* and *robustness* constraints. The *performance* constraints concern the output tracking characteristics ( $P_1$ ) and maximum control signal rates ( $P_2$ ) whereas the *robustness* constraints concern stability margins/high frequency open loop attenuation ( $R_1$ ) and robust stability under aerodynamic coefficient perturbations ( $R_2$ ). These constraints are taken from the benchmark paper [103] and are the following:

$P_1$ : Track step commands on  $\eta_r(t)$  of various amplitudes with a time constant<sup>6</sup>  $\tau \leq 0.35\text{s}$ , overshoot  $M_p \leq 10\%$  and steady state error  $e_{ss} \leq 1\%$ .

$P_2$ : The tail elevator angle deflection rate  $|\dot{\delta}|$  should be inferior to 25deg/s for 1g step reference commands.

$R_1$ : The missile should exhibit robust stability inside all its flight envelope when the pitching moment coefficients ( $a_m, b_m, c_m$ ) and  $d_m$  vary around their nominal values by  $\pm 25\%$ .

$R_2$ : The linearized system should maintain at least 30dB attenuation for the gain amplitude of the obtained open loop transfer function, when the loop is opened just before the actuator.

**Performance**

The first two performance characteristics denote objectives that in the benchmark paper [103] are tested for a given simulation profile on the nonlinear system. This profile may sometimes be too favorable for the autopilot since variations of big amplitude are demanded for relatively high values of the Mach number where the controllers are in general more performing. The procedure often used by the designers is to calculate a small number of controllers and try to adapt the performance objectives only for these points, hoping that the design will carry on to the nonlinear system. There is no indication if the number of points considered is too small, too big or even if the points themselves are chosen in places where the nonlinear dynamics need treatment.

**Robustness**

The situation for the robustness objectives is the same: these objectives are most of the time satisfied (with a relative margin) at the synthesis points and the nonlinear gain-scheduled controller is exhaustively tested for stability when the aerodynamic coefficients are perturbed. This analysis is done using a Monte Carlo procedure for hundreds (in real world systems maybe several thousands) of operating points; thus guaranteeing in a way a posteriori the well-behaved of the controller. Other tests performed include linearization of the total open loop of the system plus the gain-scheduled controller in order to check its stability margins for frozen values of time along the desired reference trajectory of the scheduling vector.

<sup>6</sup>To avoid confusion the time constant is here defined as the time it takes the tracking output to reach the 63.2% of its final value.

## 5.4 Controller Blending

In this section the results from the first ad-hoc scheduling method, based on *controller blending*, will be presented. Recall from Section 1.3.2.2, that this method uses LTI controllers designed around a family of operating points inside the system's operating domain and then interpolates the *outputs* of adjacent controllers in order to provide the final control signal.

Initially in this work, only four controllers at the corners of the flight envelope  $\mathbf{\Gamma}_{\text{fe,lin}}^{[\eta,M]}$  were used, but the performance of the gain-scheduled controller was not satisfactory. This was due to the fact that such a small number of controllers was inadequate to capture the variation of the plant dynamics inside the operating domain. Thus, it was decided to divide the flight envelope in four scheduling regions  $\Gamma^1, \Gamma^2, \Gamma^3, \Gamma^4$  formed by nine synthesis points. These synthesis points, for each value  $\varrho^i, i = 1, \dots, 9^7$ , are shown in Table 5.1.

Synthesis  
points

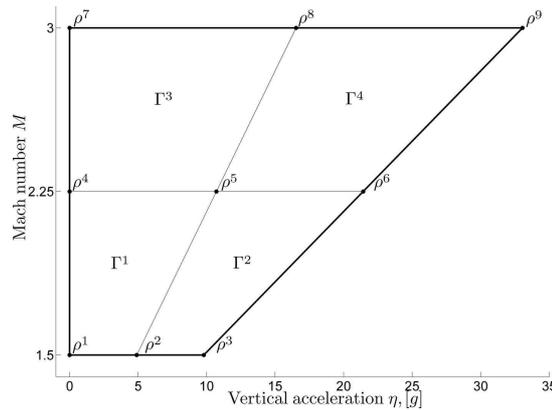
**Table 5.1:** Controller synthesis points<sup>(i),(ii)</sup>.

Point	1	2	3	4	5	6	7	8	9
$\eta$	0	4.898	9.797	0	10.713	21.426	0	16.528	33.056
$M$	1.5	1.5	1.5	2.25	2.25	2.25	3	3	3

(i) Each point corresponds to a value  $\varrho^i$ .

(ii) The values for  $\eta$  are approximated to the third digit.

The flight envelope along with the four scheduling regions are illustrated in Fig. 5.2 (the symmetrical part for negative values of the vertical acceleration is not here shown). Before proceeding to the LTI controller calculation, it should be stressed out that the same nine synthesis points are used for every type of interpolation method in the rest of this work.



**Figure 5.2:** Flight envelope, synthesis points and scheduling regions.

<sup>7</sup>Instead of noting each synthesis point as  $\varrho^{i,j}$  (where  $i = 1, \dots, 4$  is the scheduling region and  $j = 1, \dots, 4$  the number of the controller in each region) as in Section 1.3.2.2, a simpler notation is used with  $i = 1, \dots, 9$  denoting globally every controller in  $\mathbf{\Gamma}_{\text{fe,lin}}^{[\eta,M]}$ .

### 5.4.1 LTI Controller Synthesis

Missile  
LPV model

In this section the LTI controller synthesis procedure (corresponding to the third step of the linearization-based gain scheduling procedure) will be detailed. Recall that in Section 4.1.3.1, an LPV model of the missile's nonlinear dynamics, parameterized by the scheduling vector  $\varrho = [\eta \ M]^T$ , was obtained. This model  $\mathcal{S}_{\text{LPV}}(\varrho)$  was of the following form:

$$\mathcal{S}_{\text{LPV}}(\varrho) : \begin{cases} \dot{x}_\delta = \mathbf{A}(\varrho)x_\delta + \mathbf{B}(\varrho)\delta_\delta \\ y_\delta = \mathbf{C}(\varrho)x_\delta + \mathbf{D}(\varrho)\delta_\delta \end{cases} \quad (5.3)$$

with  $x = [\alpha \ q]^T$ ,  $y = [\eta \ q]$  and:

$$\begin{aligned} x_\delta &= x - x(\varrho) \\ \delta_\delta &= \delta - \delta(\varrho) \\ y_\delta &= y - y(\varrho). \end{aligned} \quad (5.4)$$

For the sake of correctness, it should be stressed that the above LPV model is in fact a family of LTI models, smoothly parameterized by fixed-equilibrium values  $\varrho_r$  of the scheduling vector. Thus, each member of this family  $\mathcal{S}_{\text{LTI}}(\varrho_r)$  of linear models describes the behavior of the initial nonlinear parameter-dependent missile model  $\mathcal{S}_{\text{pd}}$ , *locally* around the equilibrium point.

The corresponding constant system matrices  $\mathbf{A}(\varrho_r)$ ,  $\mathbf{B}(\varrho_r)$ ,  $\mathbf{C}(\varrho_r)$  and  $\mathbf{D}(\varrho_r)$  are of course obtained by Jacobian linearization using appropriate trim values for the state and the input. The corresponding to each frozen state space description  $\mathcal{S}_{\text{LTI}}(\varrho_r)$ , I/O matrix transfer function  $G(s)$  is written as:

$$\begin{bmatrix} \eta_\delta(s) \\ q_\delta(s) \end{bmatrix} = \begin{bmatrix} G_\eta(s) \\ G_q(s) \end{bmatrix} \delta_\delta = G(s)\delta_\delta. \quad (5.5)$$

LTI  
synthesis

In the context of controller synthesis, the plant was preceded by the actuator transfer function  $G_a(s)$  (see Eq. 4.11) and also augmented by an integrator acting on the tracking error  $e_\delta = \eta_r - \eta_\delta$  (in order to ensure proper reference trajectory following). A robust  $\mathcal{H}_\infty$ , S/KS-type mixed sensitivity control strategy was then selected in order to treat the tracking error dynamics on the one hand, but limit on the other hand the control effort rate, conformably to the performance objectives  $P_1, P_2$ . Appropriate constant (for simplicity) weights  $k_e, k_{\dot{\delta}}$  were added for use with this method on each signal  $e_\delta, \dot{\delta}_\delta$ .

In addition to the  $\mathcal{H}_\infty$  optimization, LMI pole placement constraints were imposed, to have a better control on the closed loop dynamics and avoid inherent inconveniences of standard  $\mathcal{H}_\infty$  control (e.g. very fast closed loop eigenvalues). Thus, an LMI region  $\mathcal{D}(\lambda_{\min}, r_{\max}, \vartheta_{\min})$  was used with  $\lambda_{\min}, r_{\max}, \vartheta_{\min}$  providing minimum decay rate, maximum undamped natural frequency and minimum damping constraints for the closed loop eigenvalues (see Fig. 3.4)<sup>8</sup>.

<sup>8</sup>For each of the nine synthesis points,  $\lambda_{\min}$  was variable in order to fine-tune the maximum control signal rate whereas  $r, \theta, k_e, k_{\dot{\delta}}$  were held constant. The values used were  $r_{\max} = 150, \vartheta_{\min} = 0.707$  (actuator undamped natural frequency/damping) and  $k_e = 1, k_{\dot{\delta}} = 3$ .

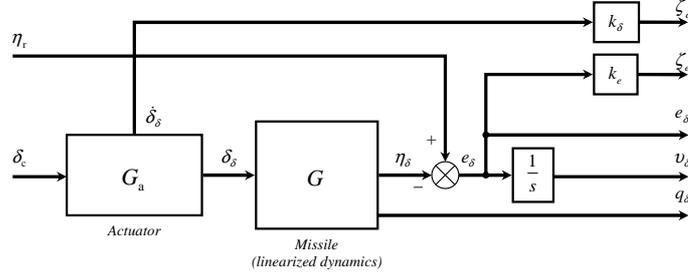


Figure 5.3:  $\mathcal{H}_\infty$  synthesis standard form.

The total synthesis block diagram used is illustrated in Fig. 5.3. The external ‘disturbance’ signal (often denoted by  $w$ ) is the output reference signal  $\eta_r$ , the ‘performance’ signals  $\zeta_\delta, \zeta_e$  are the weighted actuator deflection rate  $\dot{\delta}_\delta$  and tracking error  $e_\delta$ . The control input is the commanded actuator signal  $\delta_c$  and the controller inputs are the tracking error  $e_\delta$ , the integral of the tracking error  $v_\delta$ , and the pitch rate  $q_\delta$ <sup>9</sup>.

The controller synthesis standard form may be written thus in the following compact state-space form:

Standard  
form

$$\begin{bmatrix} \dot{x}_{\text{aug}} \\ \zeta_\infty \\ y_{\text{aug}} \end{bmatrix} = \mathbb{P} \begin{bmatrix} x_{\text{aug}} \\ w \\ u_{\text{aug}} \end{bmatrix} \quad (5.6)$$

where:

$$\mathbb{P} = \begin{bmatrix} \mathbf{A}_{\text{aug}} & \mathbf{B}_w & \mathbf{B}_u \\ \mathbf{C}_\zeta & \mathbf{D}_{w\zeta} & \mathbf{D}_{u\zeta} \\ \mathbf{C}_y & \mathbf{D}_{wy} & \mathbf{D}_{uy} \end{bmatrix}. \quad (5.7)$$

The vectors in the above equations are:  $x_{\text{aug}} = [\dot{\delta}_\delta, \delta_\delta, \alpha_\delta, q_\delta, v_\delta]^T \in \mathbb{R}^{5 \times 1}$  is the augmented (without the controller) state vector,  $\zeta_\infty = [\zeta_\delta, \zeta_e]^T \in \mathbb{R}^{2 \times 1}$  is the performance vector,  $y_{\text{aug}} = [e_\delta, v_\delta, q_\delta]^T \in \mathbb{R}^{3 \times 1}$  is the controller input vector and  $u_{\text{aug}} = \delta_c$  is the controller output and thus evidently  $\mathbb{P} \in \mathbb{R}^{10 \times 7}$ . The latter may also be written as:

$$\begin{bmatrix} \zeta_\infty \\ y_{\text{aug}} \end{bmatrix} = \begin{bmatrix} P_{w\zeta}(s) & P_{u\zeta}(s) \\ P_{wy}(s) & P_{uy}(s) \end{bmatrix} \begin{bmatrix} w \\ u_{\text{aug}} \end{bmatrix}. \quad (5.8)$$

The standard form synthesis matrices of the above Eq. 5.7 are given by the following equations:

$$\mathbf{A}_{\text{aug}} = \begin{bmatrix} -2\zeta\omega_n & -\omega_n^2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & B_{11} & A_{11} & A_{12} & 0 \\ 0 & B_{21} & A_{21} & A_{22} & 0 \\ 0 & -D_{11} & -C_{11} & -C_{12} & 0 \end{bmatrix} \quad (5.9)$$

<sup>9</sup>Note that the ‘ $\delta$ ’ notation is maintained to emphasize that the signals are in fact perturbation ones around equilibrium points.

$$\mathbf{B}_w = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{B}_u = \begin{bmatrix} \omega_n^2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (5.10)$$

$$\mathbf{C}_\infty = \begin{bmatrix} k_\delta & 0 & 0 & 0 & 0 \\ 0 & -k_e D_{11} & -k_e C_{11} & -k_e C_{12} & 0 \end{bmatrix} \quad (5.11)$$

$$\mathbf{C}_y = \begin{bmatrix} 0 & -D_{11} & -C_{11} & -C_{12} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{D}_{w\zeta} = \begin{bmatrix} 0 \\ k_e \end{bmatrix} \quad \mathbf{D}_{u\zeta} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (5.12)$$

$$\mathbf{D}_{wy} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{D}_{uy} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

The additional constants  $A_{11}, A_{12}, A_{21}, A_{22}, B_{11}, B_{21}, C_{11}, C_{12}, D_{11}$  are in fact the (frozen) state-space matrix components of the missile linearized dynamics around an equilibrium point (see Eq. 5.3) with:

$$\mathbf{A}(\varrho_r) = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad \mathbf{B}(\varrho_r) = \begin{bmatrix} B_{11} \\ B_{21} \end{bmatrix} \quad (5.13)$$

$$\mathbf{C}(\varrho_r) = \begin{bmatrix} C_{11} & C_{12} \\ 0 & 1 \end{bmatrix} \quad \mathbf{D}(\varrho_r) = \begin{bmatrix} D_{11} \\ 0 \end{bmatrix}.$$

Furthermore, due to the special form of the missile nonlinear dynamics,  $A_{22} = C_{12} = 0$  and  $A_{12} = 1$ . It should be also noted that the minus signs inside the matrices are due to the negative addition of the reference signal  $\eta_r$  to the vertical acceleration error  $\eta_\delta$  (see Fig. 5.3).

The goal for the robust controller  $K(s)$  now is to ensure closed stability, minimization of the  $\mathcal{H}_\infty$  norm of the transfer function from the disturbance to the performance vector, and also ensure a correct eigenvalue placement inside the LMI region  $\mathcal{D}$ . Briefly this can be denoted as:

*$\mathcal{H}_\infty$  synthesis:* Calculate a linear MISO, dynamic output feedback controller  $K(s) = \begin{bmatrix} \mathbf{A}_k & \mathbf{B}_k \\ \mathbf{C}_k & \mathbf{D}_k \end{bmatrix}$  with  $u_{\text{aug}}(s) = K(s)y_{\text{aug}}(s)$ , so that  $\|T_{w\zeta_\infty}\|_\infty < \gamma$  with  $T_{w\zeta_\infty}$  stable, and additionally  $\lambda(\mathbf{A}_{\text{cl}}) \in \mathcal{D}(\lambda_{\min}, r_{\max}, \vartheta_{\min})$ <sup>10</sup>.

Now for each synthesis point, it has been tried to compute a feedback controller in such a way that the step response time constant is minimized while the control effort rate is limited (according to performance objectives  $P_1, P_2$ ). The parameters used are shown in Table 5.2.

<sup>10</sup>Recall that  $T_{w\zeta_\infty} = \mathcal{F}_l(\mathbf{P}, K) = \mathbf{C}_{\text{cl}}(s\mathbf{I} - \mathbf{A}_{\text{cl}})^{-1}\mathbf{B}_{\text{cl}} + \mathbf{D}_{\text{cl}}$ .

**Table 5.2:** Controller synthesis parameters<sup>(i)</sup>.

Points	1	2	3	4	5	6	7	8	9
$\lambda_{\min}$	5.831	5.922	6.025	11.23	11.52	11.90	18.18	18.27	19.09
$\tau$	361.9	339.3	325.6	199.2	191.3	182.2	129.2	128.9	122.0
$t_s$	616.4	576.6	557.7	332.1	316.1	300.6	209.6	208.2	196.3
$M_p$	0	0	0	0.041	0.054	0.051	0.068	0.076	0.108
GM	27.93	20.93	19.75	11.66	16.43	15.06	6.93	14.86	13.83
PM	72.53	70.67	68.10	63.99	70.75	68.01	56.02	67.25	67.89
$\omega_c$	12.82	18.19	20.54	18.11	27.23	31.63	23.80	27.59	37.41
$d_{att}$	54.66	49.23	48.21	48.67	43.25	40.52	43.69	43.88	39.28
$\gamma$	2.243	2.249	2.286	2.576	2.562	2.603	2.995	2.957	3.013

<sup>(i)</sup> The time constant  $\tau$  and the settling time  $t_s$  (taken for 95% of the final value) are measured in ms, the overshoot  $M_p$  in %, the gain margin (GM) and the open loop magnitude attenuation  $d_{att}$  in dB, the phase margin (PM) in degrees and the gain crossover frequency  $\omega_c$  in rad/s.

<sup>(ii)</sup> The GM, PM,  $\omega_c$  and  $d_{att}$  are all computed for the open loop transfer function, with the loop opened before the actuator.

The poles of each  $\mathcal{H}_\infty$  controller are shown in Table 5.3 whereas the poles and transmission zeros for each of the three I/O channels are shown in Fig. 5.4<sup>11</sup>. It may be observed that the synthesis algorithm provides well-behaved controllers in terms of pole location (avoids excessively fast modes) and I/O zeros (except for some cases in controllers No. 8, 9 where some non-minimal phase zeros appear).

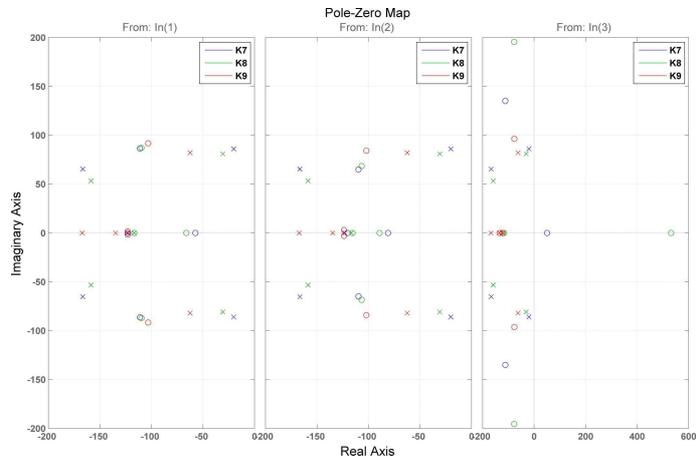
Controller  
results

In Figs. 5.5a-5.5b, simulations of each of the nine closed loops (corresponding to the synthesis points) are demonstrated. In the first figure, step responses of the vertical acceleration are shown whereas in the second the corresponding control signal rates are presented.

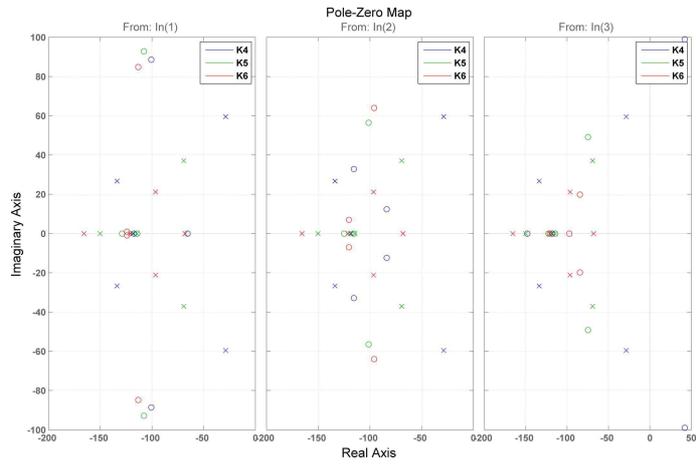
**Table 5.3:** Controller poles.

Points	Poles
1	$-48.71 \pm 9.63j, -120.94, -129.22 \pm 8.01j$
2	$-38.42, -103.16 \pm 25.32j, -123.32, -130.64$
3	$-55.76, -99.39 \pm 35.15j, -123.19, -136.45$
4	$-28.66 \pm 59.56j, -118.1, -133.8 \pm 26.71j$
5	$-69.23 \pm 37.07j, -114.51, -118.92, -150.16$
6	$-68.02, -96.49 \pm 21.16j, -121.05, -165.68$
7	$-20.08 \pm 86j, -123.41, -166.74 \pm 65.34j$
8	$-30.75 \pm 80.9j, -116.81, -158.64 \pm 53.34j$
9	$-62.46 \pm 82.08j, -123.01, -134.77, -167.26$

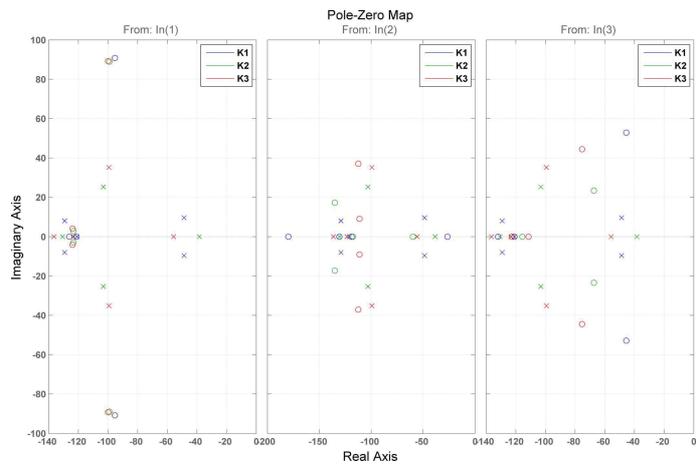
<sup>11</sup>The controllers are depicted in triplets, corresponding to constant Mach numbers:  $M = 1.5, 2.25, 3$ .



(a) Controllers No. 7,8,9



(b) Controllers No. 4,5,6



(c) Controllers No. 1,2,3

Figure 5.4: Controllers' I/O poles & zeros.

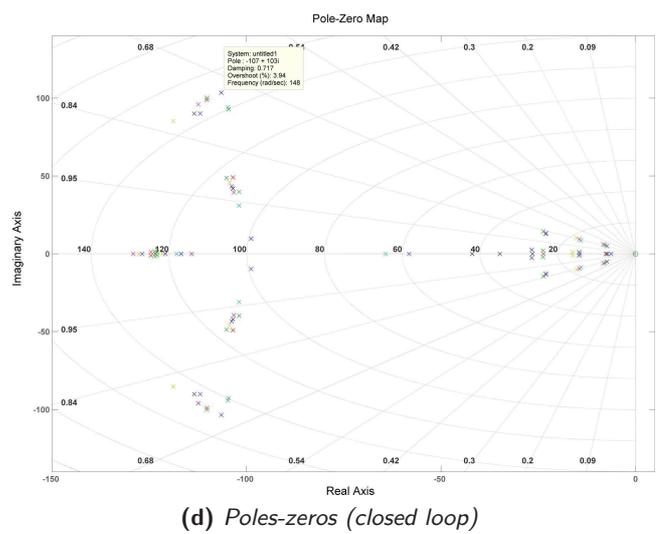
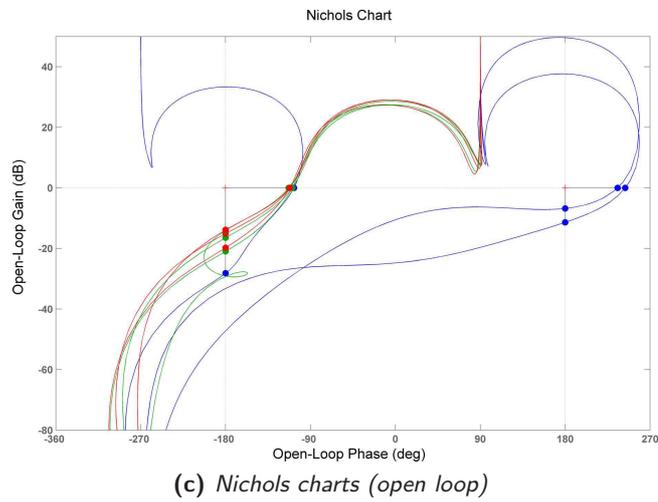
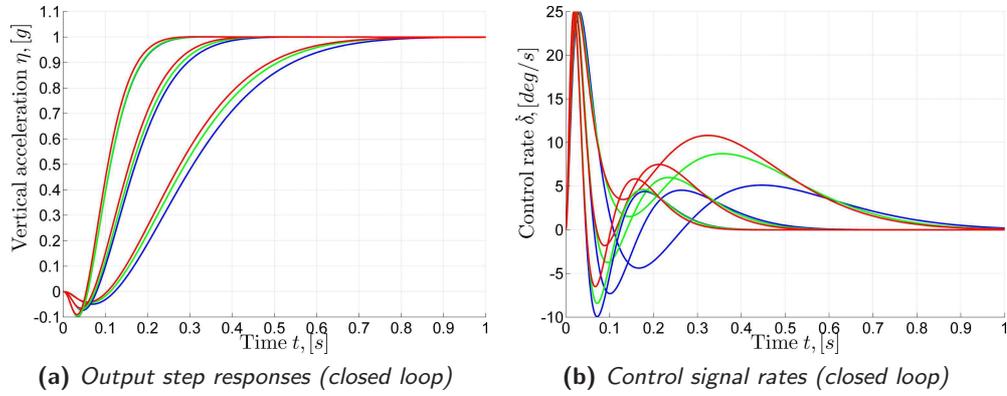
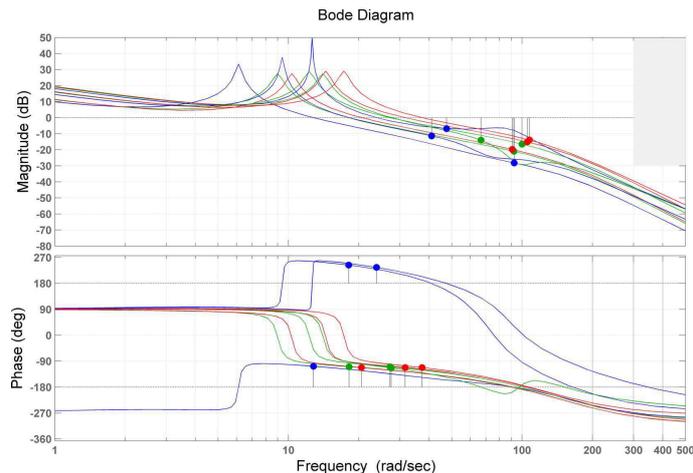


Figure 5.5: Simulation results for the linear plants.

Concerning the outputs<sup>12</sup> (see Fig. 5.5a), it may be remarked that the controllers provide excellent damping and thus the corresponding step responses demonstrate practically no overshoot (see also Table 5.2). In addition, it may be remarked from Fig. 5.5b that the controllers exploit all available bandwidth (reaching the maximum of 25deg/s) and thus provide a time constant in most cases well below the performance specifications ( $\tau \leq 350\text{ms}$ )<sup>13</sup>.

Concerning now the Nichols charts, they present a very good visualization of the open loop gain & phase margins for every linear synthesis point. Once again the smallest, but still adequate values, are obtained for synthesis point No. 7. The stability margins may also be seen from the corresponding Bode diagrams depicted in the following Fig. 5.6. In addition to these margins, and the gain & phase crossover frequencies, the very good magnitude attenuation margins  $d_{\text{att}}$  of the robustness objective  $R_2$  can be observed (the grey box shows the attenuation constraint). Finally, in Fig. 5.5d are shown the closed loop poles of the transfer function  $T_{w\zeta_\infty}(s)$  for every synthesis point. It may be observed that they are indeed inside the convex LMI region  $\mathcal{D}(\lambda_{\min}, r_{\max}, \vartheta_{\min})$ <sup>14</sup>.



**Figure 5.6:** Bode diagrams (open loop).

As a last comment, it should be stressed that even though the  $\mathcal{H}_\infty$  controllers are really of very good performance, they remain complex since the final controller to be implemented is the fifth order controller  $K(s)$  plus the integrator; all this for a second order plant. In addition, it remains to be seen in the next section if the interpolation strategy chosen (controller blending) merits such a complicated LTI synthesis approach.

<sup>12</sup>All the time responses are again grouped in triplets for constant Mach numbers (blue corresponds to synthesis points No. 1,4,7, green to No. 2,5,8 and red to No. 3,6,9 respectively).

<sup>13</sup>Except for synthesis point No. 7 where there is a small violation. Now this may be expected since the open loop dynamics are here *unstable* (for more details see the missile open loop stability discussion of Chapter 4).

<sup>14</sup>The minimum decay rate  $\lambda_{\min}$  changes for each synthesis point according to Table 5.2.

### 5.4.2 Gain-scheduled Controller

In this section the gain-scheduled controller, using the controller blending method and the  $\mathcal{H}_\infty$  controllers of the previous section, will be detailed. The analysis starts with some details on some practical issues concerning the controller interpolation/realization; the next section presents simulation results.

#### 5.4.2.1 Practical Issues

The Simulink block diagram of the total simulated plant and the gain-scheduled controller is presented in Fig. 5.7. The grey boxes correspond to the missile nonlinear dynamics, Mach number generation trajectory and their appropriate initialization blocks. The blue boxes are the error integrator, the gain-scheduled controller and a block providing trim values for the pitch rate  $q$  according to the scheduling vector value in the missile flight envelope<sup>15</sup>. The red box is the reference trajectory generating block and finally the yellow boxes are the actuator and a first order filter acting on the reference trajectory. The latter is used in order to smoothen the passage of the scheduling vector through the four scheduling regions  $\Gamma^1, \Gamma^2, \Gamma^3, \Gamma^4$  and thus facilitate the interpolating procedure.

Inner  
control  
structure

The gain-scheduled controller block interior is more complicated and is shown in Fig. 5.8; its major functions are performed mainly by four block families depicted in different colors. The *blue* colors depict the four adjacent controllers the interpolation procedure needs, in order to interpolate between their signals and provide the final control law. Each block realizing a dynamic controller, needs the appropriate controller matrices  $\mathbf{A}_k, \mathbf{B}_k, \mathbf{C}_k$  and  $\mathbf{D}_k$ , appropriate state initialization and reset when moving from one scheduling region to the next, according to the values of the scheduling vector  $\varrho = [\eta_r \ M]^T$ .

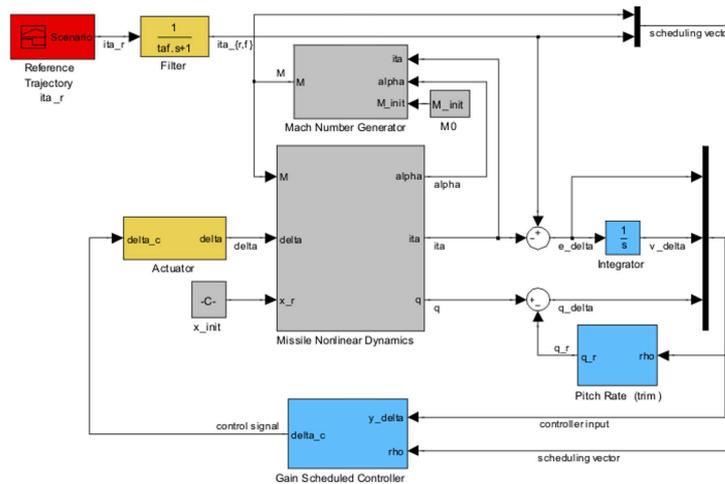


Figure 5.7: Total simulation block diagram (controller blending).

<sup>15</sup>The last is a necessary box since the input to the controller is  $q_\delta = q - q_r$ , where  $q_r = q(\varrho_r)$ .

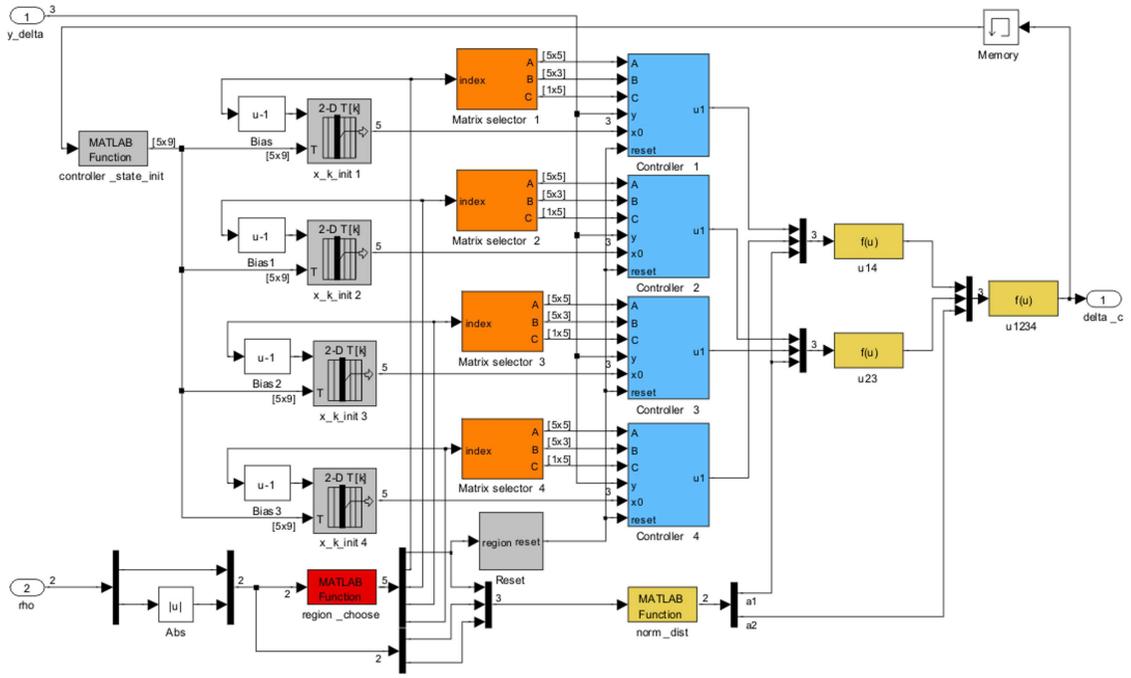


Figure 5.8: Total simulation block diagram (controller interior).

The controller state matrices of each controller are provided by an *orange* block that is in fact a look-up table. Inside this table are stored the controller matrices of all nine synthesis points and they are appropriately selected according to a signal *index* that dictates the current scheduling region. This signal is the output of another block (red color) that outputs this region number ( $\Gamma^1, \Gamma^2, \Gamma^3$  or  $\Gamma^4$ )<sup>16</sup> according to the value of the scheduling vector.

State  
matrices

Now for each region number corresponds a different quartet of controllers (in total there are nine controllers and each time only four are used). This is important when coming to interpolation since each time, in order to compute the final interpolated signal, the controllers on the left (respectively right) side corners of the current scheduling region are first combined, each combination providing a control signal. Then, these two interpolated signals are once again combined to compute the final control law (see Eqs. 1.88-1.90 for more details). This quartet of controllers in the lower left (ll), lower right (lr), upper right (ur) and upper left (ul) corners of the scheduling region is also the output of the red block that gives the scheduling region number<sup>17</sup>.

Interpolation

Recall now from the analysis of Section 1.3.2.2, concerning the controller blending method, that when changing scheduling regions and charging different controller matrices, the control signal will be discontinuous due to the incompatibility of the DC gains of the various controllers entering and leaving the algorithm. This may be corrected by guaranteeing a bump-less control signal using the analysis of the aforementioned section. Briefly, this is done by *re-initializing* all controllers of the new scheduling region at the transition time to an appropriate state. This state is calculated from Eq. 1.91 using the control signal at the transition time and the output controller matrices of the newly entered scheduling region. This is done using the *grey* blocks of Fig. 5.8: the four vertically aligned ones re-initialize the controllers at the appropriate state using the grey block on the upper left whereas the state reset command is given by the grey block on the center.

Controller  
reset

Finally, the yellow blocks calculate the interpolated final control signal. The two vertically aligned yellow boxes on the left blend the controllers' signals by pairs as detailed above (ll-ul and lr-ur respectively) and the one in the right blends these two, to provide the final control signal. The yellow block on the page center computes the normalized distances  $a_1, a_2$  (with  $0 \leq a_i \leq 1$ ) used by the aforementioned control signal interpolation blocks (see Eqs. 1.86-1.87)<sup>18</sup>.

Normalized  
distances

<sup>16</sup>See Fig. 5.2.

<sup>17</sup>For example, for the scheduling region  $\Gamma^4$ , the ll, lr, ur, ul, corners correspond to controllers No. 5, 6, 9, 8 respectively.

<sup>18</sup>As a last comment it should be stressed out that given the fact that the interpolation regions  $\Gamma^i$  are not rectangular as assumed in Section 1.3.2.2 (for simplicity) but trapezoids, with the upper and lower sides being parallel to each other, the interpolation procedure needs some more trigonometry. For the generic trapezoid of Fig. 5.9, the normalized distances are defined as:  $a_1 = l^1 / l^{\{1,4\}}$  and  $a_2 = l^2 / l^{\{1,4\} \leftrightarrow \{2,3\}}$ . The difference from a rectangular region is that the distances  $l^{\{1,4\}}$ ,  $l^{\{1,4\} \leftrightarrow \{2,3\}}$  are also time varying (except for the other ones  $l^1, l^2$  that are always time varying because of the scheduling vector motion inside the flight envelope).

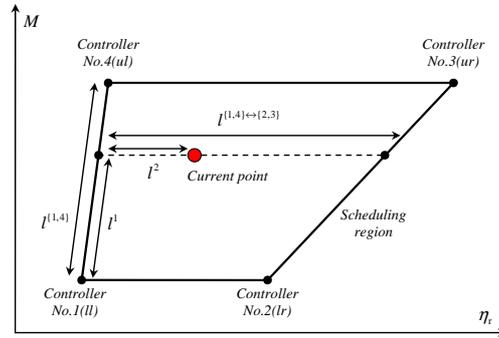
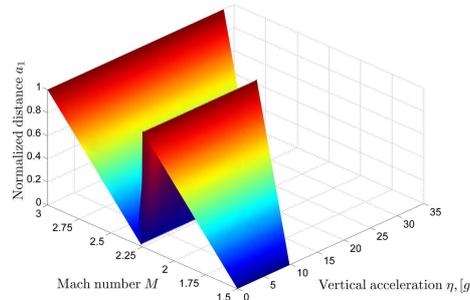


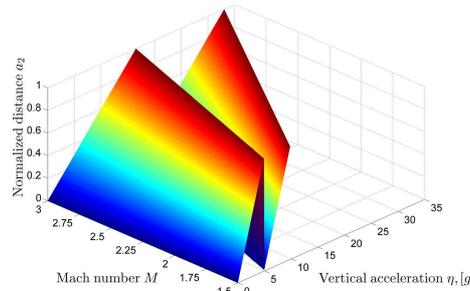
Figure 5.9: Generic trapezoidal scheduling region.

These normalized distances  $a_1, a_2$  are visualized for the whole missile flight envelope in Figs. 5.10a-5.10b. As expected, the first one is clearly augmenting inside the scheduling regions  $\Gamma^1, \Gamma^2$  and  $\Gamma^3, \Gamma^4$  as the Mach increases, whereas the second one is augmenting inside the regions  $\Gamma^1, \Gamma^4$  and  $\Gamma^2, \Gamma^3$  as the vertical acceleration increases.

As a last comment it may be added that the trajectory reference tracking is assured by the integrator at the input of the controller. An alternative to that is to decompose the control signal  $\delta_c$  into an open loop control signal providing a trim control input  $\delta_{c,r}$  (as a function of the scheduling vector) and a closed loop interpolated control signal  $\delta_{c,\delta}$  stabilizing the missile (see for example the conference paper [137]).



(a) Normalized distance  $a_1$



(b) Normalized distance  $a_2$

Figure 5.10: Scheduling region normalized distances.

### 5.4.2.2 Simulation Results

In this section, the simulation results of the nonlinear controller will be presented. Contrary to other existing works, where the nonlinear controller is not thoroughly tested due to scheduling vector reference trajectories not covering the whole missile flight envelope, here some rather stringent scenarios are considered.

The Mach number trajectory given by Eq. 4.10 (taken from the original benchmark paper [103]) covers only a small part of the missile operating domain when combined to the corresponding vertical acceleration reference trajectory given in the same work. To obtain a more realistic scenario, here the drag coefficient  $A_x$  (see Table 4.1) has been augmented in order to provide a steeper descent for the Mach number. In addition, the vertical acceleration profile has been slightly modified with respect to the aforementioned work.

In Fig. 5.11a this Mach trajectory is visualized whereas in Fig. 5.11b the vertical acceleration reference trajectory  $\eta_r(t)$  (black), the filtered reference trajectory  $\eta_{r,f}(t)$  (red) and the actual response of the system  $\eta(t)$  (blue) are demonstrated. Finally in Fig. 5.11c, the output trajectories are plotted on the missile flight envelope (using the same controllers). The general behavior of the gain-scheduled controller is rather acceptable; however there many issues that will be detailed further on.

In Fig. 5.12a is shown the total control command  $\delta_c(t)$  of the global gain-scheduled controller given to the actuator (red) and the filtered one  $\delta(t)$  that is the actual input to the system. In the same figure is also depicted the scheduling region number (either 1, 2, 3 or 4 corresponding to regions  $\Gamma^1, \Gamma^2, \Gamma^3$  or  $\Gamma^4$ ).

In Fig. 5.12b are shown the four controller outputs (corresponding each time to each of the four corners of each scheduling region) and once again the interpolated control signal  $\delta_c(t)$  (in red and blue respectively). In addition, the controller state reset signal is shown (being in fact accordant to the scheduling region number signal of the previous figure). Finally, in Fig. 5.12c the normalized distance signals  $a_1, a_2$ , that are used to interpolate the four controller signals, are illustrated; taking obviously values between zero and one.

### 5.4.2.3 Discussion

Even though the time performance of the controller blending method is good (see Fig. 5.11b), there exist several inconveniences due to the fact that the controllers need to re-initialize when changing scheduling region.

This fact causes control signal transients, chattering and degrades the overall time performance of the system<sup>19</sup>. Consider for example to different cases observing Figs. 5.11-5.12: switching due to the vertical acceleration (Case 1) and due to the Mach number (Case 2).

<sup>19</sup>When referring to time performance, output tracking is considered in most cases since this is the primary goal of control systems in this work.

Simulation  
scenario

Mach &  
output

Control  
signals

*Case 1.* The simulation scenario chosen involves four changes on the desired output vertical acceleration (see Fig. 5.11b):

$$\eta_r(t) : 0g \rightarrow 25g \rightarrow -15g \rightarrow -10g \rightarrow 5g$$

These changes correspond to the first three and the fifth state resets of Fig. 5.12b (see reset signal impulses). Consider now just one of them (the first) since the controller behavior is similar for all four. At  $t = 0s$ , the output reference signal (see Figs. 5.11b-5.11c) changes its value. Given that the state reset signal is based on the *filtered* reference signal, it gives the command to reset the states of the four controllers approximately at  $t = 0.15s$ , the time it takes the filtered signal to cross the boundary of the fourth and third scheduling regions respectively (see Fig. 5.11c).

As a consequence, the control signals exhibit a transient behavior right afterwards (see Fig. 5.12a) before settling down and controlling the plant. This fact may be observed equally on the four controller signals of Fig. 5.12b. This behavior is clearly undesirable and undermines the plant performance but is unavoidable if this interpolation method is used.

This is once again due to the need to change the *whole* controller when passing on to a subsequent region; in general two controllers are switched on and off respectively, except for the extremely improbable case that the scheduling vector crosses the scheduling boundary at a synthesis point and thus three controllers need to be switched.

*Case 2.* Consider now a scheduling region change due to the Mach number. Refer again to Figs. 5.11b-5.11c and consider the crossing of the reference trajectory at  $t = 2.3s$  (this can be equally seen from the reset signal of Fig. 5.12b) due to the Mach number falling below the value 2.25.

This causes the scheduling vector passing from region  $\Gamma^4$  to region  $\Gamma^2$  and thus the controllers switch and re-initialize<sup>20</sup>. The latter fact causes a transient behavior and the output  $\eta$  demonstrates a small oscillation around its steady state value (that had already been established at  $t = 2.3s$ ).

It is clear that this issue is even more important; if the user is not so lucky and this switch due to the Mach number falls during the transient of a switch due to the vertical acceleration also, then the performance of the system is even more deteriorated because of the combination of both effects.

These two study cases demonstrate in practice the greatest disadvantage of the *controller blending* method: controller re-initialization. In the next section an alternative method is considered based on state feedback/observer interpolation, solving this problem since the controllers are not switched but rather structurally modified.

<sup>20</sup>Recall that the missile flight envelope is symmetrical and at the switching time  $\eta_r < 0$ ; thus the system functions on the left symmetric side of the flight envelope.

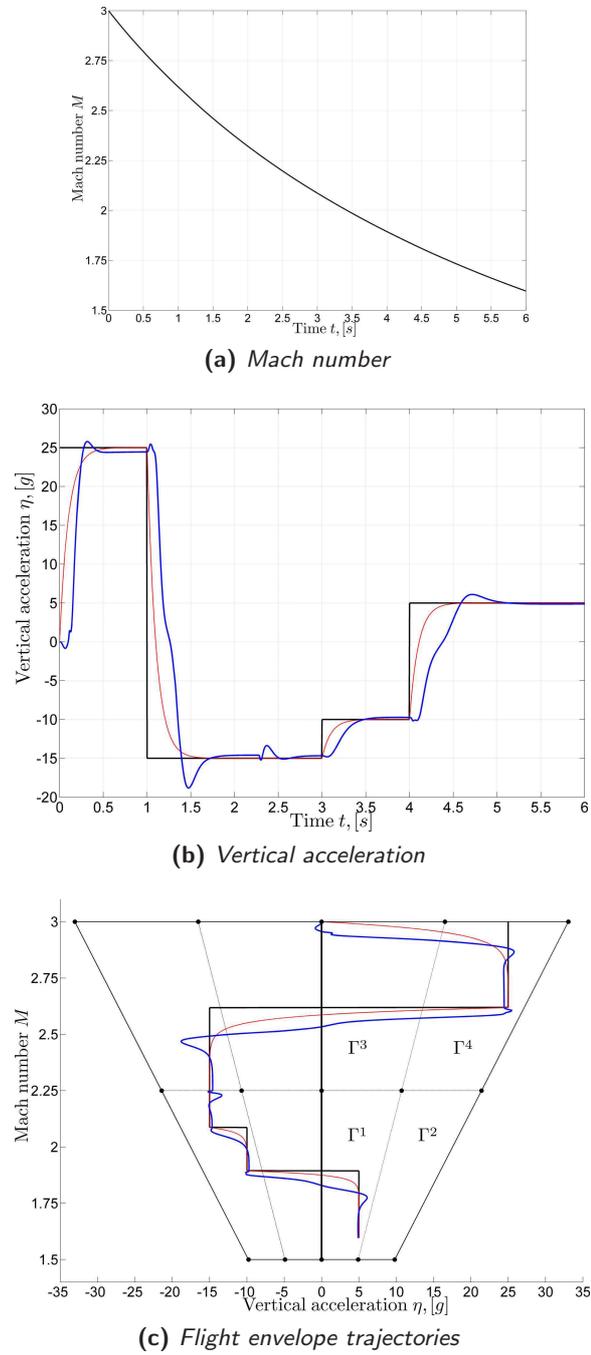
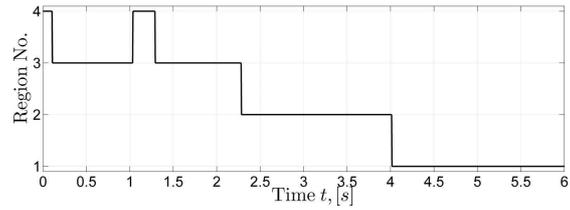
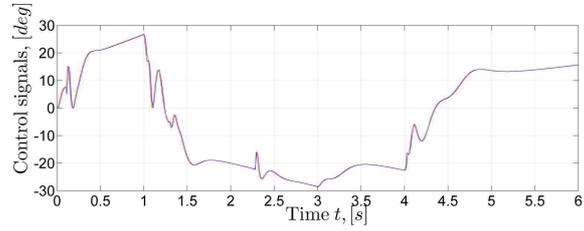
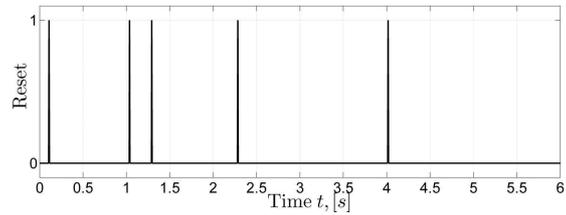
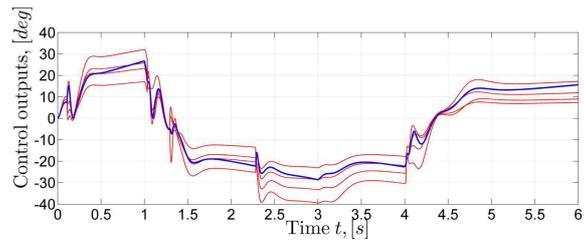


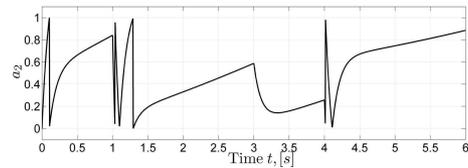
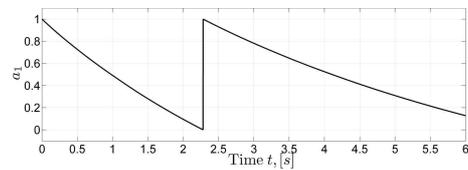
Figure 5.11: Controller blending nonlinear simulations (output).



(a) Actuator command and region number



(b) Controller outputs and reset signal



(c) Normalized distances

Figure 5.12: Controller blending nonlinear simulations (control).

## 5.5 Observer/State Feedback Interpolation

In the previous section,  $\mathcal{H}_\infty$  controllers were computed at nine operating points and then their outputs interpolated at four scheduling regions, in order to compute the final gain-scheduled control law. In this section another interpolation method for the control of the Reichert missile will be detailed, based on the state feedback/observer interpolation technique presented in Section 1.3.2.6.

### 5.5.1 LTI Synthesis

The scheduling regions and the interpolation geometry used for this method are the same as the ones used with the controller blending method. The key difference is in the *interpolation* method itself. The idea here is to convert all nine  $\mathcal{H}_\infty$  controllers of the previous section in estimator-controller form (see Section 3.2) and then obtain a gain-scheduled controller by updating/interpolating its inherent structure (gains, matrices).

LTI  
control  
structure

As far as LTI controllers are concerned, they have the following general standard form:

$$\dot{\hat{x}}_\delta = \mathbf{A}\hat{x}_\delta + \mathbf{B}u_\delta + \mathbf{K}_o(y_\delta - \mathbf{C}\hat{x}_\delta) \quad (5.14)$$

$$u_\delta = -\mathbf{K}_c\hat{x}_\delta. \quad (5.15)$$

The matrices  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$  are in fact the open loop dynamics matrices  $\mathbf{A}_{\text{aug}}$ ,  $\mathbf{B}_u$  and  $\mathbf{C}_y$  of the standard model of Eq. 5.7. As a result the estimator shall inevitably reconstruct all the state vector  $x_{\text{aug}}$  of the open loop system and perform then a state feedback pole placement in order to control it.

### 5.5.2 Gain-scheduled Controller

A simplified block diagram of the state feedback/observer-based gain-scheduled controller is shown in Fig. 5.13. The controller is constructed using a standard state space realization of a Kalman observer (see Eq. 5.14 above). The inputs to the observer are the plant's input and outputs plus the structure of the observer (state feedback/observer matrices and plant matrices). Now all these matrices should be supplied to the observer/controller according to the system's operation using the block named 'Interpolator' for each corner of the scheduling region and the value of the scheduling vector  $\rho$ .

Controller  
realization

Consider first the state feedback/observer matrices  $\mathbf{K}_c$ ,  $\mathbf{K}_o$ : the dimensions of these matrices is  $1 \times 5$  and  $5 \times 3$  respectively, thus the total number of coefficients needing interpolation is twenty (!), which is a rather big number for real world implementation<sup>21</sup>. This is of course done with the same procedure as before concerning controller blending, using normalized distances between controllers for each synthesis points and linear interpolation.

Method  
issues

<sup>21</sup>Compare with the controller blending interpolation where only four signals were interpolated.

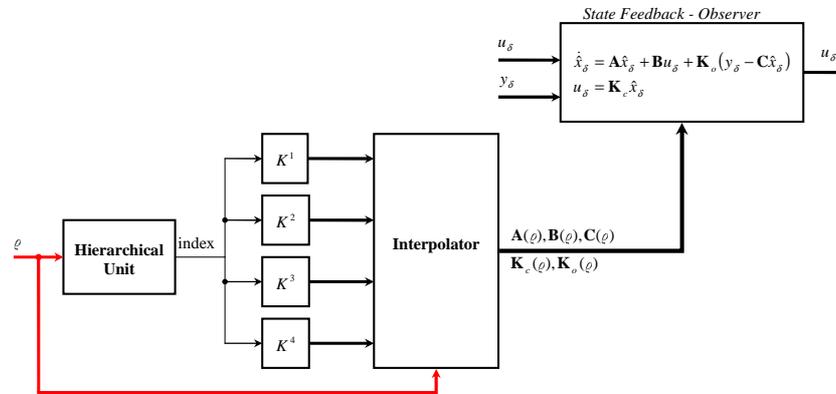


Figure 5.13: State Feedback - Observer Gain-Scheduled Controller.

Inter/tion

The most unrealistic thing however concerning this method is the fact that in order to reconstruct the state of the plant, the observer needs information on the system's structure, that is the matrices  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$ . In order for the observer to reconstruct the state, these matrices should be computed for each reference/operating point of the plant. This may be done in three different ways: *symbolic computation*, *tabulation* or *interpolation*.

Symbolic computation means that the formulas in Chapter 4 (e.g. Eqs. 4.36-4.44) concerning in fact the LPV model of the missile should be evaluated in real time for each reference/equilibrium point. Tabulation means that the corresponding surfaces of these equations for every value of the scheduling vector (see Fig. 4.6) should be stored in memory and retrieved using the value of the scheduling vector. Finally interpolation means that this storing procedure may be done only for the nine synthesis points and then use interpolation for every other intermediate point.

Implem/tion

It is clear that each of these methods presents advantages and disadvantages but globally, all three are not so realistic. The first one is evident that is totally not feasible for real world implementation (even though it offers the best results) since symbolic calculations are very costly in terms of hardware implementation. The second one could be considered but it would be also costly in terms of storage memory for a real world system or for a system that has more than two scheduling variables where coefficients hyper-surfaces should be calculated. Finally, the third method lacks precision because it is clear that nine points could not necessarily cover the whole domain of operation correctly. In addition, interpolation would be needed in order to obtain a value for the system matrix coefficients for every operating point.

This method however, even though it presents all these disadvantages demonstrates a clearly better performance than the controller blending one on various aspects. Simulation results will not be presented here since they may be found in the comparison paper [137] where the two approaches are put side by side and all their advantages and disadvantages stressed out.

Concerning the advantages of this method, it should be at least cited its excellent output time performance and the avoidance of problems regarding controller re-initialization as with the controller blending method. Clearly, it has been already stressed out in the previous section that the main disadvantage of the controller blending method is the not coherent state initialization when passing from one scheduling region to the next and the resulting control transients during this procedure. With this method, this annoying fact is avoided since the controller structure is fixed and only its inherent parameters interpolated whereas with the controller blending method, four controllers need to be realized at each scheduling region with possibly inconsistent I/O representations.

Performance

A last delicate matter concerning this interpolation method based on state feedback/observer control has to do with the issue mentioned in the discussion at the end of Section 3.2.2, concerning the partitioning of the closed loop eigenvalues between the controller and observer. It is really important that this partition be done in an automatic way since this is the essence of gain scheduling: have a systematic and repetitive manner of doing things for a generic parameter-dependent system. However, this fact is not always easy since the closed loop poles resulting from the  $\mathcal{H}_\infty$  controllers may be also complex conjugate or real or a mixture of two and of *different* multiplicity/speed for each synthesis point. Thus the choice of closed loop assignment is clearly not trivial as it may be also seen from the analysis in [4] or even in [20, 21].

Pole  
partitioning

## 5.6 Conclusions

In this chapter two ad-hoc interpolation strategies using the *controller blending* and *state feedback/observer scheduling* methods were tested and compared. Much attention has been given on issues when using these approaches for the control of real world systems.

As far as these issues are concerned, it has been stressed out that the controller blending method seems to be the simplest one in terms of calculations (only four signals are interpolated) with respect to the state feedback/observer one (all control/system matrices are interpolated). However the latter one is easier to implement since only one controller is considered and only its structure interpolated whereas the first one needs always four controller that run in real time. The state feedback/observer method is also of higher performance since it avoids control signal transients and chattering caused by controller re-initialization. However, it remains an open issue on how its LTI controllers should be calculated: should they be chosen in a standard ‘controller poles three times slower than the observer ones’ or via the Youla parametrization-based conversion detailed in the previous section. If the first method is used, the user loses the highly desirable robustness properties of  $\mathcal{H}_\infty$  control theory, whereas if the second method is used, the partitioning of the closed loop poles is not trivial when more than one operating/synthesis points are considered.

Seeing things in a global manner, it is evident that these two methods are both complicated and ad-hoc. The latter issue comes from the fact that regarding LTI controller computation there is practically *no guideline* if the nine synthesis points considered were too few, too many, appropriately partitioned in the flight envelope etc. In addition, the control structure used is too complicated; a sixth order controller for a second order plant.

As a result, a simpler, more efficient and more systematic way to treat the problem is needed that will offer an elegant and practical solution for the control of this type of nonlinear parameter-dependent systems. The next and most important chapter of this work is exactly devoted to that.