Systematic Control Strategies

Overview

In this chapter the main results of this work are presented. A new, systematic and easy to implement gain-scheduled control strategy is proposed: the extended Loop Shaping Design Procedure (or *e*-LSDP). This method is based on the *gain blending* interpolation method and uses the McFarlane&Glover standard LSDP coupled with the gap metric theory in order to provide a gain-scheduled controller that takes into account the nonlinear dynamics variation as a function of the system's operating conditions. This procedure is applied both to the Reichert and ARV benchmark examples to obtain a gain-scheduled autopilot; for both systems an exhaustive analysis is presented focusing on the features and advantages that this method presents over the *ad-hoc* ones analyzed in the previous chapter.

Chapter contents

6.1 Introduction

Novel approach

In this chapter a novel method, for the control of the two benchmark examples (the 'Reichert' missile model and the atmospheric re-entry vehicle) introduced in Chapter 4, will be presented. This method is the *gain blending* method detailed in Section 1.3.2.7 and is coupled with the $\mathscr{H}_\infty$ loop shaping and gap metric theory of Sections 3.3-3.4, in order to provide a systematic control strategy that treats the inconveniences of the two methods detailed in the previous chapter.

Motivation

As it has been already remarked, the two major disadvantages of these methods are *complexity* and *lack of performance-robustness*. Complexity results from two factors: relatively high-order LTI controllers to be interpolated and unknown number of synthesis points required in order to obtain a good coverage of the system's operating domain. Lack of performance results from the interpolation strategy chosen (see for example control transients from the controller blending method) and from the number of synthesis points considered, whereas lack of robustness results from the absence of a systematic/general way to take into account uncertainty in the feedback loop of the gain-scheduled controller.

Solution

The solution of these problems is not easy; to the author's opinion, one straightforward way to treat complexity is by selecting simple to tune yet performing controllers such as PID in order to obtain a basic compensation of the nonlinear system at a relatively small number of synthesis points (e.g. using the corners of the flight envelope or an intuitive selection like in the previous chapter). By interpolating the gains of such controllers an acceptable compensation is taken as a basis, in order to compute an *additional* number of synthesis points where simple enough static $\mathscr{H}_\infty$ controllers (based on the initial loop shaping by the PID ones) are obtained. These $\mathscr{H}_\infty$ controllers capture the nonlinearity of the system by the use of the notion of the gap metric as it will be shown in this chapter. Thus, they correct in a way the somewhat 'ad hoc' loop shaping of the PID controllers since they act on the control signal components and are afterwards interpolated using a triangulation of the system's operating domain. In order to show in fact this amelioration, the 'ad hoc' controllers are compared to the combination of these loop shaping controllers and the additional $\mathscr{H}_\infty$ ones.

The control strategy proposed is applied to the control of the 'Reichert' missile benchmark and to the ARV provided by the EADS foundation. It is clear that the work presented here corresponds to the three last steps of the LBGS procedure detailed in Section 1.3.1; namely to the *LTI Controller Synthesis*, *Gain Interpolation* and *Global Controller Implementation & Validation* steps.

This chapter is divided in two parts: in the first one the gain blending method is applied on the ARV to obtain a regulating autopilot[1], whereas the second is devoted to the missile where the same control strategy is adapted with slight modifications (due to the different control objectives).

[1] The regulating autopilot is computed for the ARV re-entry vehicle using only the systematic *gain blending* method and not the ad-hoc methods of Chapter 4 since this corresponds to the main work of this thesis.

6.2 Gain Blending (re-entry vehicle)

The ARV vehicle

Recall from Section 4.2 that the ARV nonlinear dynamics are described by a state vector that is comprised by the AoA α (in rad) and the pitch rate q (in rad/s). The control input is elevator deflection signal δ_{e} (in rad) whereas the measured output is the AoA. The elevator dynamics are governed by a second order ODE and the rectangular flight envelope $\mathbf{\Gamma}_{\mathrm{fe}}^{[\alpha,M]}$ is a function of the Mach number M and the AoA. The nonlinear aerodynamic functions $C_{\mathrm{m0}}(\alpha, M), C_{\mathrm{me}}(\alpha, M)$ associated with the pitch rate are tabulated for every value of the scheduling vector $\varrho = [\alpha\ M]^T$ inside this envelope[2].

Control objectives

The problem here is to obtain a regulating autopilot that will maintain the AoA to a constant value α_{r} for a given variation profile of the Mach number and the dynamic pressure $Q(M)$, which is an additional non-measurable time-varying variable.

The ARV control objectives are mostly precision and robustness ones and are the following:

Regulation & Flight Envelope. The autopilot should be able to regulate the AoA around a pre-defined reference value α_{r} with $\pm 1\%$ step response steady-state error accuracy when the Mach number follows a given time trajectory inside the vehicle flight envelope.

LTI Synthesis Objectives. The linear controllers designed on the synthesis points should provide at least 50° phase margin and 12dB gain margin. In addition, the dominant closed loop poles must have a damping of at least 0.45 and natural frequencies greater or equal to the dominant open loop ones. Finally, the control effort should be minimized in terms of variation rate.

Gain-scheduled Controller Objectives (nominal). The gain-scheduled controller should be implemented with a sampling period $T_{\mathrm{s}} < 0.15$s and the frozen time open loops during all the Mach time variation range should have at least 30° phase margin, 6dB gain margin and 1 control period delay margin.

Gain-scheduled Controller Objectives (uncertain). The gain-scheduled controller should provide at least 3dB gain margin and one half control period delay margin when heavy additive uncertainties are introduced on the dynamic pressure, on the moment of inertia and on the aerodynamic functions.

The next section details the extended loop shaping design procedure (*e*-LSDP) that corresponds to the third step of the LBGS procedure (LTI Controller Synthesis) of Section 1.3.1. Of course all trim analysis-control, linear models are based on the discussion in Section 4.2 and may be equally found in [139].

[2]For more details see Section 4.2.1.

6.2.1 LTI Controller Synthesis

The e-LSDP (extended-Loop Shaping Design Procedure) devised in this work is based on the standard one of Section 3.3.2 and is used to compute a global gain-scheduled controller for a nonlinear parameter-dependent plant. It incorporates not only the LTI controller synthesis phase of Section 1.3.1 but also a systematic way of choosing the operating points where this synthesis will occur.

The e-LSDP corresponds as said, to the third step of the LBGS procedure of Section 1.3.1 concerning the LTI controllers used in order to cover the operating domain of the system. Additionally, in the robust $\mathscr{H}_\infty$ loop shaping - gap metric context of Chapter 3 an operating point choice algorithm is proposed. The e-LSDP is decomposed in three steps:

The e-LSDP

Step 1 - Loop Shaping. A linearized model of the vehicle $G(s)$ is obtained (see Section 4.2.3) for five synthesis points inside the flight envelope, namely on the corners and on the center of the envelope (see Fig. 6.1):

$$[\alpha, M] : [30^\circ, 4], [50^\circ, 4], [30^\circ, 26], [50^\circ, 26], [40^\circ, 15]$$

The linearized plant is then augmented using a pre-compensator $W_1(s)$ (which is actually the actuator dynamics) and a post-compensator $W_2(s)$ (which is a filtered PID controller) in order to provide basic performance and robustness requirements for the aforementioned five synthesis points corresponding linearized plants. The analysis on how to obtain these controllers is detailed in the next section. For any other point of the flight envelope, a linear interpolation of these gains is used by considering the four corresponding triangular interpolation regions $\Gamma^1, \Gamma^2, \Gamma^3$ and Γ^4.

Step 2 - Operating Point Algorithm. Given that a simple linear interpolation of the f-PID controllers is not sufficient as it will be seen in Section 6.2.3, an additional number of controllers should be used in order to treat the variation of the nonlinear dynamics of the vehicle. A variation indicator is the undamped natural frequency of the complex conjugate poles of the linearized plant for every value of the scheduling vector $\varrho = [\alpha \ M]^T$ (see Fig. 4.15). However, given that a *closed loop* criterion would be preferred, here the *gap metric* notion is used to quantify such variation.
Recall once again from Section 3.4.2, Theorem 3.6, that the gap metric between a nominal LTI plant G and a perturbed one G_Δ are closely related to the stabilizability of both plants by the same $\mathscr{H}_\infty$ controller K_∞ (designed for the nominal plant), and the robustness margin $\epsilon_{\max}$[3].
The autopilot regulates α around a constant α_{r} for all M; thus these additional synthesis points are sought on this line of the flight envelope; for the rest of the analysis concerning this algorithm see Section 6.2.1.2.

[3] Recall that the maximum robustness margin is smaller in the static case than in the full order one (see Section 3.3.4).

Step 3 - $\mathscr{H}_\infty$ Controller Synthesis. After the algorithm using the gap metric described in Section 6.2.1.2 has found the additional set of synthesis points, robust $\mathscr{H}_\infty$ controllers are computed at these points to increase the robustness of the final gain-scheduled controller. The additional controllers are static ones to reduce complexity and are of course also interpolated at intermediate points using the Mach number (and for constant α), in the fourth step of the LBGS procedure (see Section 6.2.2).

These two last steps of the e-LSDP are closely connected; the robust controller K_∞ may be computed after all synthesis points are found (since the robustness margin associated to a given synthesis point depends *only* on the initial loop shaping) or immediately when this point is found.

The nominal open loop plant $G(s)$ when using the operating point choice algorithm, is in fact the augmented plant $G_\mathrm{s}(s)$ obtained by the series interconnection of the pre/post compensators of Step 1 and is written as $G_\mathrm{s}(s) = W_2(s)G(s)W_1(s)$, for any operating point. As it has already been mentioned, the linearized vehicle model $G(s)$ varies as function of ϱ and so does the post-compensator $W_2(s)$, since its gains are the interpolation of the controller gains at the initial five synthesis points given in Step 1.

The final implemented controller $K_\mathrm{s}(s)$ for a synthesis point issuing from the algorithm is in fact the series interconnection of the pre/post compensators and the static robust controller. The robust controller becomes itself also interpolated at the fourth step (Controller Interpolation) of the LBGS procedure as it will be presented in Section 6.2.2.

`Discussion` The operating point choice algorithm treats in fact the variation of the open loop shaped linear dynamics $G_\mathrm{s}(s)$ between a nominal operating point and a subsequent one, as uncertainty. This uncertainty is in fact visualized in the gap δ_g between these two systems. Leaving details for Section 6.2.1.2, it will only be mentioned that by computing the gap between subsequent 'uncertain' systems and a nominal one, the designer can find out until what point this dynamics variation is tolerable by a robust controller K_∞ designed for the nominal point.

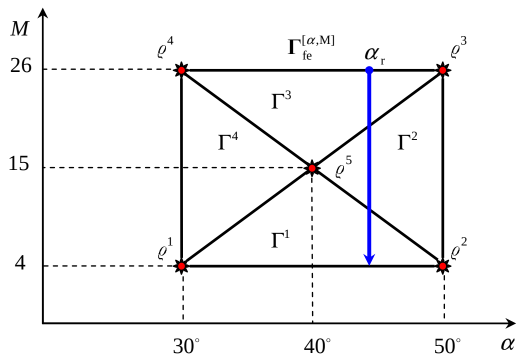


Figure 6.1: ARV flight envelope.

6.2.1.1 Loop Shaping

As it has been mentioned in the previous section, in order to obtain a basic control action for the ARV, five f-PID controllers are designed. These controllers are designed using the linearized, scheduling vector-dependent state space model $\mathcal{S}_{\mathrm{LPV}}(\varrho_{\mathrm{r}})$ of the ARV at these points given by Eqs. 4.58-4.68. The equivalent matrix transfer function of the vehicle $G(s)$ at each operating point gives the I/O relation of the control input δ_{e} to each state variable (AoA and pitch rate). Given that only the first state variable is measured, the plant is SISO with $G_\alpha(s)$ being the transfer function from the control input to the AoA.

Basic control action

In the loop shaping context (see also Section 3.3), the transfer function $G_\alpha(s)$ is augmented by the actuator acting as a pre-compensator (with $W_1(s) = G_{\mathrm{a}}(s)$) on the control signal, and by a filtered PID controller (f-PID) $G_{\mathrm{c}}(s)$ acting on the regulation error $e_\delta = \alpha_\delta - \alpha_{\mathrm{r}}$[4]. The control signal $\delta_{\mathrm{e,c}}$ ('c' for commanded) is fed to the actuator that produces the final control stabilizing control signal $\delta_{\mathrm{e},\delta}$. The loop shaping block diagram is shown in Fig. 6.2 with the corresponding open loop transfer function being:

Loop shaping

$$G_{\mathrm{s}}(s) = G_{\mathrm{c}}(s) G_\alpha(s) G_{\mathrm{a}}(s). \tag{6.1}$$

The filtered PID controller used for the AoA regulation has the following transfer function:

f-PID controller

$$G_{\mathrm{c}}(s) = \frac{1}{\frac{1}{N}s + 1} \left(K_{\mathrm{p}} + \frac{K_{\mathrm{i}}}{s} + K_{\mathrm{d}} s \right) \tag{6.2}$$

with $K_{\mathrm{p}}, K_{\mathrm{i}}, K_{\mathrm{d}}$ being the PID gains and N the filter's time constant inverse.

As it has been detailed in Chapter 4 concerning the stability analysis of the linearized models of the vehicle, its open loop dynamics are conditionally unstable given that the poles are purely imaginary. The nonlinear control problem is challenging since their undamped natural frequency ω_0 changes as a function of the Mach number and the AoA, as it may be observed from Fig. 4.15. Here however the latter is mostly important since the autopilot is a regulation one around a constant α_{r}.

Control challenge

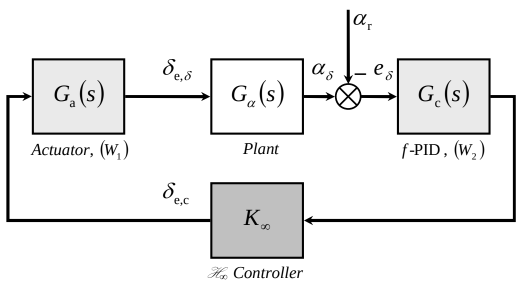


Figure 6.2: Loop shaping block diagram.

[4]The subscript 'δ' notation is used to emphasize around equilibrium operation.

The control goal is somewhat different from the Reichert missile one; the need for a classic signal tracking performance with appropriate rise times, settling times etc. is not crucial. Here the autopilot should mostly provide good stability, delay margins and damping while minimizing the control effort.

Consider now one synthesis point (e.g. point No. 1 for $\varrho^1 = [30°, 4])$ in order to detail the correction needed by the f-PID controller. The vehicle's open loop transfer function $G_\alpha(s)$ presents two complex conjugate poles with an undamped natural frequency ω_0 and zero damping since these poles are purely imaginary. The PID controllers shall correct this fact by using its two complex conjugate zeros (adjusted by the three gains $K_\mathrm{p}, K_\mathrm{i}, K_\mathrm{d}$) and attract these poles into the negative complex plane. The integrator of the controller will provide zero steady-state error whereas the filtering part that is a first order transfer function limits the control effort and adjusts the bandwidth of the system.

In terms of frequency response, the choice of the controller's parameters is not trivial; here a classic Bode response correction is used to provide an initial adjustment whereas fine-tuning is performed by using MATLAB® Simulink Control Design toolbox and its optimization routines.

Controller zeros influence

The natural frequency of the controller's complex zeros $\omega_{0,z}$ is equal to $\sqrt{K_\mathrm{i}/K_\mathrm{d}}$ and plays a significant role in providing the correct gain and phase margins for the open loop plant as well as the bandwidth, combined by the filter action. It should be chosen near but a bit smaller than the open loop natural frequency of the plant's complex conjugate poles. Reducing this frequency by moving the zeros nearer the origin, the gain magnitude increases starting from a lower frequency and thus the gain crossover frequency ω_gc is increased[5]. In addition, given that the ω_gc increases and the phase continuously decreases to $-180°$, the phase margin gets smaller[6]. The phase crossover frequency ω_pc is almost one decade further on and is not so much influenced by the movement of the zero, however given that by reducing the zeros' frequency the loop gain increases, the gain margin decreases.

The damping now of the controller zeros is governed by K_p if the other two gains are fixed; its influence is more complicated on the frequency response. In general, if the damping is increased, the step performance of the plant is ameliorated with the cost of deteriorating the stability margins and augmenting the control signal amplitude needed.

[5]The open loop magnitude starts at low frequencies from a value dictated by the controller zeros natural frequencies and drops with -20dB/dec until the point where the zeros start to act and increase the gain. Then the gain increases even more (mathematically to infinity around ω_0) due to the imaginary poles of the plant before falling once again due to the filter pole with -40dB/dec for higher frequencies (-80dB/dec if the actuator poles are added).

[6]The open loop phase starts from $-90°$ due to the integrator and then starts to increase due to the complex controller zeros until ω_0; then it suddenly loses $-180°$ due to the plant's complex conjugate purely imaginary poles. However the phase remains sufficient due to the total phase added until ω_0 by the zeros (about $150°$). It then continues to decrease due to the filter pole until $-180°$; if the actuator is counted also, then it continues to drop further on until $-360°$ at high frequencies.

Controller pole influence

Concerning now the influence of the filter's pole, things are also a bit complicated: given that the pole's frequency is bigger than the gain crossover frequency, it does not directly affect it for small displacements. If it is reduced, it starts also reducing the total phase added by the controller (being the combination of the phase due to the complex zeros and the pole) and thus deteriorate the phase margin. However, once this frequency is chosen (roughly at the middle of the zone $[\omega_{\mathrm{gc}}, \omega_{\mathrm{pc}}]$) it may be fine-tuned using Simulink Control Design.

Controller results

These concepts may be seen in Fig. 6.3[7] where the transfer functions of the open loop plant $G_\alpha(s)G_\mathrm{a}(s)$, the compensator $G_\mathrm{c}(s)$ and the combined, corrected (or 'shaped') open loop $G_\mathrm{s}(s) = G_\mathrm{c}(s)G_\alpha(s)G_\mathrm{a}(s)$ are shown together.

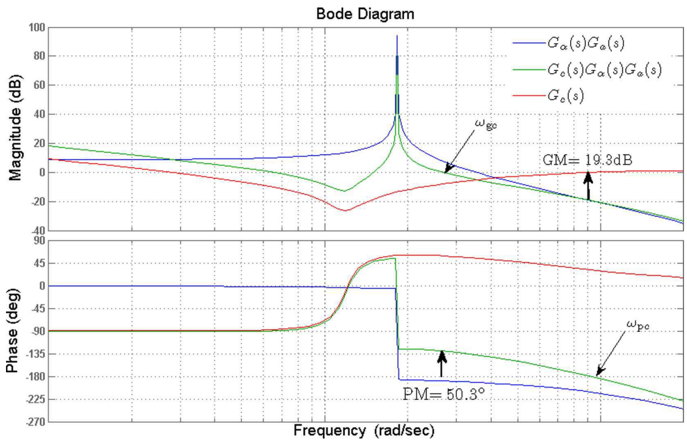


Figure 6.3: Correction open loop transfer function.

In Fig. 6.4 is shown the root locus diagram of the closed loop; in order to view the closed loop poles, the loop gain should be chosen as unitary.

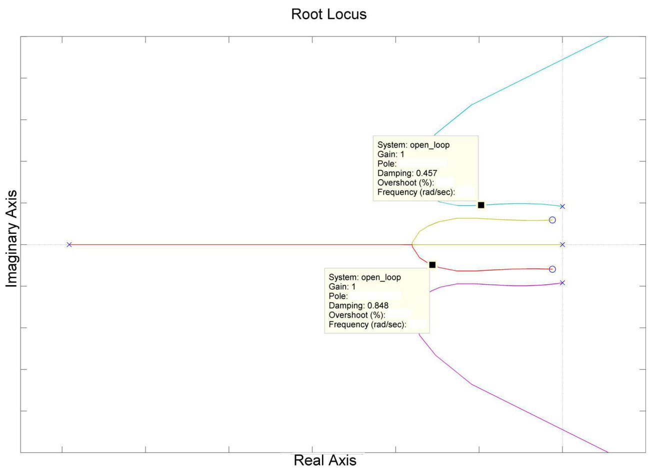


Figure 6.4: Open/closed loop poles diagram.

[7] Frequency values are omitted for confidentiality reasons.

The upper box in the previous figure shows the location of the vehicle's closed loop poles when the loop gain is unitary; the damping is satisfactory (0.457) as is demanded by the LTI synthesis objectives of Section 6.2. In addition, the controller closed loop poles are shown by the lower box; for high gains they tend to the open loop complex zeros. Here the actuator poles are not shown since they are much further on the left.

Fine-tuning & final results

Concerning now the fine-tuning performed using MATLAB® Simulink Control Design toolbox, it should be pointed out that it permits to optimize all four controller gains by putting constraints on the closed loop pole minimum damping and natural frequencies as well as on the stability margins. Performing this optimization for all points yields the results of Table 6.1.

Finally the five Nichols charts of the open loop systems are shown in Fig. 6.5a; the correct GM, PM achieved may be observed. In Fig. 6.5b are also shown the closed loop poles and zeros for all synthesis points.

Table 6.1: Loop shaping results(i),(ii)

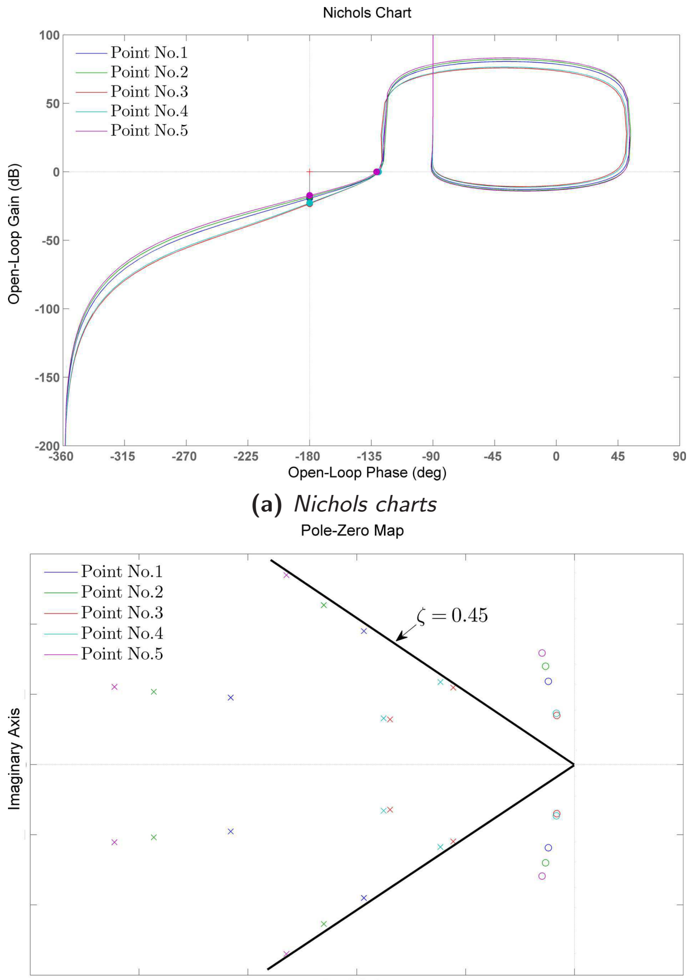
Points	**1**	**2**	**3**	**4**	**5**
K_{p}	0.04904	0.04879	0.06209	0.06868	0.06867
K_{i}	0.29011	0.36260	0.18882	0.22324	0.58750
K_{d}	0.20454	0.18348	0.38169	0.41269	0.23058
N	5.91630	7.43050	3.04130	3.25040	8.55560
GM	19.3	18.1	23.2	22.7	17.3
PM	50.3	50.2	50.3	50.4	49.0
ω_{pc}	2.65	3.15	1.54	1.64	3.59
ω_{gc}	9.18	10.2	6.71	6.93	10.8
t_{s}	4.77	3.98	8.31	7.84	3.46
$\lvert\dot{\delta}_{\mathrm{e},\delta}\rvert_{\max}$	12.7	13.8	12.8	14.8	20.4

(i) The gain margin (GM), phase margin (PM), phase crossover frequency (ω_{pc}), gain crossover frequency (ω_{gc}), settling time (t_{s}) and maximum control rate ($\lvert\dot{\delta}_{\mathrm{e},\delta}\rvert_{\max}$) are measured in dB, deg, rad·s^{-1}, rad·s^{-1}, s and deg·s^{-1} respectively.

(ii) The settling time is measured within a 2% envelope.

If a gain-scheduled controller is constructed using only the family of five points and the corresponding interpolated controller gains are computed using the four scheduling regions of Fig. 6.1, then the results are not satisfactory in terms of stability margins both for the nominal and uncertain cases, as it will be presented in Section 6.2.3.

For this reason, some additional points are added on the line for constant α_{r} and robust $\mathscr{H}_\infty$ controllers are designed and also interpolated using the Mach number. The operating point choice algorithm is detailed in the next section.



(a) *Nichols charts*

(b) *Pole-zero maps*

Figure 6.5: Loop shaping results.

6.2.1.2 Operating Point Algorithm

Motivation The operating point choice algorithm will add some additional synthesis points on the line α_r in order to ameliorate the robustness of the gain-scheduled controller detailed in Section 6.2.2. On these additional points, static $\mathscr{H}_\infty$ controllers are designed using the analysis of Section 3.3.4. The algorithm is based on the analysis of Section 3.4.2 that merges the loop shaping control theory and the gap metric.

Discussion The general idea behind the algorithm is the following: the designer computes the open loop corrected transfer function of the system $G_\mathrm{s}(s)$ at a nominal point (e.g. $\alpha = \alpha_\mathrm{r}, M_0 = 26$), using interpolation for the PID gains and computes the corresponding robustness margin for this point. This is done of course *after* a static robust controller K_∞ is computed using the analysis of Section 3.3.4[8]. Then, for a neighbor operating point (say for $M_0 + \delta M$), the new corrected open loop $G_{\mathrm{s},\Delta}(s)$ is computed and then the gap between these two open loops is calculated. If this gap is smaller than the robustness margin associated with the nominal point, then this means that the robust controller is satisfactory for the neighbor point; if not a new operating point is chosen, a new robust controller and robustness margin computed and the algorithm continues until the flight envelope is covered. The algorithm is formally divided in the following steps:

Step 1 - Initialization. Choose a gridding (e.g. equidistant) over the Mach number range [26,4] and thus obtain a set of candidate synthesis points $\Sigma_M = [M_1, \ldots, M_k]$. Then take as the initial operating point P^j (corresponding to a scheduling vector value ϱ^j) the one corresponding to $M = 26$.

Step 2 - Interpolated Loop Shaping. For the operating point P^j, compute the open loop shaped plant $G_\mathrm{s}^j = G_\mathrm{c}^j G_\alpha^j G_\mathrm{a}$. The plant G_α^j is simply the linearization of the nonlinear parameter-dependent vehicle model at $\varrho = \varrho^j$, whereas the f-PID controller gains are obtained using a triangular interpolation of the five synthesis points of the previous section[9]. For the shaped plant G_s^j, a static $\mathscr{H}_\infty$ controller is calculated using Theorem 3.5 and the corresponding to the point P^j robustness margin ϵ^j is computed.

Step 3 - Line Search or Reset. Performing a line search using subsequent candidate points belonging to Σ_M, successive shaped plants G_s^f are computed, until the gap $\delta_\mathrm{g}(G_\mathrm{s}^j, G_\mathrm{s}^f)$ between the nominal initial plant and the successive one is greater or equal than the robustness margin ϵ^j. If this is the case, then a new operating point P^j is chosen and then the algorithm jumps back to Step 2, except for the case when the end of the flight envelope is reached. In this case, even if $\delta_\mathrm{g}(G_\mathrm{s}^j, G_\mathrm{s}^f) < \epsilon^j$, the final point is selected and the procedure terminated.

[8] Recall that in the full order case the robustness magin is computed before actually computing the controller; however here it is not possible.

[9] For more details on the triangulation process see Section 6.2.2 further on.

Table 6.2: Robust controller gains

Mach	$K_{\mathrm{p},\infty}$	$K_{\mathrm{i},\infty}$	$K_{\mathrm{d},\infty}$	ϵ
26	0.5181	0.5965	0.9424	0.3297
23.5	-0.1472	0.7028	1.0769	0.3374
20.75	-0.1559	0.6507	1.0353	0.3363
18.25	0.4020	0.5894	0.8927	0.3358
16	-0.0899	0.5609	0.8148	0.3162
4	-0.0318	0.6395	0.9429	-

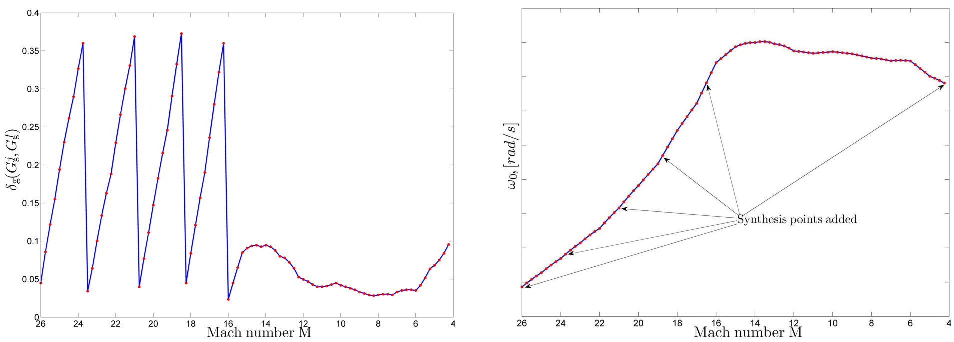
Algorithm results

Using this algorithm, totally six additional $\mathscr{H}_\infty$ controllers are computed. The static controller $K_\infty \in \mathbb{R}^{3\times 1}$ treats each of the three channels of the f-PID controller[10]. Each of the gain elements $K_{\mathrm{p},\infty}, K_{\mathrm{i},\infty}, K_{\mathrm{d},\infty}$ (one for each channel) as well as the corresponding Mach numbers are given in Table 6.2.

In Figs. 6.6a-6.6b some results on the operating point choice algorithm are presented. In the first figure, the gap $\delta_{\mathrm{g}}(G_{\mathrm{s}}^j, G_{\mathrm{s}}^f)$ evolution with respect to M is given for a gridding performed each 0.25 units (totally 88 points) whereas in the second, the natural frequency ω_0 evolution with respect to M is shown.

From Fig. 6.6a it may be observed that the gap increases until the first robustness margin $\epsilon^1 = 0.3297$ (see Table 6.2) is surpassed; the algorithm is then re-initialized until all Mach range is covered. It may also be observed that all re-initializations take place (and thus synthesis points added) until approximately $M = 16$; further on, the gap is rather small. The algorithm thus continues until the flight envelope is finished and adds the final point at $M = 4$.

This behavior is explained from Fig. 6.6b showing the linearized plant's natural frequency ω_0 variation, being an indicator of the system's 'nonlinearity'. This frequency increases rapidly until $M = 16$ but then remains almost constant; this is captured by the algorithm which decides that the plant's dynamics do not change significantly to justify another synthesis point until $M = 4$.



(a) *Gap between subsequent shaped plants* **(b)** *Natural frequencies & synthesis points*

Figure 6.6: Operating point choice algorithm results.

[10]For more details on the synthesis scheme refer to the next section.

6.2.1.3 $\mathscr{H}_\infty$ Controller Synthesis

`Control goal` In this section, the synthesis procedure concerning the static, robust $\mathscr{H}_\infty$ controllers of the previous section is detailed. Recall from Section 3.3.2 concerning the LSDP that the robust controllers are in fact designed for a shaped open loop plant $G_\mathrm{s}(s)$; additionally nothing changes for the synthesis problem in terms of posing (except of course for the LMI's) be the designed compensator of full or zero order. The final goal is to compute a static controller K_∞ for $G_\mathrm{s}(s)$ in order to guarantee a stable loop and additionally:

$$\left\| \begin{bmatrix} K_\infty \\ \mathbb{I} \end{bmatrix} (\mathbb{I} - G_\mathrm{s} K_\infty)^{-1} \tilde{M}^{-1} \right\|_\infty \leq \gamma, \quad \gamma = \epsilon^{-1} > 0. \tag{6.3}$$

Recall from the previous section that these controllers computed totally at six additional synthesis points (see Table 6.2) yield a robustness margin ϵ^j for the corresponding linearized shaped plants $G_\mathrm{s}(s)$; neighbor plants are also well-behaved under the same corresponding controller due to the gap metric theory.

`Synthesis structure` The open loop shaped plant is SISO and thus a robust controller would be a simple gain on the output of the f-PID controller, thus not permitting significant amelioration on the feedback loop. However, if the f-PID controller's control signal is broken in three parts (proportional, integral and derivative) then the robust controller is a three element matrix $(K_{\mathrm{p},\infty}, K_{\mathrm{i},\infty}, K_{\mathrm{d},\infty})$. The synthesis block diagram corresponding to Fig. 6.2 is shown in Fig. 6.7[11].

As a final comment concerning the robust controllers, it is clear that the control structure is really simple (compared to a dynamic robust controller that would be of order five), easy to implement-interpolate and of high performance, as it will be presented in Section 6.2.3.

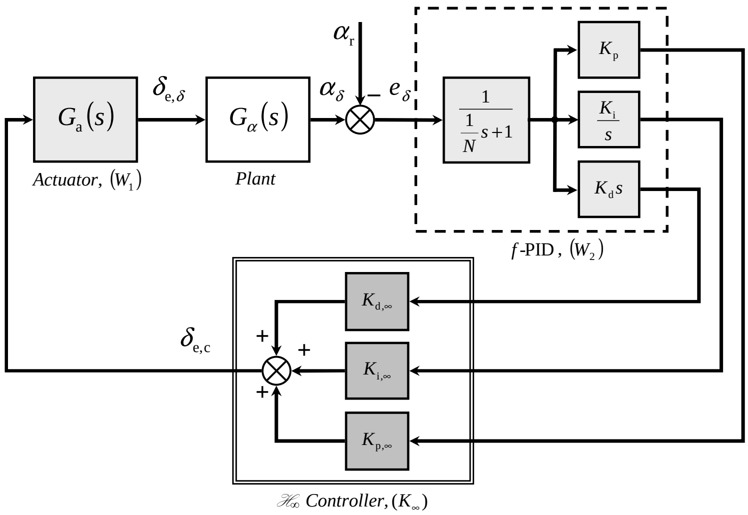


Figure 6.7: Robust controller synthesis block diagram.

[11] For details on solving the synthesis problem refer to Section 3.3.4.

6.2.2 Gain Interpolation

The gain-scheduled controller presented in the next section uses gain interpolation in order to update the LTI controllers' parameters, based on the scheduling vector values. The parameters interpolated are the f-PID gains $K_{\mathrm{p}}, K_{\mathrm{i}}, K_{\mathrm{d}}, N$ and additionally the robust controller gains $K_{\mathrm{p},\infty}, K_{\mathrm{i},\infty}, K_{\mathrm{d},\infty}$.

The first are interpolated using the five initial synthesis points of Table 6.1 and the corresponding four triangular scheduling regions $\Gamma^1, \Gamma^2, \Gamma^3, \Gamma^4$. To create the scheduling regions, Delaunay triangulation is used (refer to Section 1.4) using the coordinates $[\alpha^i\ M^i]^T, i = 1, \ldots, 5$ of all five points. Each gain is interpolated by considering the corresponding plane equation defined by the coordinates of each triangle corner of every scheduling region. `f-PID interp/tion`

Consider for example the f-PID controller derivative gain K_{d}, the three corner gains $K_{\mathrm{d}}^1, K_{\mathrm{d}}^2, K_{\mathrm{d}}^5$ of scheduling region Γ^1 (see Fig. 6.1) and the scheduling vector coordinates $\varrho^1 = [\alpha^1\ M^1]^T, \varrho^2 = [\alpha^2\ M^2]^T, \varrho^5 = [\alpha^5\ M^5]^T$. Then the interpolated gain $K_{\mathrm{d}}(\varrho)$ with $\varrho \in \Gamma^1$ is computed by solving the plane equation leading to the following solution[12]:

$$K_{\mathrm{d}}(\varrho) = \frac{c_1 - c_2\alpha(t) - c_3 M(t)}{c_4}. \tag{6.4}$$

The constants c_1, c_2, c_3 and c_4 are dependent only to the data concerning the synthesis points and are calculated as:

$$c_1 = \begin{vmatrix} \alpha^1 & M^1 & K_{\mathrm{d}}^1 \\ \alpha^2 & M^2 & K_{\mathrm{d}}^2 \\ \alpha^5 & M^5 & K_{\mathrm{d}}^5 \end{vmatrix}, c_2 = \begin{vmatrix} 1 & M^1 & K_{\mathrm{d}}^1 \\ 1 & M^2 & K_{\mathrm{d}}^2 \\ 1 & M^5 & K_{\mathrm{d}}^5 \end{vmatrix}, c_3 = \begin{vmatrix} \alpha^1 & 1 & K_{\mathrm{d}}^1 \\ \alpha^2 & 1 & K_{\mathrm{d}}^2 \\ \alpha^5 & 1 & K_{\mathrm{d}}^5 \end{vmatrix}, c_4 = \begin{vmatrix} \alpha^1 & M^1 & 1 \\ \alpha^2 & M^2 & 1 \\ \alpha^5 & M^5 & 1 \end{vmatrix}. \tag{6.5}$$

The three $\mathscr{H}_\infty$ controller gains are linearly interpolated as a function of the Mach number for the constant regulation value of the AoA α_{r}, considering the five intervals formed by the six additional synthesis points added by the gap metric operating point choice algorithm of Section 6.2.1.2. Consider for example the proportional channel gain $K_{\mathrm{d},\infty}$, the middle interval $[M^3, M^4]$ and the corresponding gains $K_{\mathrm{d},\infty}^3, K_{\mathrm{d},\infty}^4$. Then the interpolated value $K_{\mathrm{d},\infty}(M)$ is given by: $\mathscr{H}_\infty$ `controller interpolation`

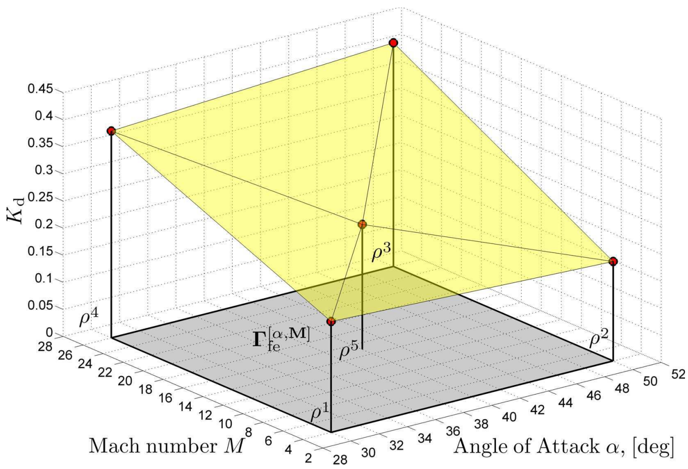
$$K_{\mathrm{d},\infty}(M) = K_{\mathrm{d},\infty}^4 a_M(t) + \big[1 - a_M(t)\big] K_{\mathrm{d},\infty}^3 \tag{6.6}$$

with $0 \leq a_M(t) \leq 1$ being the normalized distance given by:

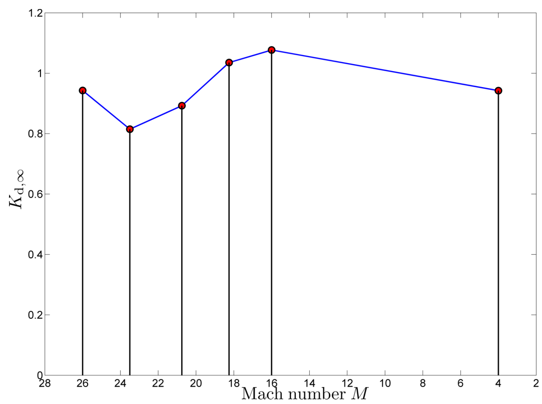
$$a_M(t) = \left| \frac{M(t) - M_3}{M_4 - M_3} \right|. \tag{6.7}$$

In Fig. 6.8a are shown the interpolation surfaces corresponding to the derivative gain K_{d} for all four triangular scheduling regions whereas in Fig. 6.8b is shown the interpolated derivative robust gain $K_{\mathrm{d},\infty}$ for all five linear interpolation regions for constant AoA.

[12]The interpolated gain may be also seen as the projection of the current scheduling vector coordinates on the plane defined by the three corners of each scheduling region.



(a) *Interpolation surfaces for* K_{d}



(b) *Interpolation line for* $K_{\mathrm{d},\infty}$

Figure 6.8: Gain interpolation results.

6.2.3 Controller Implementation & Validation

The global gain-scheduled controller will be detailed in this section; in the first section its structure and some other minor issues will be detailed whereas in the following section some simulation results will be presented.

6.2.3.1 Nonlinear Gain-scheduled Controller

The global gain-scheduled controller is implemented by discretizing the f-PID controller using bilinear transformation[13] with a sampling time $T_s < 0.15s$ and an ideal sampler. All seven gains are then interpolated using the procedure described in the previous section.

Global controller

The total control signal $\delta_{e,tot}$ supplied to the actuator is the sum of the trim control signal $\delta_{e,r} = \delta_e(\varrho_r)$ (see Eq. 4.57 in the trim analysis Section 4.2.2) and the closed loop scheduled stabilizing signal $\delta_{e,c}$. The transfer function $\mathcal{K}(s, \varrho)$ providing the gain-scheduled control signal before discretizing is:

$$K(s,\varrho) = \frac{1}{\frac{1}{N(\varrho)}s + 1}\left[K_p(\varrho)K_{p,\infty}(\varrho) + \frac{K_i(\varrho)K_{i,\infty}(\varrho)}{s} + K_d(\varrho)K_{d,\infty}(\varrho)s\right] \quad (6.8)$$

Controller block diagram

The following figure shows the Simulink diagram of the gain-scheduled controller. The big grey block represents the ARV nonlinear dynamics, the small one generates the Mach reference time trajectory illustrated in Fig. 4.12a and the red block generates the AoA reference value. The upper small blue block generates the trim control signal whereas the other two big blue blocks represent the discretized controller of Eq. 6.8 and the interpolating functions used to update its gains as a function of the scheduling vector. Finally the yellow block represents the actuator dynamics.

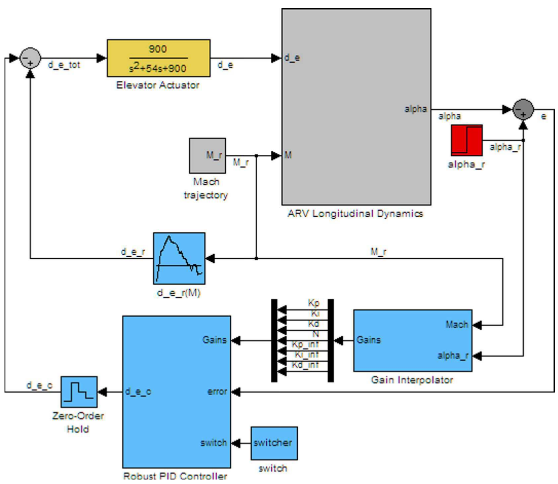


Figure 6.9: Robust controller synthesis block diagram.

[13]Recall that using this transformation the Laplace variable is replaced with $s = \frac{2}{T_s}\frac{z-1}{z+1}$.

6.2.3.2 Simulation Results

In this section some simulation results will be presented both for the nominal and the uncertain case corresponding to the control objectives of Section 6.2, demonstrating the effectiveness of the proposed control scheme.

Nominal case results

For the *nominal case*, the simulation profiles used for the Mach and the dynamic pressure are already presented in Figs. 4.12a-4.12b. The goal for the autopilot is to regulate the AoA around the reference value with $\pm 1\%$ steady-state accuracy. The time simulation of the gain-scheduled controller is shown in Fig. 6.10; the blue curve shows the response if only the f-PID controller is used whereas the red one if both the f-PID controller and the $\mathscr{H}_\infty$ controllers are used. The steady state margins are satisfied for both cases with slight differences in the amplitude; however the stability margin performance is not good if the robust controllers are not used, as it is demonstrated further on.

To test these stability margins, the gain-scheduled system is linearized every 10s and thus totally 57 frozen time open loop systems are obtained[14] when using either the f-PID controller or both the f-PID and $\mathscr{H}_\infty$ controllers. In Fig. 6.11a the Nichols charts for these two cases are shown whereas in Figure 6.11b the corresponding gain & phase margins (GM & PM) are plotted for each system. Using these figures it may be seen that the GM lower limit of 6dB is never violated and the robust controllers provide an amelioration of up to 2dB's. The results are even better for the PM since the f-PID controller by itself is not sufficient with the lower limit of 30° being violated for $350 \lesssim t \lesssim 435$s and reaching its worst point of 24° at $t \simeq 380s$. The robust controller, using the additional synthesis points, succeeds at augmenting the PM up to 7.5° and thus helps the gain-scheduled controller meeting the robustness constraints[15].

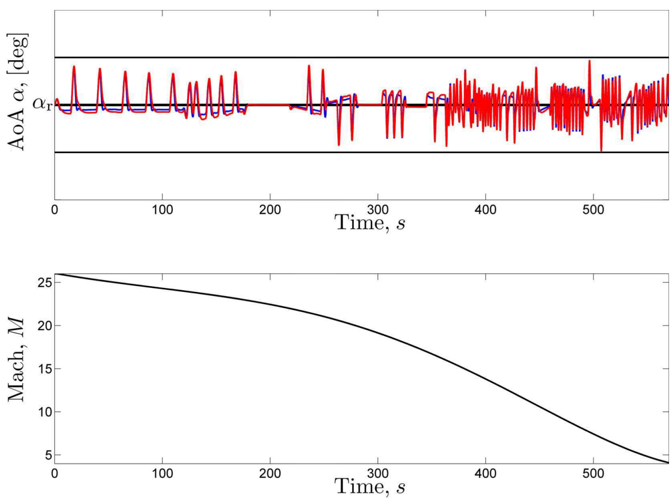
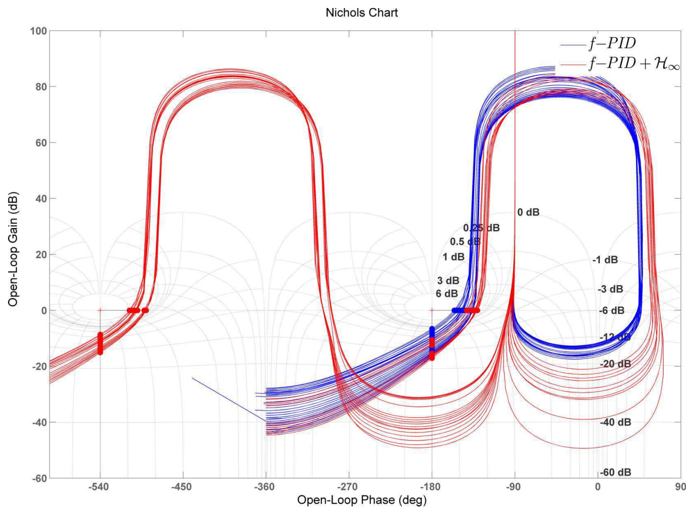
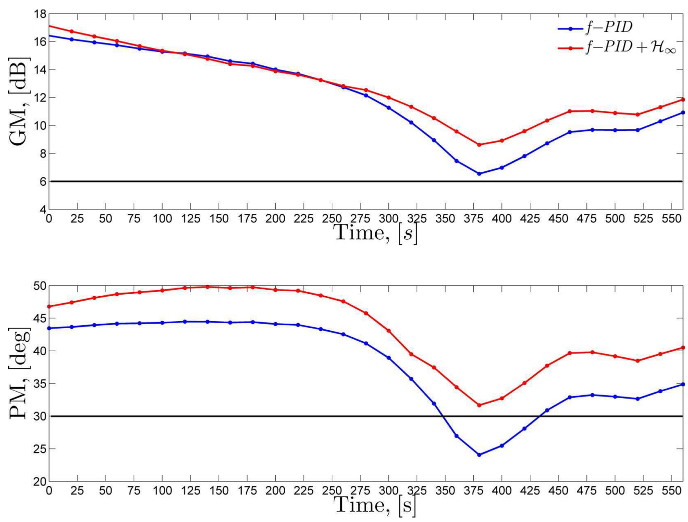


Figure 6.10: Gain-scheduled controller time performance.

[14] This is done using the MATLAB® Simulink Control Design toolbox.
[15] The biggest augmentation is observed in fact for the worst point ($t \simeq 380s$).



(a) *Nichols charts*



(b) *Gain & Phase Margins*

Figure 6.11: Simulation results (nominal).

Uncertain case results

Concerning now the *uncertain* case, the gain-scheduled autopilot is also tested in the face of uncertainties over the dynamic pressure Q, the moment of inertia $\mathrm{I_{yy}}$ and the aerodynamic functions C_{m0} and C_{me}. Two worst case scenarios are considered in total; $\pm 35\%$ and $\mp 50\%$ additive uncertainties on C_{m0} and C_{me}, $+35\%$ on Q and $+10\%$ on $\mathrm{I_{yy}}$.

The delay margin (DM) is tested for both scenarios by considering the *nominal* case plus the *robust* cases (with totally 9+9 uncertain runs obtained by increasing the uncertainty over the four variables by 10% each time until reaching the maximum uncertainty limit) and the results are shown in Figures 6.12a, 6.12b for all 57 frozen time models. Moreover, the Nichols charts (see Figures 6.13a, 6.13b) show the stability margins of these linearized open loops for the worst cases (maximum uncertainty norm) of both uncertain scenarios.

The minimum gain, phase and delay margin (in sampling periods) for the nominal case and both uncertain scenarios[16] are found in Table 6.3. It can be observed that the additional points added with the synthesis point selection algorithm of Section 6.2.1.2 have clearly assisted the gain-scheduled controller meeting the specifications imposed in Section 6.2 with only small violations on the delay margins. Obviously, no uncertain cases are considered for the simple PID tuning since not even the nominal ones are satisfied; this in fact shows the necessity of the $\mathscr{H}_\infty$ controllers.

Table 6.3: Stability margin results[(i)]

Study case	**GM**	**PM**	**DM**
PID (nominal)	6.5(6.0)	24.1 (30)	0.70 (1.0)
PID+$\mathscr{H}_\infty$ (nominal)	8.7(6.0)	32.2 (30)	1.00 (1.0)
PID+$\mathscr{H}_\infty$ (uncertain case No.1)	6.4(3.0)	17.3 (na)	0.44 (0.5)
PID+$\mathscr{H}_\infty$ (uncertain case No.2)	3.4(3.0)	15.9 (na)	0.40 (0.5)

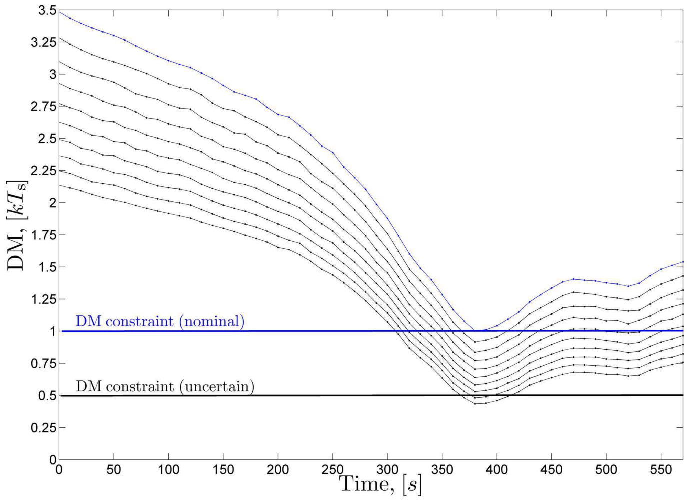
(i) The constraint values are given in parentheses.

6.2.3.3 Discussion

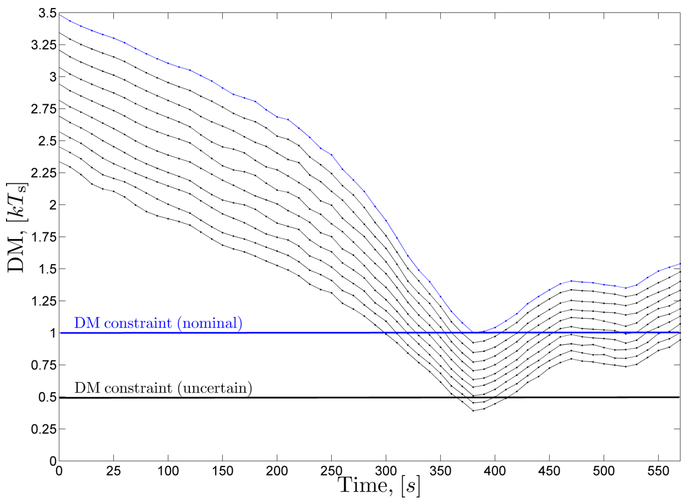
The autopilot designed in this section is used for the regulation of the ARV AoA around a reference value α_{r} during the atmosphere re-entry phase, when the Mach number is time-varying and this reference value must be held constant during the flight phase considered.

However, the procedure used here may be applied for any other reference AoA; this is easily done by re-running the e-LSDP operating point selection algorithm for this new value and re-storing the $\mathscr{H}_\infty$ controller gains. This fact really proves the generality of the approach followed here since the operating point choice algorithm is designed to fine-tune the f-PID controllers for *any* value of $30 \leq \alpha_{\mathrm{r}} \leq 50$ of the ARV's flight envelope.

[16] In the uncertain cases only the combined loop shaping plus robust controller based gain-scheduled controller is considered.

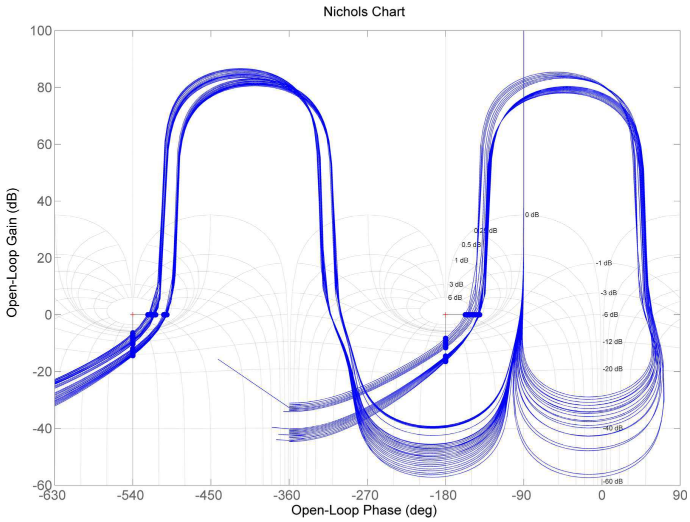


(a) *Uncertain case No.1*

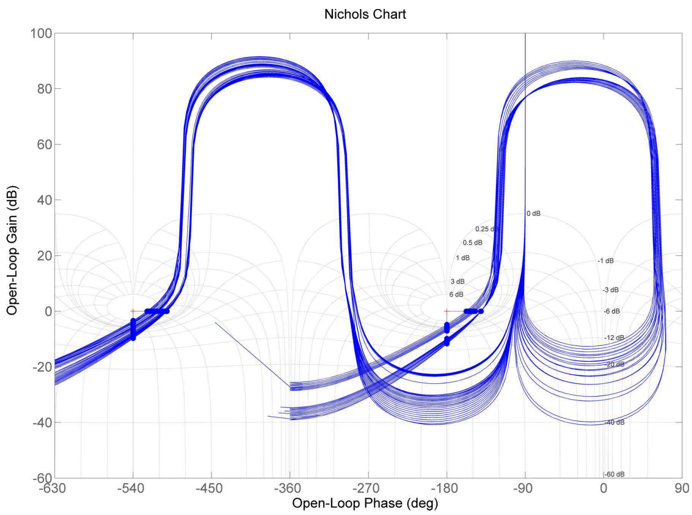


(b) *Uncertain case No.2*

Figure 6.12: Delay margins (nominal & uncertain).



(a) *Uncertain case No.1*



(b) *Uncertain case No.2*

Figure 6.13: Nichols charts (worst cases).

6.3 Gain Blending (missile)

Reminder & motivation

The Reichert benchmark missile autopilot problem has been analyzed in Chapter 5 and two scheduling methods have been applied to obtain a gain-scheduled autopilot. These methods (the controller blending and the state feedback/observer-based interpolation) are of common use in the gain scheduling practice; however they result to a conservative and complicated controller due to the fact that they give no indication on the number of synthesis points needed, due to the high order of the LTI controller and also due to practical issues concerning the methods (e.g. initialization, eigenvalue partitioning etc.). A new method is proposed here based on *gain blending* interpolation and on an extended loop shaping procedure (the e-LSDP) permitting to conceive an interpolation strategy that addresses the aforementioned issues.

In this second part of the chapter this procedure is detailed, following the analysis of the first part concerning the ARV autopilot design. The e-LSDP is adjusted to take into account that the additional points needed must now be added on a plane and not only across a line as with the ARV since the scheduling vector ϱ may follow any trajectory on the missile flight envelope.

As it has been already detailed in Section 5.3, the missile control objectives are performance ones (adequate time constant, overshoot, steady-state error & control rate) as well as robustness ones (robust stability in the face of parametric uncertainties & high frequency open loop magnitude attenuation). The difference between the missile autopilot problem and the ARV one is mainly that here the problem is a tracking and not a regulation one and that stability margin constraints do not appear explicitly; even though it is desired that they are maximized as for any feedback control system.

6.3.1 LTI Controller Synthesis

The e-LSDP (re-visited)

Following the discussion for the ARV autopilot of Section 6.2.1, the LTI controllers for the Reichert benchmark problem are calculated using the e-LSDP procedure that is divided in the three standard steps: *loop shaping*, *operating point algorithm* and finally $\mathscr{H}_\infty$ *controller synthesis*. Briefly these steps involve the following analysis:

Step 1 - Loop Shaping. A linearized model of the missile $G(s)$ is obtained for a small number of synthesis points (9) on the flight envelope, the same as in Chapter 5 (see Table 5.1). The missile linearized dynamics are preceded by the actuator dynamics $G_{\mathrm{a}}(s)$ (acting as a pre-compensator $W_1(s)$) and followed by a specific *outer-inner loop* PI/P controller (acting as a post-compensator $W_2(s)$), in order to shape the open loop frequency response. This procedure corresponds to the LSDP of Section 3.3.2 and provides some basic compensation for these synthesis points; additional synthesis points are added using the algorithm that follows.

Step 2 - Operating Point Algorithm. Similarly to the ARV autopilot problem considered in the first part of this chapter, if the loop shaping controllers only are used to obtain an interpolated gain-scheduled controller, the results are not satisfactory. Once again, the *gap metric* coupled with $\mathscr{H}_\infty$ *loop shaping* theories are used to devise an operating point choice algorithm that will capture the nonlinear dynamics variation; a glimpse of this variation may be observed from the linearized dynamics results presented in Fig. 4.7. As it has been already mentioned, the algorithm chooses points for a family of values for the scheduling vector ϱ, inside all the flight envelope and not only across a line as with the ARV problem ; for additional details on the algorithm see Section 6.3.1.2.

Step 3 - $\mathscr{H}_\infty$ Controller Synthesis. The robust controller synthesis algorithm follows closely the theory presented in the first part and thus the static $\mathscr{H}_\infty$ controller synthesis of Section 3.3. The static controllers are once again designed at the synthesis points deducted from the previous step and then interpolated to provide an additional corrective action over the loop shaping PI/P controllers; for more details see Section 6.3.2.

6.3.1.1 Loop Shaping

As it has been detailed in the previous section, the first step of the e-LSDP is the initial loop shaping performed over a small number of synthesis points, using corresponding transfer functions $G(s) = [G_\eta(s)\ G_q(s)]^T$ issued from the initial missile nonlinear parameter-dependent model $\mathcal{S}_{\text{pd}}$[17].

The control structure chosen is a special type of external/internal (PI/P type) compensation; this strategy has been chosen among others due mainly to its simplicity and ease of tuning as it will be shown in the following analysis. It is evident that in terms of performance and robustness it may be inferior than the full-order $\mathscr{H}_\infty$ controllers considered in the previous chapter; however in terms of implementation and aided by the additional static $\mathscr{H}_\infty$ controllers designed in the next sections, it results to a better gain-scheduled controller.

`Control structure` The control structure used is depicted in Fig. 6.14; an inner simple proportional feedback (P controller) is applied first on the pitch rate q_δ with positive feedback[18] in order to reduce its corresponding open loop gain and augment its gain margin. Then, an external proportional plus integral feedback (PI controller) is added to the tracking error $e_\delta = \eta_\delta - \eta_\text{r}$ in order to achieve good tracking performance[19]. The three gains $K_\text{p}, K_\text{i}, K_q$ are first adjusted in a two step procedure considering first the inner and then the outer loop, using standard frequency domain techniques, and then are optimized using MATLAB® Simulink Control Design and Simulink Response Optimization routines.

[17] For details on the missile trim analysis and linearization refer to Sections 4.1.2-4.1.3.

[18] Positive feedback is used since the pitch rate open loop gain is negative (see Eq. 6.10).

[19] The feedback sign convention for the tracking error is conformable to a standard robust control notation maintaining *positive feedback*.

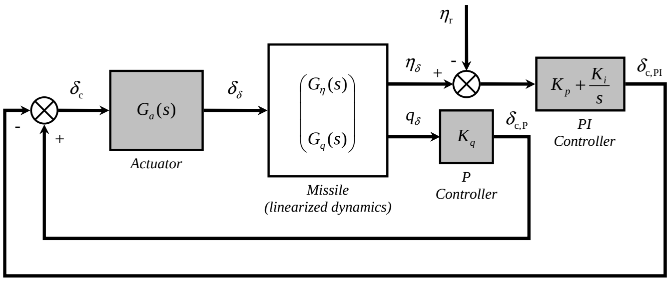


Figure 6.14: PI/P compensation block diagram.

To illustrate the PI/P controller tuning, consider the missile linearized state-space model $\mathcal{S}(\varrho_r)$ for the fifth synthesis point ($\varrho_r = [\eta_r \; M_r]^T = [10.7132 \; 2.25]^T$) corresponding to the middle of the missile flight envelope:

Tuning example

$$\mathcal{S}_{LPV}(\varrho_r) \overset{ss}{:} \begin{cases} \begin{pmatrix} \dot{\alpha}_\delta \\ \dot{q}_\delta \end{pmatrix} = \begin{pmatrix} -0.9945 & 1 \\ -151.03 & 0 \end{pmatrix} \begin{pmatrix} \alpha_\delta \\ q_\delta \end{pmatrix} + \begin{pmatrix} -0.0888 \\ -73.611 \end{pmatrix} \delta_\delta \\ \begin{pmatrix} \eta_\delta \\ q_\delta \end{pmatrix} = \begin{pmatrix} -75.873 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha_\delta \\ q_\delta \end{pmatrix} + \begin{pmatrix} -6.5705 \\ 0 \end{pmatrix} \delta_\delta \end{cases} \tag{6.9}$$

or the corresponding matrix transfer function $G(s)$, with:

$$G = \begin{bmatrix} G_\eta \\ G_q \end{bmatrix} = \frac{\begin{bmatrix} -6.57s^2 + 0.1996s + 4593 \\ -73.61s - 59.8 \end{bmatrix}}{s^2 + 0.9945s + 151}. \tag{6.10}$$

The aforementioned linearized system presents two badly damped but stable poles $p_{1,2} = -0.4972 \pm 12.279$ whereas the transmission zeros are $z_{1,2} = \pm 26.453$ and $z_3 = -0.812$ for the vertical acceleration and pitch rate channels respectively.

The tuning of the pitch rate K_q is done by considering the inner loop that may be seen as the positive *feedback* interconnection of the actuator transfer function $G_a(s)$ with the *series* interconnection of the pitch rate transfer function $G_q(s)$ and the P controller. The input to this loop is the output of the PI controller $\delta_{c,PI}$ with negative sign whereas the output is the filtered total control signal δ_δ; the block diagram of this loop is shown in Figure 6.15.

P controller tuning

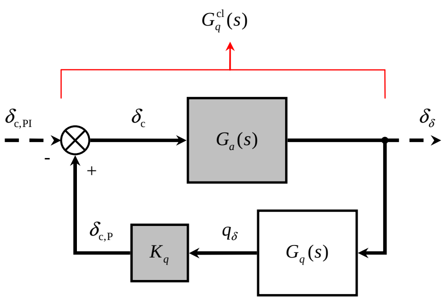


Figure 6.15: Inner loop block diagram.

The closed loop transfer function $G_q^{\mathrm{cl}}(s)$ formed by this interconnection is:

$$G_q^{\mathrm{cl}}(s) = \frac{G_a(s)}{1 - K_q G_q(s) G_a(s)}. \tag{6.11}$$

The pitch rate feedback gain K_q permits to play on the magnitude of the corresponding open loop transfer function $G_q^{\mathrm{ol}}(s)$ given by:

$$G_q^{\mathrm{ol}}(s) = K_q G_q(s) G_a(s). \tag{6.12}$$

This gain is computed by the Evans root locus method; the open loop poles of $G_q^{\mathrm{ol}}(s)$ are comprised by the badly damped missile ones plus the actuator's; given that the latter are very fast, only the former are considered for the tuning. The gain is chosen so that these poles obtain a good damping corresponding to the 10% overshoot constraint P_1 of the missile control objectives; this gives a damping of 0.59. The root locus diagram for synthesis point No. 5 is shown in Fig. 6.17a and the gain computed is $K_q = 0.183$; resulting to a loop gain decrease of $20\log_{10}(K_q) = -14.75$dB (see also the Bode diagram of Fig. 6.17b).

`PI controller tuning`

Once the pitch rate loop is tuned, the vertical acceleration loop is corrected by adjusting the gains $K_{\mathrm{p}}, K_{\mathrm{i}}$ of the PI controller. The open loop $G_\eta^{\mathrm{ol}}(s)$ now is formed by the series interconnection of the PI controller transfer function $G_{\mathrm{PI}}(s)$ (with a negative sign, corresponding to the negative feedback of Fig. 6.14), the adjusted closed loop pitch rate transfer function $G_q^{\mathrm{cl}}(s)$ and the missile vertical acceleration transfer function $G_\eta(s)$:

$$G_\eta^{\mathrm{ol}}(s) = -G_\eta(s) G_q^{\mathrm{cl}}(s) G_{\mathrm{PI}}(s). \tag{6.13}$$

The closed loop transfer function $G_\eta^{\mathrm{cl}}(s)$ is obtained by the unitary positive feedback interconnection of the open loop transfer function $G_\eta^{\mathrm{ol}}(s)$[20]:

$$G_\eta^{\mathrm{cl}}(s) = \frac{G_\eta(s) G_q^{\mathrm{cl}}(s) G_{\mathrm{PI}}(s)}{1 + G_\eta(s) G_q^{\mathrm{cl}}(s) G_{\mathrm{PI}}(s)}. \tag{6.14}$$

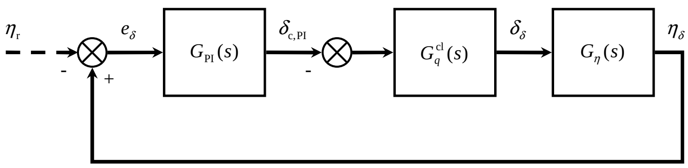
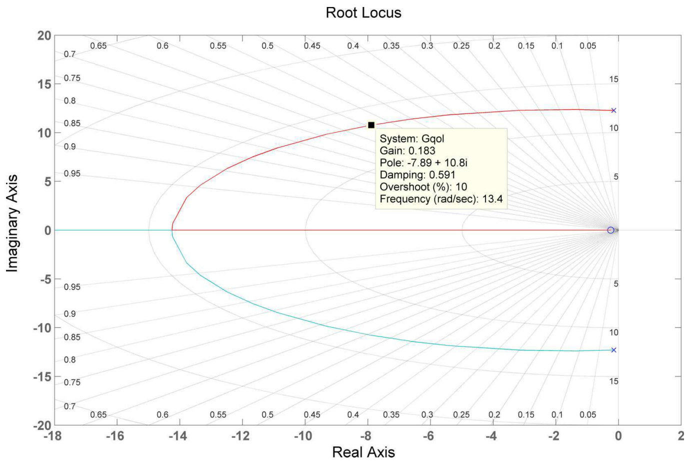


Figure 6.16: Outer loop block diagram.

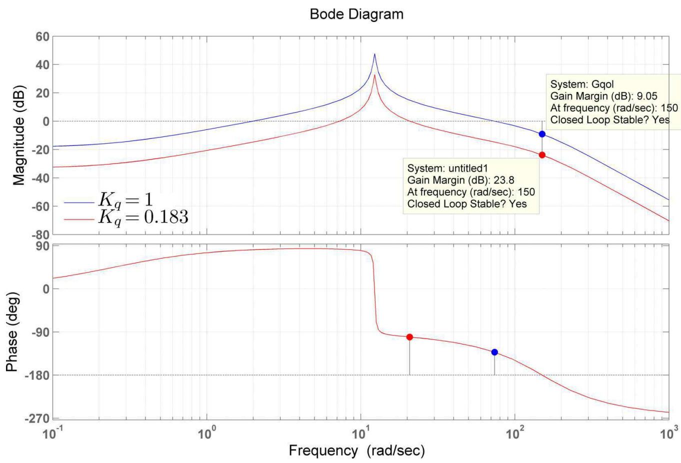
The PI controller transfer function is given by the following equivalent formulations:

$$G_{\mathrm{PI}}(s) = K_{\mathrm{p}} + \frac{K_{\mathrm{i}}}{s} = K_{\mathrm{i}} \frac{1 + \frac{K_{\mathrm{p}}}{K_{\mathrm{i}}} s}{s}. \tag{6.15}$$

[20]Note that the reference signal η_{r} is applied using a negative sign.



(a) *Evans root locus (pitch rate loop)*



(b) *Bode diagram (pitch rate loop)*

Figure 6.17: Pitch rate pre-tuning.

`Pre-tuning` The PI controller is tuned in two phases: first only the *integral action* is added and the integral gain K_i is adjusted so as obtain a satisfying step response performance[21]. By augmenting the gain, the response becomes more rapid (the time constant τ is reduced) but also more oscillatory. The integral gain is chosen trying to minimize the time constant while respecting the maximum overshoot constraint. The resulting gain is $K_\mathrm{i} = 0.15314$, giving a time constant $\tau = 254.6ms$ (less than 350ms imposed by the performance objectives) and a settling time $t_\mathrm{s} = 552ms$[22] with the overshoot being $M_\mathrm{p} = 10\%$ (equal to the constraints). The results using only the integral action are good but not all the control bandwidth is used. Indeed the maximum control rate $\dot{\delta}_\delta$ is approximately 9deg/s, about three times less than the limit of 25deg/s of the performance objective P_2 of Section 5.3. The *proportional action* now of the PI controller will add a zero on the open loop transfer function $G_\eta^{\mathrm{ol}}(s)$ of Eq. 6.13 permitting a more rapid step response.

`Final tuning` Based on the pre-tuning of the integral gain, both the *integral* and *proportional* gains are optimized using MATLAB® Simulink Control Design and Simulink Response Optimization; the strategy used is to try and minimize the vertical acceleration step response time constant τ while not violating the overshoot and control rate constraints. The gains obtained are $K_\mathrm{i} = 0.18593$ (re-tuned) and $K_\mathrm{p} = 0.0052173$[23] and the two step responses (integral and re-tuned integral plus proportional) are shown in Fig. 6.18a whereas the two corresponding corrected open loop Bode diagrams are visualized in Fig. 6.18b.

The faster step response with PI controller (red line) with respect to the I controller (blue line) is evident (see Fig. 6.18a); this may in fact be explained by the increased open loop bandwidth. In the second case the gain crossover frequency ω_gc is 4.87rad/s whereas in the first case 6.19rad/s (27% bigger) (see Fig. 6.18b). The time constant τ is in the second case 254.6ms (as mentioned before) whereas in the first case 203ms (25.5% faster). Despite the system being significantly faster, the gain margin is also slightly ameliorated (8.71dB compared to 7.84dB initially) whereas the phase margin is similar since the damping in both cases is the same.

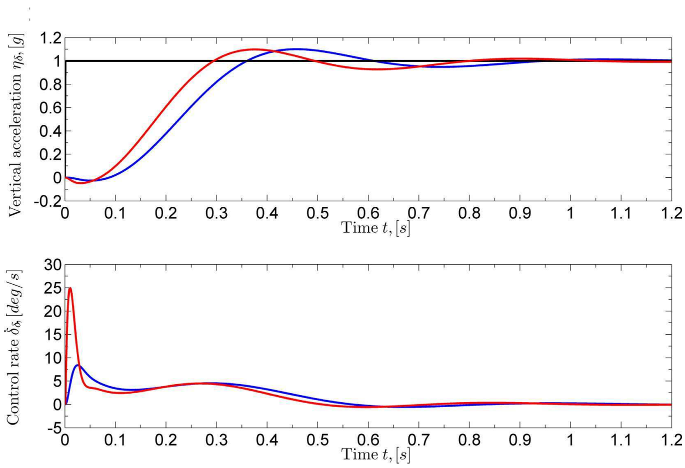
The closed loop dynamics are of fifth order; the poles (system, controller, actuator) and zeros (total outer feedback loop) are the following (the non-minimum phase step response of Fig. 6.18a may now be justified by the positive closed loop I/O zero):

$$\begin{aligned} \text{poles}: &\quad -3.84 \pm 11.6j, -6.8, -97.9 \pm 96.5j \\ \text{zeros}: &\quad \pm 26.4, -35.6. \end{aligned}$$

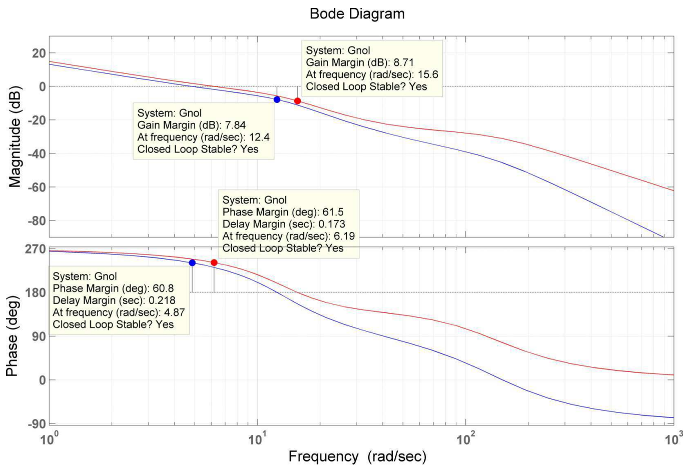
[21] This adjustment is done using MATLAB® SISOTool, permitting to observe in real-time the influence of the integral gain on the step response.

[22] Note that no settling time constraints are imposed; trying to minimize the time constant does not always mean that the settling time is also minimized. In fact this results to a greater overshoot and thus the settling time is finally augmented.

[23] This corresponds to a controller zero added at $s = -K_\mathrm{i}/K_\mathrm{p} \simeq -35.64$.



(a) *Step responses*



(b) *Open loop Bode diagrams*

Figure 6.18: PI controller tuning comparison results (blue: integral action, red: integral plus proportional actions).

Table 6.4: PI/P controller tuning results

Points	K_q	K_p	$K_\mathrm{i}^{(\mathrm{i})}$	
1	0.214	0.0056002	0.11111	0.12988
2	0.307	0.0042098	0.28452	0.30545
3	0.351	0.0031131	0.37043	0.38815
4	0.144	0.0057412	0.07762	0.10476
5	0.183	0.0052173	0.15314	0.18593
6	0.213	0.0046419	0.21069	0.24660
7	0.106	0.0057885	0.05839	0.08913
8	0.117	0.0056409	0.08030	0.11584
9	0.141	0.0053212	0.12486	0.16721

(i) The first column gives the values for the pre-tuning whereas the second the final values after the optimization.

The same procedure is applied iteratively for all synthesis points (see Section 5.4) and the results for the gains are shown in Table 6.4. The pitch rate and integral channel gains increase with the vertical acceleration for a given Mach number whereas the proportional channel gain decreases.

`Tuning results & discussion`

The results from the PI/P controller shaping are shown in Table 6.5. The second line shows the time constant achieved; the missile's performance ameliorates with increasing Mach and vertical acceleration. The synthesis is done trying not to violate $M_\mathrm{p} = 10\%$ and $|\dot{\delta}_\delta|_{\max} = 25\mathrm{deg/s}$; the rapidity constraint is not achieved only at the first synthesis point where $\tau = 440\mathrm{ms}(> 350\mathrm{ms})$. Comparing with the $\mathscr{H}_\infty$ controllers of Section 5.4.1 the results are really good, taking also into account the controller order considered in both cases.

Table 6.5: PI/P controller tuning results(i)

Points	1	2	3	4	5	6	7	8	9
τ(ms)	440	321	286	240	203	181	146	126	141
GM(dB)	9.62	8.15	7.78	10.5	8.71	8.00	10.3	9.44	8.25
PM(deg)	61.1	61.2	61.2	60.8	61.5	61.2	58.8	60.5	61.7
DM(ms)	370	276	248	196	173	155	109	112	106
ω_gc(rad/s)	2.89	3.87	4.31	5.44	6.19	6.88	9.39	9.39	10.1
GM(dB)	26.3	25.0	24.6	21.2	20.4	20.0	17.2	17.0	16.4
PM(deg)	78.5	79.9	79.9	67.2	69.6	70.0	52.0	54.3	55.8
DM(ms)	144	103	90.1	81.7	66.7	57.7	47.0	45.0	38.5
ω_gc(rad/s)	9.51	13.5	15.5	14.3	18.2	21.2	19.3	21.1	25.3
d_att(dB)	43.5	41.9	41.2	38.2	37.7	37.3	34.1	34.0	33.8

(i) Lines 2 $\rightarrow$ 5 give the frequency results with the outer loop opened, whereas lines 6 $\rightarrow$ 10 when the actuator loop is opened.

All nine step responses of the missile vertical acceleration and control rate are visualized in Figs. 6.19a-6.19b; the red lines correspond to the faster re-tuned PI controller and the blue ones to the I controller only. The uniformity of the curves is apparent with the system becoming faster through points $1 \rightarrow 9$; in addition, each response demonstrates the same overshoot and control rate (for the PI case only of course).

In Figs. 6.20a-6.20b the closed loop poles (missile+controller+actuator) and η-channel transmission zeros of each of the nine corrected systems using the final PI/P controller are presented; the first figure shows the big picture whereas the second zooms on the missile's poles only. The rapidity of the system's dominant poles (namely the missile ones) clearly increases through the synthesis points $1 \rightarrow 9$ (see Fig. 6.20b). These poles exhibit a rather constant damping (between 0.283 and 0.333) and an increasing natural frequency (between 5.59rad/s and 19.2rad/s).

The open loop frequency results with the loop opened *before the PI controller* (see Fig. 6.16) are given in lines $3 \rightarrow 6$ of Table 6.5 and the corresponding Bode and Nichols diagrams in Figs. 6.21a-6.21b. The results include the gain, phase and delay margins (GM, PM, DM) and gain crossover frequencies for all open loop transfer functions $G_\eta^{\mathrm{ol}}(s)$ of Eq. 6.13. Again the PM is almost constant (around 60°) since all time responses exhibit the same overshoot whereas the GM seems adequate ranging from 8 to 10.5dB. The DM follows the same pattern as the time constant one and corresponds from 75% to 87% of the time constant.

The open loop frequency results with the loop now opened *before the actuator* (see Fig. 6.14) are also given in lines $7 \rightarrow 11$ of Table 6.5 and the corresponding Bode and Nichols diagrams in Figs. 6.21a-6.21b. Suppose the total PI/P controller matrix transfer function from the missile outputs η_δ, q_δ to the control input δ_{c} is denoted by:

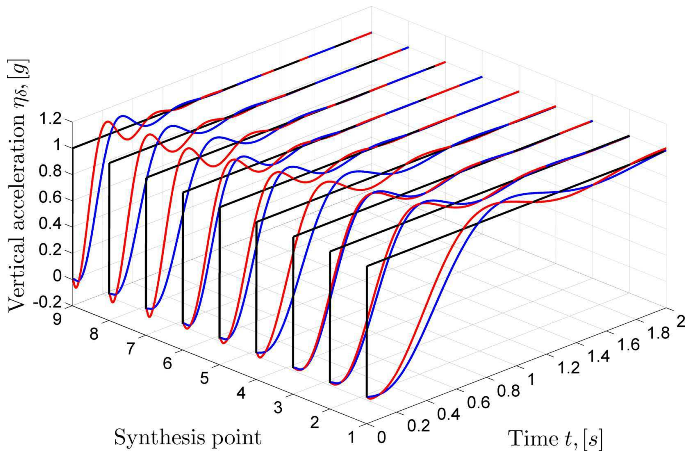
$$G_{\mathrm{c}}(s) = \begin{bmatrix} -G_{\mathrm{PI}}(s) & K_q \end{bmatrix} = \begin{bmatrix} -K_{\mathrm{p}} - \dfrac{K_{\mathrm{i}}}{s} & K_q \end{bmatrix}. \tag{6.16}$$

Then the aforementioned open loop (stabilized always with a positive feedback) is a SISO transfer function G_a^{ol} comprised by the series interconnection of the actuator, the missile linearized dynamics and the controller[24]:

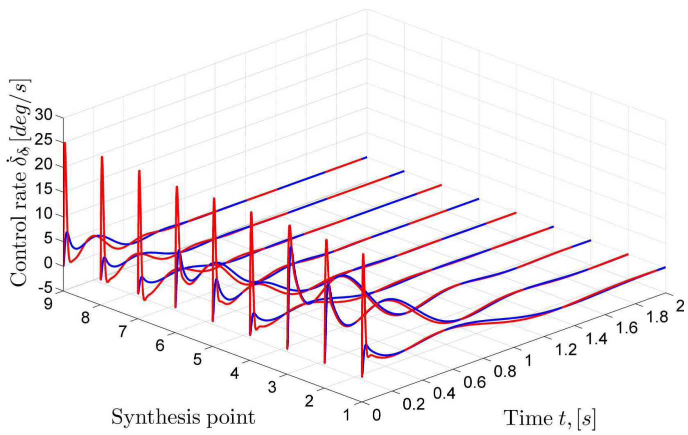
$$G_a^{\mathrm{ol}} = G_{\mathrm{c}}(s)G(s)G_a(s). \tag{6.17}$$

Besides the GM, PM, DM and gain crossover frequency, the open loop magnitude attenuation d_{att} (robustness constraint R_2 of Section 5.3) is given; the latter is maintained for all synthesis points ($d_{\mathrm{att}} > 30db$). The GM and DM decreases for synthesis points $1 \rightarrow 9$ since the system becomes more rapid with the latter being approximately one third of the system's time constant; finally, the PM is also very good.

[24]Note that a *positive feedback* is always assumed for the calculation of the closed loop transfer function.

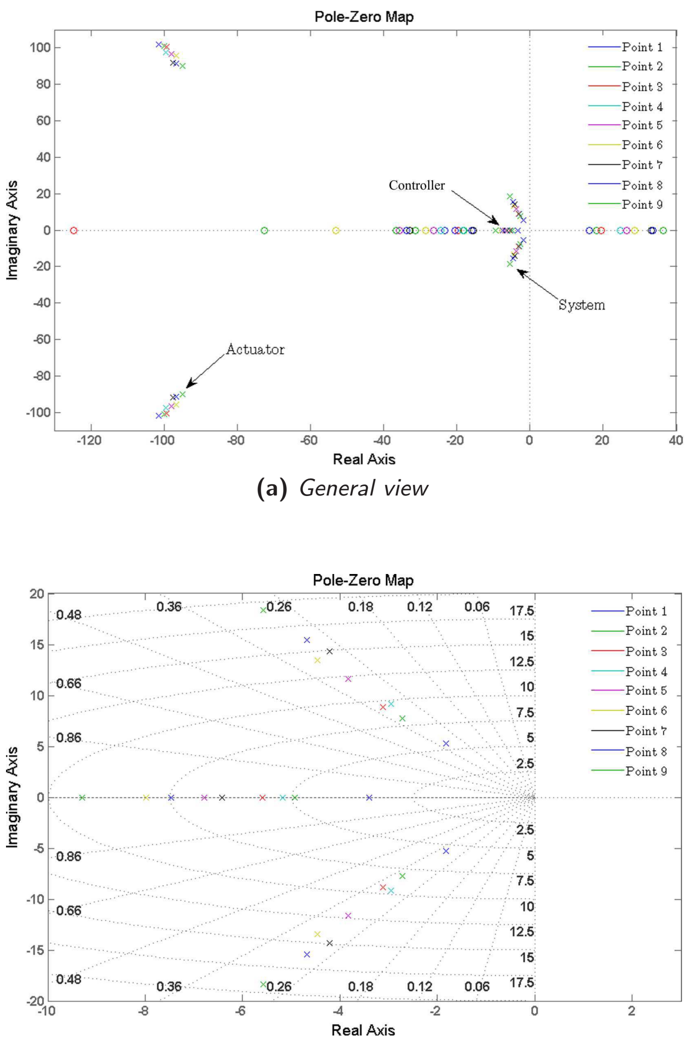


(a) *Step responses*



(b) *Open loop Bode diagrams*

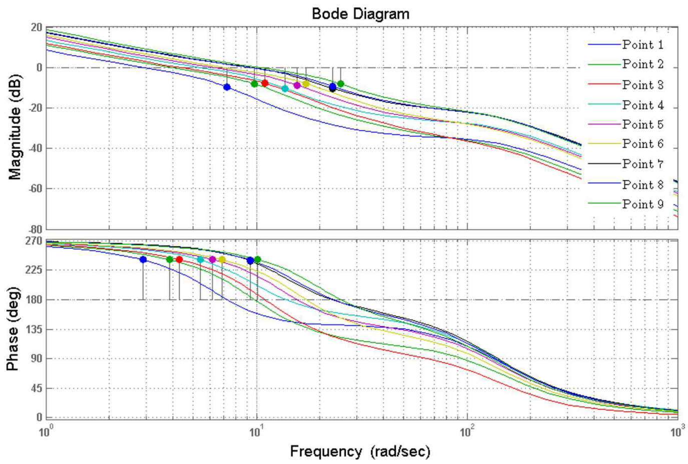
Figure 6.19: Total controller tuning results (blue: integral action, red: integral plus proportional actions).



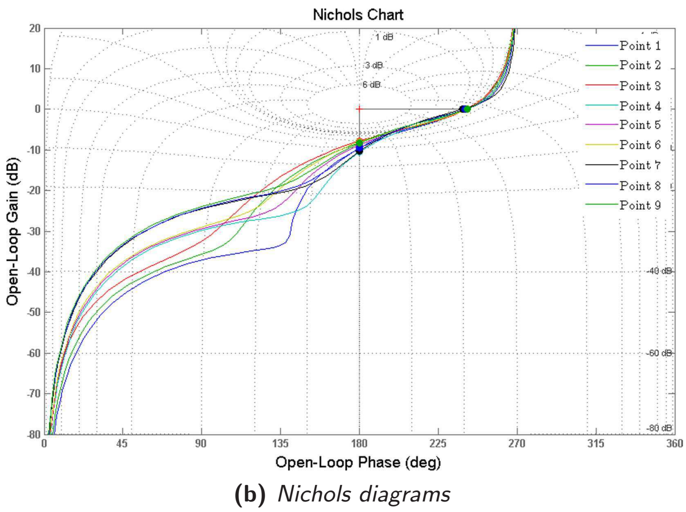
(a) *General view*

(b) *Zoomed view*

Figure 6.20: Closed loops pole-zero map.

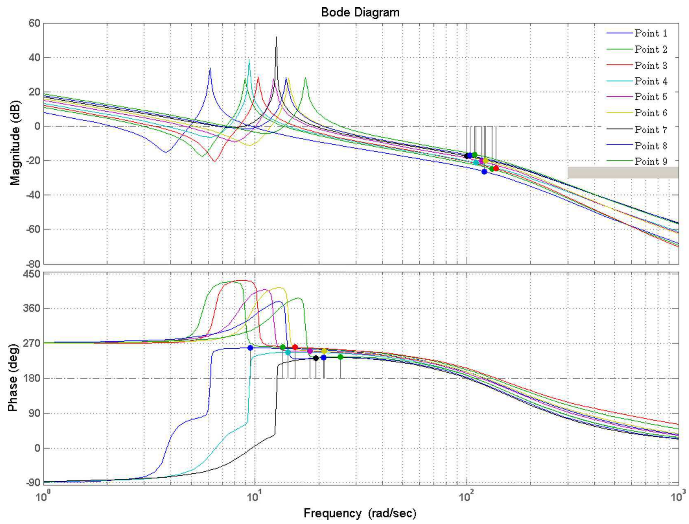


(a) *Bode diagrams*

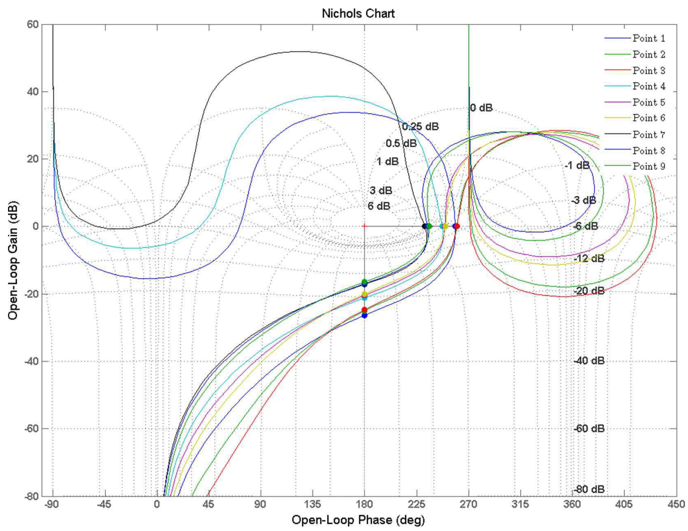


(b) *Nichols diagrams*

Figure 6.21: Open loops analysis (outer feedback loop).



(a) *Bode diagrams*



(b) *Nichols diagrams*

Figure 6.22: Open loops analysis (actuator loop).

6.3.1.2 Operating Point Algorithm

Based on the loop shaping performed in the previous section, an additional set of synthesis points for the $\mathscr{H}_\infty$ controllers will be computed, similarly to the analysis of Section 6.2.1.2 concerning the ARV benchmark model. The major difference between the two is that the missile's nonlinear dynamics will be treated in *two dimensions* and not only across a line as with the ARV.

Discussion The algorithm proposed in this section is essentially the same as the one in Section 6.2.1.2, with the difference that here the linear search is performed for m *sets* of equidistant values $\Sigma_\eta^m = [\eta_1, \ldots, \eta_k^m]$ for the vertical acceleration. The index 'm' defines the value M^m of the Mach number corresponding to each set, since the algorithm is performed iteratively for a gridding $\Sigma_M = [M^1, \ldots, M^m]$ over the Mach number. Given the fact that for each value of Σ_M, the corresponding final value η_k^m is different because of the trapezoidal form of the flight envelope, the size of each set Σ_η^m will be different for each m. The algorithm used here proposes only three values for Σ_M; the same used for the loop shaping of the previous section (i.e. 1.5,2.25,3) and thus the additional synthesis points will be added across these three (constant Mach) lines[25]. Once again, the operating point algorithm is divided into three distinct steps:

Step 1 - Initialization. Choose an equidistant gridding (e.g. $\Sigma_M = [1.5, 2.25, 3]$) over the Mach range and then a second equidistant gridding Σ_η^m for all $m = 1, \ldots 3$ over the vertical acceleration, thus creating a planar gridding of candidate synthesis points. Take then as the initial synthesis point P^j the one corresponding to $\eta = 0$, for each value of Σ_M[26].

Step 2 - Interpolated Loop Shaping. For the initial operating point P^j (for a corresponding scheduling vector value $\varrho = \varrho^j$ with $m = 1$), compute the open loop shaped plant $G_\mathrm{s}^j = G_\mathrm{c}^j G^j G_\mathrm{a}$ (see Eq. 6.17). The loop shaping PI/P controller gains at this point are computed using linear interpolation of the nine initial synthesis points and the corresponding four trapezoidal scheduling regions (see Fig. 5.2). The missile model is simply the linearization of the initial parameter-dependent model for $\varrho = \varrho^j$, and the open loop shaped plant is finally completed by adding the actuator model. For this shaped plant G_s^j an $\mathscr{H}_\infty$ static loop shaping controller is computed following Theorem 3.5 and the discussion of the next section; in addition, the corresponding robustness margin ϵ^j is obtained.

Step 3 - Line Search or Reset. Performing a line search for subsequent values of η belonging to Σ_η^m, successive shaped plants G_s^f are computed, until the gap $\delta_\mathrm{g}(G_\mathrm{s}^j, G_\mathrm{s}^f)$ between the nominal initial plant and the successive one is greater or equal than the robustness margin ϵ^j.

---

[25] Of course more points could be used; however three seem to be adequate in this case.

[26] Note that the right extremal values of the flight envelope are computed using Eq. 4.22; for more details concerning the missile operating domain refer to Section 4.1.2.3.

If this is the case, then a new operating point P^j is chosen and the algorithm jumps back to Step 2, except for the case when the end of the flight envelope is reached. In this case, even if $\delta_{\mathrm{g}}(G_{\mathrm{s}}^j, G_{\mathrm{s}}^f) < \epsilon^j$, the final point is selected and the procedure jumps to Step 2 where a new set value Σ_M is chosen ($m = m + 1$) and the algorithm continues for all m.

`Algorithm results`

Using this algorithm, totally twelve synthesis points are computed. The $\mathscr{H}_\infty$ controller treats the internal/external loops outputs and $K_\infty = [K_{\mathrm{PI},\infty}\ K_{q,\infty}] \in \mathbb{R}^{1\times 2}$ (for more details see the next section). The coordinates of the synthesis points, the corresponding gains and robustness margins are shown in Table 6.6.

Table 6.6: Robust controller gains

Points	$[\eta, M]$	$K_{\mathrm{PI},\infty}$	$K_{q,\infty}$	e
1	[0 1.5]	0.594	1.098	0.369
2	[4.20 1.5]	0.626	1.099	0.378
3	[9.7969 1.5]	0.657	1.129	0.381
4	[0 2.25]	0.645	1.131	0.391
5	[4.20 2.25]	0.643	1.093	0.387
6	[12.91 2.25]	0.720	1.152	0.396
7	[21.43 2.25]	0.745	1.155	0.389
8	[0 3]	0.569	0.960	0.292
9	[4.20 3]	0.544	0.874	0.369
10	[11.11 3]	0.628	1.006	0.399
11	[21.01 3]	0.770	1.167	0.395
12	[33.0559 3]	0.738	1.133	0.383

The gap evolution is shown in Fig. 6.23 for $M = 1.5$ (for the rest of the values for M the profile is similar). For this Mach value the operating point algorithm finds three points (see Table 6.6). Take for example the first one ($\eta = 0$); the robustness margin ϵ^j and the maximum robustness margin $\epsilon^j_{\max}$ (corresponding to the full order robust controller) are shown with black points. The red points show the gap between subsequent *candidate* synthesis points and the initial one; once the gap becomes greater than ϵ^j, a new synthesis point is selected.

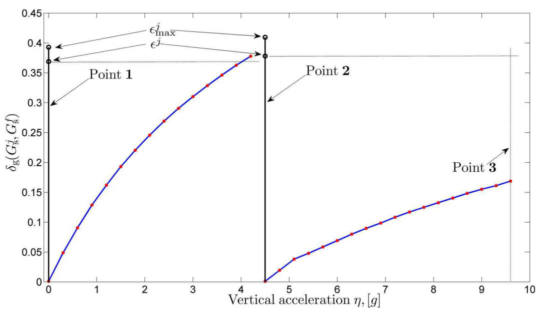


Figure 6.23: Gap evolution for $M = 1.5$.

6.3.1.3 $\mathscr{H}_\infty$ Controller Synthesis

The analysis concerning the computation of the robust $\mathscr{H}_\infty$ controllers is essentially the same as in Section 6.2.1.3 with the corresponding problem being given by Eq. 6.3, and will not be repeated.

`Control structure`

The robust controller structure uses a slightly different form here, comparing to the one concerning the ARV benchmark, where all three channels of the PID controller were treated (see Fig. 6.7). The loop now is opened using *directly* the output of the PI controller $\delta_{\mathrm{c,PI}}$ (instead of e.g. separately the proportional & integral channels) and the output of the P controller $\delta_{\mathrm{c,P}}$. Two static gains $K_{\mathrm{PI},\infty}, K_{q,\infty}$ are thus computed treating each controller output and their outputs are summed in order to provide the final control signal δ_c (see Fig. 6.24).

The step responses for the PI/P shaped plant (blue) and the PI/P shaped plant plus the robust controller (red) for the synthesis point (No. 12) are shown in Fig. 6.25 (similar behavior holds for all points); with the output now being more damped. The PI/P controllers could have very well been adjusted initially to provide such damping; however it should not be forgotten that the robust controllers, coupled with the gap metric theory, additionally capture the plant's nonlinear dynamics variation using the algorithm of the previous section.

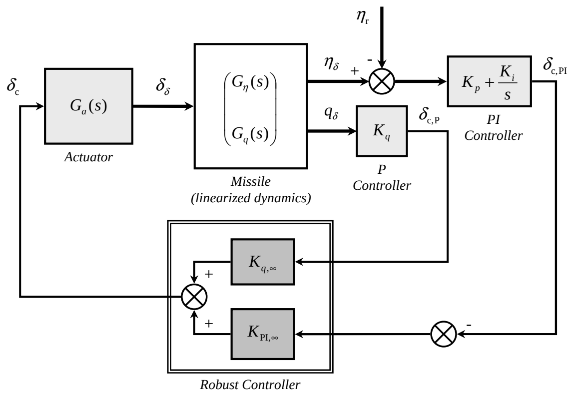


Figure 6.24: Robust controller synthesis block diagram.

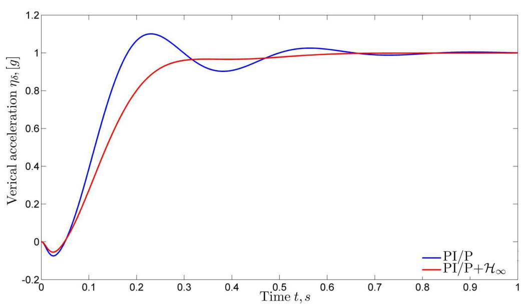


Figure 6.25: Step response comparison.

6.3.2 Gain Interpolation

The gain-scheduled controller detailed in the next section uses gain interpolation in order to update its parameters as a function of the scheduling vector $\varrho = [\eta \; M]^T$. Gain interpolation is also used by the operating point selection algorithm of Section 6.3.1.2 in order to provide PI/P controller gain values for the loop shaping needed at any point on the missile's flight envelope.

The PI/P controller gains $K_q, K_\mathrm{p}, K_\mathrm{i}$ are designed using the analysis of Section 6.3.1.1 at nine synthesis points (see Table 6.4) forming four *trapezoidal* scheduling regions $\Gamma^1, \Gamma^2, \Gamma^3$ and Γ^4 (the same used in Chapter 5). The robust controller gains $K_{\mathrm{PI},\infty}, K_{q,\infty}$ are designed at twelve synthesis points (see Table 6.6) forming twelve *triangular* scheduling regions.

`PI/P gain interpolation`

Concerning the PI/P controller gains, the trapezoidal interpolation is performed like the one used for the controller blending method in Section 5.4.2.1 (see especially Fig. 5.9). Consider for example the gain K_q and the first scheduling region Γ^1; define as $K_q^\mathrm{ll}, K_q^\mathrm{lr}, K_q^\mathrm{ur}, K_q^\mathrm{ul}$ the gain values at the lower-left, lower-right, upper-right and upper-left corners of the scheduling region. The normalized quantities a_1, a_2 (with $0 \le a_i \le 1$) give the relative distance of the current interpolated value $K_q(t)$ from the left (lower & upper) points and from the lower (left & right) points respectively. These distance need some trigonometry to be computed and the calculations will not be given here; it must however be stressed out that once the normalized quantities are found, the interpolated value for the controller gain is simply obtained by[27]:

$$K_q(t) = \big[1 - a_1(t)\big] K_q^\mathrm{l}(t) + a_1(t) K_q^\mathrm{u}(t) \tag{6.18}$$

where:

$$K_q^\mathrm{l}(t) = \big[1 - a_2(t)\big] K_q^\mathrm{ll} + a_2(t) K_q^\mathrm{lr} \tag{6.19}$$

$$K_q^\mathrm{u}(t) = \big[1 - a_2(t)\big] K_q^\mathrm{ul} + a_2(t) K_q^\mathrm{ur}. \tag{6.20}$$

To illustrate the interpolation method used here, a spiral scheduling vector trajectory is shown in Fig. 6.26 (red points).

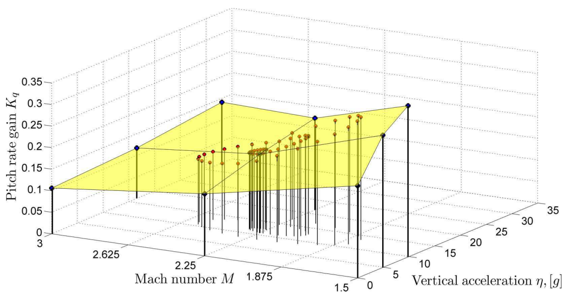


Figure 6.26: Pitch rate gain interpolation.

[27] Dependence on the scheduling vector is omitted; only time dependence is used for simplicity.

$\mathscr{H}_\infty$ controller interp/tion

The robust controller gains $K_{\mathrm{PI},\infty}, K_{q,\infty}$ now are interpolated using triangular scheduling regions as a result of Delaunay triangulation. In order to ensure a more correct triangulation of the missile flight envelope, the scheduling regions are obtained by considering only the portion of the flight envelope that corresponds to two subsequent values of the Mach number gridding.

For example, if the current value of the scheduling vector for the Mach number is $M = 1.6$ and given that the gridding values are $1.5, 2.25, 3$, only the portion of the flight envelope corresponding to $1.5 \leq M \leq 2.25$ is triangulated. In this case the triangular scheduling regions are illustrated in Fig. 6.27.

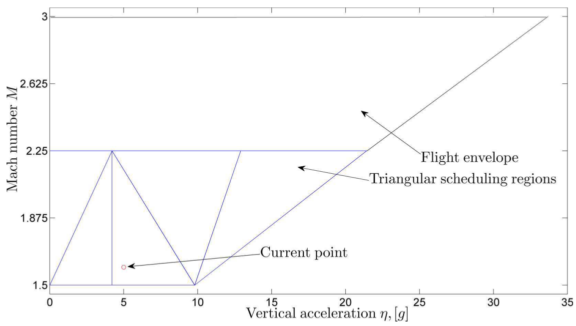
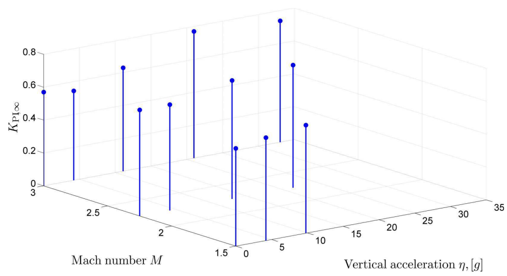
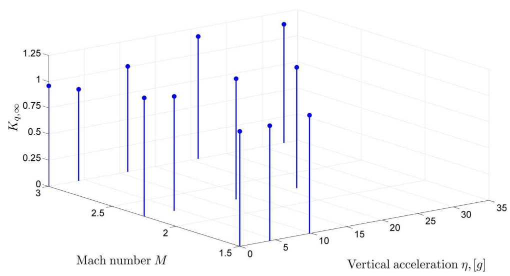


Figure 6.27: Flight envelope triangulation.

Finally in Figs. 6.28a-6.28b, the robust controller gains $K_{\mathrm{PI},\infty}, K_{q,\infty}$ are shown in 3D (see also Table 6.6).



(a) *PI-channel gain*



(b) *Pitch rate-channel gain*

Figure 6.28: Robust controller gains.

6.3.3 Controller Implementation & Validation

The missile gain-scheduled controller will be detailed in this section. The first subsection presents the controller structure whereas the second one presents the main results concerning the nominal and uncertain behavior of the missile under the specific controller; the chapter ends with a short discussion.

6.3.3.1 Nonlinear Gain-scheduled Controller

The nonlinear gain-scheduled controller block diagram is shown in Fig. 6.29. The grey blocks represent the nonlinear missile pitch-axis dynamics, the actuator dynamics and the Mach dynamics (see Eqs. 4.1-4.9, Eq. 4.11 and Eq. 4.10 respectively). The total control signal δ is the sum of the trim control δ_{r} and the closed loop control signal δ_{δ}.

`Trim & feed/ward control`

The trim control $\delta_{\mathrm{r}} = \delta(\varrho_{\mathrm{r}})$ is computed using the analysis of Section 4.1.2.2 as a function of the scheduling vector $\varrho = [\eta_{\mathrm{r}}\ M_{\mathrm{r}}]^T$. A feedforward controller is added before the trim control block in order to 'schedule on a slow variable' as it is often the case in gain scheduling. This controller is a simple first-order filter acting on the reference signal η_{r} dampening the system's output $\eta(t)$; as a result, the trim control is computed using the filtered reference signal $\eta_{\mathrm{r},f}$.

`Fast loop`

The closed loop control signal δ_{δ} is the sum of the outputs of the PI and the pitch rate controllers (see in Fig. 6.14), and scaled by the robust $\mathscr{H}_{\infty}$ controller. The inputs to these controllers are the 'error signals' η_{δ} and q_{δ}; the former is computed by subtracting the missile output by the reference output signal whereas the latter is also computed in a similar way[28]. This closed loop is the *fast* one since it stabilizes the system and ensures trajectory following.

`Slow loop`

The additional scheduling loop is the *slow* one, and uses the scheduling vector ϱ and the 'Interpolation Mechanism' block (in red) in order to update all five gains of the inner control loop as a function of the system's operating conditions (see Section 6.3.2). A similar feedforward controller as the trim control one (but with a different time constant) is used to smooth the gain variation[29]. The gain-scheduled controller is thus governed by the following equations[30]:

`Global controller`

$$\begin{aligned} \dot{x}_{\mathrm{c}} &= \mathbf{A}_{\mathrm{c}}(\varrho)\big[x_{\mathrm{c}} - x_{\mathrm{c}}(\varrho)\big] + \mathbf{B}_{\mathrm{c}}(\varrho)\big[y - y_{\mathrm{r}}(\varrho)\big] \\ \delta &= \mathbf{C}_{\mathrm{c}}(\varrho)\big[x_{\mathrm{c}} - x_{\mathrm{c}}(\varrho)\big] + \mathbf{D}_{\mathrm{c}}(\varrho)\big[y - y_{\mathrm{r}}(\varrho)\big] + \delta(\varrho_{\mathrm{r}}) \end{aligned} \tag{6.21}$$

with $y = [\eta \quad q]^T, \dot{x}_{\mathrm{c}} = \eta_{\delta}$ and:

$$\mathbf{A}_{\mathrm{c}}(\varrho) = 0 \tag{6.22}$$

$$\mathbf{B}_{\mathrm{c}}(\varrho) = [1 \quad 0] \tag{6.23}$$

$$\mathbf{C}_{\mathrm{c}}(\varrho) = K_{\mathrm{PI},\infty}(\varrho)K_{\mathrm{i}}(\varrho) \tag{6.24}$$

$$\mathbf{D}_{\mathrm{c}}(\varrho) = [K_{\mathrm{PI},\infty}(\varrho)K_{\mathrm{p}}(\varrho) \quad K_{q,\infty}(\varrho)K_{q}(\varrho)]. \tag{6.25}$$

[28] To compute q_r, an additional 'Trim Values' block is used (see Section 4.1.2.2 & Eq. 4.20).
[29] For details on tuning both feedforward controllers, see [140], §VI.A.
[30] The feedforward filters are not considered in the equations for simplicity.

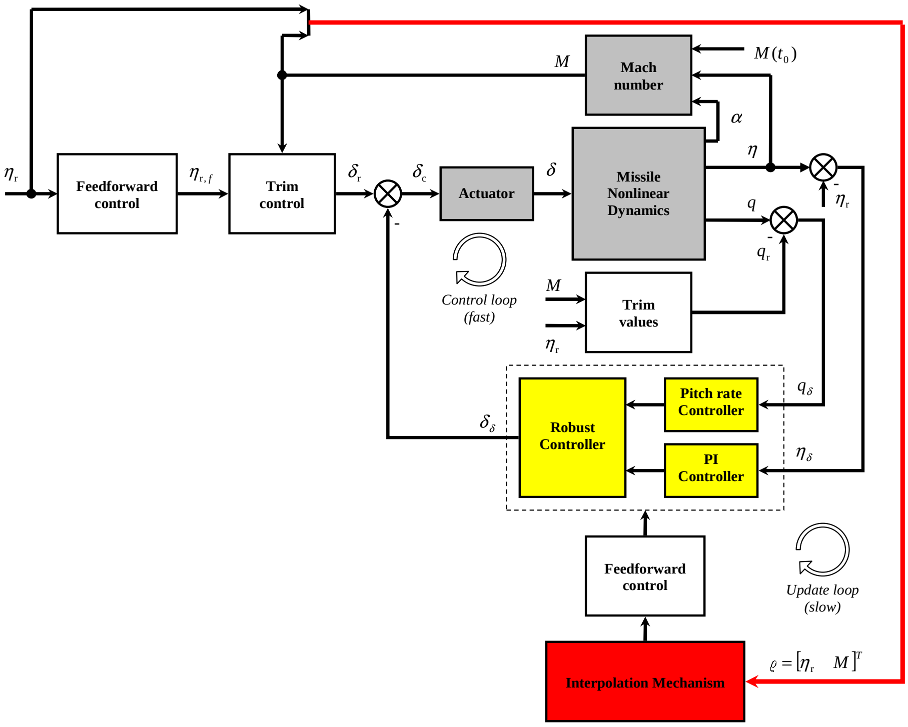


Figure 6.29: Gain-scheduled controller block diagram.

6.3.3.2 Simulation Results

The simulation results presented in this section are obtained under the same conditions as the ones in Chapter 5 (Mach number trajectory, reference signal scenario); for more information refer to Figs. 5.11a-5.11c.

The first figure presented here (see Fig. 6.30) illustrates the missile's output vertical acceleration η as a function of time, when applying two gain-scheduled controllers: the PI/P scheduled controller using the nine LTI controllers of Section 6.3.1.1 (blue line), and the full PI/P plus robust $\mathscr{H}_\infty$ scheduled controller detailed in Sections 6.3.1.2-6.3.1.3 (red line). The tracking reference η_{r} and the filtered tracking reference $\eta_{\mathrm{r},f}$ are also shown (black lines). Nominal case results

When observing the responses, the robust controller is clearly superior to the simple PI/P controller, providing adequate damping to the system's output while retaining excellent time constants and steady-state errors and satisfying the missile control objectives of Section 5.3[31]. The response characteristics for the robust gain-scheduled controller are also presented in detail in Table 6.7 for all four reference operating points.

Table 6.7: Controller nonlinear performance

Points	No. 1	No. 2	No. 3	No. 4	**Limit**
τ (ms)	260	283	250	319	**350**
M_{p} (%)	3.88	6.25	0.00	5.33	**10**
e_{ss} (%)	0.19	0.00	*1.71*	*1.05*	**1**
$\lvert\dot{\delta}\rvert$ (deg/s)$^{(\mathrm{ii})}$	4.44	6.03	6.64	7.20	**25**

$^{(\mathrm{i})}$ The violated constraints are shown in italics.
$^{(\mathrm{ii})}$ The control rate is normalized by the amplitude of the reference signal η_{r}.

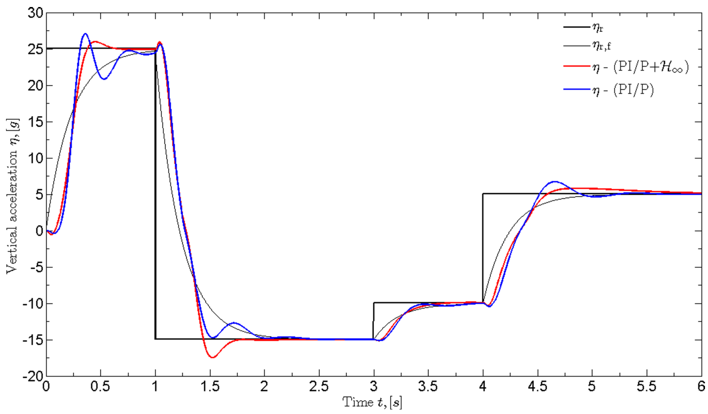
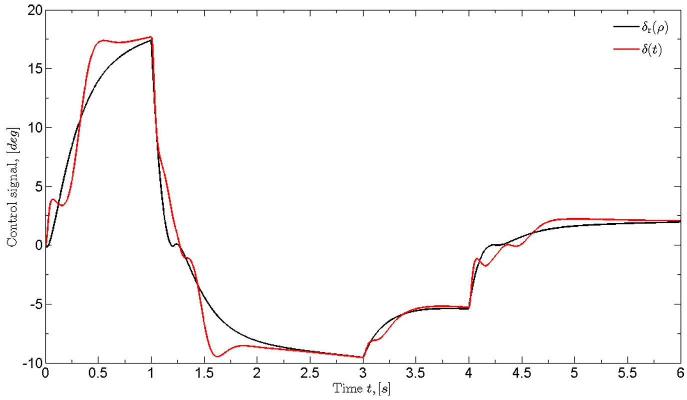
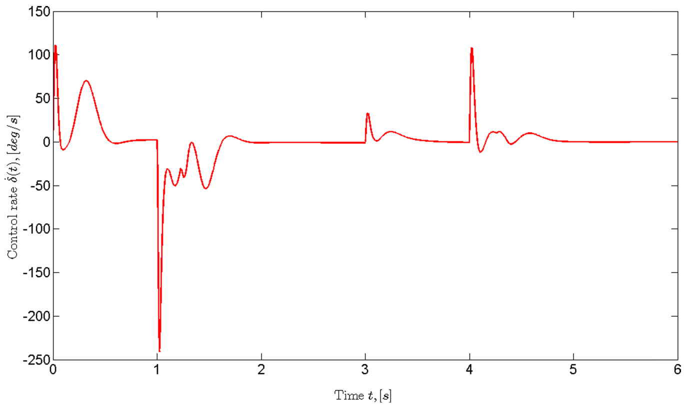


Figure 6.30: Vertical acceleration (comparison).

[31]There is a small violation on the steady-state errors for points No. 3 & 4.



(a) *Control signal*



(b) *Control rate signal*

Figure 6.31: Controller outputs.

The two figures in the previous page present the gain-scheduled controller output characteristics. In Fig. 6.31a, the total control signal $\delta(t)$[32] after the actuator is shown (in red)[33] along with the trim control signal (in blue). It may be observed that the control signal does not present any discontinuities or transients as with the *controller blending* method.

An interesting phenomenon may be also observed from Fig. 6.31a concerning moving equilibria. Take for example the transient response for the first reference point $0 \rightarrow 25$g; the output $\eta(t)$ has settled down for $t > 0.7$s (see Fig. 6.30), however the control signal $\delta(t)$ continues to increase (see Fig. 6.31a). This may be explained from the fact that during the system's operation the Mach number continues to drop rapidly and thus the trim control also augments according to Fig. 4.4c.

The control rate $\dot{\delta}(t)$ illustrated in Fig. 6.31b is also well inside the constraints (25deg/s for 1g reference commands) as it is also seen from Table 6.7. The obvious question is whether the response could be faster and exploit all the available bandwidth; the answer is of course positive but with the expense of smaller stability margins presented further down. In any case, a major role in this issue is played by the feedforward controllers that provide damping and do not let the control signal be too aggressive; the time constants however remain fast for all reference points considered in the benchmark tests (see Fig. 6.30).

The missile state vector $x = [\alpha \ q]^T$ is depicted in Fig. 6.32; the angle of attack α lies well within its domain of operation (recall from Section 4.1.1 that the nonlinear pitch-axis missile model is valid for $|\alpha| \leq 20°$) and reaches its equilibrium values corresponding to the output reference trajectory, practically with no overshoot and in a smooth way. In addition, the pitch rate q is rapidly augmenting when there is a change to the output operating point and then reaches asymptotically its equilibrium value (see also Fig. 4.3c).

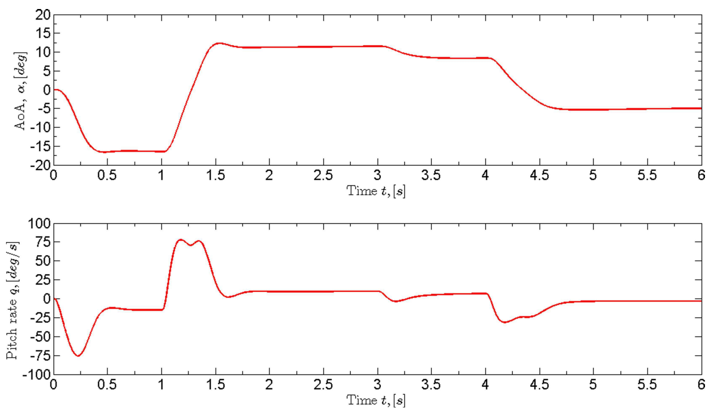


Figure 6.32: Missile state vector.

[32] Recall that the total control signal is the sum of the nominal (or trim) control signal δ_r plus the closed loop stabilization signal δ_δ.

[33] The control signal *before* the actuator δ_c is not shown here since it is very close to δ.

The robust controller gains $K_{\mathrm{PI},\infty}, K_{q,\infty}$ time evolution for the simulation scenario chosen is shown in Fig. 6.33[34]. The gain evolution is smooth and the transition rate between operating points is influenced by the feedforward controller applied on the output of the interpolation mechanism. The same phenomenon as with the control signal is also observed here; the gains do not reach steady-state values since they are interpolated not only as a function of η, but also as a function of the Mach number M that continues to drop during the system's operation.

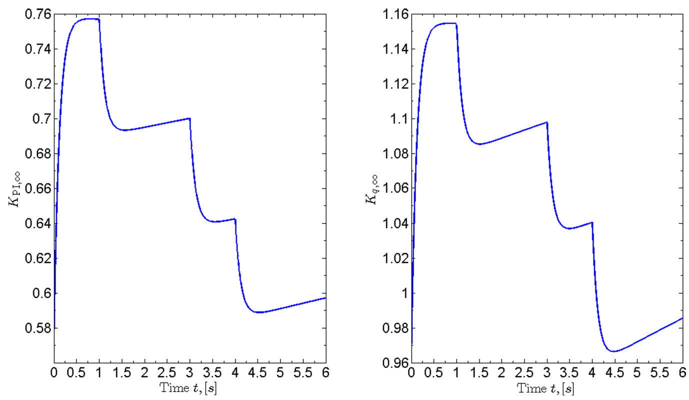


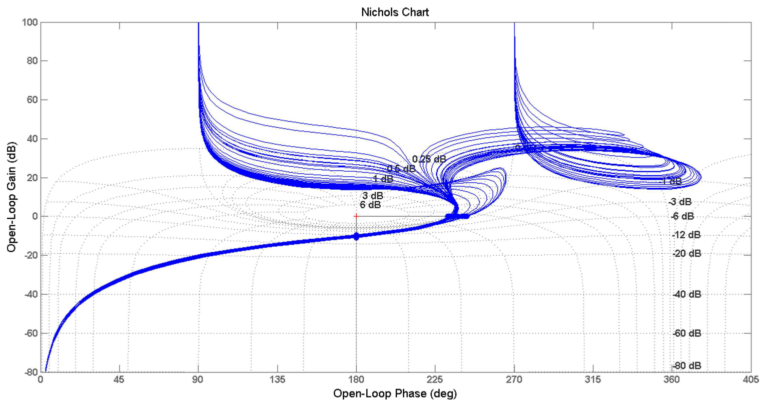
Figure 6.33: Robust controller gains time evolution.

In order to test the stability of the missile under the gain-scheduled controller, the loop is opened before the actuator and the plant is linearized freezing the time each 0.1s during the benchmark scenario of Fig. 6.30 (totally 60 open loop models are obtained). Then the Nichols & Bode diagrams of these open loops are superimposed and illustrated in Figs. 6.34a-6.34b. The worst gain and phase margins are 9.5dB and 52° respectively whereas the worst magnitude attenuation at high frequencies[35] d_{att} is approximatively 27.5db; slightly violating the 30dB robustness constraint limit R_2 (see Section 5.3).

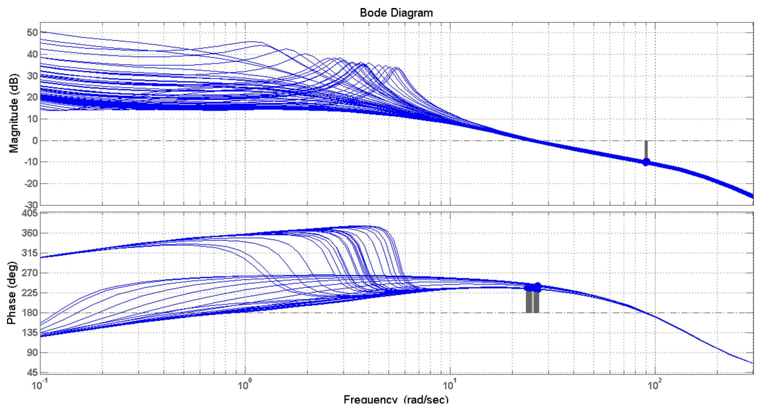
`Uncertain case results`

So far, the gain-scheduled controller's performance has been tested for the two performance and the second robustness constraint of the control objectives described in Section 5.3. The last test concerns the controller's robustness in the face of disturbances in the missile's pitch aerodynamic coefficients $a_{\mathrm{m}}, b_{\mathrm{m}}, c_{\mathrm{m}}$ and d_{m} (see Table 4.1). These coefficients are independently perturbed in two groups (the first three in the same manner and the fourth separately) with a maximum deviation of 25% of their nominal values and all resulting outputs $\eta(t)$ are superimposed in Fig. 6.35a. The independent perturbations follow a Gaussian distribution with zero mean and standard deviation $\sigma = 0.25/3$, with totally 150 cases being considered[36]. The envelope created around the nominal response is not too large and the plant demonstrates robust stability.

[34] The PI/P gains follow similar patterns and are not shown here for brevity.
[35] Recall from Section 5.3 that this corresponds to the robustness objective R_2.
[36] This scenario corresponds to a Monte-Carlo analysis.

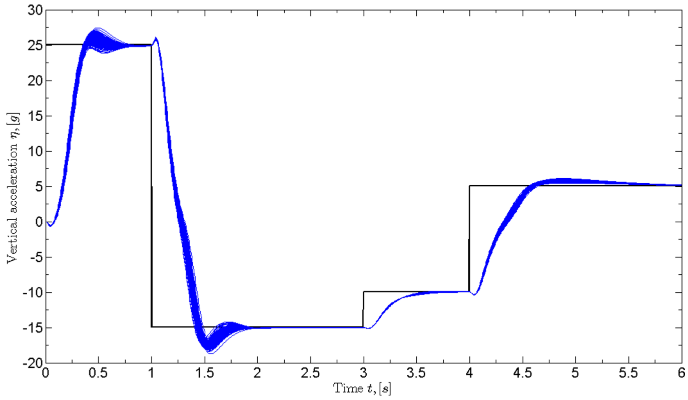


(a) *Nichols diagrams*

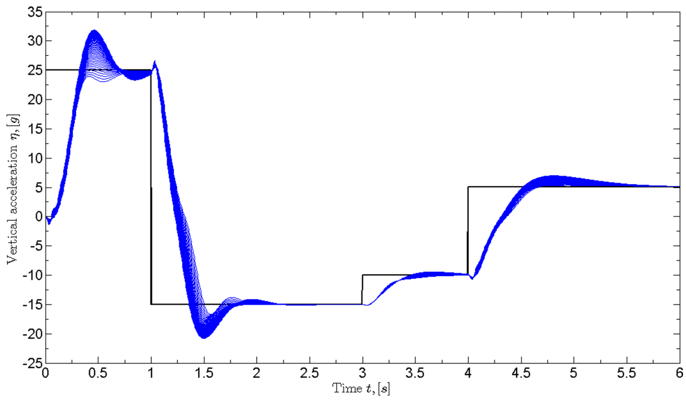


(b) *Bode diagrams*

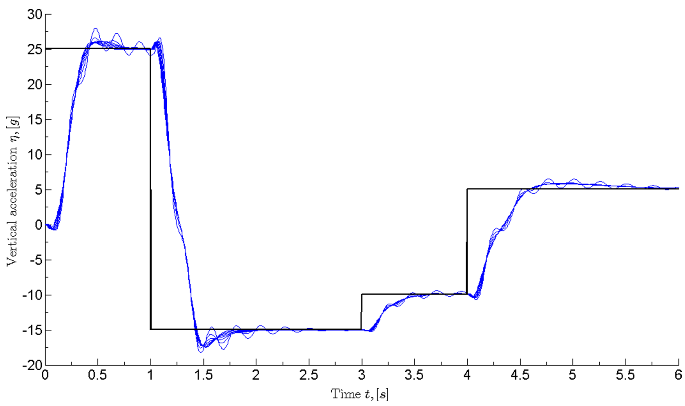
Figure 6.34: Open loop linearization results.



(a) *Parametric robustness*



(b) *Gain robustness*



(c) *Delay robustness*

Figure 6.35: Additional robustness tests.

Some additional tests were performed to demonstrate the robust stability of the missile in the face of open loop gain & delay augmentation. First, a variable gain was added before the actuator and the gain margin of the loop was verified by augmenting the gain from 0.7 to 3 (corresponding to the 9.5db of the worst GM of the previous frozen-time linearization analysis) and taking totally 47 simulation cases. The responses were superimposed in Fig. 6.35b; it is apparent that the output becomes more oscillatory as the loop gain increases but the system remains stable. Second, a variable loop delay was added, taking values from 0 (nominal case) to 35ms (worst case), in order to test the plant's delay margin. The totally 8 cases are also superimposed and illustrated in Fig. 6.35c; the limits of stability are clearly demonstrated as the output starts to oscillate as the loop delay increases. Additional tests

6.3.3.3 Discussion

In this second part of the chapter, a novel gain-scheduling approach for the control of the Reichert missile benchmark model, based on the e-LSDP, was proposed and extensively tested in order to demonstrate both its good time performance and robustness.

The advantages of the proposed method are its simple structure (low order controllers), its ability to take into account the plant's nonlinear dynamics variation (gap metric algorithm) and its simplicity of interpolation (gain interpolation). The gain-scheduled controller depicted in Fig. 6.29 is thus very straightforward to implement on a real system since it does not demand any complex calculations or great memory as for example other gain-scheduled controllers found in the bibliography (see for example [17, 74, 103] and other approaches detailed in Section 5.2).