Robust Autopilot Design by Structured H_∞ Loop Shaping

An Agile Aerospace Vehicle Application

MSc Thesis

Chari Loukisas

Faculty of Aerospace Engineering

TU Delft

Robust Autopilot Design by Structured H_∞ Loop Shaping

An Agile Aerospace Vehicle Application

by

Chari Loukisas

ter verkrijging van de graad van Master of Science
aan de Technische Universiteit Delft,
in het openbaar de verdedigen op woensdag 31 januari om 13:00 uur.

Student number:	5185521	
Project duration:	February, 2023 – January, 2024	
Thesis committee:	Prof. dr. ir. M. Mulder,	TU Delft, AE C&S, chair
	Dr. Eng. S. Theodoulis,	TU Delft, AE C&S, supervisor
	Dr. ir. E. Mooij,	TU Delft, AE C&S, external examiner

Cover: Air Demonstration Team Thunderbirds, ”Precision Loop” by U.S. Air Force CC BY-NC 2.0

TU Delft Delft University of Technology

Preface

As an end to my study journey, I took the challenge of diving deep into the robust control discipline, while hopefully making our understanding of it a tiny bit better. I am proud to state that I will be among the first students in recent years to graduate in this subject with the guidance of my supervisor Spilios Theodoulis. I was additionally given the opportunity to conduct my thesis at TNO. The interesting projects I witnessed will have a lasting impression on me. I am delighted to state that the project has become a success.

This marks the end of a long journey of finding my way into the world of Aerospace Engineering, the discipline I wanted to become proficient at since I was a 10 year old boy living on the small island of Zakynthos. And what a journey it has been. I am humbled and sincerely thankful I have been given the opportunity to pursue this goal. This was made possible by very special individuals who have supported me unconditionally.

Do not go gentle into that good night,
Old age should burn and rave at close of day,
Rage, rage against the dying of the light. - Dylan Thomas

The words to my great friends, my inspiration Georges, my sister Katiana, my girlfriend Andrea, my stepfather Peter and of course my loving mother Margit: I stand here today for you, and because of you. Look at where we started, look where we are now. On the shoulders of giants indeed we stand to reach higher. I am eternally grateful for you.

It is my sincere hope I have contributed to extending humanity's knowledge by even a tiny bit, a contribution to the betterment of our understanding, a betterment of life.

Chari Loukisas
Delft, January 2024

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
SISO	Single Input Single Output
SIMO	Single Input Multi Output
MIMO	Multi Input Multi Output
NCF	Normalized Coprime Factor
DOF	Degree of freedom
PID	Proportional-Integral-Derivative

Symbols

Symbol	Definition	Unit
α	Angle of attack	deg
γ	Uncertainty margin	$-$
$\bar{\gamma}$	Flight path angle	deg
δ	Actuator deflection	deg
δ_c	Commanded actuator deflection	deg
ϵ	Worst-case NCF magnitude	$-$
θ	Attitude	deg
a_x	Longitudinal acceleration	g
a_z	Normal acceleration	g
C_m	Aerodynamic coefficient of the pitching moment	$-$
C_x	Aerodynamic coefficient of the longitudinal force	$-$
C_z	Aerodynamic coefficient of the normal force	$-$
g	Gravitational constant	m/s^2
M	Mach number	$-$
q	Pitch rate	deg/s
$\bar{q}$	Dynamic pressure	kg/ms^2
u	Body longitudinal velocity	m/s
v	Body normal velocity	m/s
V	Airspeed	m/s
$\vert\vert \, . \, \vert\vert_\infty$	H_∞-norm	$-$

1 Introduction

1.1. Context & Goal

Control of (agile) aerospace vehicles is an active and well-known research field, with no exception to the rule in recent years [36, 18, 29] as they pose a challenging (robust) control problem. Robust control is a discipline of special significance for Aerospace Vehicles (manned and unmanned) due to the often critical operating conditions and situations inherent to them. Exposure to (unexpected) uncertainty is therefore a major challenge for control system designers, to which rigorous research has been dedicated to in the last decades and still poses a challenge in modern days [30, 10].

Robust linear control combined with gain scheduling, despite being the industry standard for agile vehicles [10], has been criticised for being complex in the design phase and challenging to implement while requiring extensive manual intervention throughout [16, 36], especially when fast dynamics are involved. A well-known issue with common robust control methods like mixed-sensitivity H_∞ is the often conflicting design requirements and other issues like pole-zero cancellations [36, 3]. In addition, classical robust control methods lead to high-order controllers which are difficult to implement [36]. With respect to these criticisms, the method of H_∞ Loop Shaping introduced in [9] is a powerful robust control method with numerous advantages. The important advantages include an inherent balance between robustness/ performance properties without the need for weighting function definitions (hence simpler design objectives compared to mixed-sensitivity); a-priori robustness measures / goals (hence no need for γ-iteration) and simultaneous gain/ phase margin guarantees [3]. Along with these advantages is the method's widely-applicable 2 stage design process [3] which inherently gives room for automation as opposed to e.g. mixed-sensitivity. Combining this method with structured robustification in addition avoids high-order controllers, which is recently made possible by non-smooth optimization [1].

The goal of this study is thus to demonstrate an effective design procedure using structured (gain-scheduled) H_∞ Loop-Shaping with a focus on automation, including its implementation for testing of the non-linear system of the selected agile aerospace vehicle model.

1.2. Report Structure

The report starts with the (already graded) literature survey in Chapter 2, which serves to cover the fundamentals of robust control theory, specifically H_∞ control, in the context of agile aerospace vehicles. The survey initiates with an introduction of H_∞ Robust Control in section 2.1, where the main aspects of the control methods are covered along with a special focus on Mixed-Sensitivity and H_∞ Loop-Shaping. Emphasis is also placed on the structure of the resulting controllers. Related to this section, model uncertainty is discussed section 2.2, giving more context into general uncertainty modelling. In order to form the research gap and questions, an overview of the state of the art of robust agile vehicle control is given in section 2.3, along with certain challenges and criticisms of the selected design method in section 2.4. Section 2.5 summarizes the insights gained from the literature survey as well as serving as input into the research goals. From the bibliographic survey, a research gap is identified from which the research goal and research questions are formulated in Chapter 3.

The main part of the study can be found in Part II of the report in the form of a scientific article. The selected model along with its requirements is discussed and analyzed, after which the theory of H_∞ is briefly covered. The controller design along with its implementation and results are discussed. Conclusions are formed and certain recommendations are described.

The report ends with a discussion on the achieved successful results. Certain limitations / improvements are identified which are subject to recommendations for future work (based on the article). Finally, a conclusion describes the results of the article and evaluates the answers to the research questions.

Part I

Literature Research

2 Bibliographic Survey

2.1. H_∞ Robust Control

The H_∞ control method is suited and specifically developed to take uncertainty into account during the design phase of a control system. The term H_∞ was first introduced in the 1980s by Zames [28], but failed to take hold to large extents initially in the aerospace industry due to complications in using it for structured controllers (see section 2.1.2), after which it was solved by Apkarian and Noll [1] using non-smooth optimization techniques, leading to it being an industry-standard robust control method, especially for aerospace applications [2]. For agile vehicle autopilots, the approach works well for specific operating conditions (despite being difficult to automate the design over the design space) [10].

The control method uses the H_∞-norm of a selected system signal for robustness and performance reasons. It provides a natural extension of classical and multivariable control concepts and allows for intuitive model formulations and transparent design processes that yield robustness guarantees [3]. This makes H_∞ Robust Control an extremely powerful tool for robust performance control systems.

Just like for H_2, H_∞ optimization problems are typically solved using LMIs and AREs for unstructured controllers, while non-smooth optimization, as mentioned, can be used for structured controllers [1]. In general, the purpose of H_∞ control is to minimize the exogenous effects w on outputs z given a plant P and controller K [3]. Referring to figure 2.1, the problem can be described as [3]:

$$z = N(P,K)w \tag{2.1}$$

where the problem to be solved is:

$$||N(P,K)||_\infty = \gamma_{min} \tag{2.2}$$

However, for practical applications it is not always necessary (or even possible) to achieve the true optimal. A sub-optimal controller may be achieved for this reason with $\gamma_{min} \leq \gamma$ [3]:

$$||N(P,K)||_\infty < \gamma \tag{2.3}$$

The solution of this generic objective depends on what signals are chosen for this objective, which in turn depend on the purpose of the design objectives.

2.1.1. Multivariable control & H_∞ Norm

The H_2-norm is directly related to the H_∞-norm of a system. Consider $G(s)$ that maps input $u(s)$ to output $y(s)$. The H_∞ norm of $G(s)$ is then defined as [18]:

$$||G||_\infty = \sup_{||u||_2 \neq 0} \frac{||y||_2}{||u||_2} = \sup_{||u||_2 \neq 0} \frac{||Gu||_2}{||u||_2} \tag{2.4}$$

Equation 2.4 implies that the H_∞-norm is the factor by which the energy of the input signal is amplified to get the output signal [1]. The direct translation in the frequency domain states that the H_∞-norm is the largest system gain over the entire frequency domain [18]:

$$||G||_\infty = \sup_\omega |G(j\omega)| \tag{2.5}$$

As can be concluded, H_∞ control is about minimizing the closed loop ∞-norm of a system's signals, thus minimizing the frequency-domain peak (i.e. minimizing the maximum (supremum) "energy" amplification in the time domain for all possible realisations). As implied by the "H" (for Hardy space) in H_∞, the H_∞ control problem is suited for the set of transfer functions with a bounded ∞-norm, implying that the transfer functions are stable and proper [30].

The SISO generalization discussed thus far provides clarity on the essence of H_∞ Robust Control. Multivariable control (multi-input multi-output, MIMO) is a highly relevant concept due the highly coupled and interdependent states that are inherent to agile vehicle control and application, next to also being highly relevant for H_∞ Robust Control. This naturally leads to the necessity to discuss the MIMO case.

Multi-input multi-output systems differ from single-input single-output (SISO) systems. In MIMO systems, there can be interactions between (multiple) inputs with (multiple) outputs. Next to this, the concept of vector directions is a relevant concept in MIMO systems unlike for the SISO case. Despite this difference, many concepts and ideas of SISO can be used in MIMO, an important example of which is that the loss of rank of $G(s)$ corresponds to the "zeros" of the transfer function in the SISO case [30]. Another important concept is the relation between scalar system magnitudes (SISO) and singular values (MIMO) [30].

Eigenvalues do not give clear indications of the gain a MIMO system applies to a certain input, as they do not account for the direction of the inputs, which can lead to completely different outputs for inputs u_i with the same magnitude [30]. For this, the concept of singular values is more useful, as the maximum singular value of a transfer matrix G is equal to the largest gain for any input direction [30]:

$$\sigma_{max}(G) = \bar{\sigma}(G) = \underset{u \neq 0}{max} \frac{||Gu||_2}{||u||_2} \tag{2.6}$$

with:

$$\bar{\sigma}(G) = \sqrt{\rho(G^H G)} \tag{2.7}$$

where $\rho = max_i(\lambda_i(G))$, i.e. the largest eigenvalue of G. Singular value decomposition can show the maximum and minimum magnitudes and the corresponding input and output directions [30]. Analogously, the minimum singular value corresponds to the smallest gain, both maximum and minimum being an indication of performance for systems (e.g. feasability of control for minimum values) [30]. A direct relationship can be concluded when this is compared to equations 2.4 and 2.5, namely [1]:

$$||G||_\infty = \underset{\omega \in \mathbb{R}}{max} \ \bar{\sigma}(G(j\omega)) \tag{2.8}$$

By equation 2.8, an H_∞ generalization for MIMO systems exists. Note that G does not necessarily have to be the closed-loop transfer matrix from input to output, but can also represent the sensitivity function, to e.g. account for measured output (sensor) noise.

It is possible to decouple MIMO systems by introducing a compensator that e.g. diagonalizes $G(s)$ at certain frequencies, despite limitations such as in compensator choice and additional model uncertainty sensitivity (which has different/worse and more complex implications than for the SISO case already), limiting its use for practical applications (unlike the case when the system is close to diagonal) [30]. Next to this, by mixed-sensitivity and output weights, emphasis can be placed on one of the output's behaviour rather than another output when using methods like H_∞.

2.1.2. Mixed-sensitivity & H_∞ Loop-shaping

The H_∞ method is powerful when it comes to setting performance requirements to systems, one of which is loop-shaping. A method used to achieve a certain loop-shaping requirement is weighted sensitivity, the "weight" representing a bound over the frequency domain that e.g. the sensitivity function of the system has to adhere to. Since a control system has to often satisfy multiple requirements, the use of mixed sensitivity is possible, in which multiple weights can be assigned and solved through stacking them. Consider the generalized feedback system:

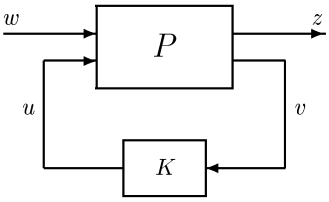


Figure 2.1: Generalized feedback system [30]

Here, w represents the exogenous signals such as reference or noise, z the ”error” signals, v the measured outputs and u the controller inputs. A problem formulation can be illustrated by the following generic case. Consider the following system, with its outputs multiplied with the loop-shaping weights:

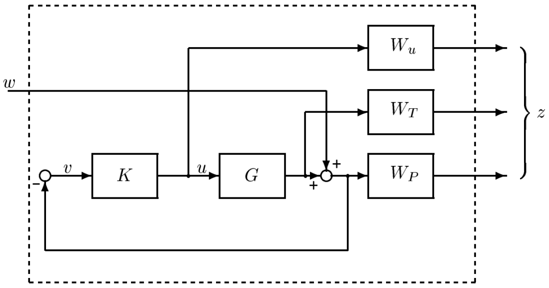


Figure 2.2: Example of block diagram with loop-shaping weights and (exogenous) output noise [30]

This corresponds to the stacked H_∞ problem [30]:

$$\underset{K}{min}||N(K)||_\infty,\ N = \begin{pmatrix} W_u KS \\ W_T T \\ W_P S \end{pmatrix} \tag{2.9}$$

with S the sensitivity function, KS the ”control effort” function and T the complementary sensitivity function. The H_∞ norms of the stacked function’s components are simultaneously optimized, the requirement being that the individual ∞-norms do not exceed the inverse of the weighting functions. MATLAB functions such as **mixsyn** create controllers that satisfy the requirements as close as possible depending on the weight selection [32].

Unlike the name might suggest, a similar yet slightly different approach is H_∞ loop-shaping controller design. The mixed-sensitivity approach involves shaping the closed-loop performance of the system by certain weights and evaluating its robustness and performance afterwards in an iterative fashion. For H_∞ loop-shaping, the procedure is slightly different, as it first ”shapes” the open loop performance of the system to meet certain requirements or good robustness traits (such as roll-off behaviour):

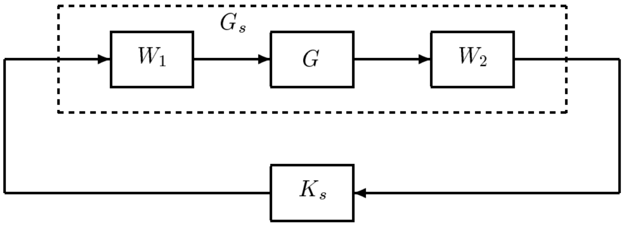


Figure 2.3: Open-loop plant shaping using central controller K_S [30]

Note that the weights W_1 and W_2 are part of the controller, the plant itself is not actually altered. After this, the procedure optimizes the closed-loop system performance for robustness against coprime factor uncertainty (see 2.2.1) while not deviating in large extents from the initially shaped plant for certain allowable values of coprime error $|\Delta N \Delta M|_\infty < \epsilon$ (i.e. the open-loop performance/behaviour of plant-controller combination $|K_S(j\omega)G_S(j\omega)|$ does not deviate much from the open-loop performance of shaped plant $G_S(j\omega)$) [3]. This allows for the initial shaping traits to not be lost depsite having been optimized.

After the shaped plant is determined, the closed loop controller K_S is then determined for a specified $\epsilon < \epsilon_{max}$ (which can already be determined beforehand) [3]:

$$\left\| \begin{pmatrix} K_S \\ I \end{pmatrix} (I - GK)^{-1} M_l^{-1} \right\|_\infty = \left\| \begin{pmatrix} K_S \\ I \end{pmatrix} (I - GK)^{-1} \begin{pmatrix} I & G_S \end{pmatrix} \right\|_\infty < 1/\epsilon \tag{2.10}$$

which is equivalent to [30] [3] [9]:

$$\left\| \begin{pmatrix} K_S S_S & K_S S_S G_S \\ S_S & S_S G_S \end{pmatrix} \right\|_\infty = \left\| \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \to \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \right\|_\infty < 1/\epsilon \tag{2.11}$$

Here the similarity to Mixed-sensitivity becomes obvious, as it takes into account all sources of uncertainty and the effect they have on all system input and output relations as illustrated (note that $K = W_1 K_S W_2$):

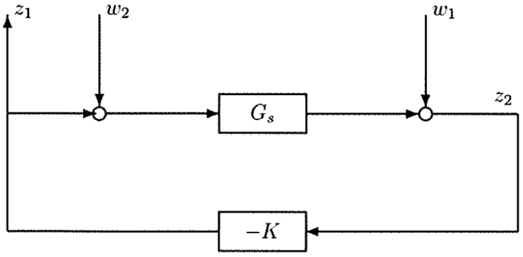


Figure 2.4: Input output effects covered by equation 2.11 [3]

The solutions to these optimization problems can be solved by Ricatti equations or non-smooth optimization for the structured case [1] [3] [31]. H_∞ loop-shaping has many benefits over Mixed-sensitivity, the primary ones being no γ iteration is needed, it enjoys the benefits of coprime factor uncertainty described in section 2.2.1, no pole-zero cancellations, clear(er) management of conflicting requirements, providing a balanced robustness performance approach, guaranteed simultaneous gain and phase margins and exact observer implementation using shaped plant matrices [3].

As described in [35], the so-called Gap-Metric can be used which gives an indication on how much plant uncertainty a given controller can tolerate.

Structured vs unstructured controllers

For agile vehicle control, the H_∞ approach (despite working well for specific operating conditions) can lead to high order controllers that can be difficult to implement [10]. This is therefore something to be considered in the design of the autopilot. In this section, the meaning and definition of controller structure along with its significance is discussed.

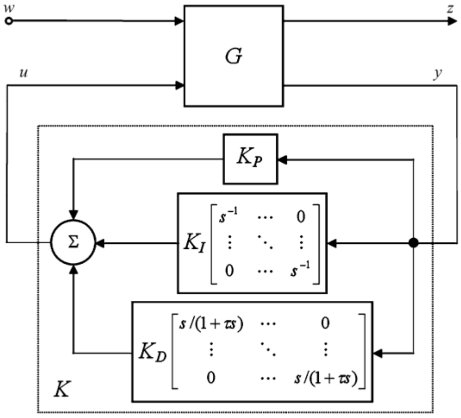
Referring to figure 2.1, controller K can be described as a state-space system [38]:

$$\begin{aligned} \dot{x}_K(t) &= A_K x_K(t) + B_K u_K(t) \\ y_K(t) &= C_K x_K(t) + D_K u_K(t) \end{aligned} \tag{2.12}$$

The control action $y_K(t)$ is related to the controller state vector $x_K(t) \in \mathbb{R}^{n_K}$, with n_K the order of the controller. If n_K is equal to the number of states of the controlled system n_x, the controller is a full-order controller, whereas for the case $0 \leq n_K < n_x$ the controller is of reduced-order (in the case where $n_K = 0$, the controller can be seen a simple gain, i.e. a feed-through system) [38]. Note that the closed loop state-space system of controller and plant can be defined due to the simplifying effect direct feed-through (D_{yu}) has. This however can only be done with unstructured controllers [38].

In the case where a controller is unstructured, the controller matrices A_K, B_K, C_K and D_K along with the order of the controller n_K are free to be chosen [38]. With this freedom, given a design optimization problem for e.g. H_∞, the optimization problem is convex and can be solved by LMIs that reach the global feasible optimum [38]. Often, however, this leads to a full-order (which is often high-order) controller, especially when performance weights are introduced [38]. LMI optimization does not allow for controller structure constraints [38]. As stated, high-order controllers are difficult to implement in already existing controllers and can lead to high computation times, which is especially problematic when speed is of the essence and computational power and memory are limited [38].

In the case when a controller is structured, the structure of the controller matrices and its order are imposed, as stated by Toscano [38]. Apkrarian gives the definition that a controller is called structured if the controller matrices depend smoothly on a design parameter vector varying in some parameter space or in a constrained subset thereof [1]. The free parameters denote the parameters that can be altered/tuned, for example the gains in a PID controller, as shown below:



w
z
G
u
y
K_P
Σ
$K_I \begin{bmatrix} s^{-1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & s^{-1} \end{bmatrix}$
$K_D \begin{bmatrix} s/(1+\tau s) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & s/(1+\tau s) \end{bmatrix}$
K

Figure 2.5: General structure of a MIMO-PID controller [38]

Despite leading to relatively lower performance compared to unstructured (and therefore unrestricted, purely optimal) controllers, structured controllers represent far more realistic controllers (e.g. due to hardware/software limitations, certification, complexity) along with certain benefits, such as the relative ease of gain-scheduling [3]. The problem with structured controllers is that the optimization is non-convex due to the controller structure constraints, making this a significantly more difficult optimization problem to solve. Apkarian and Noll solved this issue in 2006 by using non-smooth optimization techniques, and their methods having been incorporated in the MATLAB function **systune** and **hinfstruct** [1]. This has enabled control system designers to find optimal H_∞, H_∞/H_2, H_∞/H_∞ (multi-plant, single controller) and μ-synthesis solutions with structured controllers [1].

Finally, another way to deal with high-order controllers (and systems with many states) is the concept of model reduction. A method of model reduction is model truncation, in which part of the dynamics are left out in the reduced model (e.g. a state is neglected). In general not a lot can be said about the differences between the original and truncated model, except that the truncated and original model are equivalent at infinite frequencies [30]. However, if the state matrix A is in Jordan form, one can quantify the differences more accurately (defined as the ∞-norm of the error $||G - G_a||_\infty$) [30]. An advantage of truncation is that the poles of the truncated model is a subset of the poles of the original, making the physical interpretations valid in the truncated case. Another method if model residualization, in which some state derivatives are put to zero and are then determined by solving for the affected state in terms of other states. The steady state response of the reduced model and the original are equivalent in this case (since only the derivative is put to zero) [30].

In general, truncation is preferred when accuracy at high frequencies is deemed important, while residualization is preferred for low frequency modelling [30]. In order to get the best of both worlds, an alternating approach between the two is possible [3]. The state selection for truncation or residualization in order to have the minimum model error can be done by balanced truncation / realization, where the states with the smallest Hankel singular values (guaranteeing that the max error is smaller than the combination of singular values of the neglected states); and the states that are the least observable and controllable are selected respectively. The method of order reduction and its specific model requirements (e.g. error versus frequency) can be defined and used by matlab functions such as **balred**. Special attention is needed for unstable model-reduction which depends on the separation of the system into stable and unstable parts (the reduction is then executed on the stable part) and the use of coprime factors [3].

It should be noted that it has been demonstrated that in general for H_∞ controllers controller order reduction does not yield optimal static controllers compared to (pre-)structured controllers that have been optimised [3], highlighting the significance of the non-smooth optimization approach by Apkarian and Noll that essentially solved the structured H_∞ controller problem. This drawback obviously does not necessarily apply to model reduction applied on the plant.

2.1.3. μ synthesis

Similar to H_∞ synthesis, μ synthesis uses the structured singular value μ to optimize for given control problem and is further elaborated in section 2.2.2. The optimization can use methods like DK-iteration, but suffers from inherent problems like non-convexity which can lead to only finding a local optimum [30]. This is especially a problem when the structure of the controller is fixed. A method to deal with this is using non-smooth optimization [1].

2.2. Model Uncertainty & Robustness Analysis

For certification and (robust) performance measurement reasons, taking into account model uncertainty is necessary for designing (automatic) control systems, the extent of which depends on the application. It gives an indication of the origin and effect that certain sources of uncertainty have on the model and its severity, in turn making this an objective for the robustness requirements of the control system. This helps to avoid getting into scenarios where the control system might become unstable for certain uncertainties or (simultaneous) combinations of those, while also taking into account to not be too conservative / pessimistic of its effects such that agility / performance is negatively impacted to a too large extent, which is of special importance for high-performance vehicles.

Classical methods like gain and phase margin are useful for SISO systems as they are even able to take into account simultaneous perturbations by e.g. using the Nichols exclusion regions [3]. However, applying these methods for the MIMO case, aspects like cross-coupling effects can give a unreliable indications of robustness (i.e. although methods like the Nichols exclusion region can be extended to MIMO, the fact that all possible gain and phase perturbation combinations need to be considered makes it a cumbersome method especially for multiloop systems with many sources of uncertainty [3]). Therefore, this section provides an overview of uncertainty modelling methods that are applicable for multivariable systems, along with what information they may provide on robustness. Note that classic SISO methods (e.g. parametric uncertainty, Nyquist uncertainty illustrations) are not mentioned here and is deemed background information.

2.2.1. Unstructured Uncertainty

Unstructured uncertainty is the type of uncertainty that makes no assumption about the exact origin or structure of a certain uncertainty source Δ, but assumes a defined effect of the system, that is also bounded [3].

Additive & multiplicative uncertainty

Additive and multiplicative uncertainty models, given a theoretical system model $P_0(s)$ and the model that includes uncertainty $P(s)$, are of the form [3]:

$$\begin{aligned} P(s) &= P_0(s) + \Delta_A(s) \\ P(s) &= P_0(s)(I + \Delta_I(s)) \\ P(s) &= (I + \Delta_O(s))P_0(s) \end{aligned} \tag{2.13}$$

The uncertainty $\Delta(s)$ is allowed to be any stable complex matrix satisfying $||\Delta||_\infty \leq 1$ [30]. Given a plant and controller combination M and uncertainty Δ, the structure of the system with uncertainty included is of the form [3]:

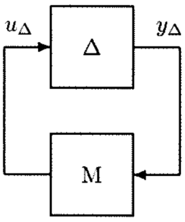


Figure 2.6: Uncertainty interconnection structure [3]

From the small gain theorem, this system fulfils the robust stability condition (the system is stable for all possible uncertainty perturbations given a controller K) when the following condition is met [3] [30]:

$$||M||_\infty||\Delta||_\infty < 1 \tag{2.14}$$

The considered uncertainties can be applied to input-output and/or output-input uncertainty e.g. in terms of the sensitivity function [3]. Methods for analyses can be applied in terms of determining the smallest destabilizing perturbation [30]. Furthermore, stability analysis can also be conducted using the MIMO disk [33].

Coprime factor uncertainty

Another method that can model certain effects that additive/multiplicative uncertainty cannot (e.g. destabilizing parameter variations, resonance frequency change etc.) is coprime factor uncertainty (albeit with the disadvantage that it can lack physical interpretation) [3]. Next to this, H_∞-loop shaping uses coprime factor uncertainty to optimize the system.

The coprime factorization is not unique and can be left or right handed for a system $G(s)$:

$$G(S) = N_r(s)M_r^{-1}(s) = M_l^{-1}(s)N_l(s) \tag{2.15}$$

for which a stable coprime factorisation is when N includes the right-half plane zeros and M the right-half plane poles [3]. The uncertainty for a perturbed system $G_p(s)$ can in turn be noted as:

$$G_p(s) = (M_l + \Delta_M)^{-1}(N_l + \Delta_N) \tag{2.16}$$

For which $||[\Delta_N \ \Delta_M]||_\infty \leq \epsilon$. The equation can be visualized:

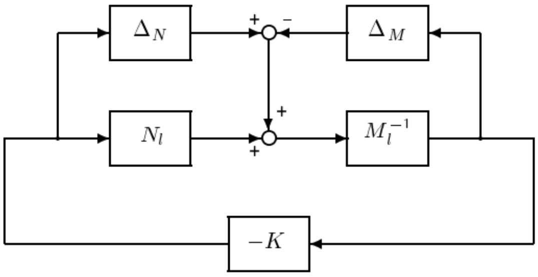


Figure 2.7: Coprime factor uncertainty block diagram [30]

Using equation 2.14 and figures 2.6 and 2.7, M and its infinity norm can be written as follows:

$$||M||_\infty = \left\|\begin{pmatrix} K \\ I \end{pmatrix}(I - GK)^{-1}M_l^{-1}\right\|_\infty < 1/\epsilon \tag{2.17}$$

where $||M||_\infty < 1/\epsilon$ ensures the condition of equation 2.14 is met (as $||[\Delta_N \ \Delta_M]||_\infty \leq \epsilon$ holds). Methods like Nichols exclusion zones can be used to visualize perturbations for the SISO system (which can tell us something about the MIMO case in turn) [3].

2.2.2. Structured Uncertainty

As mentioned in the introduction of this section, it is important to not be too optimistic or pessimistic of the severity of uncertainty in a system. Being too optimistic is dangerous for obvious reasons, but too pessimistic (i.e. being too conservative) may come at the price of performance to account for the (too) severe uncertainties. Structured uncertainty may give way of avoiding the relatively pessimistic approach of unstructured uncertainty.

The ”problem” is in the structure of Δ. Unlike the case for unstructured uncertainty where the singular values of full-block uncertainty are considered, the structured singular values allow for tighter robust stability by taking into account the structure of the uncertainty (e.g. when Δ is block-diagonal, it is structured) [30]. The structured singular value μ is used to evaluate structured uncertainty and it is the generalisation of the singular value and spectral radius of a system [30]. The larger the value of μ the smaller the perturbation that can cause the system to turn unstable. Structured singular values give the generalized small-gain theorem [30]:

$$\mu(M(j\omega))\bar{\sigma}(\Delta(j\omega)) < 1 \;\; \forall\omega \tag{2.18}$$

The structured singular value can provide a variety of information (e.g. factor by which the uncertainties have to be reduced to guarantee stability, or part of uncertainties by using the skewed-μ) and can be used for different perturbations, such as complex perturbations along with a variety of useful properties [30]. Therefore, the structured singular value gives a solid indication of robust stability when detailed uncertainty analysis is conducted. It also forms the basis of μ-synthesis where robustness/performance requirements can be imposed.

2.3. State of the art H_∞ Robust control

2.3.1. Overview

The most recent study on state of the art robust agile vehicle control is given in [29], where an LPV/LFT approach is taken to construct a fixed-structure controller that uses the bounded-range angle of attack as the LPV gain-scheduling parameter (and the bounded-range mach number as uncertainty) to meet certain requirements posed by integral quadratic constraints (IQC) and H_∞ mixed-sensitivity weights for robustness. This study builds upon a previous study given in [26], this time putting the emphasis on maintaining a very simple controller (in terms of order and the scheduling parameter) by using non-smooth optimization. Despite the controller order being significantly lower and therefore simpler, identical performance was reached. This study demonstrated the power of the LPV/LFT methods to meet certain criteria posed by mixed-sensitivity weights, despite the controller being limited to being simple. This method was applied to a fixed-altitude 2-dimensional model first given in [23] (a model used by many academic studies and by industry). A similar study is given in [34]. A LPV application (mixed-sensitivity) with a 5-DOF model is given in [14]. In a previous study described in [4], an introduction is given to using LPVs for this purpose, yet giving the disclaimer that the technology is not on point yet and development is needed, highlighting the significance of the previously mentioned studies.

Using the same model, a gain-scheduled controller is designed in studies like [37] and [36]. Using a Mixed-Sensitivity approach, a gain-scheduled controller is designed by using point-linearizations solved in two methods: point-by-point (a posteriori) and all simultaneously (a priori). Note that the effect of nonlinearities are treated as uncertainties in the design. On top of the H_∞ approach that is used, a reference model is used (also the case in a previous study in [27]). This lead to the design of an effective and robust autopilot. An important part of the studies is that the controller structure is fixed and therefore uses non-smooth optimization. In fact, the controller structure is such that it uses the improved version of the classical 3-loop autopilot form as proposed in [15]. The latter study improved the classical autopilot by investigating a formed Mixed-Sensitivity autopilot and incorporating its ”effect” in the classical autopilot.

In [39], stabilization and control of a highly unstable supersonic air-breathing vehicles is described using a mixed LQR and mixed-sensitivity H_∞ approach, the former being used to guarantee stability and good gain and phase margins of the nominal plant, while the latter is used for robustness. The control method was successfully implemented. A mixed H_2/H_∞ approach is used in [40].

In [8] a minimum H_∞ Entropy method is used by minimizing an entropy function, after which it is compared to a classical LMI method and applying both on a model. The latter still outperforms the the entropy method which is largely to be attributed to the lack of transient response specifications, yet is robust. The paper suggests more studies to be conducted into information theoretic functions being used for this control purpose. An application using neural networks is also suggested.

Unlike most studies, in [21] a robust controller is developed and applied to a 6-DOF model and evaluated in a realistic simulation. The controller is developed using a mixed-sensitivity approach. Emphasis is put into model order reduction techniques such that the reduced model still represents the true model sufficiently. The same model order-reduction technique is also used on the determined controllers. In terms simulation scenarios, synthesis and order reduction a similar study is presented in [25]. In addition, different weighting functions are used in this study. An interesting aspect is its clear demonstration of how a robust controller has a consistent performance compared to classical controllers. Unlike the mentioned studies so far that used mixed-sensitivity, another study applying control to a 6-DOF model is presented in [19] in which two controllers are developed for different mach numbers using H_∞ Loop-Shaping. Order reduction is applied on the controller. The controller is compared to and outperforms a backstepping controller, primarily in terms of stable commands and in terms of the amount of feedback information needed. A simulated scenario is created to demonstrat its capabilities. An interesting aspect to note lies in how two robust controllers are combined for smooth controller transfer which avoids gain-scheduling needs to a certain extent, however clearly shows a sub-optimal controller (next to the fact that it is order-reduced).

On H_∞ Loop-Shaping, a comparison of the performance of a H_∞ Loop-Shaping and a mixed sensitivity controller are compared when applied to a pitch model in [12]. A more systematic procedure for gain-scheduled (by linearization) H_∞ Loop-Shaping 2-DOF controllers is given in [35]. The gap-metric is used in this study to evaluate uncertainty tolerances and extensive procedures are described in terms of e.g. flight envelope synthesis points. In [7] H_∞ Loop-Shaping is applied on a 6-DOF model while also using the ν-tool for analysis. An interesting (and highly related to H_∞ control) method is μ-synthesis as studied in [24] and [5]. For this, the type of (structured) uncertainty needs to be accurately modelled as opposed to e.g. H_∞ Loop-Shaping where a more generic (also more conservative) unstructured uncertainty method used, namely coprime-factor uncertainty.

Other methods such as machine learning and adaptive control are covered in [41, 17, 20, 6]. It has been mentioned however that H_∞ control and Non-linear control are deemed of special interest in the control of agile vehicles, with H_∞ control being an industry standard [10].

Conclusion

This section has provided a range of state of the art studies when it comes to using H_∞ Robust control methods and has proven its usefulness in terms of guaranteed stability and performance . Mixed-sensitivity applied on 2-DOF models is well represented in recent studies and vary from describing effective agile vehicle controller synthesis procedures to evaluating appropriate weighting functions. LPV also holds high potential and is used in extremely recent state of the art studies. Significant emphasis is placed on designing and optimizing fixed-structure and simple controllers which represent realistic controllers that could be used in real vehicles, which is fairly recently made possible by non-smooth optimization [1]. More recent control methods like dynamic-inversion methods and artificial intelligence control show great potential in this field [10], with active research being conducted for various applications. However, these methods have as of now their own drawbacks which prevent it from becoming extensively used in industry.

2.4. Challenges

Classical (robust) control methods function by linearizing a to be controlled non-linear system around a set of design points that define the flight envelope [22]. The individual controllers are then to be combined, such that the implemented controller uses the right linear controller depending on its flight condition and its corresponding values of the scheduling variables. This process is known as gain scheduling.

Robust linear control combined with gain scheduling has been criticised for being complex in the design phase and challenging to implement while requiring extensive manual intervention throughout [36, 16], especially when fast dynamics are involved as is the case in agile vehicles. A major reason for these issues is the often high order of the robust controller (as discussed earlier) using classical robust control (convex optimization problems solved by Linear Matrix Inequalities and Algebraic Ricatti Equations), requiring order-reduction that often leads to loss of robustness [1, 3, 36]. Robustifying controllers with pre-imposed structures suffer from being non-convex optimization problems that cannot be solved using conventional optimization methods [3, 1]. This has been solved in [1] and used in recent studies, it is to be explored further in other applications such as H_∞ Loop-Shaping.

A well-known issue with common robust control methods like Mixed-Sensitivity H_∞ is the often conflicting design requirements and other issues like pole-zero cancellations [36, 16]. Among other reasons, the H_∞ control methods are challenging to automate [36, 16], which is a major issue in terms of design difficulty when in combination with many design points, which is often the case with gain scheduling. In addition, H_∞ control methods are known to be quite conservative [30].

Finally, implementing gain-scheduled controllers with fast dynamics pose issues in terms of predictability, for which special implementation methods can be used as covered in e.g. [11].

2.5. Conclusion

The literature research started off with the basic concepts of robust control methods. The concepts discussed are potent and well-established control methods and allow model-based control design, a fundamental part of today's control industry. The strength of robust control methods lies in their ability to have guaranteed stability for part of the/ the entire flight envelope with performance requirements taken into account. Despite the methods being well understood with old roots and largely applied, it is still is a very active research field, especially for agile vehicles.

The combination of robust control methods with gain scheduling do involve certain limitations and challenges. The most important challenges involve the complexity of the design phase and the implementation issues that arise from high-order controllers. Simplicity of the design procedure combined with low-order controllers are therefore important points that are to be improved upon.

3 Research Goals

3.1. Research Gap and Contribution

It is clear from the previous section that robust control methods are well-established and form the industry standard for agile vehicle control. Along with this, certain challenges and limitations to the methods are covered in recent studies. With respect to these challenges, certain subjects are less represented in recent literature, offering a research gap. H_∞ Loop-Shaping, despite being used in some studies and having been proven to be a potent synthesis tool [3], seems to be less represented in its application to agile vehicles. Additionally, using the specific combination of H_∞ Loop-Shaping and structured controllers seems to be relatively uncovered ground.

As mentioned in the literature survey, H_∞ Loop-Shaping has certain benefits over Mixed-Sensitivity, its most important aspects being its a-priori guarantees of robustness and its inherent balance between robustness and performance. Its 2-stage design process additionally offers an attractive design philosophy which has high potential for design automation with minimal manual intervention. The combination of the design procedure with structured robustification in turn may yield a widely-applicable design procedure that offers easy automation and implementation into the non-linear system. The latter in turn offers a test of the quality of the design procedure along with insights into the challenges that this procedure poses while implementing the controller. It is expected that since the control of agile aerospace vehicles is an active research field, the application of the design procedure will additionally yield insight into the control of such systems.

It is for these reasons chosen to demonstrate and introduce a design method using H_∞ Loop-Shaping in combination with structured controllers. Specifically, the design method is applied on an agile aerospace vehicle due to the fast dynamics involved, posing a challenging control problem. The most important robust control limitations that this combination thus tackles are the limitation of extensive manual intervention and difficulty of implementation, next to offering additional insight into the robust control of aerospace vehicles with fast dynamics.

3.2. Research Questions

The goal is to develop and demonstrate an effective design procedure which combines H_∞ Loop-Shaping with structured controller design. The method is to be applied on an agile aerospace vehicle, for which the non-linear implementation and non-linear results are to be covered. In summary, the research objective is thus defined to be:

Research Objective

Develop and demonstrate an effective design procedure using structured H_∞ Loop-Shaping with a focus on automation; implemented into and evaluated against the non-linear system

For an effective overview of the research objective, the following main- and sub-questions need to be answered. The questions are tackled in the scientific article of Part II and discussed in the closure, Part III.

Research Questions

Main research question:
What is an effective agile vehicle autopilot design method using Structured H_∞ Loop-Shaping with a focus on automation?
Sub-questions:

1. **What agile aerospace vehicle model is used and what are its control requirements?**

   1.A What are the non-linear dynamics of the model?
   1.B What are the linearized models/ design points to be used?
   1.C What are the specific robustness and performance requirements of the final system?

2. **How is H_∞ Loop-Shaping applied?**

   2.A What is the magnitude of the uncertainty?
   2.B How are shaping functions W_1 and W_2 defined and applied for Loop-Shaping?
   2.C How is the shaped plant robustified and what is the structure of the robustifying controller?
   2.D How is the time-domain performance enhanced?
   2.E How well are the requirements met by the individual design points?

3. **How is the the autopilot implemented and what is its non-linear performance?**

   3.A What implementation methods are required and used?
   3.B What is the time-domain performance of the autopilot?
   3.C How does the system handle uncertainty?
   3.D How does the system perform in a practical scenario?

Part II

Scientific Article

Robust Gain-Scheduled Autopilot Design by Structured H_∞ Loop Shaping for an Agile Aerospace Vehicle

Chari Loukisas
Delft University of Technology
Faculty of Aerospace Engineering, Department of Control & Simulation

This study demonstrates an effective systematic control design procedure by applying H_∞ Loop-Shaping with a structured controller on an agile aerospace vehicle with a focus on automation. The gain-scheduled implementation is additionally described and tested with non-linear simulations, including a realistic moving point-hit scenario with guidance. The imposed robustness and performance requirements are met for most linear design points and for the non-linear simulations. The resulting autopilot design procedure is deemed effective in both the design procedure and implementation. It is subject to certain recommendations for improvement and extension.

Nomenclature

$\|\| \cdot \|\|_\infty$	=	H_∞-norm
α	=	Angle of attack
$\bar{\gamma}$	=	Flight path angle
$\bar{q}$	=	Dynamic pressure
δ	=	Actuator deflection
δ_c	=	Commanded actuator deflection
ϵ	=	Worst-case normalized coprime factor uncertainty magnitude
γ	=	Stability margin
$\overline{\sigma}$	=	Maximum singular value
θ	=	Attitude
$\underline{\sigma}$	=	Minimum singular value
a_x	=	Longitudinal acceleration
a_z	=	Normal acceleration
C_m	=	Aerodynamic coefficient of the pitching moment
C_x	=	Aerodynamic coefficient of the longitudinal force
C_z	=	Aerodynamic coefficient of the normal force
g	=	Gravitational constant
M	=	Mach number
q	=	Pitch rate
u	=	Body longitudinal velocity
V	=	Airspeed
w	=	Body normal velocity

I. Introduction

Control of (agile) aerospace vehicles is an active and well-known research field, with no exception to the rule in recent years [1–3]. The essence of modern control is based on mathematical models of the controlled vehicles that are used in the design phase [2]. The natural disadvantage of this approach is that models are not capable of fully capturing the complexity and detail of real life's dynamics and (environmental) effects [2]. This misalignment of models and real system behaviours can be seen as "uncertainties", which can lead to unexpected control system behaviours if not accounted for properly. Uncertainties may find their origin from different sources, including misalignment in plant-parametric models and real life due to modelling errors, sources of noise or environmental factors.

Robust control is a discipline of special significance for Aerospace Vehicles (manned and unmanned) due to the often critical operating conditions and situations inherent to them. Exposure to (unexpected) uncertainty is therefore a major challenge for control system designers, to which rigorous research has been dedicated to in the last decades and still poses a challenge in modern days [4, 5]. The essence of many robust control methods is to account for these uncertainties by incorporating their worst-case condition in the design phase [2, 6], with the goal to design a control system that remains stable with adequate stability margins while also still being able to complete their control task successfully despite these uncertainties.

Robust control of agile aerospace vehicles vary across studies in methodology from Linear / LPV methods [1, 3, 7–10] to Non-Linear (and comparisons between the two) [11–14] and other methods such as machine learning and adaptive control [15–18]. In particular H_∞ control and Non-linear control are deemed of special interest in the control of agile vehicles , with H_∞ control being an industry standard [1, 5].

Robust linear control combined with gain scheduling has been criticised for being complex in the design phase and challenging to implement while requiring extensive manual intervention throughout [1, 11], especially when fast dynamics are involved as is the case in agile vehicles. A major reason for these issues is the often high order of the robust controller using classical robust control (convex optimization problems solved by Linear Matrix Inequalities and Algebraic Ricatti Equations), requiring order-reduction that often leads to loss of robustness [1, 19, 20]. Robustifying controllers with pre-imposed structures suffer from being non-convex optimization problems that cannot be solved using conventional optimization methods [19, 20]. Additionally, a well-known issue with common robust control methods like mixed-sensitivity H_∞ is the often conflicting design requirements and other issues like pole-zero cancellations [1, 19].

With recent advances in non-smooth optimization techniques, the non-convexity issue has been resolved and incorporated in commercially available software packages, namely the functions `systune()` and `hinfstruct()` [20]. This implies simpler / lower-order (i.e. fixed-structure) controllers can be optimized with multiple objectives and can in turn often get close to the robustness of the unstructured / full-order controllers, as effectively demonstrated in e.g. [1, 3]. This ability is of special significance to H_∞ problems, given that these inherently are used in combination with gain scheduling where controller order / structure plays a significant role.

The method of H_∞ Loop Shaping introduced in [21] is a powerful robust control method with numerous advantages. The important advantages include an inherent balance between robustness/ performance properties without the need for weighting function definitions (hence simpler design objectives compared to mixed-sensitivity); a-priori robustness measures / goals (hence no need for γ-iteration) and simultaneous gain/ phase margin guarantees [19]. Along with these advantages is the method's widely-applicable 2 stage design process which inherently gives room for automation as opposed to e.g. mixed-sensitivity [19].

With respect to the mentioned criticisms, the goal of this article is to demonstrate a systematic, widely applicable autopilot design procedure that combines H_∞ Loop Shaping with structured robustification applied on an agile aerospace vehicle, due to the numerous automation and implementation benefits this combination yields. Along with this, this combination applied on an agile aerospace vehicle is not largely represented in literature to the best of the author's knowledge, and will therefore yield useful insights from the application alone. This article hence aims to demonstrate an effective application of this design procedure and its result, showing its relative simplicity and automation capability. Additionally, next to the description of the non-linear implementation, an unconventional but intuitive approach is covered to solve the issue related to the neglected dynamics-induced steady-state error. The linear design points are analysed individually with respect to the imposed requirements. The non-linear results are compared to a reference model for time domain performance, while the uncertainty handling qualities are shown for a number of uncertainty realizations.

The article initiates with a description of the non-linear model's dynamics and its imposed requirements in section II. The process of trimming and subsequent linearizations are additionally discussed and analysed. Classic H_∞ Loop Shaping theory is in turn briefly covered for the unstructured robustification case in section III. The linear control design procedure in section IV initiates by converting the sets of linearized models into time-scale separated and stabilized SISO systems. The model's imposed parametric uncertainty is analysed with respect to Normalized Coprime Factor uncertainty and its magnitude (with respect to the SISO systems). The H_∞ Loop Shaping procedure with structured controller optimization is then applied on the linearized models and extensively analysed, the sets of which are eventually used for direct gain-scheduling. The article finishes with the non-linear implementation of the scheduled autopilot along with its full non-linear simulation results, showing the autopilot's uncertainty handling and (time-domain) agility in section V. In the same section, as a demonstration of the successful autopilot design, a more realistic non-linear scenario is created which involves proportional guidance with the goal that the vehicle hits a rapidly-moving point in space. The observations and conclusions are subsequently covered, along with certain recommendations for future work in section VI. The software used is `MATLAB` and *Simulink* by MathWorks.

II. Aerospace Vehicle Model

The non-linear agile aerospace vehicle model that is used is given in [22] and further analysed in [1], the results of which (including the body reference frame conventions) are used throughout this study. The tail-controlled (hence non-minimum phase) vehicle is described by its non-linear 2-dimensional dynamics at fixed altitude at supersonic air speeds. The vehicle is in a so-called burn-out phase, where no thrust is being produced, implying no shift in the center of gravity is taking place and the aerodynamic drag is the dominant longitudinal force. However, high performance time-domain requirements are imposed. The model is used in numerous studies like [1, 3, 8, 22] and therefore suitable for cross-verification of model results and control performance. Its simplifying assumptions allow for an effective demonstration, yet still pose a challenging control problem given its dynamics and requirements as discussed later.

As linear control is being applied, the non-linear model needs to be linearized at a defined set of points within the considered flight envelope. These linear sub-models are subject to requirements that should be satisfied as closely as possible by the local linear controller covered in section IV, such that the non-linear gain-scheduled autopilot also satisfies these requirements as close as possible by locally reflecting the behaviour of the linearized system.

A. Non-Linear Flight Dynamics

The non-linear model consists of (symmetrical) non-linear airframe dynamics and a linear second-order actuator model. They hold for a flight envelope defined by the mach-number and angle of attack for domains $1.5 \leq M \leq 3$ & $-20^\circ \leq \alpha \leq 20^\circ$. The analysis of [1] is used and described.

1. Airframe & Actuator

The pitch-axis state dynamics of the airframe are a function of the angle of attack α and the pitch rate q given by:

$$\begin{cases} \dot{\alpha} = \left(\frac{\bar{q}S}{mV}\right) C_z\left(\alpha(t), \delta(t), M(t)\right) \cos\alpha(t) + q(t) \\ \dot{q} = \left(\frac{\bar{q}Sl}{I_y}\right) C_m\left(\alpha(t), \delta(t), M(t)\right) \end{cases} \tag{1}$$

The aerodynamic coefficients of the normal force C_z and pitch moment C_m are given by:

$$C_z(\alpha, \delta, M) = C_{z_a}(\alpha, M)\alpha + d_n\delta \tag{2}$$

and:

$$C_m(\alpha, \delta, M) = C_{m_a}(\alpha, M)\alpha + d_m\delta \tag{3}$$

The respective angle of attack derivatives are given to be:

$$C_{z_a}(\alpha, M) = a_n\alpha^2 + b_n sgn(\alpha)\alpha + c_n\left(2 - \frac{M}{3}\right) \tag{4}$$

and:

$$C_{m_a}(\alpha, M) = a_m\alpha^2 + b_m sgn(\alpha)\alpha + c_m\left(-7 + \frac{8M}{3}\right) \tag{5}$$

The Mach number is given by $M = V/a$, with a the speed of sound. The dynamic pressure is given by $\bar{q} = \frac{1}{2}\rho V^2$. The atmospheric variables are given for a fixed altitude of $h = 6096m$ $(20000ft)$. These are given along with the model parameters in Table 1. The rate of change of the Mach number (with C_x constant):

$$\dot{M} = \left(\frac{\bar{q}S}{ma}\right)\left(C_x \cos a + C_z \sin a\right) \tag{6}$$

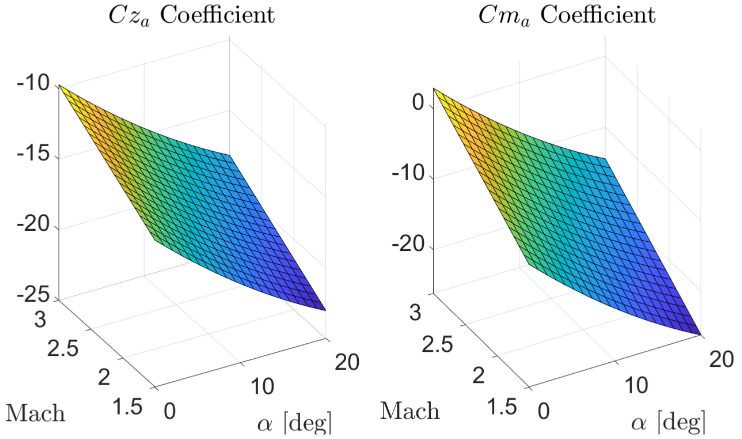
The output dynamics (i.e. observable/measured states) of the system consist of the pitch rate q and the normal acceleration a_z, the latter of which given by:

$$a_z = \left(\frac{\bar{q}S}{m}\right) C_z\left(\alpha(t), \delta(t), M(t)\right) \tag{7}$$

The actuator of the model controls a tail-fin, which is modelled by the following linear second-order system:

$$\begin{pmatrix} \dot{\delta} \\ \ddot{\delta} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\omega_a^2 & -2\zeta_a\omega_a \end{pmatrix} \begin{pmatrix} \delta \\ \dot{\delta} \end{pmatrix} + \begin{pmatrix} 0 \\ \omega_a^2 \end{pmatrix} \delta_c(t) \tag{8}$$

In Figure 1 the angle of attack derivatives of the aerodynamic normal force and moment coefficient are shown, giving the same results as [1].



Cz_a Coefficient

Cm_a Coefficient

Figure 1. Angle of attack derivatives of aerodynamic coefficients

A noteworthy result from C_{m_a} can be seen, as $C_{m_a} > 0$ holds for a small region of the flight envelope, indicating an unstable region.

B. Linearized Flight Dynamics

The model is extensively analysed in terms of linearization in [1], the main points of which are described here. Jacobian linearization is used as opposed to velocity-based or LPV methods for simplicity's sake, which has implications that are to be addressed for the non-linear implementation in section V.

Trimming is conducted such that the dynamics at a certain to-be-linearized point do not suffer from fast changes (i.e. the changing speed from a given confidence region of a linear point is minimized) which can lead to rapid deviation from the confidence region and yielding unrepresentative dynamics [23]. Due to the nature of the vehicle considered, a trim point is differently defined than for a conventional aircraft. Since the flight envelope is defined by α and M, their values are imposed. With the exception of the Mach number (due to drag), the angle of attack is imposed to be steady state. Additionally, q is imposed to be steady state only. For a given combination of trim variables that satisfy the impositions for a given deflection δ, let $\bar{\mathbf{T}}$ be the trim vector defined as:

$$\bar{\mathbf{T}} = \begin{pmatrix} \alpha \\ M \end{pmatrix}_{\dot{\alpha},\dot{q}=0} \in \mathbf{F} \tag{9}$$

with $\mathbf{F}$ the set of all α and M that define the flight envelope. The `MATLAB` functions `operspec()`, `findop()` and `linearize()` can be used for this purpose. As derived in [1], the linearized airframe dynamics are given in state-space form, where the state values form the differences from their trim values:

$$\begin{pmatrix} \dot{\tilde{\alpha}} \\ \dot{\tilde{q}} \end{pmatrix} = \begin{pmatrix} Z_\alpha(\bar{\mathbf{T}}) & 1 \\ M_\alpha(\bar{\mathbf{T}}) & 0 \end{pmatrix} \begin{pmatrix} \tilde{\alpha} \\ \tilde{q} \end{pmatrix} + \begin{pmatrix} Z_\delta(\bar{\mathbf{T}}) \\ M_\delta(\bar{\mathbf{T}}) \end{pmatrix} \tilde{\delta} \tag{10}$$

$$\begin{pmatrix} \tilde{a}_z \\ \tilde{q} \end{pmatrix} = \begin{pmatrix} A_\alpha(\bar{\mathbf{T}}) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{\alpha} \\ \tilde{q} \end{pmatrix} + \begin{pmatrix} A_\delta \bar{\mathbf{T}}) \\ 0 \end{pmatrix} \tilde{\delta} \tag{11}$$

It is for the purposes of this project assumed there is access to the observable states without noise, hence state-feedback is assumed possible. The respective derivative maps are shown below in Figure 2, giving the same results as given in [1].

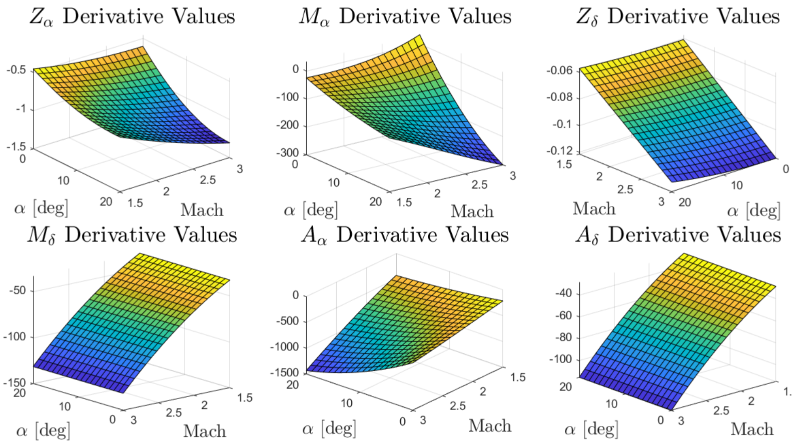


Figure 2. Angle of attack derivatives of Aerodynamic Coefficients

As shown clearly for M_α in Figure 2, there is a small region for which the derivative is positive, implying instability. Figure 3 shows the complete unstable region for the airframe. The unstable region has a substantial impact on the design process as discussed in section IV.

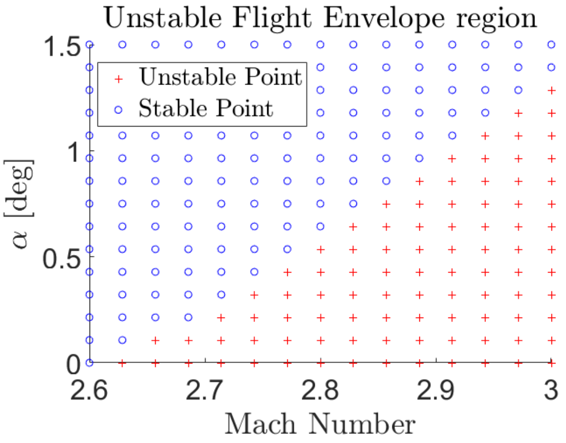


Figure 3. Unstable flight envelope region

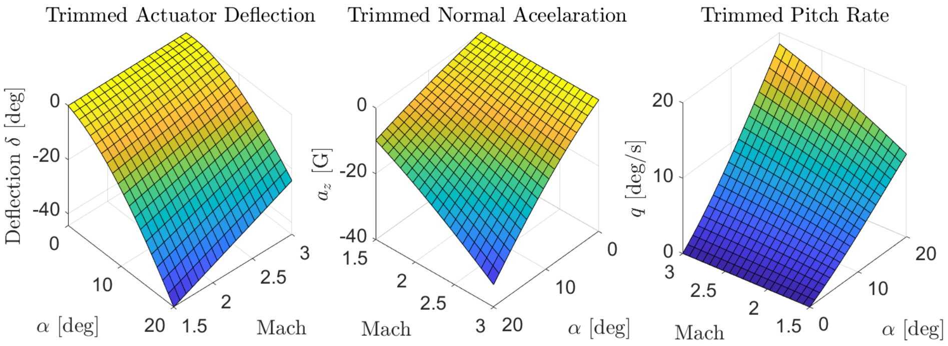


Figure 4. Trim values for linearized points

The derivatives were determined with the trim values shown in Figure 4, giving similar smooth results as in [1].

C. Autopilot Requirements

The requirements that have to be satisfied by the autopilot can be categorized into robustness requirements and (time-domain) performance requirements. The requirements are drawn from the same source as the vehicle model [22]:

- Robustness requirements:
    1) Maintain robust stability for the complete flight envelope defined by $1.5 \leq M \leq 3$ & $-20° \leq \alpha \leq 20°$. The requirement refers to robustness against uncertainty in the aerodynamic pitching moment coefficient C_m, with aerodynamic parameters a_m, b_m, c_m & d_m varying independently by up to 25%.
    2) Sufficient attenuation should be achieved at higher frequencies, namely at least -30 dB at 300 rad/s with the loop opened at the actuator input. This is to avoid exciting unmodeled structural dynamics in a potentially real vehicle.
- Performance requirements:
    1) The maximum actuator deflection rate should not exceed 25 deg/s for a 1 g step command which scales proportionally for step commands of different magnitudes.
    2) Step commands are to be tracked within 5% of the reference signal within 0.35 s where possible given the actuator rate saturation.
    3) Maximum overshoot should not exceed 10%

It is worth noting that instead of a maximum actuator deflection, it is logical from a control design point of view to use an actuator rate limit. This is because the actuator rate can be controlled in the linear design process by e.g. imposing maximum frequency domain gains, thus indirectly controlling the maximum deflection angle for a certain state and reference signal combination. Imposing an actual maximum deflection angle directly cannot be taken into account in linear control. In addition, actuator deflection saturation is a non-linear phenomenon that can be taken into account using anti-windup techniques, which is out of the scope of this study.

III. H_∞ Loop Shaping Theory

The method of H_∞ Loop Shaping was introduced in [21] and further described in [4] and [19], consisting of a systematic two-step process. In the first step the open loop performance of the plant is shaped using weighting functions for desirable characteristics in the frequency domain (which in turn affect the time-domain), guided by the desired robustness and performance characteristics. The second step involves robustifying the shaped plant with a controller in terms of normalized coprime factor (NCF) uncertainty [19]. The section provides a brief overview of H_∞ Loop Shaping theory, starting by an introduction of the uncertainty modelling method used, after which the loop shaping procedure is described followed finally by robustification.

A. Normalized Coprime Factor Uncertainty

As described in [4] and [19], a given plant G can be decomposed in its unique (left) normalized coprime factors (NCF):

$$G = M^{-1}N \tag{12}$$

For the purposes of this study, using the right NCF is identical to using the left NCF, since an optimal controller for NCF uncertainty is optimal for both left and right variants [8]. The set of plants G_p perturbed with bounded NCF uncertainty is given to be:

$$G_p = \{(M + \Delta_M)^{-1}(N + \Delta_N)\} : \left\| \begin{bmatrix} \Delta_N & \Delta_M \end{bmatrix} \right\|_\infty < \epsilon\} \tag{13}$$

with Δ_M and Δ_N being unknown stable transfer functions and ϵ their maximum magnitude. Including a controller K the description of Figure 5 holds.

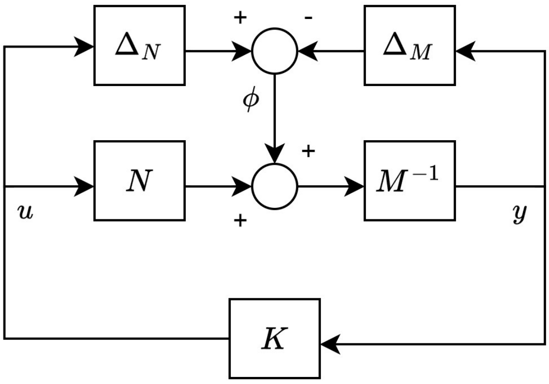


Figure 5. Left normalized coprime factor uncertainty

The best achievable stability margin γ_{min} given the optimal controller K from the set of all stabilizing controllers, the following holds:

$$\gamma_{min} = \left\| \begin{bmatrix} K \\ I \end{bmatrix} (I - GK)^{-1} M^{-1} \right\|_\infty \leq \epsilon_{max}^{-1} \tag{14}$$

The uncertainty modelling method is extremely generic given its ability to model plant-destabilizing uncertainties or e.g. to model uncertainty-induced resonance frequency shifts, unlike multiplicative and additive uncertainty [19]. The uncertainty modelling method forms the basis of H_∞ Loop Shaping.

B. Loop Shaping

The first step in the design process is to shape the open-loop frequency-domain (in terms of the plant's maximum and minimum singular-values $\overline{\sigma}$ & $\underline{\sigma}$) behaviour of a given plant G using two pre- and post-weighting functions, W_1 and W_2 respectively, giving the shaped plant:

$$G_s = W_2 G W_1 \tag{15}$$

As proven in [21], in order to achieve good robustness against NCF uncertainty along with desirable performance behaviour, the following summarized guidelines may be followed [4, 19]:

- High gain at low frequencies for good reference tracking
- Low gain at high frequencies for noise attenuation
- Roll-off rate of approximately -20 dB/dec at the desired crossover region

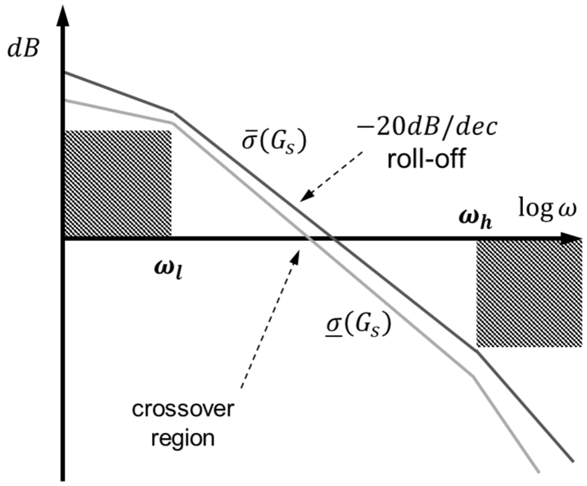


Figure 6. Loop shaping guideline

Figure 6 summarises the guidelines. In addition to these, no hidden dynamics should be present in G_s. The robustness of the final system is heavily dependent on the quality of the loop shape. A noteworthy result from [21] is essentially that the less the robustifying controller needs to "intervene", that is, the more similar the loop shape post-robustification is to the loop shape of the shaped plant G_s, the higher the achieved robustness. In general however, a trade-off needs to be made between desirable time-domain performance and robustness, an example of which is the selection of the crossover frequency: a higher crossover frequency constitutes to better tracking performance but comes at the cost of robustness. The quality of the loop shape can be determined by the alternate expression of Equation 14 [4].

$$\gamma_{min} = \epsilon_{max}^{-1} = \left[1 - \left\|\begin{bmatrix}N & M\end{bmatrix}\right\|_H^2\right]^{-\frac{1}{2}} = (1 + \rho\,(XZ))^{\frac{1}{2}} \tag{16}$$

With $||.||_H$ signifying the Hankel norm, ρ the spectral radius and X and Z the solutions to the Algebraic Ricatti Equations given in Equations 19 and 20 respectively. Depending on the robustification problem to be solved, general guidelines of good loop shapes are given in [21] and [4]. In the case that insufficient robustness is achieved, the process starts over again given the new knowledge.

C. Robustification

In this section the unstructured robustification is covered as a fundamental introduction. Structured robustification is covered in section IV. With loop shaping completed, the shaped plant G_s is used for robustification. For this, the normalized coprime factors of G_s are used in order to find a controller K_∞ that satisfies [4, 21]:

$$\left\|\begin{bmatrix}K_\infty \\ I\end{bmatrix}(I - G_s K_\infty)^{-1} M_s^{-1}\right\|_\infty \leq \gamma \tag{17}$$

for a selected $\gamma > \gamma_{min}$ from Equation 16. In the unstructured case, the selection of the factor of $\frac{\gamma}{\gamma_{min}}$ can be drawn from guidelines (e.g. 10% as suggested in [4] to avoid near-singular matrices), but in practice is dependent on the numerical capabilities of the software used. Figure 7 illustrates the robustification. Equation 17 can also be interpreted as the H_∞ norm from plant input and output to and from controller input and output, i.e. a 4-block system [21].

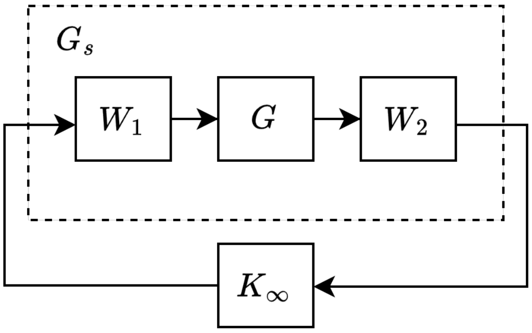


Figure 7. Shaped plant with robustifying controller

As given in [4], the analytical solution of K_∞ to Equation 17 is determined from solution to the following Algebraic Riccati Equations, for a given minimal realization of plant G, for X:

$$\left(A - BS^{-1}D^TC\right)^T X + X\left(A - BS^{-1}D^TC\right) - XBS^{-1}B^TX + C^TR^{-1}C = 0 \tag{18}$$

with $R = I + DD^T$ and $S = I + D^TD$, which for strictly proper plants simplifies to $R = I$ and $S = I$, yielding:

$$\left(A - BD^TC\right)^T X + X\left(A - BD^TC\right) - XBB^TX + C^TC = 0 \tag{19}$$

and for Z:

$$\left(A - BD^TC\right) Z + Z\left(A - BD^TC\right)^T - ZC^TCZ + BB^T = 0 \tag{20}$$

Finally, the controller K_∞ is given by:

$$K_\infty = \left[\begin{array}{c|c} A + BF + \gamma^2(L^T)^{-1}ZC^T(C + DF) & \gamma^2(L^T)^{-1}ZC^T \\ \hline B^TX & -D^T \end{array}\right] \tag{21}$$

where $F = -S^{-1}(D^TC + B^TX)$ and $L = (1 - \gamma^2)I + XZ$. The controller that results has an order equal to that of the shaped plant. This in turn leads to high order controllers, conventionally needing order reduction for implementation, which is important for simplicity of implementation in real systems while also providing transparency and clarity of its functionality [19].

D. Aspects

The main benefits of H_∞ Loop Shaping for this study make it an extremely powerful robust control method [19]:

- Normalized coprime factor uncertainty is a very generic modelling method. Its ability to model destabilizing perturbations and complex dynamic alterations makes it unique compared to other uncertainty modelling methods.
- The uncertainty magnitude and therefore the robustness goal is clear from the start of the design process. No γ-iteration is required, which is conventionally the case for solving H_∞ problems.
- No pole-zero cancellations, avoiding hidden dynamics in the robustification process, as opposed to Mixed SensitivityH_∞ problems.
- Balanced robustness and performance, next to providing a clear method to manage the trade-off between robustness and performance

- Guaranteed simultaneous disk gain and phase margins. Using the stability margin $\gamma = \epsilon^{-1}$ for the Single-Input Single-Output (SISO) case (the SIMO case is half of that of the SISO case):

$$-20\log_{10}\frac{1+\epsilon}{1-\epsilon}\ dB \leq GM \leq 20\log_{10}\frac{1+\epsilon}{1-\epsilon}\ dB \tag{22}$$

$$-2\sin^{-1}\epsilon\ deg \leq PM \leq 2\sin^{-1}\epsilon\ deg \tag{23}$$

H_∞ control in general combined with NCF uncertainty has been criticised for being conservative as opposed to other methods using e.g. the structured singular value μ [4]. However, this can be used to a control designer's advantage if a simple control method is preferred that gives clear guarantees, and therefore inherently providing high robustness against potentially unforeseen or more complex situations.

IV. Flight Control Design

The design procedure involves H_∞ Loop Shaping on the set of linearized plants that cover the entire flight envelope. The aim is to make the set of linear models satisfy the requirements, such that the same behaviour is reflected in the non-linear system while also making sure the process is suitable for automation. The plant linearization and control design is performed for 100 points in the flight envelope (determined by the similarity of the systems and the resulting smooth controller gains), divided in 10 x 10 points for the Mach number and angle of attack. The resulting points are combined to form the gain surfaces to be implemented in the gain-scheduled autopilot. Minor point-surface deviations are smoothed by averaging.

A. Pitch Rate Stabilization

The linearized vehicle system (Airframe and Actuator) is a Single-Input Multiple-Output (SIMO) system as seen in Equations 8, 10 and 11. Due to the nature of the controlled vehicle, the reference signal to be followed is a normal acceleration command, hence a_z is the reference-tracking signal.

In order to gain transparency into the loop-shaping process (with respect to a_z) and in order to provide time-scale separation from the fast dynamics of q and the slower dynamics of tracking signal a_z, the system loop is closed with proportional feedback of q by K_q, yielding a stabilized SISO plant a shown in Figure 8.

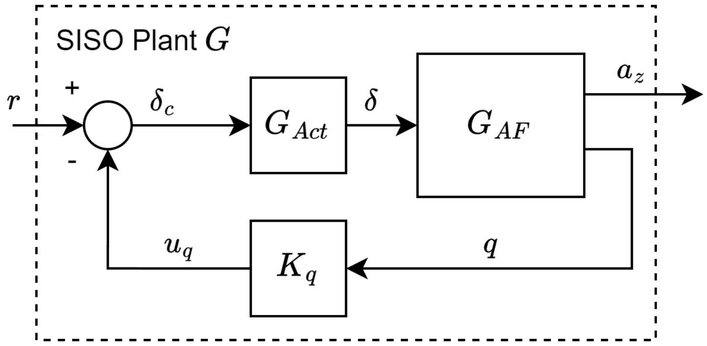


Figure 8. Stabilized plant

The fixed-structure controller is designed by imposing the requirement that the transfer function from $r \longrightarrow q$ has a minimum damping ratio of $\zeta = 0.7$, as this is commonly known to be a balanced value for reference tracking and fast enough damping (which constitutes to robustness). The optimization problem can be solved by using the function `systune()`. The requirement applies for the dominant poles of the system, which originate from the airframe dynamics G_{AF}. However, as `systune()` considers all poles involved equally, a virtual control problem is solved instead which ignores the actuator dynamics in order to isolate the dominant airframe poles as shown in Figure 9.

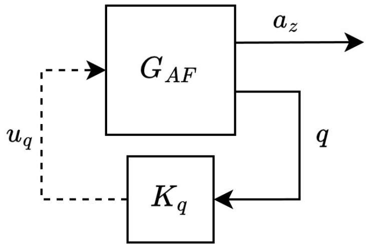


Figure 9. Virtual control problem

The gain surface for the flight envelope is shown in Figure 10 , the minimum damping ratio having been met for all design points. The effect of the unstable region is clearly shown in the jump of the gain surface, yet is required for good handling of the non-linear system as was seen during the non-linear implementation covered later.

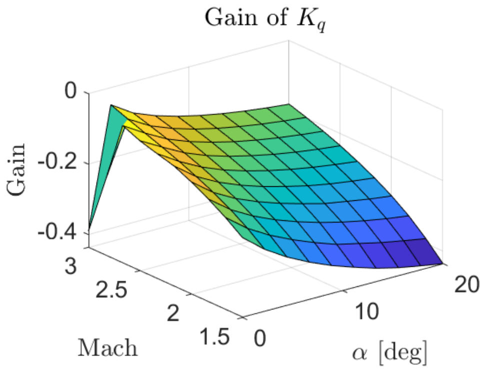


Figure 10. Gain Surface of K_q

The fact that the nominal plant is equated to the SISO stabilized system has implications for the design procedure, namely that the guarantees that the H_∞ Loop Shaping gives applies for the SISO plant only, and thus not the SIMO plant. However, given that the goal of this study is to describe a widely applicable design procedure, starting the design phase in this manner yields transparency and can form the basis for the SIMO plant robustification. General aerospace robustness guidelines can be used to determine the robustness for the SIMO case as well as covered in IV.D. In this case, the conservative nature of H_∞ Loop Shaping provides an indication the the SIMO plant will likely be robust aswell.

B. Normalized Coprime Factor Uncertainty Modelling

The aerodynamic uncertainty introduced on the vehicle as described in section II.C is to be translated into its NCF uncertainty description. This is done by taking the worst-case combination of the parametric uncertainty in terms of NCF uncertainty. In an iterative method, the set of perturbed plants are compared in terms of their coprime factors ($G_p = M_p^{-1}N_p$) to the nominal plant's coprime factors ($G_{nom} = M_{nom}^{-1}N_{nom}$):

$$\begin{aligned}\Delta_M &= M_p - M_{nom}\\ \Delta_N &= N_p - N_{nom}\end{aligned} \tag{24}$$

As can be derived from section II.B and [1], the uncertainty affects the state dynamics of derivatives M_α and M_δ of Equation 10. Hence, a two-dimensional iteration is used in terms of the minimum and maximum values of the derivatives, which in turn is determined using the function `wcgain()`. The worst-case NCF uncertainty magnitude ϵ is determined:

$$||\Delta(s)||_\infty = \left|\left|\begin{bmatrix}\Delta_N & -\Delta_M\end{bmatrix}\right|\right|_\infty < \epsilon \tag{25}$$

Note that the nominal plant used is the closed-loop SISO system as covered in section IV.A. The magnitude of epsilon is given for the flight envelope's local plant linearizations in Figure 11.

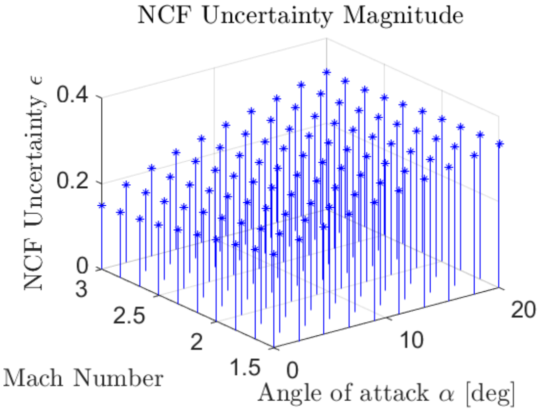


Figure 11. NCF Uncertainty magnitude for flight envelope

The magnitude of uncertainty is relatively large, the maximum point occurring at $M = 1.5$ & $\alpha = 20°$ with $\epsilon \approx 0.34$.

C. Loop Shaping

The SISO plant of Figure 8 is used for loop shaping, implying that the tracking signal a_z is shaped for performance and robustness. Since SISO is used, W_1 and W_2 may be used interchangeably and W_1 is chosen to be the main loop shaping function. Commonly used functions for W_1 are of a lead-lag filter form [8, 21, 24] which gives freedom to alter the crossover slope as well as the crossover frequency itself while being able to provide high frequency attenuation. An important function of W_1 is also the integrating action on the tracked signal-error.

A trade-off between simplicity of the loop shaping process and the loop shape quality becomes apparent, depending on the controlled system. For effective automation, i.e. shifting the robustification process to K_∞ as much as possible, it was chosen to make W_1 of a pure integrator form shown in Equation 26 for which only the desired crossover frequency needs to be selected,

$$W_1 = \frac{K_i}{s} \tag{26}$$

The gain K_i allows for the choice of crossover frequency, for which a simple bisection search-algorithm is used shown in Algorithm 1 in the appendix. In order to avoid gain-surface chattering which may induce issues in the non-linear implementation (such as unexpected fluctuation behaviours), the crossover frequency is chosen to be the same for all linear systems. The exact selection of crossover frequency is dependent on the following considerations:

- The crossover frequency for which the system remains robust, i.e. a good quality of the loop shape

- The crossover frequency for which (most of) the linear systems have at least -30 dB attenuation from section II.C (partially dependent on the robustifying controller K_∞)
- The frequency range for which the crossover slopes of most linear systems approaches -20 dB/dec
- The resulting K_∞ controller should not exceed 25 deg/s for disturbances at input and output, while giving decent disturbance-rejection speeds

Note that for the last consideration, no explicit requirement is defined, but it is taken into account that fast disturbance rejection speeds are preferred. Also note that reference tracking performance is not considered here but covered in section IV.E. The aim of this Loop Shaping method is to minimize manual interventions of the mentioned considerations by only having to select the crossover frequency, albeit that a designer becomes more dependent on the controlled system. Since satisfying the design requirements for as many points as possible requires design iterations (i.e. trial of different crossover frequencies), the automation accommodated by the simplicity of W_1 is deemed an important benefit.

From the design iterations performed, it was concluded that a crossover frequency of 3 rad/s offered a balanced trade-off between the design aspects. The resulting bode plots of all linear points being shown in Figure 12.

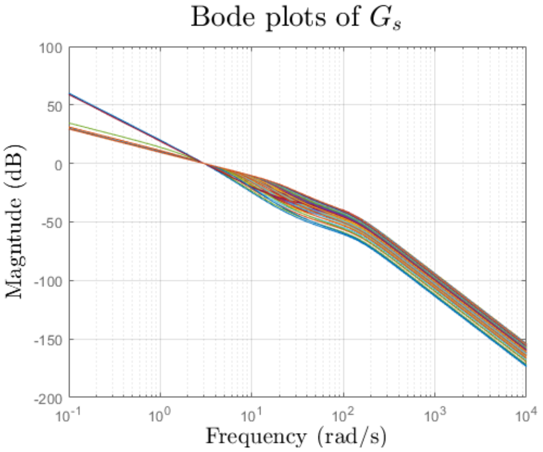


Figure 12. Loop shapes of linear systems

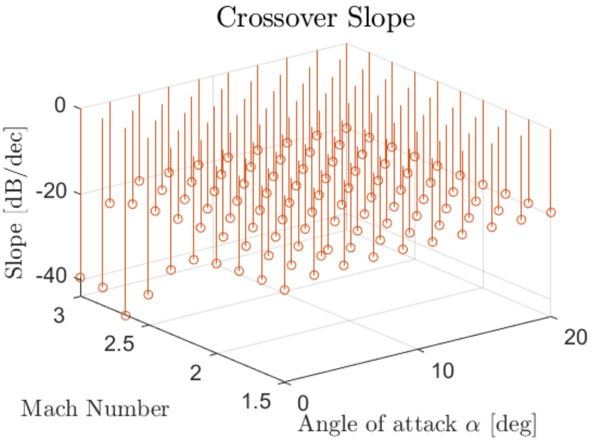


Figure 13. Achieved crossover slopes of the open loop of G_s

The majority of the linear points share a similar crossover slope for the selected crossover frequency, with the exception of the linear points of the unstable region as shown in Figure 13. Figure 13 thus indicates that most points are sufficiently close to the desired roll-off rate, with the exception of the unstable region for which robustification needs to compensate. The best achievable stability margin γ per linear point is shown in Figure 14.

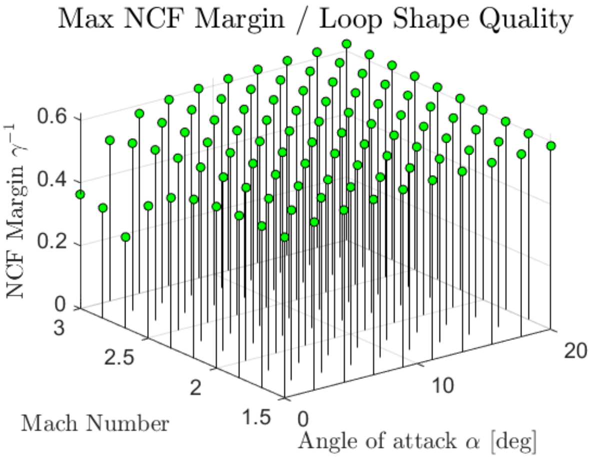


Figure 14. Loop shape robustness quality

The gain surface of K_i from Equation 26 is shown in figure 15, its smoothness indicating suitability for direct implementation.

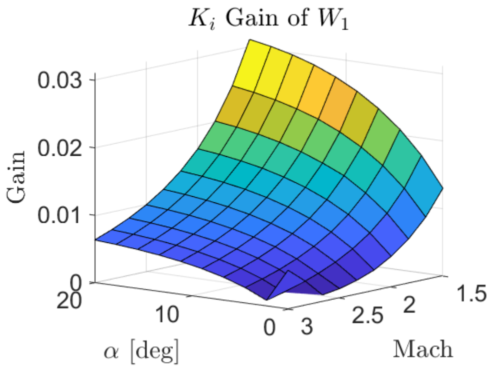


Figure 15. Gain surface of K_i for W_1

On a general applicability level, the loop-shaping results indicate that the method gives desirable results for the majority of design points despite its simplicity. The limitation however is that there may be exceptions in terms local controller quality for points in the unstable region, which in turn depends on the severity of the lower quality and the context of the control problem.

D. Robustification

It is important for the implementation of the gain-scheduled autopilot to have the least number of gains that is achievable in order to have desirable and predicable behaviour of the non-linear system [19]. For this reason, a structure is imposed on K_∞ of the lead-lag form shown in Equation 27.

$$K_\infty = K_s \frac{s+z}{s+p} \tag{27}$$

with K_s the controller gain, z the zero and p the pole. The structure is selected such that the freedom for effective H_∞ Loop Shaping is maintained as discussed in section IV.C. It is also the lowest order controller that has the freedom to robustify the system other than lowering its gain (by using its pole and zero), as opposed to a 0^{th} order (proportional) controller. Since the latter is thus only able to lower the gain of the controller for robustification, this comes directly at the cost time-domain performance. Another benefit of the combination of a simple W_1 in combination with the lead-lag controller being subject to robustification, is that there is no risk of pole-zero cancellation (which induces hidden dynamics) [19], which would be the case if the zero and pole had to be manually selected (requiring pole-zero analysis post-design).

Imposing a structure on the controller yields a non-convex optimization problem for which conventional optimization methods cannot be used [19, 20]. In [20] the issue is resolved using non-smooth optimization techniques embedded in the function `systune()`. The H_∞ problem posed from Equation 17 can be re-written as shown in Equation 28, with S_s the sensitivity function of the shaped plant[4, 21].

$$\left\| \begin{bmatrix} K_\infty \\ I \end{bmatrix} (I - G_s K_\infty)^{-1} M_s^{-1} \right\|_\infty \equiv \left\| \begin{bmatrix} K_\infty & K_\infty S_s G_s \\ S_s & S_s G_s \end{bmatrix} \right\|_\infty \equiv \left\| \begin{bmatrix} d_i \\ d_o \end{bmatrix} \rightarrow \begin{bmatrix} u \\ y \end{bmatrix} \right\|_\infty \tag{28}$$

Equation 28 essentially minimizes the H_∞ norm of the shaped plant G_s at its input and output as shown in Figure 16.

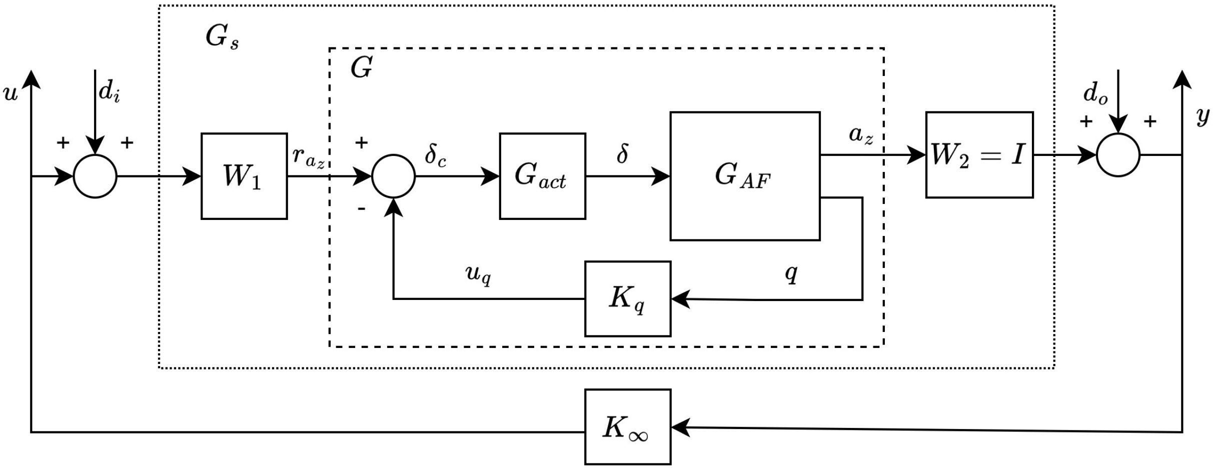


Figure 16. Equivalent H_∞ problem

The achieved robustness margins across the flight envelope are illustrated in Figure 17 together with the NCF uncertainty magnitude. For comparison, the robustness of the unstructured controller as derived in section III.C is additionally shown ($\gamma = 1.01\gamma_{min}$ selected for numerical stability).

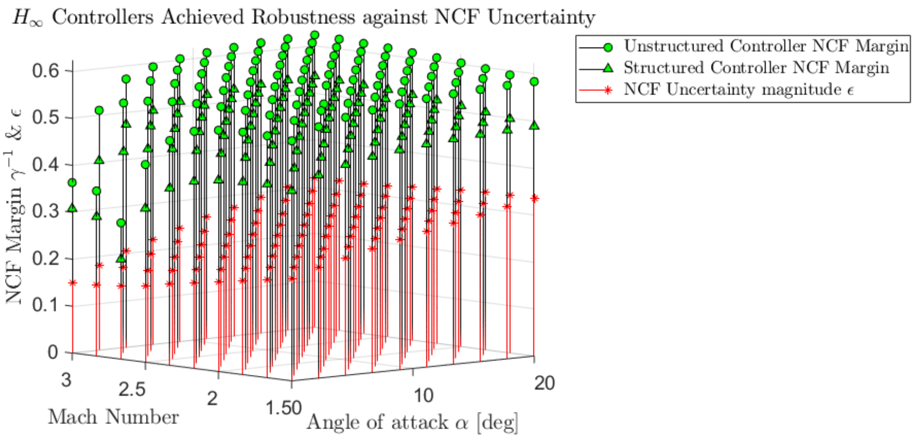


Figure 17. Achieved robustness of structured & unstructured controllers

Figure 17 shows that the system is robust against NCF uncertainty for the SISO system. This can be concluded as the values of the NCF uncertainty margins of the structured and unstructured controllers ($\gamma^{-1}_{str/uns}$) of are larger than the NCF magnitude ϵ. As can be expected, the lower-order controller gives lower robustness margins than the unstructured controller, yet high robustness in absolute terms along with being significantly less challenging to implement. The general robustness is lower for small angles of attack near the stable-unstable transition at approximately Mach 2.6, yet sufficient. This is a direct effect of the loop shaping quality in the unstable region and was to be expected. However, given that this occurs at small angles of attack, it may be expected that the controller in the non-linear implementation "catches" a potential near-instability as the angle of attack increases into the more robust regions for higher angles of attack for all Mach numbers.

The figures below show the effects of the structured and unstructured controllers on the loop shapes, yielding similar results to each other. The crossover frequency is minimally altered for both controllers. The most profound alterations of the controllers on the initial loop shapes are for the roll-off rates of certain systems and the high frequency attenuation, the latter of which both controllers show the largest difference from each other.

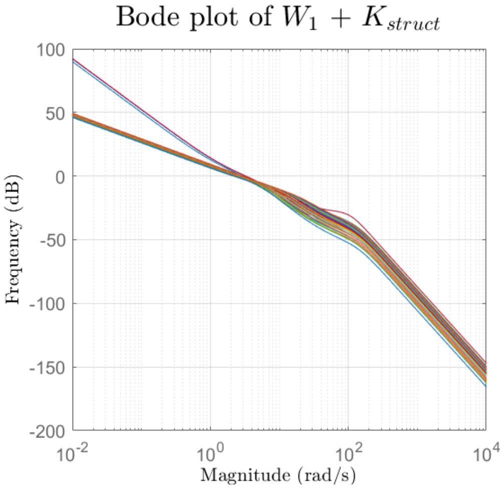


Figure 18. Loop shapes of W_1 combined with the structured robust controller

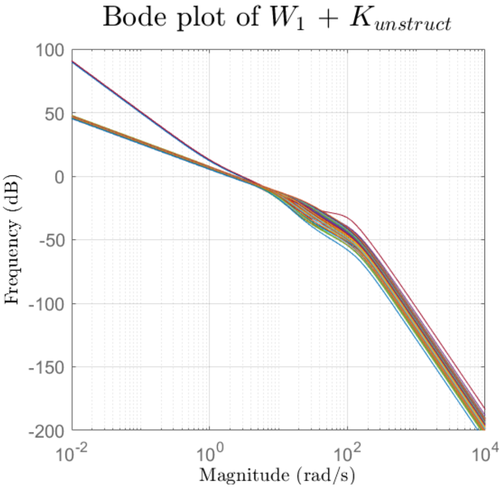


Figure 19. Loop shapes of W_1 combined with the unstructured robust controller

The figures in turn indicate by similarity the effectiveness of the structured robustification compared to the "ideal case" unstructured controller. In addition, the effect the structured robust controller has on the roll-of rates of the design points is shown in Figure 20. The robust controller clearly "pushes" the roll-off rate to its desired value of $-20\ dB/dec$, serving as an additional verification of the robustification procedure.

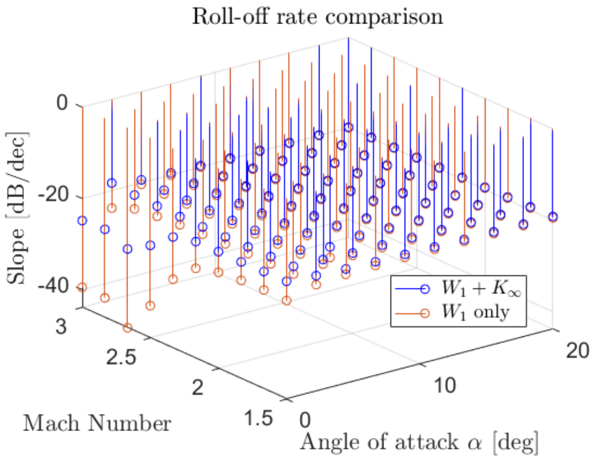


Figure 20. Roll-off rate comparison of loop shapes pre- and post structured robustification

The structured controller is placed in the feedback path in order to avoid the reference signal directly exciting the controller's dynamics [4], requiring reference scaling by the scheduled gain K_{dc} as shown in Figure 21.

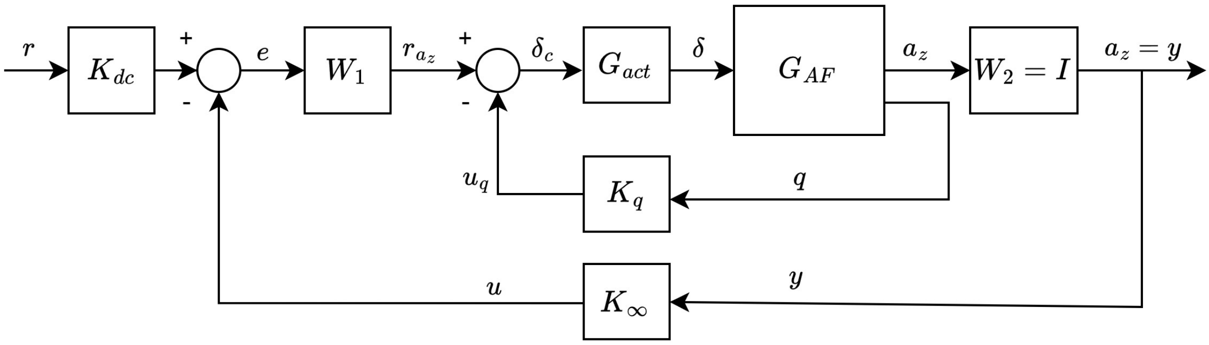


Figure 21. K_∞ implementation

The respective gains of K_∞ are shown in Figure 22. The gain surface of K_{dc} is given in Figure 46 in the appendix. The smoothness of the gains indicate consistent results for the robustification across all design points (i.e. a verification of the procedure and robustification) as well as sufficient smoothness for direct implementation. Note that Num s / Gain K_∞ required slight normalization along the $\alpha = 0$ line by averaging. Points like these can likely be avoided by using more initial points within the function `systune()`, though may also be an effect of the unstable region.

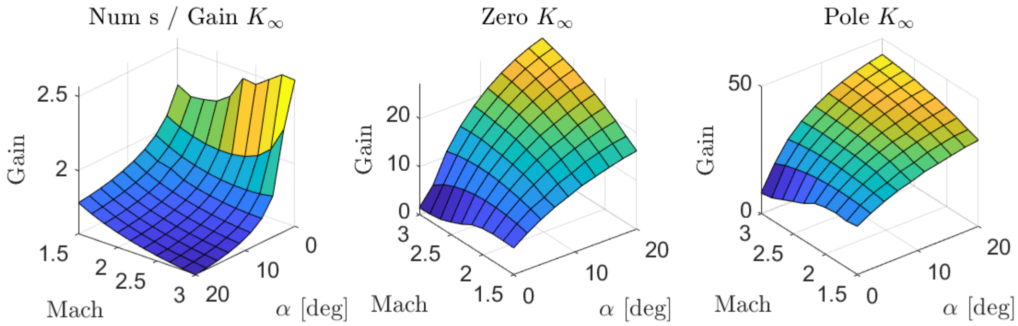


Figure 22. K_∞ gains

In order to verify the robustness, both the SISO case and the full system in the SIMO case need to considered. For the SISO case, the gain and phase margins are guaranteed (as mentioned in section III.D). The respective disk gain and phase margins are given in fFigure 23. The disk gain and phase margins are used for insight in the simultaneous gain and phase variations as opposed to gain / phase variations only. As shown in Figure 23 and a-priori guaranteed from Figure 17, the gain and phase margins exceed the required margins by a substantial amount, guaranteeing local robustness against the NCF uncertainty for the SISO plant. To verify the robustness for the full SIMO system, the loops are opened at both outputs of the plant and at the actuator input (loop-at-a-time), for which the disk gain and phase margins are shown in Figure 24.

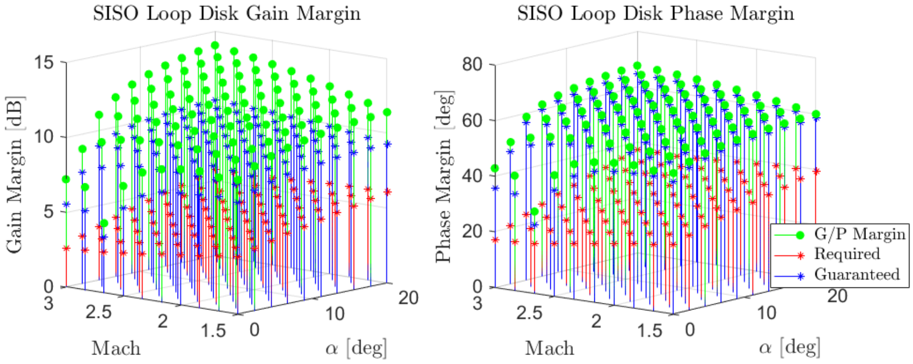


Figure 23. SISO disk gain and phase margins

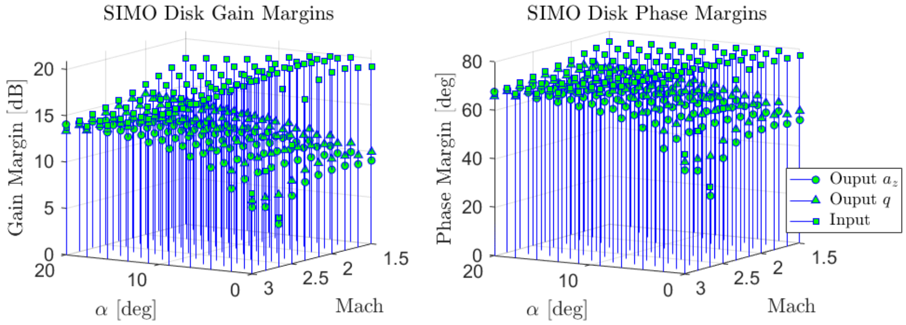


Figure 24. SIMO loop-at-a-time disk gain and phase margins

Although a SIMO design procedure is required in order to have guarantees of robustness against the

NCF uncertainty, Figure 24 show significant amounts of gain and phase margins coherent with the common loop-at-a-time aerospace requirement of 6 *dB* of gain margin and 38° of phase margin. The requirement holds for the entire flight envelope, with the exception of $M \approx 2.6$ and $\alpha = 0$, which is not deemed an issue as explained earlier for Figure 17, despite it being a limitation of the procedure. For additional insight into the stability performance of the SIMO systems, the Nichols chart for the input and outputs with corresponding Nichols exclusion zones for all design points (with the lower-robustness point excluded) is shown in Figure 25.

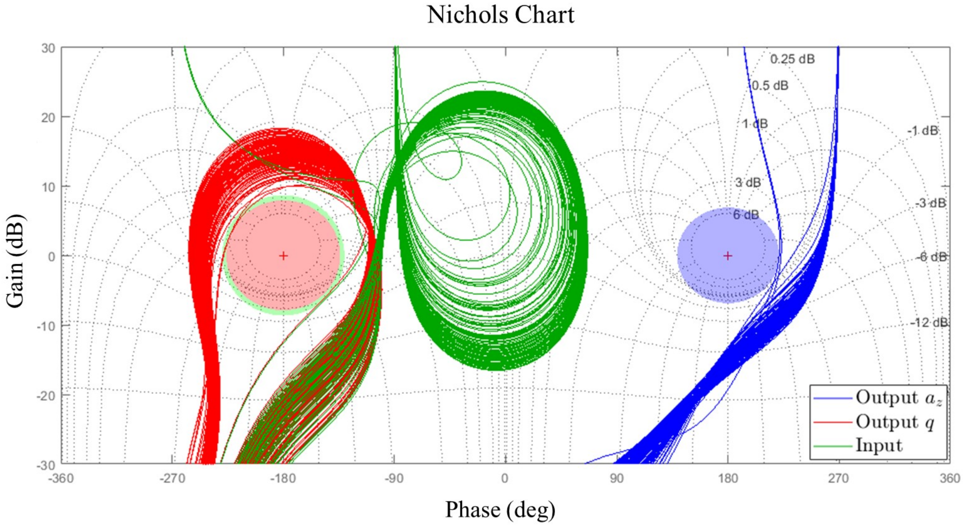


Figure 25. Nichols chart for SIMO system input and outputs

With the loop opened at the actuator input, the frequency for which the attenuating gain of −30 *dB* is first crossed is shown in Figure 26. The figure shows that the attenuation requirement is met for the whole flight envelope, with the exception occurring at $M = 3$ and $\alpha = 0$. As only one point violates the requirement along with this issue also being addressed in [8], the attenuation quality is deemed sufficient.

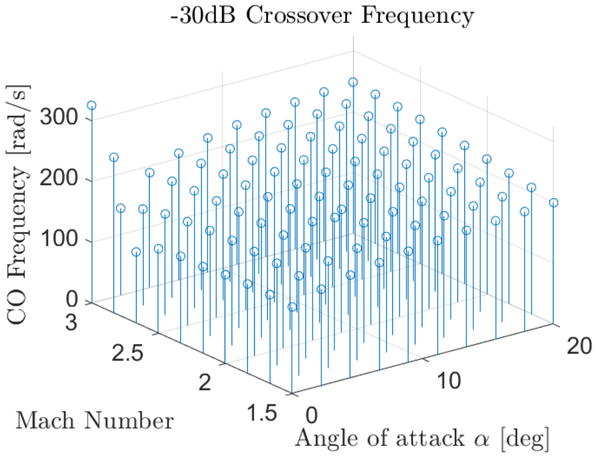


Figure 26. High frequency attenuation verification

The input-output disturbance-rejection performance is shown in Figures 47 and 48 in the appendix. The output disturbance (e.g. a wind gust effect) is translated into a change in the angle of attack, the most directly relevant effect of which is the change in the normal acceleration, which is deemed the most interesting signal due to its reference-tracking function. For this reason, an a_z output disturbance of 1 g is deemed representative. Additionally an input disturbance is introduced which showed good results for $\delta = 0.01\ rad$ ($\approx 0.57°$), which induces a maximum a_z disturbance of approximately 0.5 g at $M = 3$ and $\alpha = 0$. For other points in the flight envelope there is room for larger disturbance sizes.

E. Feed-Forward controller

The robustness and disturbance-rejection properties of the controller are discussed thus far. In order to meet the time-domain (tracking) performance requirements of the system, a separate feed-forward controller is introduced and designed in order to create a 2 degree-of-freedom (2-DOF) controller. Unlike the method where the 2-DOF controller is taken directly into the H_∞ process (as shown in [4] and applied in [24]) the approach taken here treats the robust optimization and time-domain improvement separately. This was chosen in order to asses the controller quality when keeping a clear distinction between agility and robustness, but it is recommended to explore the latter method in future similar studies.

The controller is constructed such that the system follows a reference model that shows similar non-minimum-phase behaviour. and satisfies the performance requirements For sufficient response speed, the output is injected into the inner loop dynamics of the system as shown in Figure 27.

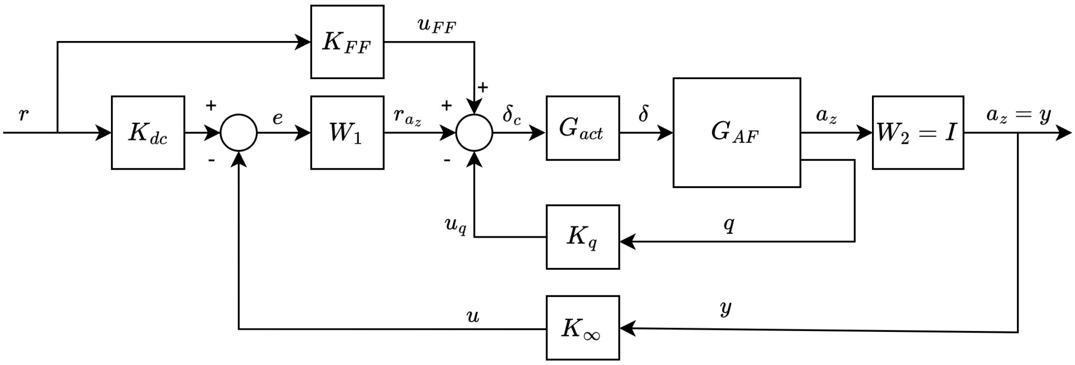


Figure 27. Complete linear autopilot with feed-forward controller

Since the feed-forward controller is not part of the feedback system, the robustness and stability of the system is unaffected. The reference model is generated by imposing a desired overshoot, settling time, non-minimum-phase zeros (from the respective linear vehicle model), damping ratio and natural frequency as shown in Table 2 in the appendix. The values are then used to find a 2^{nd}-order system with "optimal" ζ and ω_n values that best satisfy the overshoot and settling time requirements (i.e. an optimization problem that can be solved with function `fminunc()`). The table values are determined by trial-error based on the combination that best represents all linear systems across the flight envelope, while satisfying the requirements the best for the derived feed-forward controller.

The controller is imposed to be of 2^{nd} order and solved for step model-following using `systune()`. Given no extra constraints, the resulting feed-forward controller satisfies the time domain requirements for all linear points, but severely violates the actuator rate constraint for lower Mach numbers (as control authority is decreased with lower air speeds). This required the additional constraint of a frequency domain maximum gain limit from the reference signal to the actuator deflection rate. As control authority (and therefore actuator rate saturation) varied substantially across the flight envelope, the gain is made a function of the Mach number as shown in Figure .49 in the appendix. A typical step response is shown in Figure 28.

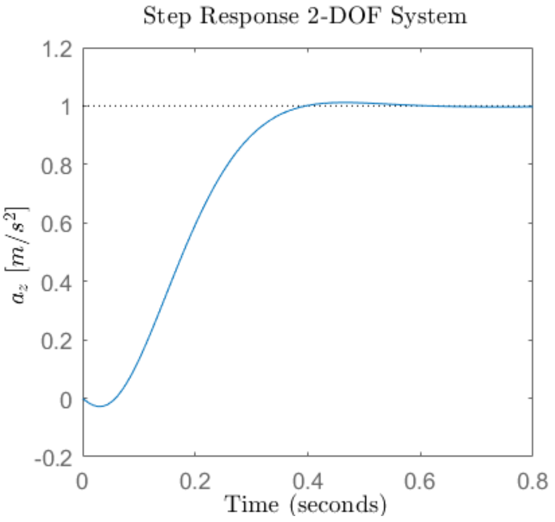


Figure 28. Typical step response ($M = 3$, $\alpha = 10$)

The achieved settling times and overshoots are shown in Figure 29.

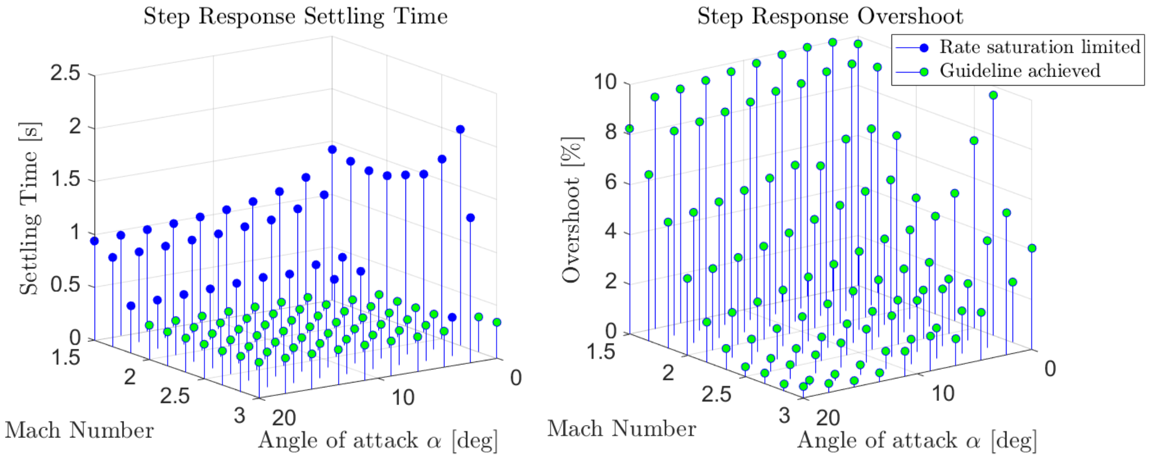


Figure 29. Settling time and overshoot across the flight envelope

The overshoot requirement is met for the entire flight envelope. The step response settling time requirement is met for most of the flight envelope, but is higher for small Mach numbers and small angles of attack due to the actuator rate saturation. The larger values for the settling time are due to the last phase of settling taking a longer time due to overshoot and undershoot, yet closely represent the other points where the settling time requirement is achieved. The respective actuator deflection (rate) is shown in Figure 30, indicating the actuator rate saturation requirement is met.

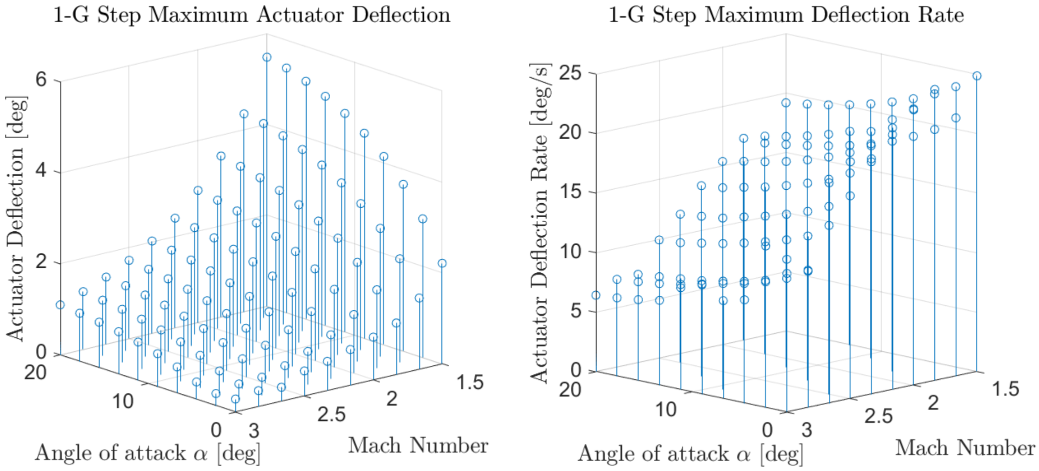


Figure 30. Actuator maximum deflections & deflection rates for a 1 g step reference

The respective smoothed gain surfaces of the 2^{nd} order feed-forward controller is shown in Figure 50 in the appendix. It was chosen to implement the feed-forward controller on an averaging basis, as described in section V.A.1.

V. Non-Linear Results

The full non-linear simulations serve the purpose of testing the uncertainty handling (stability & handling similarity of random uncertainty realizations) and performance qualities of the gain-scheduled autopilot. Since linear short-period dynamic models were used for the controllers, an unknown confidence region around each linearized point is present [23] that was compensated by the relatively numerous linearized models / grid points. This is especially relevant due to the fast dynamics that are involved. Therefore, a number of randomized-parameter simulations is performed to verify that the system performs as desired. The step reference signals from [1] and [22] are used, along with a more realistic scenario with simple guidance incorporated. The latter is to intuitively asses the uncertainty handling performance of the autopilot with a more practical example.

A. Implementation

The implementation of the gain-scheduled controller required special points of attention and minor manual corrections due to the neglected dynamics and other non-linear effects which are hard to predict beforehand.

1. Feed-forward controller

The feed-forward controller performed poorly in its gain-scheduled form largely due to the relatively high order and due to some regions of the gain surface not being smooth enough as shown in Figure 50. However, it is clear that the gains are smooth and highly similar for large portions of the flight envelope with a clear Mach number-based distinction. It was therefore chosen to average the values, giving the transfer function shown in Equation 29 (the respective bode plot of K_{FF} is given in Figure 51 in the appendix).

$$K_{FF} = \frac{7.7s^2 + 121.1s + 5686.1}{s^2 + 41.1s + 529.3} 10^{-4} \tag{29}$$

Additionally, the transfer function is made semi-dynamic based on minor manual tuning as a function of the Mach number, the result of which can be found in Figure 52 in the appendix. The implementation scheme is shown in Figure 31.

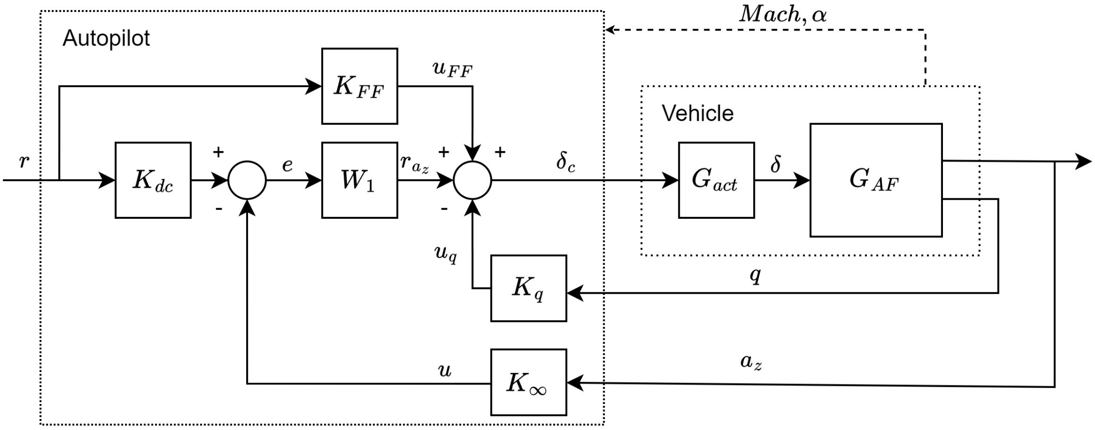


Figure 31. Base autopilot implementation

2. Velocity-based implementation

The initial non-linear results showed high overshoots and steady-state errors for high reference values. A method to improve this behaviour is to use the velocity-based implementation introduced in [25]. This involves differentiating the secondary (non-reference-tracking) state (in this case pitch rate q) and using the integrator of the state tracking controller to integrate this value again. The velocity-based autopilot is implemented by approximating $\dot{q}$ with the transfer function $\frac{s}{\epsilon s+1}$ with $\epsilon << 1$ as shown in Figure 32.

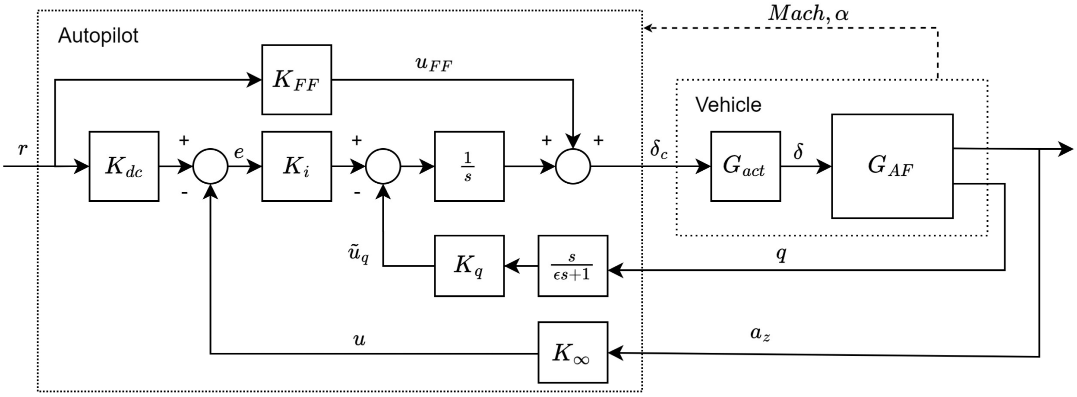


Figure 32. Velocity-based implementation

The integrator of W_1 is moved forward, leaving K_i behind. The effect of the velocity-based implementation is shown in Figure 33.

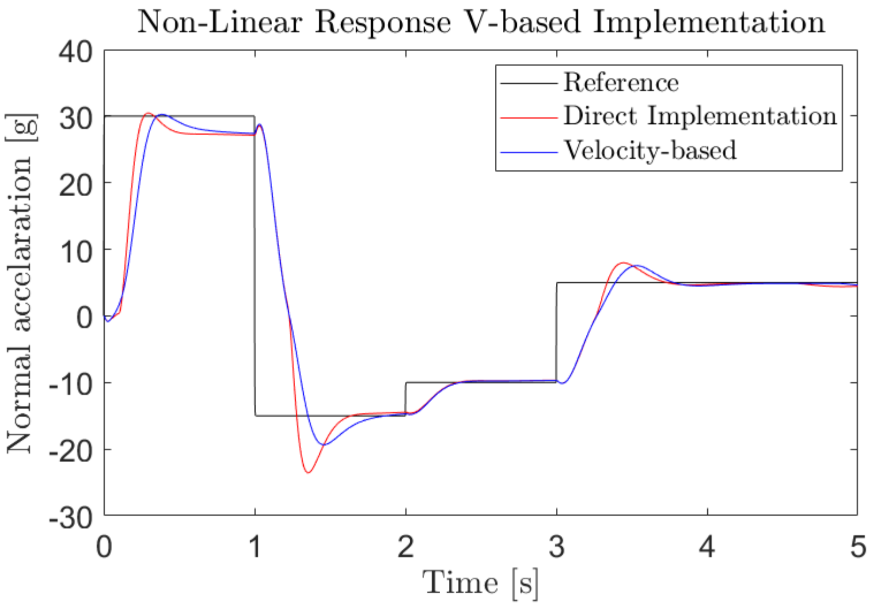


Figure 33. Effect of velocity-based implementation

The overshoot of the system is improved by using the velocity-based implementation. The steady-state error is improved in section V.A.3.

3. Mach correction

Neglected dynamics and hidden coupling terms are common problems with gain-scheduling and covered in literature in e.g. [26]. The issue posed in Figure 33 is however resolved by a simple observation.

The steady-state error occurs more severely for high reference values. The effect of high angles of attack and therefore force generation causes the Mach number to decrease more rapidly than for smaller reference values (Equation 6). The observation is made that the actuator "recognises" the steady-state error, but compensates for this too slowly, likely due to the relatively low crossover frequency chosen during loop-shaping. From this it is theorized that the Mach number rate has an influence on the steady-state error, since it is dropped as a state in the linearized (short-period) dynamics. Assuming this is the case, Equation 30 holds:

$$e_{ss} = rK_{dc}(\alpha, M) - a_z K_\infty(\alpha, M) + f(\dot{M}) \tag{30}$$

If it is assumed that $f(\dot{M}) = c * \dot{M}(t)$ & $\dot{M} \neq 0$, rewriting yields:

$$c = \frac{rK_{dc} - a_z K_\infty}{\dot{M}} \tag{31}$$

for which $c \approx 90$ for $r > 20$ and $c(t) \approx 30$ for $r < 20$ while remaining relatively constant. In essence, this means a Mach number rate-based correction can be added to the scaled reference signal, which can be scheduled and has no effect on robustness. The adjusted autopilot system is shown in Figure 34.

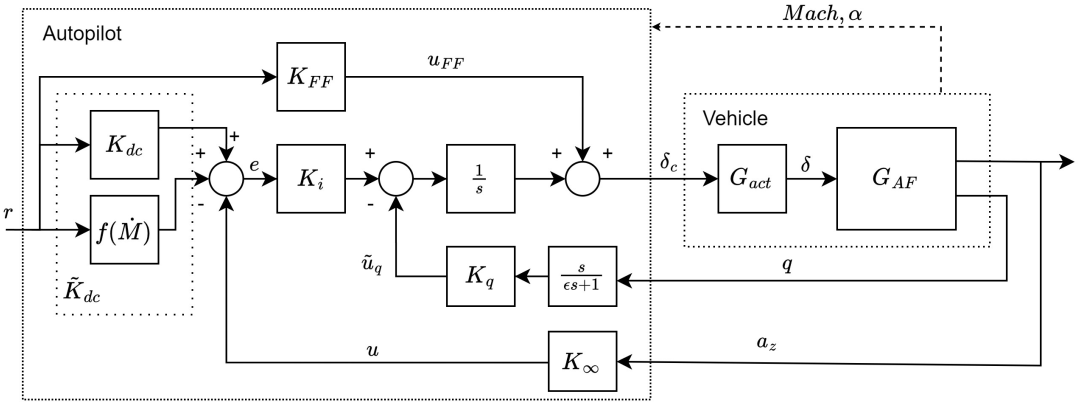


Figure 34. Neglected dynamics correction

The effect of this neglected dynamics correction using only the 2 scheduled values is shown in Figure 35.

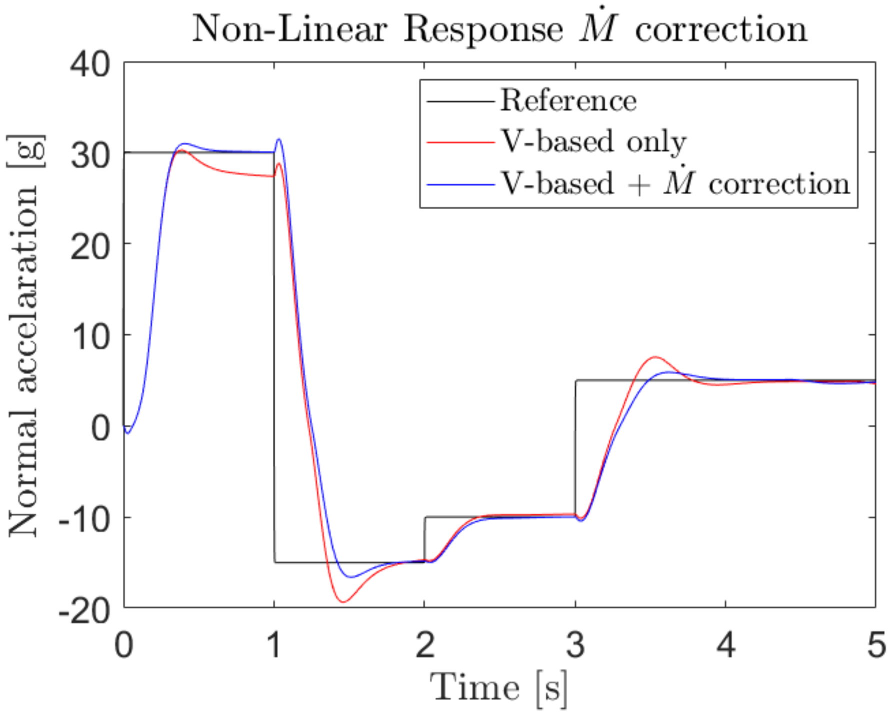


Figure 35. Neglected dynamics correction comparison

It is clear that there is a substantial improvement in both the steady-state error and the overshoot.

B. Simulation Results

1. Staircase reference

The following simulations use a "staircase" reference signal from [22] and an "extended staircase" from [1]. The reference signal profiles allow for representative flight envelope coverage. The time-domain performance is compared against an ideal 2^{nd} order reference model determined as discussed in IV.E while using the values of Table 2, with this time the desired overshoot being 4.9% and the settling time being 0.349 s. The results are shown in Figure 36, from which it is clear that the reference model is closely followed by the system (even outperforming the reference model in the first step), while gradually becoming slower due to reduced control authority by the reduced airspeed.

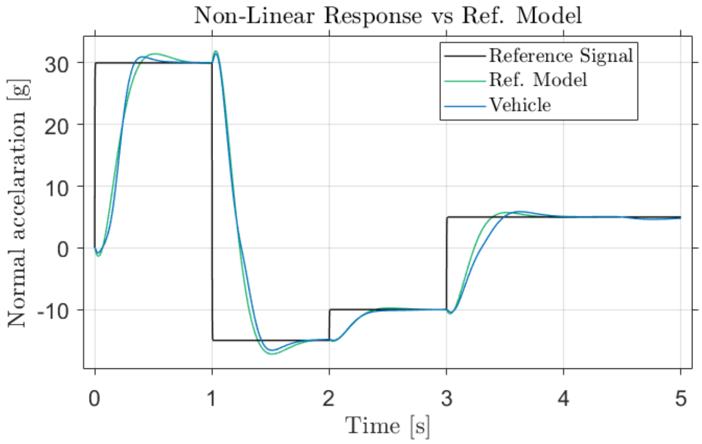


Figure 36. Reference model comparison

The respective actuator dynamics are shown in the appendix in Figure 53. For the actuator rate saturation requirement, the normalized actuator deflections shown in Figure 38 indicate that the requirement is satisfied by a large margin (max 10 deg/s versus the 25 deg/s requirement). This is deemed a benefit as well as a drawback, as better performance could be demanded from the system at the cost of (feasible) higher actuator rates.

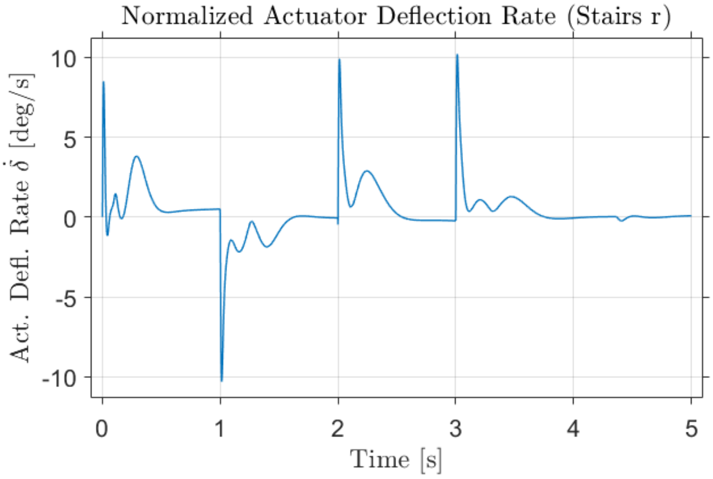


Figure 37. Normalized actuator deflection for the Stairs reference

The behaviour is consistent across the wider flight envelope, for which the extended stairs reference signal is used in Figure 38. The system closely represents the reference model in most cases and gradually gets slower due to the actuator rate saturation requirement.

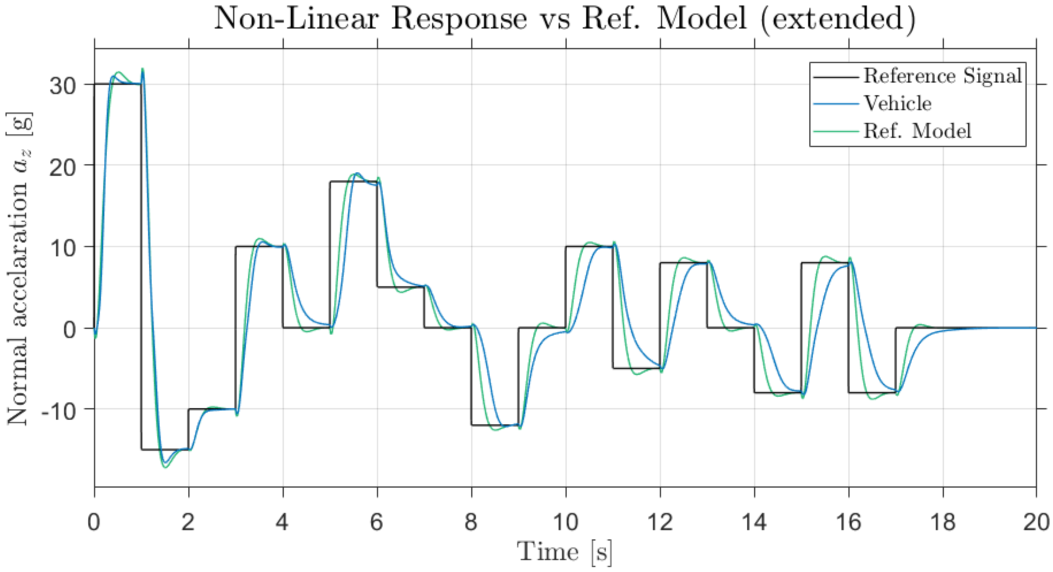


Figure 38. Reference model comparison extended stairs reference signal

The respective actuator dynamics and covered Mach number - angle of attack domain are shown in Figures 54 and 55 respectively in the appendix.

In order to asses the autopilot's uncertainty handling, the system is tested by performing 100 iterations with randomized combinations of the uncertain aerodynamic parameters as mentioned in II.C. The responses are shown in Figure 39 and the respective actuator deflections in Figure 40.

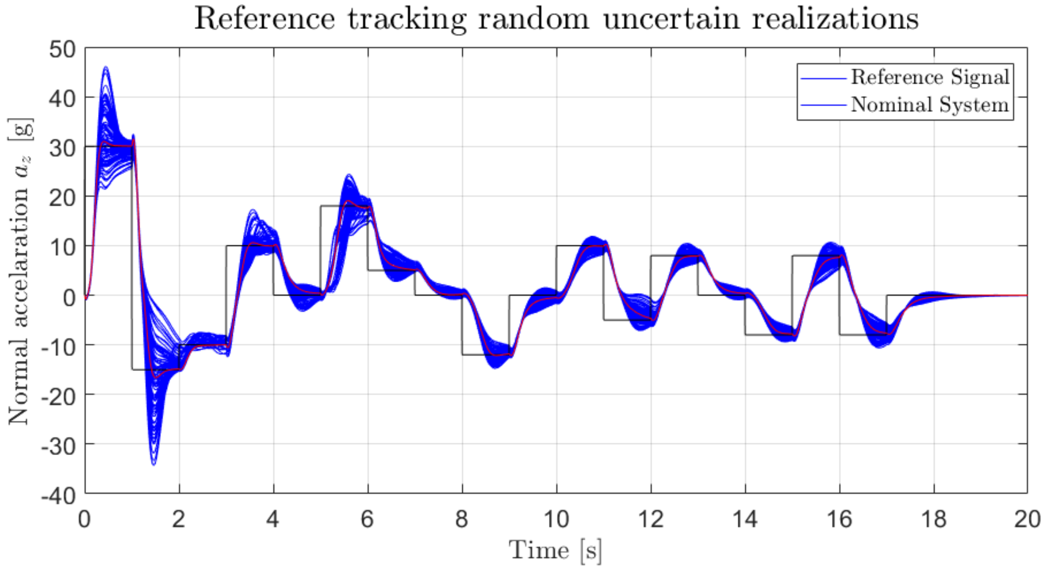


Figure 39. Non-linear model robustness, 100 random uncertain realizations

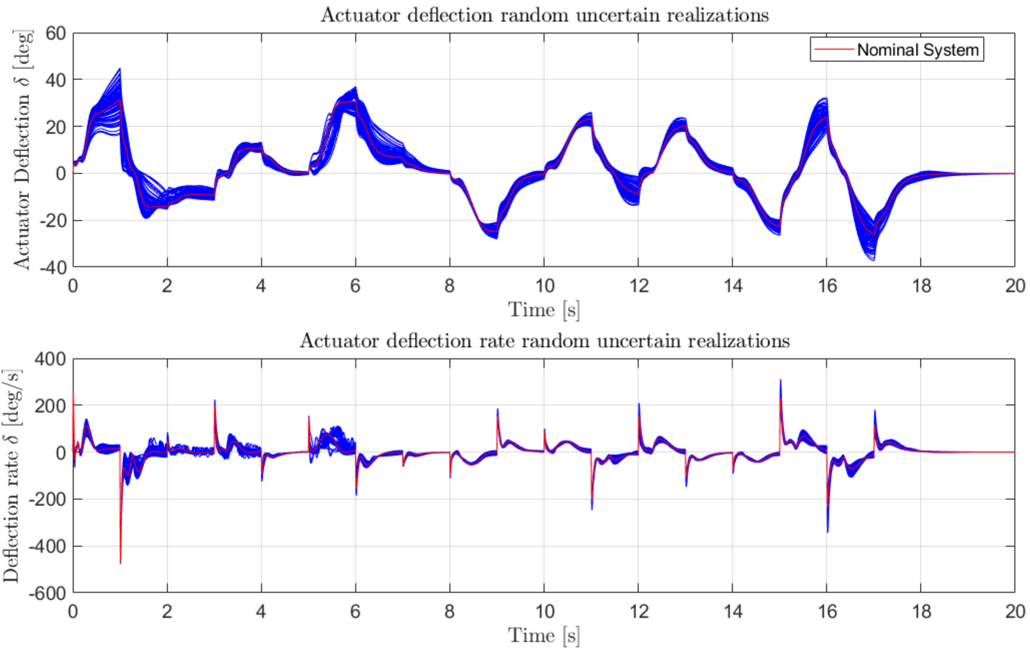


Figure 40. Actuator deflections of random uncertain realizations

The Figures show the autopilot behaviour under uncertainty, its impact becoming less profound as the velocity decreases, yet always being consistently controlled for the relatively large magnitude of the uncertainty imposed, implying desirable uncertainty-handling qualities. For instance, no sinusoidal fluctuations are taking place, which often indicate near-instability. Initially, a high overshoot is taking place for certain realizations due to the high control authority combined with high uncertainty. However, most realizations show similar behaviour to the agile nominal system while providing stability for all of the realizations. Note that to fully verify robustness of the non-linear system, methods like Monte Carlo simulations can be used. The effect of the deviations from the nominal system in a practical scenario is discussed and contextualised in section V.B.2.

The system shows similar disturbance-rejection properties to the linear design points even with uncertainty present, a realization of which is shown in Figure 56 in the appendix.

2. Guidance scenario

The autopilot follows commands, i.e. the reference normal acceleration signal, for navigation from the guidance system. Given the agility of the vehicle, a realistic application of the system would be a guidance system that gives aggressive maneuvering commands to which the vehicle needs to respond rapidly. An example of such situations is when the guidance system attempts to take the vehicle as close as possible to a moving point in space. The conversion of the body reference frame to the earth reference frame is described in Equations 32 and 33 [27, 28].

$$\begin{pmatrix}\dot{u}\\ \dot{w}\end{pmatrix} = \begin{pmatrix}a_x - qw - g\sin\theta\\ a_z + qu + g\sin\theta\end{pmatrix} \tag{32}$$

$$\begin{pmatrix}\dot{X}_e\\ \dot{Z}_e\end{pmatrix} = \begin{pmatrix}u\cos\theta + w\sin\theta\\ -u\sin\theta + w\cos\theta\end{pmatrix} \tag{33}$$

The guidance system uses proportional navigation as described in [29]. The guidance law essentially uses a constant factor between the rate of the angle between the vehicle and the point in space $\dot{\lambda}$ and the rate of the vehicle flight path angle $\dot{\gamma}$, i.e. $\dot{\gamma} = N\dot{\lambda}$. From the relation $a_z = V\dot{\gamma}$, the reference signal is determined by Equation 34.

$$r = a_{z_c} = NV\dot{\lambda} \tag{34}$$

An approximation of $\dot{\lambda}$ is used and $N = 4$. The guidance system is activated after 1 second and the simulation halts when the rate of the relative distance becomes positive: $\dot{R} > 0$. The full system is shown in Figure 41.

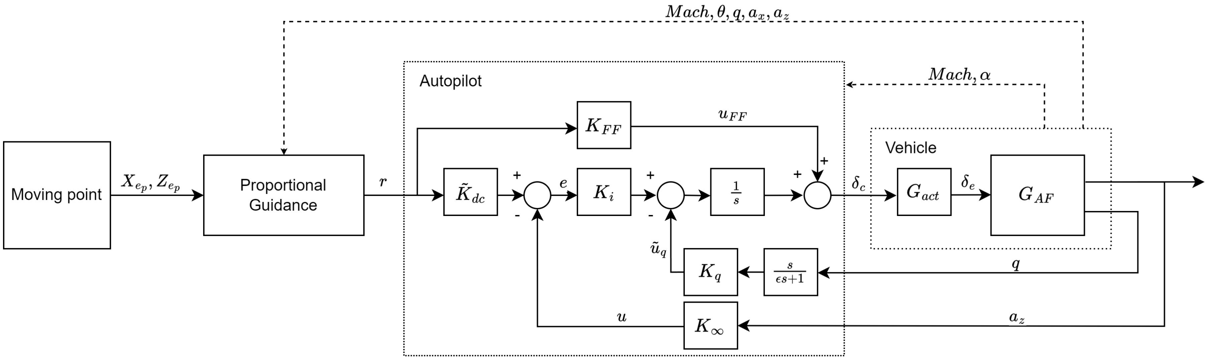


Figure 41. Full system with guidance

The vehicle is initialized at an altitude of $Z_e = 6000\ m$ and a horizontal distance of $X_e = 0\ m$, with the moving point in space being initialized at $Z_{e_p} = 6500\ m$ and $X_{e_p} = 8000\ m$ with an initial velocity towards to the vehicle $V_p = -350 m/s$ ($\gamma_p = \pi$). For a moving point in space with a fixed velocity vector, the result is shown for 100 random realizations of the aerodynamic uncertainty in Figure 42. The respective reference following performance and actuator dynamics are shown in Figure 43.

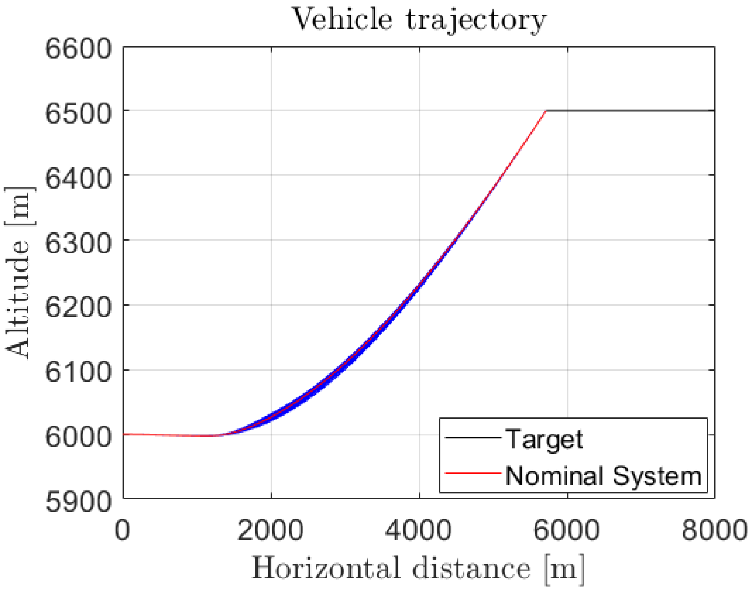


Figure 42. Vehicle trajectories for 100 random realizations with moving point in space

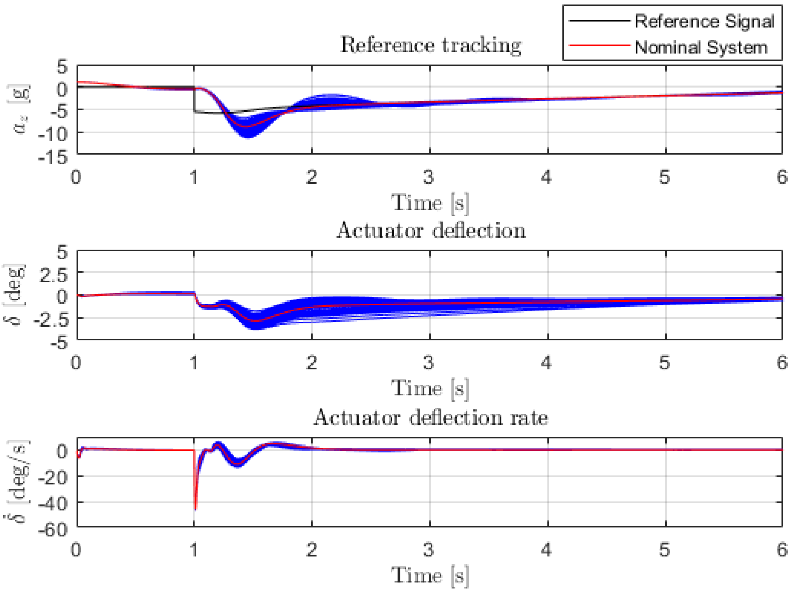


Figure 43. Reference tracking and actuator dynamics

For a more aggressively moving point in space, the results are shown in Figures 44 and 45.

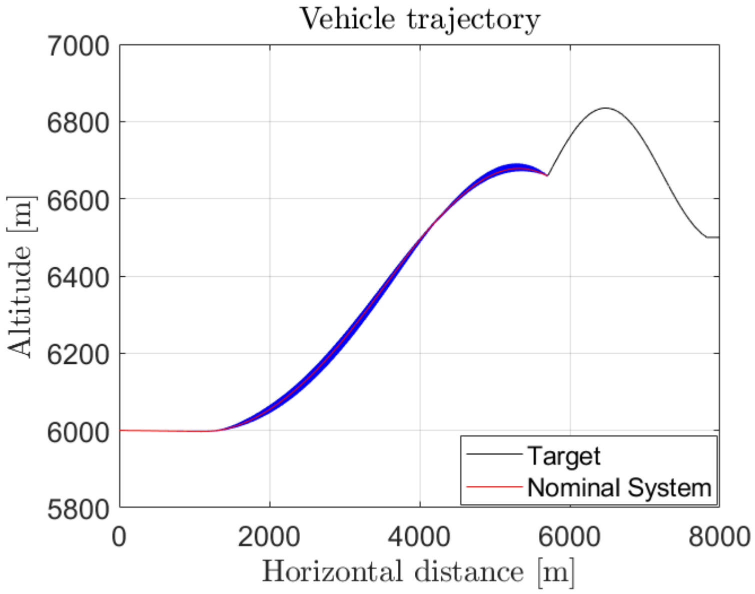


Figure 44. Vehicle trajectories for 100 random realizations with aggressive target

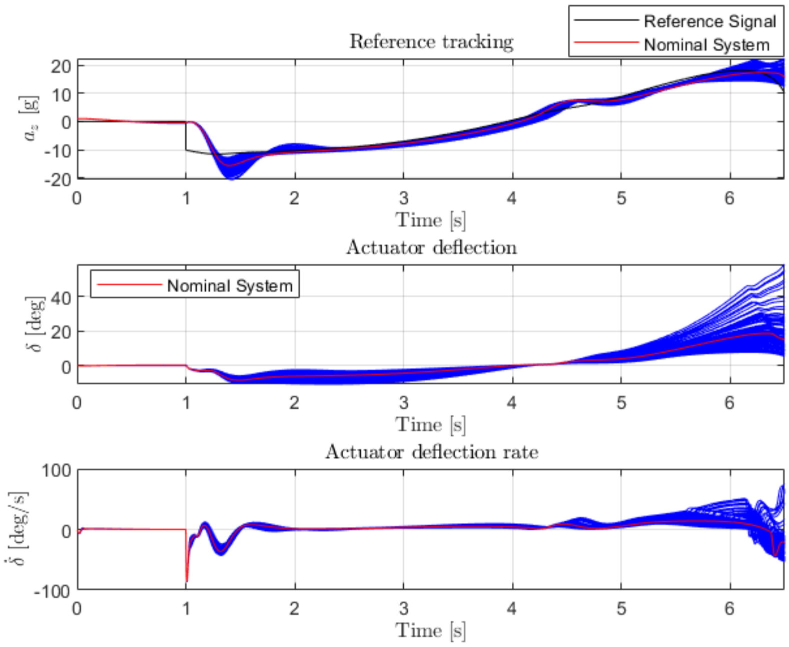


Figure 45. Reference tracking and actuator dynamics (aggressive target scenario)

Note that the reference signal differs per uncertain realization, due to the guidance law adjusting the reference per unique trajectory. It can be concluded that the autopilot performs accurately in a practical scenario with the present uncertainty. The "miss-distance" for both scenarios vary by small margins, while the vehicle performs in an agile way even under uncertainty. Despite the large overshoots for certain realizations as shown in Figure 39, which are induced by the large uncertainty magnitude, the deviation's effect for a more practical point-hit application is limited. This section thus offers an intuitive insight into the uncertainty handling and performance of the autopilot for a more practical application suitable for agile vehicles.

VI. Conclusions & Recommendations

A. Summary and Conclusion

A robust autopilot is designed for an agile aerospace vehicle; selected for its use in other (robust) control studies, thus yielding the possibility for comparison and cross-verification while also posing a challenging control problem given its requirements. The model's non-linear dynamics are analysed and verified, after which trimming and point-linearization is applied for the flight envelope domain, totalling 100 points. The linearized points are stabilized into SISO plants by proportional feedback, which allows for time-scale separation as well as loop shaping the reference-tracking output a_z. H_∞ Loop-Shaping is applied, with pre-filter W_1 being of an integrator form for simplicity and ease of automation. Having chosen the crossover frequency through trial and error (simplified by its automation capability), structured robustification is applied by using non-smooth optimization. A feed-forward controller is in turn designed to enhance reference tracking performance and speed. The resulting gains are implemented, for which the velocity-based implementation and the $\dot{M}$-correction allowed for accurate tracking of the reference signal. The feed-forward controller is averaged and implemented with a semi-dynamic transfer function. The autopilot is tested in non-linear simulations with reference signal profiles and a point-hit scenario with guidance included, showing desirable behaviour even under uncertainty.

The robustness requirements are met and verified for the linear SISO plants, while the linear SIMO plants are verified for having high robustness values. The non-linear simulations showed desirable uncertainty handling. The actuator saturation rate requirements along with the overshoot limits are met for all linear design points and non-linear simulations. The settling time guideline is met for the majority of linear design points while being subject to actuator rate saturation for design points with small mach numbers and angles of attack, which is also reflected in the non-simulations. High frequency attenuation is met (except for 1 point).

It can be concluded that the given design procedure, combining H_∞ Loop-Shaping and structured robustification, is extremely potent in its ability to be automated while yielding desirable robustness and performance results despite its simplicity.

B. Limitations

Despite the simple and effective design structure and the requirements having been met, there are certain limitations to this design technique which require improvements. An important limitation is the procedure's dependence on the similarity of the linearized systems / design points across the flight envelope. This dependency causes a small number of lower-quality controllers to be formed, which form "exceptions to the rule" in an otherwise successful set of local controllers as seen for the loop shape qualities and attenuation values. Next to this, more detailed designs that take into account very specific types of uncertainty may suffer from the conservatives of H_∞ Loop Shaping [4]. Compared to mixed-sensitivity, H_∞ Loop Shaping offers less freedom in tuning a system's individual loops, adding an additional layer of possible conservativeness. As such, a trade off is to be made between the freedom of design required against the desired automation, which is in turn dependent on the specific control design problem.

In terms of the model used, it is only representative for a fixed-altitude situation with no shift in its centre of gravity (due to e.g. fuel burn). These simplifications of reality may yield additional complexities in the design phase for models that cover an extended flight envelope, for which variables like altitude and fuel mass have to be used as scheduling parameters. Moreover, access to the states might be limited in other applications, requiring an observer or countermeasures to noise, for which methods can be used specifically for H_∞ Loop-Shaping as discussed in e.g. [4]. This is out of the scope of this study.

C. Recommendations

Future work will involve tackling the discussed limitations. Smart design-point gridding using the gap metric as explored in e.g. [8] may yield useful insights with respect to flight envelope division across multiple autopilot channels, with each its own set of design points and loop-shaping filters. For the feed-forward controller it is advised to use a multi-model approach (see [20]) given the high controller order that is required, instead of the averaging and manual-tuning approach taken here. Furthermore, alternative methods are recommended to be explored for this application, e.g. the 2-DOF H_∞ Loop-Shaping optimization method as given in [4, 24].

Next to the recommendations to resolve the limitations, it is recommended to extend the design procedure and apply it on more complex models, yielding a more representative insight into the proposed design procedure. An important extension will be the robustification of the SIMO plant (that is, involving q in the robustification), given the loop shaping functions $\frac{K_i}{s}$ and K_q, which together could form W_2 and with W_1 e.g. a scaling function. This will yield guarantees on the robustness against the NCF uncertainty for the SIMO system, while potentially providing more control freedom through K_q (which may in turn improve the consistency of controller quality in the unstable region). Furthermore, it is recommended to extend the non-linear simulations by incorporating actuator deflection saturation, for which anti-windup techniques can be explored. Finally, LPV methods are recommended to be explored in combination with H_∞ Loop-Shaping and structured robustification due to the potentially numerous benefits (e.g. regional guarantees as opposed to point guarantees) as shown in e.g. [3] and [30].

References

[1] Theodoulis, S., and Proff, M., "Robust flight control tuning for highly agile missiles," American Institute of Aeronautics and Astronautics Inc, AIAA, 2021, pp. 1–26. doi: 10.2514/6.2021-1568.

[2] Liu, K.-Z., and Yao, Y., *Robust Control Theory And Applications*, Wiley, 2016. doi: 0.1002/9781119113072.

[3] Simões, A. M., and Cavalcanti, V. M. G. B., "Missile Autopilot Design via Structured Robust Linear Parameter-Varying Synthesis," *Journal of Guidance, Control, and Dynamics*, 2023. doi: 10.2514/1.G007580, URL `https://arc.aiaa.org/doi/10.2514/1.G007580`.

[4] Skogestad, S., and Postlethwaite, I., *Multivariable Feedback Control Analysis and design*, Wiley, 2005.

[5] Jackson, P. B., "Overview of Missile Flight Control Systems," *Johns Hopkins APL Technical Digest: HOMING MISSILE GUIDANCE AND CONTROL*, Vol. 29, 2010, pp. 9–24. URL `https://secwww.jhuapl.edu/techdigest/Home/Detail?Journal=J&VolumeID=29&IssueID=1`.

[6] Lavretsky, E., and Wise, K. A., *Robust and Adaptive Control With Aerospace Applications*, Springer, 2013. URL `https://link.springer.com/book/10.1007/978-1-4471-4396-3`.

[7] do Valle, R. C., Menegaldo, L. L., and Simões, A. M., "Smoothly Gain-Scheduled Control of a Tri-Turbofan Airship," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 1, 2015, pp. 53–61. doi: 10.2514/1.G000453, URL `https://doi.org/10.2514/1.G000453`.

[8] Theodoulis, S., and Duc, G., "Missile autopilot design: Gain-Scheduling and the gap metric," *Journal of Guidance, Control, and Dynamics*, Vol. 32, 2009, pp. 986–996. doi: 10.2514/1.34756.

[9] Vincent, R. V., Economou, J., Wall, D. G., and Cleminson, J., "H/LQR optimal control for a supersonic air-breathing missile of asymmetric configuration," Elsevier B.V., 2019, pp. 214–218. doi: 10.1016/j.ifacol.2019.11.245.

[10] Arikapalli, V., Bhowmick, S., Rao, P., and R., A., "Investigative design of missile longitudinal dynamics using LQR-LQGcontroller in presence of measurement noise and inaccurate model," , 2 2022. doi: https://doi.org/10.1007/s12046-021-01784-0, lQR-LQG.

[11] Lee, C.-H., Jun, B.-E., and Lee, J.-I., "Connections Between Linear and Nonlinear Missile Autopilots via Three-Loop Topology," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 6, 2016, pp. 1426–1432. doi: 10.2514/1.G001565, URL `https://doi.org/10.2514/1.G001565`.

[12] Devaud, E., Harcaut, J.-P., and Siguerdidjane, H., "Three-Axes Missile Autopilot Design: From Linear to Nonlinear Control Strategies," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 1, 2001, pp. 64–71. doi: 10.2514/2.4676, URL `https://doi.org/10.2514/2.4676`.

[13] Mattei, G., and Monaco, S., "Nonlinear Autopilot Design for an Asymmetric Missile Using Robust Backstepping Control," *Journal of Guidance Control and Dynamics*, Vol. 37, 2014. doi: 10.2514/1.G000434.

[14] Mahmood, A., Kim, Y., and Park, J., "Robust H autopilot design for agile missile with time-varying parameters," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 50, 2014, pp. 3082–3089. doi: 10.1109/TAES.2014.130750.

[15] Yan, L., Chang, X., Wang, N., Zhang, L., Liu, W., and Deng, X., "Aerodynamic Identification and Control Law Design of a Missile Using Machine Learning," *AIAA Journal*, 2023, pp. 1–21. doi: 10.2514/1.J062801, URL `https://arc.aiaa.org/doi/10.2514/1.J062801`, nN.

[16] Lin, S. Y., Wang, J., and Wang, W., "Barrier Lyapunov Function Based Integrated Missile Guidance and Control Considering Phased Array Seeker Disturbance Rejection Rate," *IEEE Access*, Vol. 10, 2022, pp. 31070–31083. doi: 10.1109/ACCESS.2022.3156292, other.

[17] McFarland, M. B., and Calise, A. J., "Adaptive nonlinear control of agile anti-air missiles using neural networks," *IEEE Transactions on Control Systems Technology*, Vol. 8, 2000, pp. 749–756. doi: 10.1109/87.865848.

[18] D'Amato, A. M., and Bernstein, D. S., "Adaptive control of a seeker-guided 2D missile with unmodeled aerodynamics," American Institute of Aeronautics and Astronautics Inc., 2012. doi: 10.2514/6.2012-4617, adaptive.

[19] Bates, D., and Postlethwaite, I., *Robust Multivariable Control of Aerospace Systems*, DUP Science, 2002.

[20] Apkarian, P., and Noll, D., "The H∞ Control Problem is Solved," *Aerospace Lab*, 2017, pp. pages 1–11. doi: 10.12762/2017.AL13-01, URL `https://hal.science/hal-01653161`.

[21] Glover, K., and McFarlane, D., "A Loop Shaping Design Procedure Using Ha Synthesis," *IEEE Transactions on Automatic Control*, Vol. 37, 1992, pp. 759–769. doi: 10.1109/9.256330.

[22] Nichols, R. A., Nichols, R. A., Reichert, R. T., and Rugh, W. J., "Gain Scheduling for H-Infinity Controllers: A Flight Control Example," *IEEE Transactions on Control Systems Technology*, Vol. 1, 1993, pp. 69–79. doi:

10.1109/87.238400.

[23] Mulder, J. A., Staveren, W. H. J. J. V., Vaart, J. C. V. D., Weerdt, E. D., Visser, C. C. D., 't Veld, A. C. I., and Mooij, E., *Lecture Notes AE3202 Flight Dynamics*, TU Delft Faculty of Aerospace Engineering, 2013.

[24] Kanade, S. P., and Mathew, A. T., "2 DOF H-Infinity Loop Shaping Robust Control for Rocket Attitude Stabilization," *International Journal of Aerospace Sciences*, Vol. 2013, 2013, pp. 71–91. doi: 10.5923/j.aerospace.20130203.02, URL `http://journal.sapub.org/aerospace`.

[25] Kaminer, I., Pascoal, A. M., Khargonekar, P. P., and Coleman, E. E., "A velocity algorithm for the implementation of gain-scheduled controllers," *Automatica*, Vol. 31, No. 8, 1995, pp. 1185–1191. doi: https://doi.org/10.1016/0005-1098(95)00026-S, URL `https://www.sciencedirect.com/science/article/pii/000510989500026S`.

[26] Lhachemi, H., Saussié, D., and Zhu, G., "Hidden Coupling Terms Inclusion in Gain-Scheduling Control Design: Extension of an Eigenstructure Assignment-Based Technique**This work was supported by NSERC under Grant RGPIN-2014-03942 and RGPIN-312116-13." *IFAC-PapersOnLine*, Vol. 49, No. 17, 2016, pp. 403–408. doi: https://doi.org/10.1016/j.ifacol.2016.09.069, URL `https://www.sciencedirect.com/science/article/pii/S2405896316315427`, 20th IFAC Symposium on Automatic Control in AerospaceACA 2016.

[27] The MathWorks Inc., "Design a Guidance System in MATLAB and Simulink," , 2023. URL `https://nl.mathworks.com/help/simulink/slref/designing-a-guidance-system-in-matlab-and-simulink.html`.

[28] USA Department of Defense, *MIL-HDBK-1211 Military handbook Missile Flight Simulation Part One: Surface-to-air Missiles*, 1995.

[29] Zipfel, P. H., *Modeling and simulation of aerospace vehicle dynamics*, American Institute of Aeronautics and Astronautics, 2007.

[30] Pellanda, P. C., Apkarian, P., and Tuan, H. D., "Missile autopilot design via a multi-channel LFT/LPV control method," *International Journal of Robust and Nonlinear Control*, Vol. 12, 2002, pp. 1–20. doi: 10.1002/rnc.612.

Appendix

Model Values

Table 1. Variables of model dynamics

Variable	Value	Units	Description
m	204.02276	kg	Mass
S	0.04104	m^2	Reference surface
l	0.2286	m	Reference length
I_y	247.43678	kgm^2	Pitch moment of inertia
a_n	19.373368	$1/rad^3$	Force coefficient
b_n	-31.022520	$1/rad^2$	Force coefficient
c_n	-9.717364	$1/rad$	Force coefficient
d_n	-1.948057	$1/rad$	Force coefficient
a_m	40.439554	$1/rad^3$	Force coefficient
b_m	-64.014724	$1/rad^2$	Force coefficient
c_m	2.922085	$1/rad$	Force coefficient
d_m	-11.802931	$1/rad$	Force coefficient
ω_a	150	rad/s	Actuator natural frequency
ζ_a	0.7	$-$	Actuator damping radio
C_x	-0.3	$-$	Longitudinal force coefficient
ρ	0.653118	kg/m^3	Air density
a	316.056116	m/s	Speed of sound
g	9.806650	m/s^2	Gravitational constant

Feed-forward design reference system

Table 2. Reference system initial values for optimization

Variable	Value	Units	Description
ST	0.349	s	Settling time
OS	1	%	Overshoot
ζ	1	$-$	Damping ratio
ω_n	9	rad/s	Natural frequency

Bisection algorithm

Algorithm 1: Loop Shaping Bisection

```
tol = 0.001
$CO = 3$
$K_{i_{a_0}} = 1e - 5$
$K_{i_{b_0}} = 1$
$K_{i_a} = K_{i_{a_0}}$
$K_{i_b} = K_{i_{b_0}}$
repeat
    $W_{1_a} = \frac{K_{i_a}}{s}$
    $W_{1_b} = \frac{K_{i_b}}{s}$
    $CO_a$ = getGainCrossover($G\,W_{1_a}$)
    $CO_b$ = getGainCrossover($G\,W_{1_b}$)

    $\Delta_a = CO_a - CO$
    $\Delta_b = CO_b - CO$

    $K_{i_c} = \frac{K_{i_a}+K_{i_b}}{2}$
    $W_{1_b} = \frac{K_{i_c}}{s}$
    $CO_c$ = getGainCrossover($G\,W_{1_c}$)
    $\Delta_c = CO_c - CO$

    if and($\Delta_a > 0, \Delta_b < 0$):
      if $\Delta_c > 0$: $K_{i_a} \leftarrow K_{i_c}$
      else if $\Delta_c < 0$: $K_{i_b} \leftarrow K_{i_c}$

    vice versa for if and($\Delta_a < 0, \Delta_b > 0$)

until min($|\Delta_a|, |\Delta_b|$) ≤ tol
if $|\Delta_a| \leq$ tol: $W_1 \leftarrow W_{1_a}$
if $|\Delta_b| \leq$ tol: $W_1 \leftarrow W_{1_b}$
```

Gain surface of K_{dc}

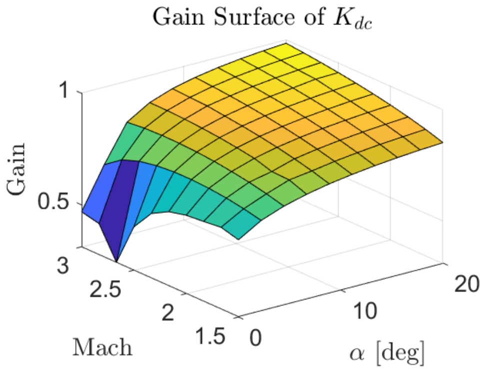


Figure 46. Scaling controller gains

Disturbance rejection performance

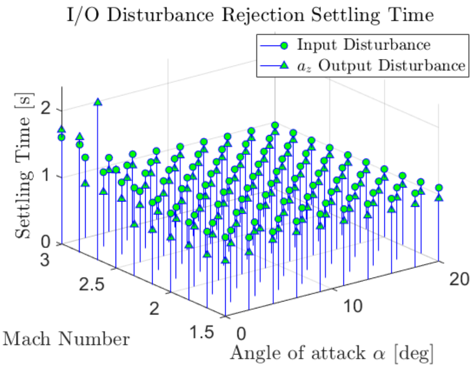


Figure 47. Input-Output disturbance-rejection settling times

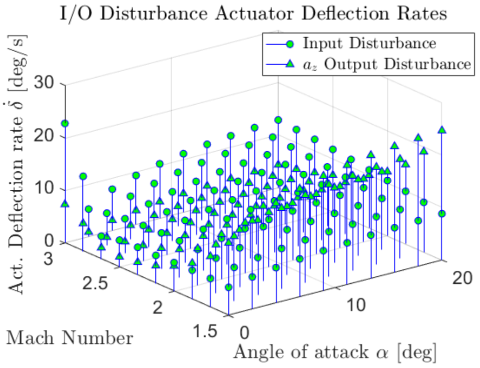


Figure 48. Input-Output disturbance-rejection actuator deflection rate

Feed-forward design

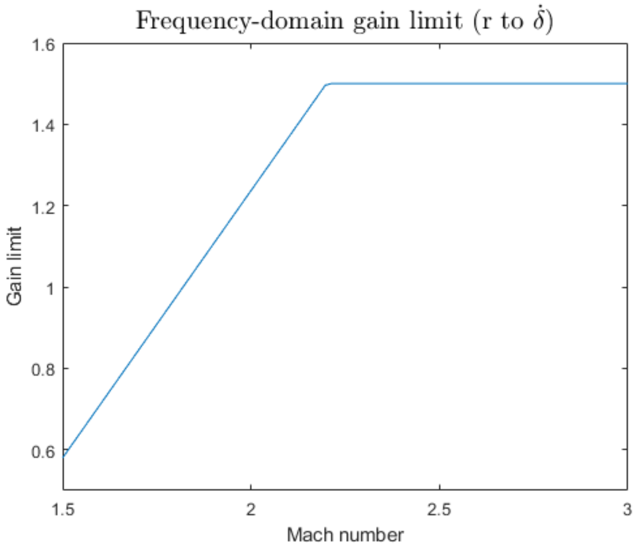


Figure 49. Mach number dependent frequency-domain gain limit

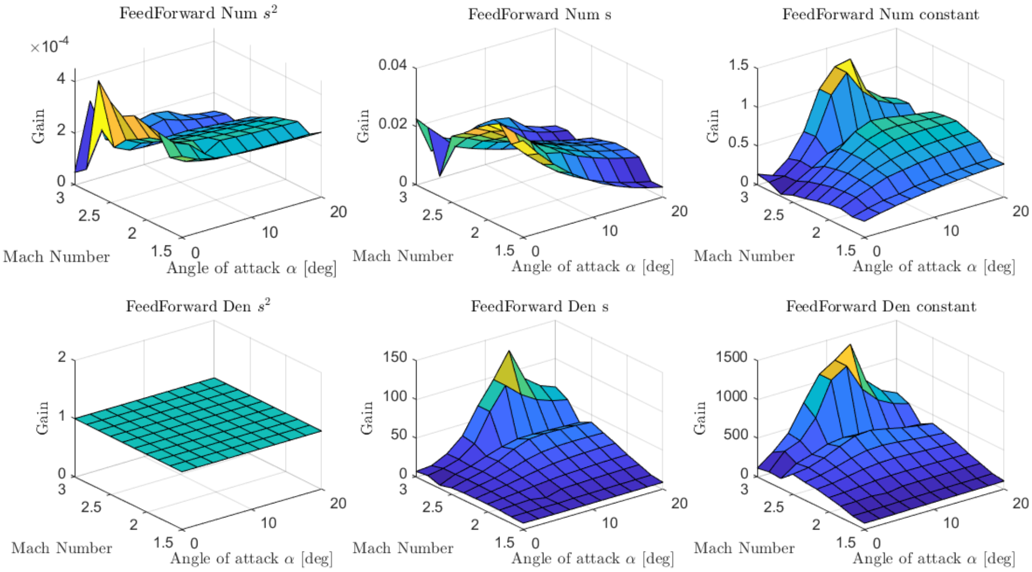


Figure 50. Gain surfaces of feed forward controller (smoothed)

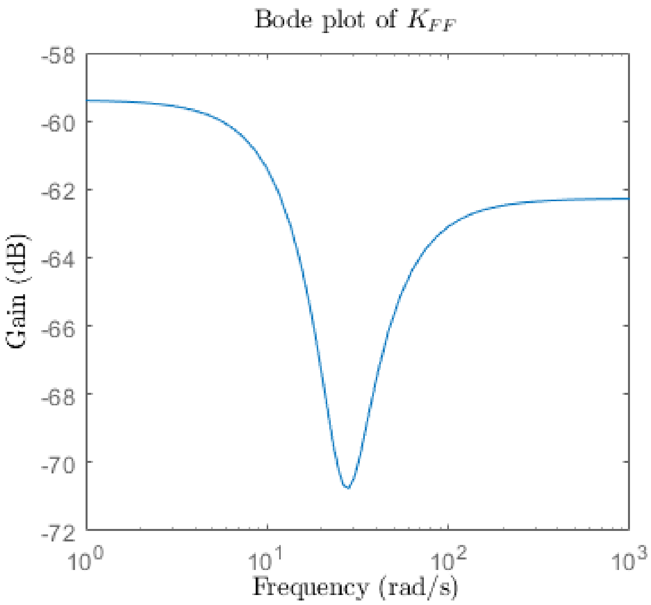


Figure 51. Bode plot of the feed-forward controller

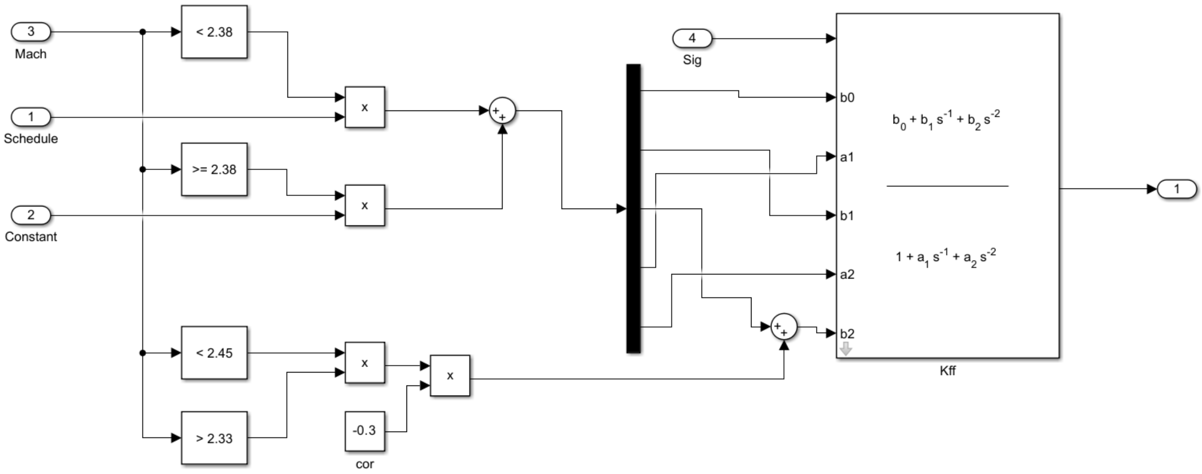


Figure 52. Simulink manual switching block of the feed-forward controller

Actuator dynamics

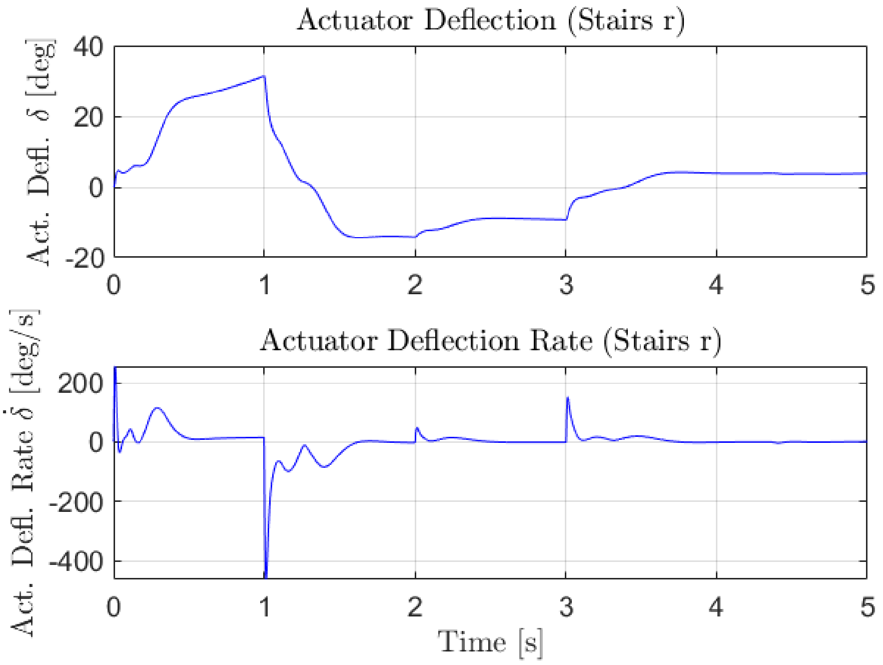


Figure 53. Actuator dynamics for stairs reference

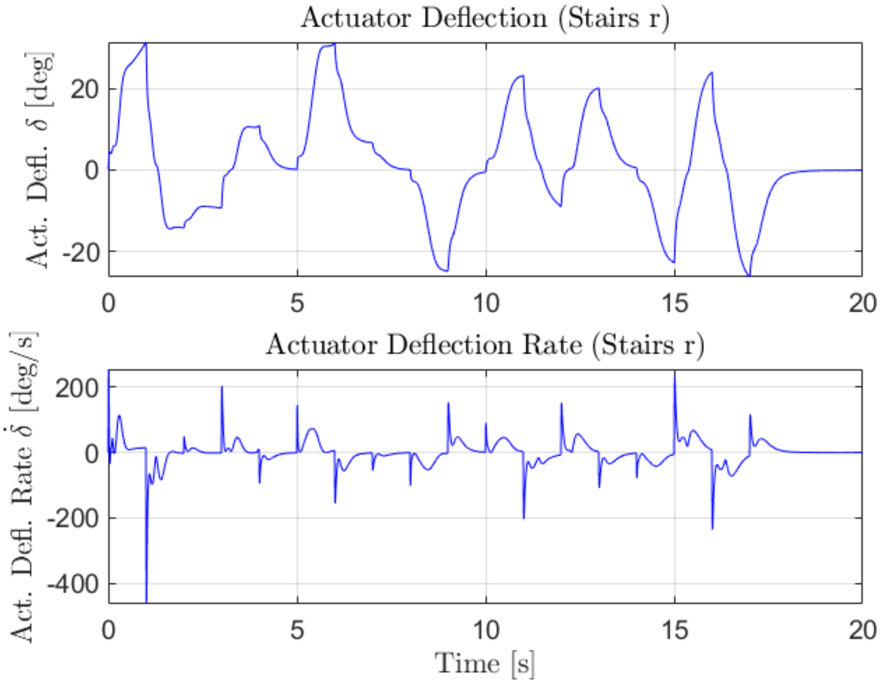


Figure 54. Actuator dynamics for extended stairs reference

Covered flight envelope

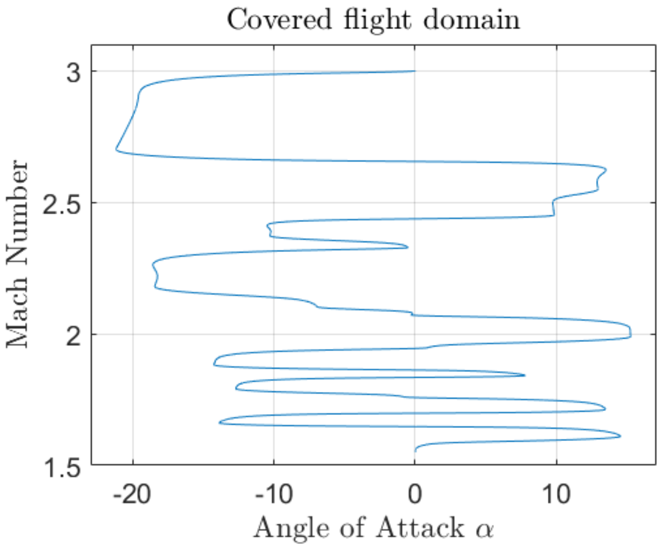


Figure 55. Mach number - α plot

Disturbance rejection of Non-Linear system

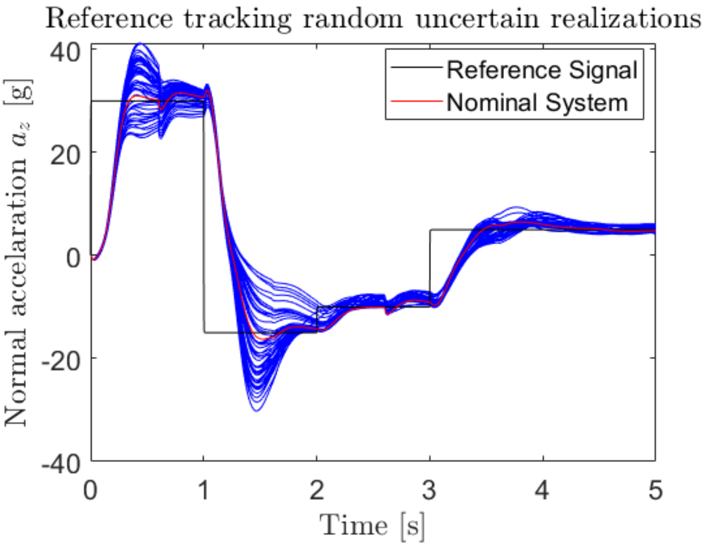


Figure 56. Disturbance rejection with 50 random realizations

The figure above consists of 50 random uncertain realizations, on top of which output disturbances (timestamps 0.6 and 2.6) and input disturbances (timestamps 1.6 and 3.6) are added.

Part III

Closure

5 Discussion & Conclusion

This chapter provides a discussion of the achieved results along with the main results and conclusions with respect to the main- and sub-research questions. The discussion summarizes and comments on the achieved results, along with certain recommendations for improvements and extensions of the methods in section 5.1. The conclusions and research questions are discussed in section 5.2.

5.1. Discussion

The requirements are met for the majority of linearized points. Robust stability against the aerodynamic uncertainty is verified for the SISO plant by the achieved stability margin of K_∞ for all points, although the margin by which this is achieved is lower for 1 point in the unstable (transition) region. Despite having only used the SISO system, the SIMO system shows high stability margins for all linear points (with the same exception as for the SISO system), which constitutes to high robustness. The robustness values satisfy common aerospace robustness requirements, despite having no NCF uncertainty stability guarantee for which SIMO robustification is needed. The attenuation requirements are met, for which only 1 point of the 100 does not satisfy the requirement. The actuator saturation requirement as well as the overshoot requirement are met for all linear design points, which is also the case for the non-linear simulations. The step-tracking guideline is met for most linear points, and is not met due to the actuator rate saturation requirement for low air-speeds and small angles of attack, which is also reflected in the non-linear simulations.

The agile vehicle model used in this study was selected for its use in other (robust) control studies, thus yielding the possibility for comparison and cross-verification. In addition, the model combined with its requirements poses a challenging control problem, certainly in combination with its unstable region. However, the model is only representative for a fixed-altitude situation with no shift in its centre of gravity (due to e.g. fuel burn). These simplifications of reality may yield additional complexities in the design phase of models that cover an extended flight envelope that includes e.g. altitude and its fuel mass as scheduling parameters. It is hence important to test and extend this method into more complex models, yielding a more representative insight into the proposed design procedure. Access to the states used in this study may also be limited or polluted with noise in other systems, which is likely more important for the angle of attack than the (slower) Mach number. The reader is referred to [30] for observer-based structures for H_∞ Loop-Shaping and it is recommended to test and address these issues further.

Despite the simple and effective design structure and the requirements having been met, there are certain limitations to this design technique which require improvements. An important limitation is the procedure's dependence on the similarity of the linearized systems / design points across the flight envelope. This dependency causes a small number of lower-quality controllers to be formed, which form ”exceptions to the rule” in an otherwise successful set of local controllers as seen for the loop shape qualities and attenuation values. Next to this, more detailed designs that take into account very specific types of uncertainty may suffer from the conservativeness of H_∞ Loop Shaping [30]. Compared to Mixed-Sensitivity, H_∞ Loop Shaping offers less freedom in tuning a system's individual loops, adding an additional layer of possible conservativeness. As such, a trade off is to be made between the freedom of design required against the desired automation, which is in turn dependent on the specific control design problem.

For the non-linear implementation, the direct use of the gain surfaces offered a straight forward procedure, for which limited intervention was required. The combination of the velocity-based implementation and mach number rate correction yielded an effective autopilot that behaved desirably for time-domain performance, despite having used trimmed point-linearization for the controller design. The uncertainty handling is analysed for 100 random uncertainty realizations. Since the robustness measures only apply to the linearized design points, this analysis yielded insight into the uncertainty handling qualities of the non-linear model, but can only be verified by more extended methods like Monte Carlo simulations. The completed non-linear system is additionally tested in a ”proof of concept” moving point-hit scenario, for which a simple guidance law is used. The scenario yielded useful additional insight into the uncertainty handling of the autopilot, which gave highly consistent results while showing high agility for aggressive guidance commands even under uncertainty.

Future work will involve tackling the limitations discussed. Smart design-point gridding using the gap metric as explored in e.g. [35] may yield useful insights with respect to flight envelope division across multiple autopilot channels, with each its own set of design points and loop-shaping filters. For the feed-forward controller it is advised to use a multi-model approach (see [1]) given the high controller order that is required, instead of the averaging and manual-tuning approach taken here; to get more performance out of the controller. Furthermore, alternative methods are recommended to be explored for this application, e.g. the 2-DOF H_∞ Loop-Shaping optimization method as given in [13, 30].

Next to the recommendations to resolve the limitations, it is recommended to extend the design procedure. An important extension will be the robustification of the SIMO plant (that is, involving q in the robustification), given the loop shaping functions $\frac{K_i}{s}$ and K_q, which together could form W_2 and with W_1 e.g. a scaling function. This will yield guarantees on the robustness against the NCF uncertainty for the SIMO system, while potentially providing more control freedom through K_q (which may in turn improve the consistency of controller quality in the unstable region). Finally, LPV methods are recommended to be explored in combination with H_∞ Loop-Shaping and structured robustification due to the potentially numerous benefits (e.g. regional guarantees as opposed to point guarantees) as shown in e.g. [26] and [29].

The main points of improvement for the non-linear simulations may be its extension into more realistic implementations, which for example can include anti-windup techniques for deflection saturation. Next to this, the feed-forward controller is of a relatively high order and required special attention in the design phase. The averaging approach of the feed-forward controller yielded stable and desirable results in the non-linear simulations, yet do not meet their full potential given the large actuator rate saturation margin.

5.2. Conclusion

This section evaluates the results and conclusions covered in part II with respect to the posed research questions. The research questions posed in Chapter chapter 3 are repeated below for convenience.

Research Questions

Main research question:
What is an effective agile vehicle autopilot design method using Structured H_∞ Loop-Shaping with a focus on automation?
Sub-questions:

1. **What agile aerospace vehicle model is used and what are its control requirements?**

   1.A What is a suitable agile vehicle model to be used?
   1.B What are the non-linear dynamics of the model?
   1.C What are the linearized models/ design points to be used?
   1.D What are the specific robustness and performance requirements of the final system?

2. **How is H_∞ Loop-Shaping applied?**

   2.A What is the magnitude of the uncertainty?
   2.B How are shaping functions W_1 and W_2 defined and applied for Loop-Shaping?
   2.C How is the shaped plant robustified and what is the structure of the robustifying controller?
   2.D How is the time-domain performance enhanced?
   2.E How well are the requirements met by the individual design points?

3. **How is the the autopilot implemented and what is its non-linear performance?**

   3.A What implementation methods are required and used?
   3.B What is the time-domain performance of the autopilot?
   3.C How does the system handle uncertainty?
   3.D How does the system perform in a practical scenario?

A robust autopilot is designed for an agile aerospace vehicle, which was selected for its use in other robust control studies, its imposed requirements making it a challenging control problem. The model non-linear dynamics are analysed and verified, after which trimming and point-linearization is applied for the flight envelope domain (**Questions 1.A-D answered**).

The linearized points are in turn stabilized into SISO plants by proportional feedback, which allows for time-scale separation as well as loop shaping the reference-tracking output a_z. H_∞ Loop-Shaping is applied, with pre-filter W_1 being of an integrator form for simplicity and ease of automation. Having chosen the crossover frequency through trial and error (simplified by its automation capability), structured robustification is applied by using non-smooth optimization. A feed-forward controller is in turn designed to enhance reference tracking performance and speed (**Questions 2.A-D answered**). The robustness requirements are met and verified for the linear SISO plants, while the linear SIMO plants are verified for having high robustness values. The actuator saturation rate requirements along with the overshoot limits are met for all linear design points. The settling time guideline is met for the majority of linear design points while being subject to actuator rate saturation for design points with small mach numbers and angles of attack. The high-frequency attenuation requirement is met by all linear design points (except for 1 point) (**Question 2.E answered**).

The resulting gains are implemented, for which the velocity-based implementation and the $\dot{M}$-correction allowed for accurate tracking of the reference signal. The feed-forward controller is averaged and implemented with a semi-dynamic transfer function (**Question 3.A answered**). The autopilot is tested in non-linear simulations with reference signal profiles. The actuator saturation rate requirements along with the overshoot limits are met. The settling time guideline is met for flight phases of high Mach numbers, with the effect of actuator rate saturation being reflected for smaller Mach numbers (**Question 3.B answered**). Finally, a point-hit scenario with guidance is created and used for testing, showing desirable behaviour even under uncertainty (**Questions 3.C-D answered**).

For the main research question; it can be concluded that the given design procedure combining H_∞ Loop-Shaping and structured robustification is very potent in its ability to be automated while yielding desirable robustness and performance results despite its simplicity. The described procedure is therefore deemed effective for its application, making the **Main Research Question answered**.

References

[1] Pierre Apkarian and Dominikus Noll. “The H∞ Control Problem is Solved”. In: *Aerospace Lab* 13 (Nov. 2017), pages 1–11. DOI: 10.12762/2017.AL13-01. URL: https://hal.science/hal-01653161.

[2] Gary J. Balas. “Flight control law design: An industry perspective”. In: *European Journal of Control* 9 (2-3 Apr. 2003), pp. 207–226. ISSN: 09473580. DOI: 10.3166/ejc.9.207-226.

[3] Declan Bates and Ian Postlethwaite. *Robust Multivariable Control of Aerospace Systems*. DUP Science, June 2002. ISBN: 90-407-2317-6.

[4] Jean-Marc Biannic, Pascal Gahinet, and Pierre Apkarian. “Gain-Scheduled H∞ Control of a Missile via Linear Matrix Inequalities”. In: (Dec. 1993). URL: https://www.researchgate.net/publication/2799959.

[5] John Eugene Bibel. “Missile Autopilot Design Using mu-synthesis”. Virginia Polytechnic Institute and State University, June 1998. URL: https://vtechworks.lib.vt.edu/items/b56bcb74-476b-4459-99f4-cbac3ad3ff1b.

[6] Anthony M. D’Amato and Dennis S. Bernstein. “Adaptive control of a seeker-guided 2D missile with unmodeled aerodynamics”. In: Adaptive. American Institute of Aeronautics and Astronautics Inc., 2012. ISBN: 9781600869389. DOI: 10.2514/6.2012-4617.

[7] Jean Patrick Friang, Gilles Duc, and Jean Paul Bonnet. “Robust autopilot for a flexible missile: loop-shaping H∞ design and real v-analysis”. In: *International Journal of Robust and Nonlinear Control* 8 (2 1998), pp. 129–153. ISSN: 10498923. DOI: 10.1002/(sici)1099-1239(199802)8:2<129::aid-rnc325>3.0.co;2-x.

[8] Bo Fu and Silvia Ferrari. “Robust flight control via minimum H∞ entropy principle”. In: American Institute of Aeronautics and Astronautics Inc, AIAA, Jan. 2018. ISBN: 9781624105265. DOI: 10.2514/6.2018-1313.

[9] Keith Glover and Duncan McFarlane. “A Loop Shaping Design Procedure Using Ha Synthesis”. In: *IEEE Transactions on Automatic Control* 37 (6 June 1992), pp. 759–769. ISSN: 15582523. DOI: 10.1109/9.256330.

[10] Paul B. Jackson. “Overview of Missile Flight Control Systems”. In: *Johns Hopkins APL Technical Digest: HOMING MISSILE GUIDANCE AND CONTROL* 29 (1 2010), pp. 9–24. URL: https://secwww.jhuapl.edu/techdigest/Home/Detail?Journal=J&VolumeID=29&IssueID=1.

[11] Isaac Kaminer et al. “A velocity algorithm for the implementation of gain-scheduled controllers”. In: *Automatica* 31.8 (1995), pp. 1185–1191. ISSN: 0005-1098. DOI: https://doi.org/10.1016/0005-1098(95)00026-S. URL: https://www.sciencedirect.com/science/article/pii/000510989500026S.

[12] Swapnil Pramod Kanade and Abraham T Mathew. “2 DOF H-Infinity Loop Shaping Robust Control for Rocket Attitude Stabilization”. In: *International Journal of Aerospace Sciences* 2013 (2 2013), pp. 71–91. DOI: 10.5923/j.aerospace.20130203.02. URL: http://journal.sapub.org/aerospace.

[13] Swapnil Pramod Kanade and Abraham T Mathew. “2 DOF H-Infinity Loop Shaping Robust Control for Rocket Attitude Stabilization”. In: *International Journal of Aerospace Sciences* 2013 (2 2013), pp. 71–91. URL: http://journal.sapub.org/aerospace.

[14] Seonhyeok Kang et al. “Roll-pitch-yaw integrated robust autopilot design for a high angle-of-attack missile”. In: *Journal of Guidance, Control, and Dynamics* 32 (5 2009), pp. 1622–1628. ISSN: 15333884. DOI: 10.2514/1.39812.

[15] Jong Han Kim and Ick Ho Whang. “Augmented three-loop autopilot structure based on mixed-sensitivity h∞ optimization”. In: *Journal of Guidance, Control, and Dynamics* 41 (3 2018), pp. 748–753. ISSN: 15333884. DOI: 10.2514/1.G003119.

[16] Chang-Hun Lee, Byung-Eul Jun, and Jin-Ik Lee. "Connections Between Linear and Nonlinear Missile Autopilots via Three-Loop Topology". In: *Journal of Guidance, Control, and Dynamics* 39.6 (2016), pp. 1426–1432. DOI: 10.2514/1.G001565. eprint: https://doi.org/10.2514/1.G001565. URL: https://doi.org/10.2514/1.G001565.

[17] Shih Yao Lin, Jiang Wang, and Wei Wang. "Barrier Lyapunov Function Based Integrated Missile Guidance and Control Considering Phased Array Seeker Disturbance Rejection Rate". In: *IEEE Access* 10 (2022). Other, pp. 31070–31083. ISSN: 21693536. DOI: 10.1109/ACCESS.2022.3156292.

[18] Kang-Zhi Liu and Yu Yao. *Robust Control Theory And Applications*. Wiley, 2016. ISBN: 978-1-1187-5437-5. URL: https://onlinelibrary.wiley.com/doi/book/10.1002/9781119113072.

[19] Arshad Mahmood, Yoonsoo Kim, and Jeongho Park. "Robust H∞ autopilot design for agile missile with time-varying parameters". In: *IEEE Transactions on Aerospace and Electronic Systems* 50 (4 Oct. 2014), pp. 3082–3089. ISSN: 00189251. DOI: 10.1109/TAES.2014.130750.

[20] Michael B. McFarland and Anthony J. Calise. "Adaptive nonlinear control of agile anti-air missiles using neural networks". In: *IEEE Transactions on Control Systems Technology* 8 (5 Sept. 2000), pp. 749–756. ISSN: 10636536. DOI: 10.1109/87.865848.

[21] A. Mohamed et al. "Robust autopilot design and hardware-in-the-loop simulation for air to air guided missile". In: *Advances in Military Technology* 12 (2 2017), pp. 281–300. ISSN: 25334123. DOI: 10.3849/aimt.01207.

[22] J A Mulder et al. *Lecture Notes AE3202 Flight Dynamics*. TU Delft Faculty of Aerospace Engineering, 2013.

[23] Robert A. Nichols et al. "Gain Scheduling for H-Infinity Controllers: A Flight Control Example". In: *IEEE Transactions on Control Systems Technology* 1 (2 1993), pp. 69–79. ISSN: 15580865. DOI: 10.1109/87.238400.

[24] Yoshimasa Ochi, Kouhei Itoh, and Kimio Kanai. "Missile Guidance Law Design via μ-Synthesis". In: *TRANSACTIONS OF THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES* 45.148 (2002), pp. 102–109. DOI: 10.2322/tjsass.45.102.

[25] A. N. Ouda. "Scalar H∞ autopilot synthesis for control systems and evaluation via HIL simulation". In: *SN Applied Sciences* 1 (12 Dec. 2019). ISSN: 25233971. DOI: 10.1007/s42452-019-1400-9.

[26] Paulo C. Pellanda, Pierre Apkarian, and Hoang Duong Tuan. "Missile autopilot design via a multi-channel LFT/LPV control method". In: *International Journal of Robust and Nonlinear Control* 12 (1 Jan. 2002), pp. 1–20. ISSN: 10498923. DOI: 10.1002/rnc.612.

[27] C.A. Rabbath et al. "Combined H-Infinity Model-Matching Control and Dual-rate Digital Redesign for Missile Acceleration Autopilots". In: *Transactions of the Canadian Society for Mechanical Engineering* 28.1 (2004), pp. 73–87. DOI: 10.1139/tcsme-2004-0007. eprint: https://doi.org/10.1139/tcsme-2004-0007. URL: https://doi.org/10.1139/tcsme-2004-0007.

[28] Michael G. Safonov. "Origins of robust control: Early history and future speculations". In: *Annual Reviews in Control* 36 (2 Dec. 2012), pp. 173–181. ISSN: 13675788. DOI: 10.1016/j.arcontrol.2012.09.001. URL: https://www.sciencedirect.com/science/article/pii/S1474667015376540.

[29] Alberto M. Simões and Vinícius M. G. B. Cavalcanti. "Missile Autopilot Design via Structured Robust Linear Parameter-Varying Synthesis". In: *Journal of Guidance, Control, and Dynamics* (Apr. 2023), pp. 1–8. ISSN: 0731-5090. DOI: 10.2514/1.G007580. URL: https://arc.aiaa.org/doi/10.2514/1.G007580.

[30] Sigurd Skogestad and Ian Postlethwaite. *Multivariable Feedback Control Analysis and design*. Wiley, Nov. 2005. ISBN: 978-0-470-01167-6.

[31] The MathWorks Inc. *Fixed-Structure H-infinity Synthesis with hinfstruct*. URL: https://nl.mathworks.com/help/robust/gs/fixed-structure-h-infinity-synthesis-with-hinfstruct.html#hinfstruct_demo-4.

[32] The MathWorks Inc. *Mixed-Sensitivity Loop Shaping*. URL: https://nl.mathworks.com/help/robust/gs/using-mixsyn-for-h-infinity-loop-shaping.html.

[33] The MathWorks Inc. *Stability Analysis Using Disk Margins*. URL: https://nl.mathworks.com/help/robust/ug/stability-analysis-using-disk-margins.html#mw_db136e84-9dff-4e5a-bc98-3dfec5ff231b.

[34] Julian Theis et al. “Linear parameter-varying feedforward control: A missile autopilot design”. In: American Institute of Aeronautics and Astronautics Inc, AIAA, 2015. ISBN: 9781624103391. DOI: 10.2514/6.2015-2001.

[35] Spilios Theodoulis and Gilles Duc. “Missile autopilot design: Gain-Scheduling and the gap metric”. In: *Journal of Guidance, Control, and Dynamics* 32 (3 2009), pp. 986–996. ISSN: 15333884. DOI: 10.2514/1.34756.

[36] Spilios Theodoulis and Michael Proff. “Robust flight control tuning for highly agile missiles”. In: American Institute of Aeronautics and Astronautics Inc, AIAA, 2021, pp. 1–26. ISBN: 9781624106095. DOI: 10.2514/6.2021-1568.

[37] Spilios Theodoulis, Micheal Proff, and Charlotte Marchand. “Robust Design for Highly Agile Missile Autopilots”. In: IEEE, Sept. 2020, pp. 67–72. ISBN: 978-1-7281-5742-9. DOI: 10.1109/MED48518.2020.9183304.

[38] Rosario Toscano. *Structured Controllers for Uncertain Systems*. 1st ed. Springer London, 2013. ISBN: 978-1-4471-5187-6. DOI: 10.1007/978-1-4471-5188-3. URL: https://link.springer.com/10.1007/978-1-4471-5188-3.

[39] Raymond Vin Vincent et al. “H∞/LQR optimal control for a supersonic air-breathing missile of asymmetric configuration”. In: vol. 52. Elsevier B.V., Oct. 2019, pp. 214–218. DOI: 10.1016/j.ifacol.2019.11.245.

[40] Dae-Yeon Won, Min-Jea Tahk, and Yoon-Hwan Kim. “Three-Axis Autopilot Design for a High Angle-Of-Attack Missile Using Mixed H 2 /H ∞ Control”. In: *International Journal of Aeronautical and Space Sciences* 11 (2 June 2010), pp. 131–135. ISSN: 2093-274X. DOI: 10.5139/jass.2010.11.2.131.

[41] Lang Yan et al. “Aerodynamic Identification and Control Law Design of a Missile Using Machine Learning”. In: *AIAA Journal* (Apr. 2023). NN, pp. 1–21. ISSN: 0001-1452. DOI: 10.2514/1.J062801. URL: https://arc.aiaa.org/doi/10.2514/1.J062801.