Review of H_∞ Static Output Feedback Controller Synthesis Methods

Application to Fighter Aircraft Control

A.D.P. Schoon

Delft University of Technology

TU Delft

Review of H_∞ Static Output Feedback Controller Synthesis Methods

Application to Fighter Aircraft Control

a M.Sc. thesis by

A.D.P. Schoon

Thesis committee:	Dr. Eng. S. Theodoulis — TU Delft, AE, Control & Simulation
	Dr. Ir. C.C. de Visser — TU Delft, AE, Control & Simulation
	Dr. Ir. O.A. Sharpans'kykh — TU Delft, AE, Air Transport and Operations
Project Duration:	December, 2022 - January, 2024
Faculty:	Faculty of Aerospace Engineering, Delft, The Netherlands

Preface

I'm proud to present this thesis report, which describes research on Static Output Feedback over a period which lasted a little over a year at the Delft University of Technology. I am pleased to say that I will be the first student to graduate on a Robust Control subject at the Aerospace Engineering Faculty at TU Delft since many years, and I'm confident more groundbreaking researches will succeed mine. I would like to express my sincere gratitude to Coen, and to Spilios, my supervisor, for masterfully guiding me through this complicated branch of control. I would also like to thank Fedrik, Yair, and Chari, my student peers, without whom this project probably would've taken me much longer to finish. Lastly, I would like to express gratitude to my family and friends as well for their emotional support and encouragement.

This thesis provides a full implementation of promising Lyapunov-based Static Output Feedback (SOF) algorithms for H_∞-performance. Additionally, the algorithms have been extended to incorporate structured SOF, and I have tried to improve upon the original algorithms in terms of computational efficiency. The algorithms have been compared to well-established non-smooth optimization methods to assess their competitiveness.

Thank you so much for your interest in my thesis and I hope it can inspire you to do more research on SOF, because it is still very much an open problem.

A.D.P. Schoon
Delft, December 2023

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Abbreviation	Definition
BMI	Bilinear Matrix Inequality
CAP	Control Anticipation Parameter
CCL	Cone Complementarity Linearization
CTM	Coordinate Transformation Matrix
DOF	Dynamic Output Feedback
DS	Direct Search
GBD	Generalized Benders Decomposition
GPM	Gradient Projection Method
H.R.	Hit-and-Run algorithm
ILMI	Iterative Linear Matrix Inequality
LFT	Linear Fractional Transformation
LHP	Left-Hand Plane
LMI	Linear Matrix Inequality
LTI	Linear Time Invariant
LQG	Linear Quadratic Gaussian
SDP	Semi-Definite Programming
SOF	Static Output Feedback

Symbols

Symbol	Definition	Unit
Latin:		
A	State matrix	-
a_n	Normal acceleration	$g -$ units
B	Input matrix	-
C	Output matrix	-
D	Feedthrough matrix	-
n	Load factor	$g -$ units
P	Generalized plant matrix	-
P, Q	Lyapunov matrices	-
q	Pitch rate	°/s
u	Input vector	-
V	Velocity	m/s
w	Exogenous input vector	-
x	State vector	-
y	Measured output vector	-
z	Evaluated output vector	-
Greek:		
α	Angle of attack	rad
α	Decay rate	-
ζ_{sp}	Short-period damping ratio	-
ω	Frequency	rad/s
ω_{sp}	Short-period natural frequency	rad/s

1

Bibliographic Survey

Introduction

In many applications, including aerospace applications, it is often not possible to measure all the states. State-feedback, i.e. full-information, controllers are therefore less realistic when compared to output feedback controllers, which are based on the measurable outputs only and are therefore more applicable in practical applications.

Furthermore, in H_∞ control, synthesized controllers are often of high order. It is often desired, however, to construct fixed-order controllers with lower order in contrast to full-order controllers, the latter of which are more expensive to tune and apply in practice. It is therefore cheaper and more practical to use fixed-order controllers that have a lower order, e.g. PI controllers. This also makes it easier to provide hardware and software maintenance in industrial applications. The simplest subset of fixed-order output feedback controllers are static output feedback (SOF) controllers, in contrast to dynamic output feedback (DOF) controllers. The simplicity of SOF controllers leads to them being desirable in practice, also because of the fact that these controllers are direct, meaning the entries of the controller have a physical interpretation, as they directly link a certain input with a certain output. Lastly, the long-term memory aspect of dynamic controllers is rendered useless when dealing with fast changing dynamics and/or random noise.

The problem of designing SOF controllers is a difficult challenge and is the reason for much of the scientific research in the late 1990s and to this day it is still an open problem in control engineering. This is due to the fact that fixed-order control problems, i.e. (Lyapunov based) Bilinear Matrix Inequality (BMI) problems or spectral abscissa minimization problems, are NP-hard to solve due to their non-convexity and/or non-differentiability, making it difficult to solve in polynomial time (Toker and Ozbay 1995).

Many attempts have been made in recent years to tackle the problem of SOF control. Methods that are highly efficient in synthesizing fixed-order controllers and SOF controllers are HIFOO (Burke et al. 2006) and Hinfstruct (Apkarian and Noll 2006b). Although their undeniable power in synthesizing SOF controllers, these methods do not allow for much flexibility and transparency in their approach as they rely on rather sophisticated theoretical tools (non-smooth optimization) without much control interpretation in their approach. So from the perspective of a control designer, it might be preferable to opt for alternative methods.

Research gap and contribution

SOF synthesis methods have been compared in papers before (Syrmos et al. 1997; Sadabadi and Dimitri Peaucelle 2016). However, there has been a great amount of research done on SOF synthesis since then, which has not been covered by the aforementioned papers, the most significant being (M. Saeki and C. M. Saeki 2018; Dingchao Ren 2022; Holicki and C. W. Scherer 2021; Harikumar, Dhall, and Bhat 2019; Chanekar, Chopra, and Azarm 2017; Cheng et al. 2022; Feng, She, and Xu 2019; Feng, Guo, et al. 2022; P. Li et al. 2021; D. Ren, Xiong, and Daniel W.C. Ho 2021; Y. Ren et al. 2021; Sereni et al. 2018; B. Sereni 2023; Sahoo et al. 2019; Gopmandal and Ghosh 2021; Chou and Wei 2020; Y. Peretz 2017; Yossi Peretz 2018; Peretz, Merzbach, and Moyall 2020; Tian et al. 2020; D. Arzelier et al. 2018; Felipe and R. Oliveira 2021). Furthermore, these comparative studies mostly compared the methods qualitatively, explaining the different advantages and disadvantages of each group of approaches. While this is useful in order to gain understanding of the methods, decisive conclusions on the competitiveness of these methods can be drawn with more confidence when accompanied with quantitative analysis of their performance. To the best of the author's knowledge, there has not been such comparative study done this to this day.

This research will try to fill this gap by investigating different SOF synthesis approaches and comparing their results to commercially available software and to other experimented SOF methods. Hence, conclusions can be drawn on the current state of SOF synthesis approaches and their competitiveness with highly efficient methods such as HIFOO and Hinfstruct. It is worth noting as well that the authors of the SOF approaches generally do not provide public domain software to reproduce the computational experiments. It is therefore already useful should implementation of the written theory be able to be performed.

Besides contributing to the field of research on SOF, this study will contribute to the field of aerospace, by analyzing the possibility of applying a SOF controller to stabilize the complex and often coupled dynamics of military fighter aircraft, which are open-loop unstable and thus more difficult to control (Stein 2003). Applying SOF controllers to aeronautical applications would greatly reduce the long and expensive process of certifying flight control systems. To showcase the effectiveness of the approaches, the methods will be applied on the F-16 Fighting Falcon. Finally, mixed sensitivity allows for a good framework for application of the SOF algorithms, due to its transparency in meeting closed-loop design objectives with weighting filter selection for multi-variable systems.

Objective and research questions

To obtain a clear purpose and define the structure of the research, the research objective and its research questions are formulated as follows:

Research Objective:

"To obtain more insight into the performance of Lyapunov-based static output feedback controller synthesis methods in the H_∞ mixed sensitivity framework."

Research Questions:

The main research question that can be formulated is as follows:

Q: ***"How competitive are Lyapunov-based static output feedback synthesis methods in an H_∞ mixed sensitivity framework for flight control of fighter aircraft, so that it is robust against uncertainties and achieves good performance?"***

The main research question can be subdivided in the following subquestions:

Q1: *What mixed sensitivity approaches are suitable for flight control of fighter aircraft?*
- **Q1.a:** *What approaches are stated in the literature?*
- **Q1.b:** *What are the strengths and limitations of each approach?*

Q2: *What are the state-of-the-art Lyapunov-based algorithms for static output feedback control?*
- **Q2.a:** *What algorithms are stated in the literature?*
- **Q2.b:** *What are the strengths and limitations of each algorithm?*

Q3: *How well do the static output feedback algorithms perform on benchmark models, in comparison to well-established methods?*

Q4: *How well do the controllers synthesized by the static output feedback algorithms perform on the nominal aircraft model?*

Q5: *How robust is the closed-loop system tuned with static output feedback synthesis to uncertainties in the aircraft model?*

Report structure

The bibliographic review can be divided into two main parts. The first part consists of covering the robust control methods. It will cover the literature concerning the mixed sensitivity and tries to compare and explain different approaches on the application on fighter aircraft. This can be found in section 1.1. After that, different algorithms for SOF controller synthesis are explained and their advantages and disadvantages are elaborated on in section 1.2.

Notation: $X^\top$ for a matrix X denotes the transpose, $\mathrm{Sym}\{X\}$ denotes $X + X^\top$, $X^\perp$ denotes the orthogonal complement, X^+ denotes the Moore-Penrose pseudo-inverse, and $||X||_\infty$ denotes the H_∞-norm of X, lastly, * in a symmetric matrix denotes the symmetric term.

1.1. Robust Control Method

In this chapter, the robust control method that is used on the F-16 is presented and elaborated on. In subsection 1.1.1, the history and theory of the H_∞ control is given. Then, in subsection 1.1.2 an overview is given of the characteristics of fighter aircraft and their implications and an overview of control methods on fighter aircraft is given. Finally, in subsection 1.1.3, the different approaches to mixed sensitivity design are presented and compared for their strengths and limitations.

1.1.1. H_∞ control

Most of the 20th century was dominated by classical and optimal control, which provided powerful methods for optimal performance and stability. However, in practice, these methods were found to be unable to withstand model uncertainty and this lead to many systems becoming unstable in practice. Therefore, a change of paradigm was on the horizon and this lead to the birth of *robust* control and with it, H_∞ control.

The introduction of the H_∞ norm was introduced in the 1960s by George Zames, but the research for H_∞ control came much later, after Zames introduced it in (Zames 1981). The core was defined by the minimization of the H_∞-norm, i.e. the maximum gain of the system over all frequencies and in all directions. Minimizing this will minimize the effect of external (exogenous) inputs on the evaluated output of the system. In (Keith Glover and J. C. Doyle 1988), the problem was proven to be solvable using Algebraic Ricatti Equations (AREs). After that in (Pascal Gahinet and Apkarian 1994), semi-definite programming was used to solve the problem using Linear Matrix Inequality (LMI) constraints.

Many methods rose from the introduction of the H_∞ control. These include for example mixed H_2/H_∞ control (J. Doyle, Zhou, and Bodenheimer 1989) and H_∞ loop-shaping (D. C. McFarlane and Keith Glover 1990; D. McFarlane and K. Glover 1992). For mixed sensitivity loop shaping, early considerations of minimization of infinity norms on weighted sensitivity functions can be traced back to Zames' paper already and early examples of mixed sensitivity control can be found in (Francis 1987; Safonov, RY Chiang, and Flashner 1988). A discussion of these methods and their different advantages and disadvantages with respect to eachother can be found in (Bates and Ian Postlethwaite 2002a).

1.1.2. Control of fighter aircraft

Before diving into the different methods for H_∞ mixed sensitivity control, first the configuration shall be investigated on which these methods will be applied. The F-16 and its flying characteristics will be described including discussion on its open-loop instability, after which an overview of control methods on fighter aircraft will be presented next.

Aircraft description and flying characteristics

The Lockheed Martin F-16 "Fighting Falcon" is a fighter aircraft, falling under Class IV of high-maneuverability airplanes in ("MIL-F-8785C" 1980). Due to the fact that it is a Class IV aircraft, high-performance precision tracking of targets is required. The F-16 is fly-by-wire and is open-loop unstable with the center of pressure lying ahead of the center of gravity, leading to the fact that it is much harder to control than statically stable aircraft. This is done in order to greatly improve maneuverability, one of the inherent requirements to Class IV aircraft. In fact, the F-16 is the first fighter aircraft where open-loop instability was used deliberately. The implications of this on the design of the controller are described in (Stein 2003) and can be summarized as follows:

- **Sensitivity equilibrium:** Attenuation of sensitivity peaks at certain (low) frequencies must be "paid for" at other (higher) frequencies, within a certain frequency range called the "available bandwidth", and this price is worse for open-loop unstable systems. The price of sensitivity reduction is directly related to the location of the unstable pole(s) and the bandwidth up to which one wants good performance.
- **Operation critical:** Controllers for open-loop unstable systems are operation critical. Therefore, for fly-by-wire systems such as the F-16, careful reliable control systems must be designed that involve redundancy management with multiple backup channels in case of failure. A voting system then takes place to identify failure channels. In case of unstable dynamics this voting process becomes troublesome due to set thresholds being exceeded on which the voting is based.
- **Local stability:** Closed-loop systems with unstable components are only locally stable. Special care must be taken to avoid position and rate limits saturation, in which case divergence would occur.

The non-linear equations of motion stem from (Zipfel 2007), the six degrees-of-freedom aerodynamic model is taken from (Stevens, Frank L. Lewis, and Johnson 2016, p. 714-723), and the flight requirements stem from ("MIL-F-8785C" 1980). These include bounds for the natural frequency and damping ratio for different flight modes, e.g. Dutch roll, short-period, etc. Knowing these, one can obtain the required placement of the poles to satisfy handling qualities. However, in case of highly augmented aircraft such as the F-16, pole-zero specifications become quickly impractical due to the contribution of additional poles and zeros by the control systems. In that case, frequency-response techniques can be applied by, for example, fitting a low-order modal system to a higher-order system (Stevens, Frank L. Lewis, and Johnson 2016). This will be further described in subsubsection 1.1.3.

Control methods overview

H_∞ robust control has been implemented widely in aerospace applications due to their robustness guarantees and performance. In (Voulgaris and Valavani 1991), mixed sensitivity control is applied on the F-18 and in (Ikeda, McDowell, and Hargis 1990), mixed sensitivity control was applied on the longitudinal dynamics of a fighter aircraft and results were compared to that of LQG design. In (Garg and Ouzts 1991; Garg 1993), mixed sensitivity was applied on a supersonic short take-off and vertical landing (STOVL) to operate in transonic flight conditions. In (Anderson, Emami-Naeini, and Vincent 1991), a mixed sensitivity approach has been used for the control law design on the lateral-directional dynamics of a hypersonic vehicle. In (R.Y. Chiang et al. 1993), a non-scheduled fixed mixed sensitivity controller is designed for a supermaneuverable aircraft in order to perform a Herbst maneuver. In (Kwakernaak 2002b), the S/KS mixed sensitivity design discussed in subsection 1.1.3 by the same author is applied on a design example of a fighter aircraft. Finally, in (Yang, Chang, and Y.-P. Sun 1996), the S/T mixed sensitivity approach was applied on a pitch-axis controller of the Rockwell HiMAT, a highly maneuverable aircraft.

Concerning the F-16, in (Alvarez and Lu 2011), comparisons between different control approaches (classical, H_∞, and LPV control) were compared in piloted simulation on the longitudinal dynamics of an F-16 and they were tested on pilot scores of handling qualities. In (Gadewadikar and F. Lewis 2006; J. Gadewadikar et al. 2006), a two-stage approach proposed by the same authors is applied, respectively, on the longitudinal and lateral dynamics of the F-16 using H_∞ control to obtain a SOF controller (see subsubsection 1.2.2 for more details). In (Lu and F. Wu 2005), with a randomized algorithm, a weighted sensitivity problem is synthesized for a longitudinal autopilot controller of the F-16 to be robust against probabilistic uncertainties. In chapter 4 of (Adams et al. 1994), using dynamic inversion and μ-synthesis, a lateral-directional controller is designed for the F-16 and lastly, in chapter 4 of (Bates and Ian Postlethwaite 2002a), a wing-leveller control law was designed for the F-16 using H_∞ loop shaping.

There is a large amount of research done on the F-16 besides H_∞ control. Some of the other methods are listed as follows: Sliding Mode Control is used for the F-16 in (Vo and Seshagiri 2008; Seshagiri and Promtun 2008; Nguyen and Tran 2020). In (Stevens, Frank L. Lewis, and Johnson 2016), LQQ/LTR design was applied to the F-16 for different autopilots. In (Tol et al. 2014), Non-linear Dynamic Inversion (NDI) was combined with control allocation based on multivariate splines to improve high-performance tracking for the F-16, even in non-linear regions such as high angle-of-attacks. In (Reigelsperger, Hammett, and Banda 1997), μ-synthesis and dynamic inversion was applied on the lateral-directional modes of an F-16. In (Keviczky and Balas 2006), model-predictive control is applied on the longitudinal axis of the F-16. In (Lu, F. Wu, and S. Kim 2006; Shin, Balas, and Kaya 2002), LPV control is used to stabilize the F-16 over the entire flight envelope and in (Lars Sonneveldt, Chu, and Mulder 2006; L. Sonneveldt et al. 2009), nonlinear constrained adaptive backstepping is applied on the F-16 to design a stability and control augmentation system.

1.1.3. Mixed-sensitivity methods

Mixed sensitivity H_∞ controller design tries to satisfy multiple design objectives, formulated in terms of the singular values of certain closed-loop transfer functions in the system. This provides a way to specify requirements to both robust stability to *unstructured* system uncertainty, and to *nominal* performance. The design objectives relevant to this type of aircraft include low-frequency disturbance attenuation and reference tracking, and the closed-loop system must be robust to model uncertainty.

The following methods will try to exploit the relevant sensitivity functions to meet these design requirements. In (Huang et al. 2010), a comparison is made between the common S/KS method, the four-block method (Skogestad and Ian Postlethwaite 2005, p.508), and an alternated S/KS method by (Kwakernaak 2002a).

S/KS methods

In general, constraining the sensitivity function S leads to better disturbance attenuation and less steady-state error, and constraining KS leads to less control effort, more measurement noise attenuation at the input, and robustness to additive uncertainty. $\bar{\sigma}(S)$ is required to be small at low frequencies, since those are the frequencies of most disturbance signals and $\bar{\sigma}(KS)$ is to be small at high frequencies since that is where the frequencies of the noise (at the plant input) lies. W_1 is generally a diagonal matrix containing low-pass filters with bandwidth equal to that of the disturbance and W_2 a high-pass filter with crossover frequency equal to the desired bandwidth (Bates and Ian Postlethwaite 2002a).

An advantage of S/KS is the simplicity of the design. With only 2 stacked cost functions as above, the bandwidth requirement is often complementary and simple filters are enough to carry out the required shaping (Skogestad and Ian Postlethwaite 2005). Disadvantages of the standard S/KS method are that pole-zero cancellations are inherent to the problem and make stability of the closed-loop system dependent on these cancellations. A further discussion of this problem can be found in (Sefton and K. Glover 1990; Tsai, Geddes, and Ian Postlethwaite 1992). Pole-zero cancellations can induce instability when unstable poles are cancelled with compensator zeros and, due to uncertainties, the zeros are slightly moved. Besides inherent pole-zero cancellations, the S/KS method fails to provide specifications on other design objectives other than those linked to $\bar{\sigma}(S)$ and $\bar{\sigma}(KS)$. So, only constraining S and KS leads to problems when considering, for example, input disturbance signals at the plant output (SG would be relevant).

An alternated S/KS method is described in (I. Postlethwaite, Tsai, and Gu 1990; Tsai, Geddes, and Ian Postlethwaite 1992; Kwakernaak 2002a; Kwakernaak 2002b), where a rational matrix V is added as a pre-filter on the exogenous signals (reference and/or disturbance). Assuming that the plant transfer matrix has the left polynomial fraction representation $P = D^{-1}N$, then V consists of a product of the inverse of D, the polynomial coprime factor of the plant and lastly M, which is a polynomial matrix chosen in order for V to be bi-proper. The authors then proceed to tackle the problem of pole-zero cancellations by partially placing the *dominant* closed-loop poles as the roots of M. *Partially*, because only the dominant poles are reassigned as the roots of M and there are other closed-loop poles present. Besides preventing pole-zero cancellations, one could alter the performance of the closed-loop system by assigning the poles.

However, according to (Huang et al. 2010), tuning of the pre-filter V is a process that takes a lot of time and effort and it is heavily dependent on the plant, so there is not a lot of freedom in tuning W_1V and W_2V in the alternate S/KS method.

S/T and S/KS/T methods

As described in (Bates and Ian Postlethwaite 2002a), constraining T is done to ensure good reference tracking, one of the design objectives, and this motivates the use of S/T and S/KS/T methods. The S/T method is less complex than the S/KS/T method. However, as mentioned above, constraining KS as well helps to be robust against additive uncertainty as well as reducing the control effort, which is important for aircraft applications, where there are physical constraints on the actuator angles and rates.

S/KS/T approaches are the most used methods from the mixed sensitivity methods, since it provides a good balance between performance (disturbance attenuation and reference tracking) and robustness against additive and multiplicative uncertainties at the plant output, while minimizing the control effort as much as possible to stay within the physical bounds of the actuators. Some of the current day literature that use S/KS/T are (Pirat et al. 2020; Ma et al. 2021; Zhao et al. 2019; M. Zhang et al. 2021; Asali, Indriyanto, Trilaksono, et al. 2019; Ji and A. Wu 2011; Nag et al. 2013).

Four block methods

Four-block methods give the designer possibility to constrain more transfer functions where the other methods above fail to do so. Attenuating KSG and SG could minimize the effect of input disturbance on the error signal, plant output, and plant input. This approach has more exact relations between the weighting functions in each block and the closed-loop performance objectives, making the tuning process more transparent. However, interference between the weighting functions are more common since different closed-loop performance objectives can be achieved through more ways than in comparison to two-block methods (Huang et al. 2010).

An example of a variant of the four-block method that was used to enhance disturbance rejection is given in (Skogestad and Ian Postlethwaite 2005, p.508). In this case, pre-filters are used that act on the reference signal and disturbance signal and act as tuning knobs for the designer. The system includes a disturbance model G_d, which allows to incorporate modelled atmospheric turbulence. A system including disturbance rejection is shown in Figure 1.1.

The four block method with disturbance rejection aims at improving passenger comfort. More importantly for fighter aircraft, it also aims at alleviating pilot workload in countering gust. Thereby, it enables aggressive maneuvers in poor weather conditions and in general, improves tracking precision during aggressive maneuvers.

Model following

This method includes a (second-order) reference model which is obtained from inspecting the desired short period characteristics, i.e. the natural frequency and damping ratio of the response.

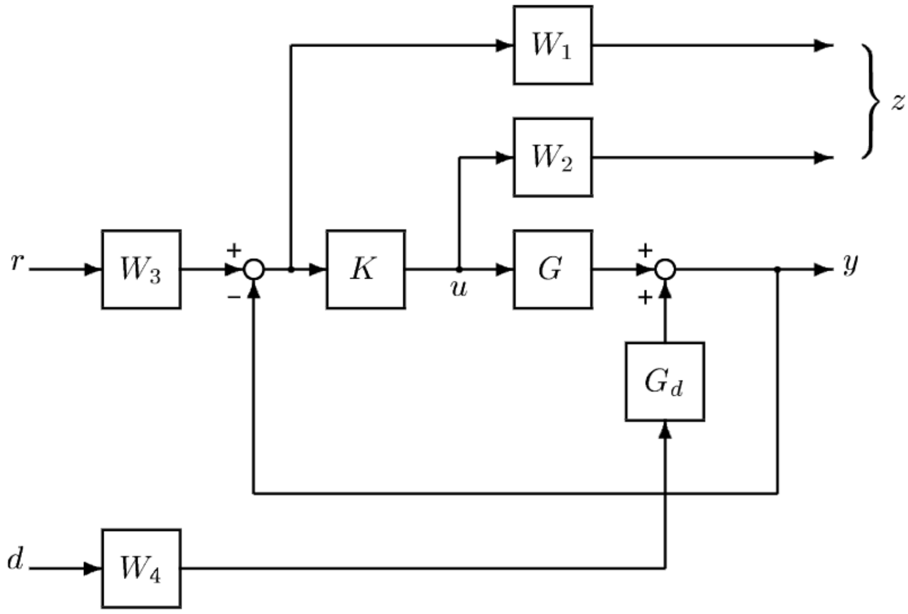


Figure 1.1: Disturbance rejection interconnection diagram (Skogestad and Ian Postlethwaite 2005, p.508)

An example of an application which includes model following is described in (Florian Sève et al. 2014; Sève et al. 2017), where the error between the output to the reference model and output to the model (which correlates with the model-matching sensitivity) is constrained by a different weighting function. When done correctly, this will lead to the system following the reference model and in turn leads to desired handling qualities, which are described in ("MIL-F-8785C" 1980). With this method, the reference model serves as a pre-compensator and due to the use of the four stacked functions instead of two for the S/KS method, it gives the designer more capabilities of loop-shaping the system. An extensive discussion on the procedure for model-following design in application to mixed sensitivity can be found in (Daniel WC Ho, James Lam, and TWK Chan 1992; D. Ho, J. Lam, and T.W.K. Chan 1994).

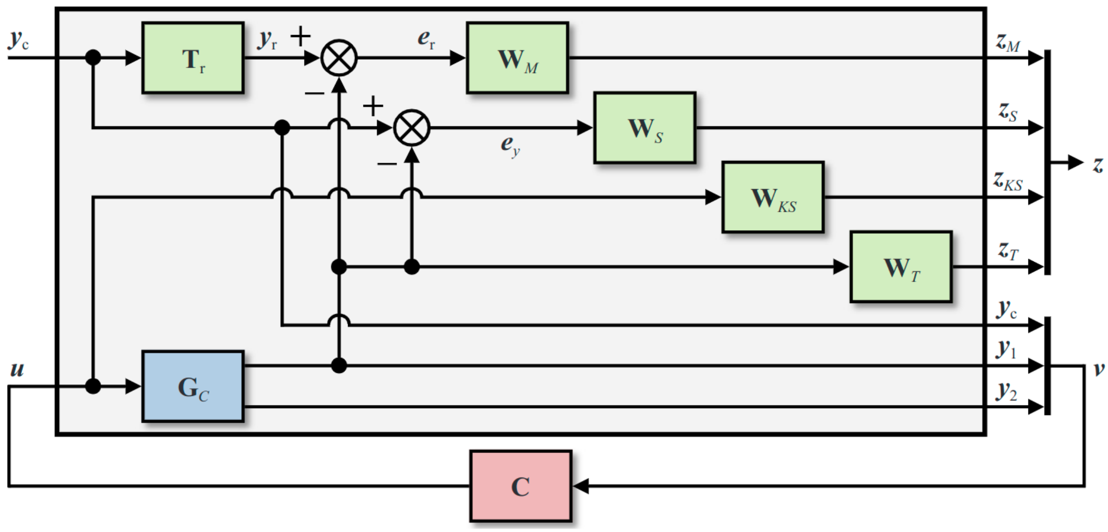


Figure 1.2: Interconnection of four-block configuration including reference model. From (Sève et al. 2017).

Weight selection

All the weights in mixed sensitivity design should be stable and proper. This is due to assumptions made on the shaped generalized plant for solving the controller K (Bates and Ian Postlethwaite 2002b). Furthermore, the weight matrices are diagonal matrices containing their filters on the diagonal elements, so that the tuning can be related to specific inputs/outputs.

For the weighting filter on the error W_s, one should bear in mind that the inverse should serve as an upper bound for the singular values of the sensitivity function S:

$$|W_s(s)S(s)| < 1 \quad \rightarrow \quad \sigma(S(s)) < \frac{1}{|W_s(s)|} \tag{1.1}$$

A similar relationship holds for the other weighting filters as well. A common choice is the following low-pass filter ((J. H. Kim and Whang 2018; Bates and Ian Postlethwaite 2002a; Feng, Guo, et al. 2022) to name a few that use this):

$$W_s(s) = M\frac{s + \omega_b}{s + \omega_b \epsilon} \tag{1.2}$$

where $M = 0.5$ is a typical value. The cross-over frequency is denoted by ω_b, and $10^{-6} \leq \epsilon \leq 10^{-4}$ a very small number to make W_s stable. This number also represents an upper bound on the steady-state error. It is known that the tuning of this weight can be a difficult task and requires skill of the designer. The aim of W_u is to constrain KS in order to limit the control effort, attenuate measurement noise signals at the plant input and improve robust stability to additive uncertainty (Bates and Ian Postlethwaite 2002a). The filter is a high-pass filter, which in case of first order looks as $M\frac{1+\tau_1 s}{1+\tau_2 s}$ where $\tau_2 >> \tau_1$.

Optimizing weighting functions

In recent years, there has been some research dedicated to the optimization of the parameters of the weighting functions of the mixed-sensitivity approach. The classical way of tuning these functions is by engineering experience and skill. However, it is not hard to understand why this would be sub-optimal, and automatic tuning of the parameters might be give a better solution. To this end, intelligent optimization methods have been researched to solve the multi-objective optimization of the weighting functions. These include genetic algorithms (Varsek, Urbancic, and Filipic 1993; Kumar and Narayan 2016), particle swarm optimization (Ali et al. 2011), ant colony optimization(Hsiao, Chuang, and Chien 2004) and quantum genetic algorithms (M. Zhang et al. 2021), just to name a few. More optimal tuning parameters have been reported, although significant computational and design effort is needed for these algorithms. Therefore, although the methods are a good option for potential fine-tuning in later stages of control design, for now it is considered beyond the scope of the current research topic, but nonetheless worth mentioning. Lastly, it is also possible to render the weighting filters parameters as individual gains and to let Systune tune these parameters. This process is called co-design and is performed in (Pérez et al. 2022).

1.2. SOF Controller Design

This chapter will give an overview of necessary and/or sufficient conditions for SOF and give an overview of the different approaches to SOF. It should be noted that the methods described below are categorized but are not exclusive to such categorization. For example, the method by (Apkarian and Noll 2006a) belongs to Direct Search methods, but can also belong to the Non-Lyapunov methods and the method by (J. Gadewadikar et al. 2006) belongs to the Ricatti methods, but can also belong to the two-stage approaches. In general there are categories where combinations of groups of methods can be used.

In subsection 1.2.1, the SOF problem is defined, after which in subsection 1.2.2 iterative LMI (ILMI) approaches are presented. Then, in subsection 1.2.3, rank minimization methods are presented, in subsection 1.2.4 two-stage approaches are covered, in subsection 1.2.5 direct synthesis methods are described, in subsection 1.2.6 decoupled Lyapunov methods are described, and finally in subsection 1.2.7 non-Lyapunov methods are elaborated on.

1.2.1. SOF problem definition

The SOF controller gain K is known to stabilize the closed-loop system if one of the following equivalent statements is true:

$$\text{The eigenvalues of } A + BKC \text{ are all in the LHP} \tag{1.3}$$

$$\text{There exist a } P = P^\top > 0 \text{ s.t. } P > 0 \text{ and } \mathrm{Sym}\{P(A + BKC)\} < 0 \tag{1.4}$$

$$\text{There exist a } Q = Q^\top > 0 \text{ s.t. } Q > 0 \text{ and } \mathrm{Sym}\{(A + BKC)Q\} < 0 \tag{1.5}$$

The methods in the following sections aim to deal with either the non-convexity and/or the non-differentiability (non-smoothness) of the conditions above.

1.2.2. ILMI heuristic approaches

In this section the most important and relevant approaches that use iterative LMI's are given. These approaches try to deal with the non-convexity of the problem above and work by alternatively freezing variables of the BMI, to render the said BMI as LMI, which is convex and solvable in polynomial time. A history is given of these approaches as well as current day developments.

P-K iteration

This is the simplest form of the proposed ILMI heuristics and it is also one of the first approaches that was proposed (El Ghaoui and Balakrishnan 1994). The idea is to alternatively freeze/fix the Lyapunov matrix P (or Q), or the controller gain K at each step in the iteration. By doing this, one renders the BMI (1.4) (or (1.5)) as a convex LMI, which can be solved for P or K in polynomial time.

One of the advantages to this approach is that structural constraints to K can be imposed, since the corresponding LMI is convex in K. However, the objective function value of the optimization problem has no control interpretation and the terminal step consequently gives no knowledge or characterization whether the solution to the objective function is a local minimum/plateau, while rapid convergence to a local plateau is very probable for most of the ILMI methods descibed in this section.

Path-following method

The problem of having no control interpretation of the objective function in P-K iteration is tackled in the path-following method that was developed a few years after P-K iteration by (Hassibi, How, and S. P. Boyd 1999). In this method, a level of decay rate is introduced, which is defined as the maximal real part of the eigenvalues of A_{cl}. The procedure tries to convexify the following BMI:

$$\mathrm{Sym}\left\{P(A + BKC - \alpha I)\right\} < 0 \tag{1.6}$$

where α is the level of decay rate. (1.6) is convexified by iteratively freezing variables, similar to P-K iteration, by a first order perturbation approximation around P (or Q) and K with a small search step. The biggest downside to this approach is that the convergence is very dependent on the step size of the approximation. If the step size is too small, the convergence might take too much time. In contrast, if it is too large, the convergence might fail. Also, the initial gain K is again chosen to be zero, there is no intelligent initialization procedure for this method and this consequently induces conservatism in the results.

Ricatti approaches

The approach in (Cao, Y.-X. Sun, and Mao 1998; Cao and Y.-X. Sun 1998) works by solving Algebraic Ricatti Equations iteratively to find a stabilizing K. However, similar to other ILMI methods, the algorithm might converge to a local minimum rapidly, without being able to conclude anything about the dead-point.

The approaches by (J. Gadewadikar et al. 2006; Jyotirmay Gadewadikar, Bhilegaonkar, and F. L. Lewis 2007; Jyotirmay Gadewadikar, Frank L. Lewis, and Abu-Khalaf 2012; Jyotirmay Gadewadikar, Frank L. Lewis, Subbarao, et al. 2009) are based on AREs as well. Advantages to these approaches are that no random or null initial gain K_0 has to be used. This is in contrast to several iterative methods such as linearized convex-concave decomposition, path-following method, etc. This leads to the algorithm being more likely to converge to a local (or global) optimum instead of a local plateau. Furthermore, one can specify the desired L_2 gain bound (γ^2) in advance, but if the value is taken too small, the ARE will not converge. γ thus serves as a design variable which can be tuned.

A disadvantage to this approach is that there is also no dead-point criterion with a control interpretation. The algorithm just stops whenever the difference in gains between iterations is sufficiently small. However, the authors claim the algorithm most often leads to convergence and good results are presented using this approach.

Convex-concave decomposition

This approach introduced by (Tran Dinh et al. 2012) works as well by solving (1.6), which is BMI due to its term containing the product of P and K. α is again the exponential decay rate, similar to the path-following method. Instead of dealing with (1.6) directly, as was done for the path-following method, convex-concave decomposition works by first decomposing (1.6) into convex and concave terms $G(.)$ and $H(.)$. The convex term $G(.)$ can be transformed into an LMI with use of the Schur complement.
This procedure has been applied to e.g. (Debrouwere et al. 2013; M. Saeki and C. M. Saeki 2018) for dynamic controllers. The procedure has been applied for the SOF in (Dingchao Ren 2022) with the H_2-norm and negativity imaginary system constraints, the latter which is defined by no poles in the RHP for all frequencies.

The procedure runs into the same problems as the approaches described in the former sections in this chapter. The solution might converge to a local plateau rapidly without any dead-point criterion characterization. In contrast to P-K iteration, the algorithm for SOF includes the exponential decay rate α, which has a control interpretation as it is the maximum real eigenvalue of the closed-loop system dynamics. Another one of the advantages is that, in contrast to the path-following method, one only needs to linearize the concave part $H(.)$ instead of the whole problem. Furthermore, the procedure has no need for tuning the step size parameter, which was the case for the path-following method.

Dual-iteration approach

Introduced by (Tetsuya Iwasaki 1999), this approach works by iterating between two stages where the first version of a dual formulation of (1.6) is first solved with a fixed state-feedback gain. After that, the second version is solved for a fixed output feedback gain.

The main disadvantage of this approach is the quasi-bilinearity with the terms αP and αQ. This requires quasi convex optimization at each step. Also, there is no possibility to impose structural constraints on the gain during iteration, except at the terminal step, which then does not guarantee convergence with this controller gain structure. However, at each step, P and Q are optimized over, which increases the designer freedom. Furthermore, the approach relies on an initial state feedback gain, which is less arbitrary than $K_0 = 0$, which is used in other ILMI methods, such as convex-concave linearization.

In (Holicki and C. W. Scherer 2021), the problem is revisited and the algorithm is extended to robust gain-scheduled controller design.

Genetic algorithms

In (Marrison and Stengel 1997; Sekaj and Veselý 2002; Arfiadi and Hadi 2001; Harikumar, Dhall, and Bhat 2019), a SOF controller is designed using a genetic algorithm. With a fixed initial matrix that is any square matrix, LMI's are solved to obtain the SOF gain and the Lyapunov matrix. The initial matrix serves as a design variable and is changed until a satisfactory solution is obtained. The fitness of the solution is judged by a penalty function. Evaluating the internal stability of the closed-loop system with the eigenvalues, one can construct a penalty function which includes penalties for instability and undesired performance characteristics of this closed-loop system. Advantage to this approach is its simplicity and ease of implementing. However, there is no clear reasoning for finding the initial design variable matrix other than randomized approaches, and convergence to a local minimum (or convergence at all) is heavily dependent on this initialization. Rapid convergence to a local plateau is a strong possibility when running this algorithm.

Remarks on ILMI approaches

ILMI approaches are in general easy to implement and coding is not that complex, with convergence to a local solution being a good possibility (Sadabadi and Dimitri Peaucelle 2016). However, for all the ILMI approaches, the results are heavily dependent on the initialization procedure for either K or the design matrices. Rapid convergence to a local plateau is common and is again dependent on that initialization step. Lastly, robust synthesis may be achieved through the so-called "Lyapunov Shaping Paradigm" where the Lyapunov matrix obtained holds for all realizations of the uncertain plant.

1.2.3. Rank minimization approaches

The following methods are based on the following inequalities which are necessary conditions for a stabilizing controller for the system.

$$B^{\perp}\mathrm{Sym}\{AQ\}B^{\perp^{\top}} < 0 \tag{1.7}$$

$$C^{\perp^{\top}}\mathrm{Sym}\{PA\}C^{\perp} < 0 \tag{1.8}$$

$$W(P,Q) = \begin{bmatrix} Q & I \\ I & P \end{bmatrix} \geq 0 \tag{1.9}$$

$$\mathrm{rank}(W(P,Q)) \leq n \tag{1.10}$$

where A, B, and C are the system equations, P and Q are the Lyapunov Matrices and n is the order of the plant.

Min/max algorithm

This algorithm from (J. C. Geromel, Souza, and Robert E Skelton 1998) works by directly and alternatively optimizing both (1.7) and (1.8) with a given P and Q, respectively. The idea exploits the fact that the rank constraint (1.10) holds if $PQ = I$.
Downsides of this algorithm is that the procedure has no dead-point criterion, no local optimum convergence is guaranteed, and the objective values for the optimization have no control interpretation and are not strictly non-increasing. Furthermore, the algorithm stops when (1.10) holds, which is not always *exactly* the case.

XY-centering algorithm

This method from (Iwasaki and R. Skelton 1995) is similar to the min/max method, but in comparison to the min/max algorithm described above, this algorithm includes strictly non-increasing criteria values during iterations. Furthermore, (1.10) (or equivalently $PQ = I$) does not need to exactly hold for the terminal step. These two things are advantages over the min/max algorithm. For this algorithm and rank minimization algorithms that follow, the stopping criterion is dependent on whether (1.4) is feasible for a given P that comes from a converged $||P_k Q_k - I||$ (at least close to zero), where k is the iteration number.

Projection methods

The *alternating* projection method (Grigoriadis and Robert E Skelton 1996) works by searching (P, Q) in the intersection of the non-convex set where (1.10) holds and the convex set where (1.8)-(1.9) hold. It finds this by taking orthogonal projections on each set alternately. The projection on the non-convex set of constraints is found by computing the eigenvalue decomposition of the positive semi-definite matrix $W(\tilde{P}, \tilde{Q})$. The projection on the convex set of constraints is the solution to an optimization problem subject to (1.8)-(1.9). Thus, as a major advantage it optimizes over both P and Q at each step, which are included in both projections. The major disadvantage is that convergence cannot be guaranteed.

The *directional* alternating projection algorithm is an alteration on the standard alternating projection method, where directions are chosen based on the geometry of the constraint sets. It is proposed in (Grigoriadis and E. B. Beran 2000) and its aim is to have a faster convergence than the non-directional alternating projection algorithm. However, convergence is still not guaranteed, and there is no indication of improvement over iterations. The algorithm was improved in (Ebihara, Tokuyama, and Hagiwara 2004) to include structural constraints on the SOF gain by keeping the controller variable in the convex LMI.

A third variant of a projection algorithm is the penalty function method by (S.-J. Kim, Moon, and Kwon 2007). For this method, the (local or global) convergence of the penalty function is also not guaranteed. Furthermore, convergence is heavily dependent on initialization of the design parameters of the penalty function for which only a trial-and-error tuning process is available based on guidelines from the Langrangian method, described later in this section. This method is easy to implement, though, due to the simplicity of the objective function.

In (E. Beran, Vandenberghe, and S. Boyd 1997; Chanekar, Chopra, and Azarm 2017), the problem of structured SOF is tackled head-on in the optimal control framework. First, the problem of SOF is formulated in a Linear Quadratic (LQ) function. Next, this problem was reformulated into an optimization problem with a linear objective function and a BMI constraint. After this, a combination of Generalized Benders Decomposition (GBD) and a Gradient Projection Method (GPM) was used in combination, iteratively, to compute a solution. Necessary conditions for the proof of K being a local minimum can be obtained, which is an advantage over other ILMI methods where no information on the minima was present. However, this approach does not include proof of convergence to a local (or global) minimum yet, which is still part of future research and convergence is limited by the first-order optimization problem. This latter problem is tackled in (Cheng et al. 2022), where convergence of the LQ problem is improved by using a second-order optimization.

Cone complementarity linearization approach

Cone complementarity linearization (CCL) was first introduced by (Ghaoui, Oustry, and Aitrami 1997). The idea of this approach works by minimizing Trace(PQ), which is minimal when (1.10) holds[1]. This problem is nonlinear, which is the reason for applying a Frank and Wolfe linearization procedure to the problem. In (Leibfritz and Optim 2001), an additional line search step is applied to the iteration to improve convergence.

An advantage of this approach is that at each step, one can optimize over the Lyapunov matrices, which allows for more design freedom. However, similar to most rank minimization methods, there exist no dead-point criteria for this approach, so it is hard to determine whether the local minimum/plateau is also a global solution to the problem.

Lagrangian methods

Proposed in (Apkarian, Noll, and Tuan 2003) and (Noll, Torki, and Apkarian 2004), the augmented Lagrangian method aims at minimizing a tangent version around some point of a penalty function including a Lagrangian multiplier matrix. As reported in the aforementioned papers, this method shows improved numerical results to CCL, but comments and remarks on the advantages and disadvantages are of the same nature.

[1] (1.10) holds iff $PQ = I$. Also, $\mathrm{Tr}(PQ)$ is minimal when (1.10) holds ($\mathrm{Tr}(PQ) = n$ iff $PQ = I$).

Another method is the partially augmented Lagrangian method proposed by (Ankelhed, Helmersson, and Hansson 2012). This method aims at minimizing a reformulation of (1.10)[2]. An advantage over the augmented method is that the rank constraint is tackled directly through the reformulation. Other than that, similar comments hold with the method described above.

T-K iteration

A novel approach is proposed in (Feng, She, and Xu 2019). T-K iteration works with two steps. First, an initial coordinate transformation matrix (CTM) is found by CCL (see subsubsection 1.2.3). Once this CTM is found and fixed, a solution is found by first optimizing for the de-composed Lyapunov matrix elements P_1 and P_2 and Y to find a gain $K = YP_1^{-1}$. With this gain, one can subsequently optimize again to find the Lyapunov matrix P for the fixed K.

Significant advantages of this approach are that, in contrast to many other methods described in this section, this method produces locally optimal solutions instead of converging to a local plateau, which would yield conservative results. Furthermore, there exists a dead-point criterion evaluation index in N. When $N = P_{11}^{-1}P_{12} = 0$ (or $P_{12} = 0$), the solution is a local optimum. For many other iterative methods, the algorithm stops when there is no better solution found, but this would provide no intelligent characterization of the dead-point. Lastly, at each iteration, P is optimized over, allowing for more design freedom, similarly to CCL. Results are shown to be promising for low- to moderate sized systems, where the algorithm is presented to be competitive with HIFOO.

In (Feng, Guo, et al. 2022), the problem is extended to a weighted sensitivity approach, and a multivariable PID controller is synthesized by first transforming the fixed-order PID problem into a SOF problem and then applying T-K iteration on it to obtain the controller.

Remarks on rank minimization methods

Advantages of rank constraints methods include that for most of the methods, P (or Q) are optimized over at each iteration, allowing for more design freedom. Drawbacks include that for most methods (with exception of T-K iteration and augmented Lagrangian methods) there exists no dead-point criterion characterization. The solutions mostly stop when there is no better solution found. Despite this, the algorithms often are shown to converge to benchmark examples from *Compl$_e$ib*[3]. Another downside to rank minimization methods is that robust synthesis is not available, since a common-to-all Lyapunov matrix[4] that satisfies (1.7)-(1.10) for all vertices on the polytopic model does unfortunately not guarantee that the resulting K satisfies (1.4) for all vertices.

1.2.4. Two-stage approaches

In recent years, the two-stage approaches have gained popularity in the field of research on SOF. A downside of many methods described before, is that no clear approach for finding initial values for the gains are used (often they are just set to null). Introduced by (D. Peaucelle and D. Arzelier 2001; Arzelier, Peaucelle, and Salhi 2003) and further researched by (Mehdi, Boukas, and Bachelier 2004; X. Li and Gao 2014; H. Zhang et al. 2014; P. Li et al. 2021) to name a few, this method works by first finding an initial stabilizing *state*-feedback gain, after which this initial gain is used to find the stabilizing SOF controller.

Two-stage approaches have received a great amount of scientific attention in recent years as well (Sereni et al. 2018; B. Sereni 2023; D. Ren, Xiong, and Daniel W.C. Ho 2021; Y. Ren et al. 2021; Denis Arzelier et al. 2010). In (Sereni et al. 2018), the two-stage approach is used to design a robust SOF controller with relaxed LMI conditions. These include parameter-dependent Lyapunov functions to reduce conservatism when dealing with uncertainties. In (B. Sereni 2023), the two-stage method is applied in a gain-scheduling framework to ensure stability in the *switching zones* between different linearized operating points. In (D. Ren, Xiong, and Daniel W.C. Ho 2021) for H_∞ performance, first an initial Output-Injection (OI) controller is found after which the SOF gain is found using this fixed OI gain. In (Y. Ren et al. 2021) first an initial state-feedback gain is found after which sufficient conditions for the SOF controller is found using the Kalman-Yakubovich-Popov lemma (Tetsuya Iwasaki and Hara 2005) and a homogeneous polynomially parameter-dependent technique to reduced conservatism. In (Denis Arzelier et al. 2010), randomized hit-and-run algorithms are used to find many guesses of initial stabilizing state-feedback gains, with shown efficiency in finding feasible solutions.

Two-stage approaches are less likely to converge to a local plateau, since the initial gain is not defined as random or zero, but rather a (quasi-)optimum initial gain is produced, which will increase the likelihood of converging to a local optimum instead of a local plateau.

[2] (1.10) is satisfied iff $a_{n-1}(I-PQ) = 0$ where a_i are the non-zero characteristic polynomial coefficients of a matrix (Helmersson 2009).

[3] Link to COMPlib

[4] According to the Lyapunov shaping paradigm.

The two-stage approaches provide only sufficient conditions for the existence of stabilizing SOF gains, since the choice of stabilizing state-feedback gain is not unique and different methods exist for finding it. These include, for example, methods with Ricatti equations (J. Gadewadikar et al. 2006), but also randomized methods such as the one described in (Denis Arzelier et al. 2010).

1.2.5. Direct synthesis methods

The methods described above use some sort of iterations to either alternatively fix P or K to render a BMI $\rightarrow$ LMI, or try to iteratively minimize the rank constraint (1.10). The following methods exploit certain conditions in which stability of the system is proven or direct search is exploited to obtain a solution without need for iterating.

Direct search methods

Direct search (DS) methods can be traced back to as early as the late 1950s when it was pioneered by algorithms such as (Hooke and Jeeves 1961; Box 1957). Direct search methods are mostly neglected by the control community in researching SOF stabilization algorithms. This statement is supported by the fact that it was not included in the papers by (Sadabadi and Dimitri Peaucelle 2016; Syrmos et al. 1997) and the fact that it was strongly criticized in (Apkarian and Noll 2006a) by its inability to deal with nonsmoothness issues such as in spectral abscissa maximization and H_∞ minimization. Nonsmooth optimization methods were therefore combined with the DS method called multidirectional search method in the paper to reach adequate H_∞ performance. Lastly, DS algorithms could not provide convergence guarantees until the early nineties in (Torczon 1989). This was also the time when (iterative) LMI methods started to arise though, bringing DS methods even more to the background in SOF research.

Two other noteworthy DS methods besides multidirectional search are mesh-adaptive DS (Audet and Dennis Jr 2006) with convergence guarantees even for nonsmooth functions and the Nelder-Mead algorithm (Nelder and Mead 1965) improved in (Emile Simon 2011) to improve convergence on nonsmooth functions. These methods were compared on performance in (Henrion 2006) on benchmark examples of *COMPlib* and it showed that the algorithms were very efficient, being able to solve the problems in seconds. However, this paper only dealt with SOF stablization and did not extend it to H_∞ performance.

Despite the call for more research in this field by (E. Simon 2011), there has been little research on DS methods for SOF since then. It can be noted that DS methods can be combined with other methods to refine their potentially conservative results. The gain K resulting from these methods could be set as initilization point for the DS algorithm to improve the solution.

Information constraints

The method by (Sahoo et al. 2019) is based on necessary and sufficient conditions introduced in (Rubió-Masssegú et al. 2013). (Sahoo et al. 2019) improves on these conditions by considering the off-diagonal terms of the Lyapunov matrices, which were simply neglected in the former paper. This used to lead to more restrictions in the developed criteria, and consequently to a more conservative result. The idea of the novel paper works by decomposing matrices corresponding to the multiplication terms in (1.4) and (1.5), that render them BMI. LMI criteria are subsequently derived, including for H_∞ control and pole placement constraints. In (Gopmandal and Ghosh 2021), the method by (Rubió-Massegú et al. 2013) is further extended to include suitable initialization matrices based on the L_2 bound γ and to include norm-bounded uncertainties in the design phase for robust synthesis.

It is worth noting that through the Lyapunov matrices (and their components) and other design variables, structural constraints can be indirectly imposed on the controller gain K. As an example, P_{22} is commonly chosen to be block-diagonal and another design variable Y_B is chosen to have the same desired structure as K, because $K = P_{22}^{-1} Y_B$ in this approach. This is an advantage over approaches that are unable to impose structural constraints on the SOF gain matrix.

Convexifying by plant structure

This group of methods aims to convexify the problem by rendering the generalized plant in a certain way, either through assuming some structure on the plant or by transforming the plant. The LMIs can then be solved using standard LMI solvers, like MOSEK, SDPT3 or SEDUMI. No iteration/optimization is necessary to go BMI $\rightarrow$ LMI, which is in contrast to the methods described in the previous sections on iterative LMI heuristics and rank constraints and it can be considered a clear advantage to these approaches.

The method by (Crusius and Trofino 1999) which yields sufficient conditions for a set of stabilizing gains assumes that BP or CQ is close to commute. However, in general this does not hold for most plants. In (Prempain and Ian Postlethwaite 2001), structural constraints for existence of a stabilizing gain include assumptions which, among others, state that the system is a minimal realization, the plant is square and CB is of full row-rank. In (Ebihara, Dimitri Peaucelle, and Denis Arzelier 2014), plant conditions for convexifying the problem are that D_{zw} and D_{yw} (or D_{zu}) are null, that the matrix B_w can be partitioned as $[B_u B_{w,2}]$ for some matrix $B_{w,2}$, which can be null, and lastly, $D_{zu}^T D_{zu} \geq I_r$. This last assumption is the most limiting assumption, as this is often not the case for generalized plants. This method allows to generate a subset of stabilizing controllers instead of one resulting controller though, which is an advantage, but this subset may be empty and is dependent on the matrix D_{zu}.

In the novel method from (Chou and Wei 2020), the generalized plant is transformed in such a manner that necessary and sufficient conditions can be derived for the existence of SOF gains. This method resembles the method described in (C. Scherer, P. Gahinet, and Chilali 1997) for the full-order Dynamic Output Feedback (DOF) controller design, but extents it to the SOF framework.

The assumptions for the structure of the generalized plant are as follows:

1. The first assumption states that there needs to be an even number of states. In case of an odd number of states, an isolated system with odd dimensions and appropriate parameters can be augmented to the given system to give it even dimensions.
2. The second assumption states that the number of inputs and number of outputs must both be greater than **half** the number of states.
3. The plant must be transformed if necessary to obtain the desired form.

The method shows great promise when the results were compared to methods such as HIFOO and Hinfstruct, with the authors presenting H_∞ norms lower than for both of the powerful non-smooth non-convex optimization methods. Unfortunately, no computational times are presented and it might be interesting to see whether the algorithm outperforms the other methods in that regard.

1.2.6. Decoupled Lyapunov matrices

The following methods will aim to reduce conservatism by decoupling the Lyapunov matrices from the system matrices thus giving the designer an extra degree of freedom by allowing to optimize over both the Lyapunov matrices and controller gain separately.

Resilient Approach

This approach proposed in (D. Peaucelle, D. Arzelier, and Farges 2004) aims at reassembling the conditions in such a manner that the Lyapunov marices are decoupled from the system gain. Subsequently, rank minimization methods can be applied since the non-LMI part of the conditions can be reformulated as a rank constraint, similar to what was done in subsection 1.2.3. A big advantage of this approach is that simultaneous stabilization and robust stabilization are easily achievable by copying the LMI's for each vertex in the polytopic model. This is done with a common-for-all Lyapunov matrix P, by considering the "Lyapunov Shaping Paradigm" (C. Scherer, P. Gahinet, and Chilali 1997).

Furthermore, compared to methods in subsection 1.2.3, the rank constraint is of much smaller size, since the rank needs to be equal to the size of the input vector instead of the size of the plant. The latter is in general much larger. Also, instead of finding only one gain, a set of controllers is obtained. This allows for imprecisions in the solution to K, since multiple options for viable controllers are obtained. Resilience in this sense differs from robustness, since robustness implies uncertainties in the modelling of the closed-loop system and resilience implies uncertainties in the implementation of the controller. Lastly, structural constraints can be easily imposed, because variables directly related to the control gain enter in the constraints explicitly.

This method has been combined with CCL (see subsubsection 1.2.3) in (D. Peaucelle and D. Arzelier 2005) for the SOF problem and in (Tian et al. 2020) for SOF control of semi-Markov jump systems.

S-variable methods

Proposed by (J. Geromel, M. C. d. Oliveira, and Hsu 1998) and extensively described in (Ebihara, Dimitri Peaucelle, and Denis Arzelier 2015), in the S-variable approaches the BMI is transformed into a variant by use of the Finsler's lemma (i.e. inverse of the elimination lemma). In this variant, closed-loop stability is established by investigating if there exist slack variables F and G, and Lyapunov matrix P such that the conditions are satisfied. The key of this method is that it reduces conservatism by decoupling the system dynamic (with the gain) and the Lyapunov matrix by introducing slack variables. This will lead to more design freedom as it inherently adds another degree of freedom by optimizing over both separately. Iterative methods described above can subsequently be applied to this BMI to convexify the problem to LMI.

S-variable methods can be easily extended to robust stabilization with the search for parameter dependent Lyapunov matrices for all the vertices of the polytopic model, with reduced conservatism compared to the BMI (1.4) and (1.5). This is done with a common gain and S-variables for each vertex. This is done in both

(Felipe and R. Oliveira 2021; Ebihara, Dimitri Peaucelle, and Denis Arzelier 2015; Sereni et al. 2018) and in (D. Arzelier et al. 2018) for probabilistic uncertainties. The biggest downside is that the size of the BMI is doubled compared to the coupled BMI's, thus increasing the computational complexity with an increase in problem size.

1.2.7. Non-Lyapunov approaches

The following methods will make no use of optimization over any Lyapunov matrices:

The sum-of-squares method by (Chesi 2013) works by making use of the Routh-Hurwitz table characteristic polynomials which define stabilization. Using convex optimization on the constraints of these polynomials, one can guarantee robust stabilization for systems that depend affinely on the uncertainties. However, this method can often not be used in practice due to the fact that dimensions of the polynomials explode with increase in size of the plant, thus rendering the method impractical for moderate to large systems.

These methods will try to optimize the following problem. This problem is non-convex and non-smooth.

Given matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times r}$, $C \in \mathbb{R}^{l\times n}$, find a matrix $K \in \mathbb{R}^{r\times l}$ such that the spectral abscissa of matrix $A + BKC$ is strictly negative.

A group of methods that has a lot of potential are the randomized algorithm (Denis Arzelier et al. 2010) and in particular the ray-shooting algorithm. Introduced in (Yossi Peretz 2016), the ray-shooting algorithm is a general randomised method that is able to find a global minimum on any non-convex set (either smooth or non-smooth). The method is extended beyond stabilization and pole placement to structured SOF in (Y. Peretz 2017), H_∞ performance is obtained on a PID controller in (Yossi Peretz 2018) and proof of convergence (by probability) in the framework of robust stabilization is shown in (Peretz, Merzbach, and Moyall 2020). Advantages of these approaches are the fact that they are able to find global minima instead of local minima. Furthermore, the algorithms are able to find a set of stablizing controllers which is an advantage over other methods that are only able to find one gain.

In the 1990s and early 2000s, the focus for SOF research was mostly on solving LMI's including Lyapunov matrices. The following methods introduced a change in paradigm. The method for HIFOO in (Burke et al. 2006) tried to tackle the non-convex non-smooth problem described above in two-phases, first by classical quasi-Newton updating and then by gradient sampling. Gradient sampling is stochastic by nature, so different runs give different solutions, even with consistent initial starting point.

The algorithms for Hinfstruct and Systune (Apkarian and Noll 2006b) work with Clarke's subgradient calculus (Clarke 1983). The methods are deterministic, in contrast to HIFOO. However, randomized initial points lead to different local minima. These algorithms have been widely incorporated in MATLAB toolboxes and are the standard for fixed-structure controller synthesis. They are able to cope with any control structure imposed by the designer (e.g. number and configuration of loops, complexity of transfer functions, gains), but also due to their computational efficiency. LMI approaches are prone to be much slower with increase in plant size due to the Lyapunov matrices containing most of the decision variables that are to be tuned. The methods described in this section only need to tune the controller parameters, which have much smaller dimensions than the Lyapunov matrices. In (Apkarian, Dao, and Noll 2015) the issue of parametric robust synthesis is addressed for the non-smooth non-convex optimization methods.

While powerful, the Hinfstruct and Systune in this section are dependent on the Robust Control Toolbox in MATLAB, which is not open-source. This is in contrast to methods above, which can be solved using open-source LMI solvers. Furthermore, the methods only produce local minima and no proof of convergence to a global minimum is present.

Conclusion and Future Research

The purpose of this survey was to provide a bibliographic review on the current state of SOF synthesis in the framework of H_∞ mixed-sensitivity control of fighter aircraft and to define a structure for future research on this matter. This was done as follows.

First, the motivation for researching SOF was mostly explained by the desire for simple controllers, which are much easier and cheaper to use in practice. This would aid the field of aeronautical engineering by simplifying the often long and expensive certification process of flight control systems. After this motivation was established, the gap in research was argued to be a lack of quantitative comparisons of SOF results, which lead to the inability to accurately draw conclusions on the competitiveness of the SOF methods with respect to already established commercially available software. The research questions were formulated to try to fill this gap and the subsequent chapters aimed at answering **Q1** and **Q2** by investigating what was done on H_∞-control in literature and investigating current state-of-the-art methods on SOF synthesis, while elaborating on their strengths and limitations. The application, the F-16, was also investigated and its characteristics and its implications on the design process were presented.

Future research will aim at answering **Q3**, **Q4**, and **Q5** by reproducing SOF algorithms on the generalized plant and analyzing their stability, performance and robustness in light of uncertainties in the model. Comparisons will be made between these methods and commercially available methods on some benchmark models to draw conclusions on their competitiveness. This will allow one to answer the main research question **Q** on the competitiveness of Lyapunov-based SOF methods for H_∞-control of fighter aircraft. In the end, this will allow one to achieve the research objective of gaining more insight into the performance of different SOF algorithms.

2

Scientific Article

Review of H_∞ Static Output Feedback Controller Synthesis Methods for Fighter Aircraft Control

A.D.P. Schoon*
Delft University of Technology, Delft, The Netherlands

To gain more insight into the performance of state-of-the-art Static Output Feedback (SOF) controller synthesis methods for H_∞-control, quantitative comparisons are made between Lyapunov methods and well-known established non-smooth optimization methods, i.e. Hinfstruct and HIFOO. Three methods were deemed to be the most promising to compete and were bundled into one toolbox named SOFHi. The algorithms were extended to incorporate structured SOF and a variant of SOFHi was proposed to significantly improve upon the computational efficiency of the original implementation. Extensive comparisons show that SOFHi was able to compete with the established non-smooth methods and even able to significantly outperform one of them. Lastly, an elaborate flight control benchmark example is given to showcase the effectiveness of the algorithms, which involves the design of a gain-scheduled normal acceleration Control Augmentation System (CAS) for a highly maneuverable fighter aircraft.

I. Nomenclature

Latin	**Description**	**Unit**
A	state matrix	
a_n	normal acceleration	[g]
B_u	control input matrix	
B_w	exogenous input matrix	
C_y	measured output matrix	
C_z	evaluated output matrix	
D_{yu}	input to measured output	
D_{yw}	exogenous input to measured output	
D_{zu}	input to evaluated output	
D_{zw}	exogenous input to evaluated output	
e	tracking error	[g]
e_{ref}	model-matching error	[g]
F	feedforward function	
F, Z	slack variable matrices	
G	aircraft model	
K	gain matrix	
m	mass	[kg]
M	model-matching sensitivity function	
N	local minimum evaluation	
P, Q	Lyapunov matrices	
q	pitch rate	[rad/s]
S_{a_n}	norm. acc. output sensitivity function	
S_i	input sensitivity function	
T_{a_n}	complementary output sensitivity function	
T_{c1}	constant matrix to build T_y	
T_i	complementary input sensitivity function	
T_{ref}	reference model	
T_y	coordinate transformation matrix	
T_{zw}	exogenous input to evaluated output	
u	input vector	
w	exogenous input vector	
W	weighting filter	
x	state vector	
X_{COG}	center of gravity position	[m]
y	measured output vector	
Y	decision variable matrix	
z	evaluated output vector	
$\mathfrak{K}, \mathfrak{C}, \mathfrak{L}$	sets of gain matrices	

Greek	**Description**	**Unit**
n/α	load factor per angle of attack	[g/rad]
γ	H_∞ performance index	[-]
ϵ	integrated control error	g·s
ζ_{sp}	short period damping ratio	[-]
ω_{sp}	short period natural frequency	[rad/s]
ω_{BW}	bandwidth	[rad/s]

*M.Sc. student, Control & Simulation division, Faculty of Aerospace Engineering, P.O. Box 5058, 2600GB Delft, The Netherlands; adpschoon@gmail.com, a.d.p.schoon@student.tudelft.nl.

II. Introduction

The problem of Static Output Feedback (SOF) controller synthesis is well-known, and has garnered a lot of attention from the field of research in recent decades. It is a challenging problem, which is why it is still an open problem to this day. The challenge lies in the fact that the problem is a Bilinear Matrix Inequality due to terms containing a product between decision variables, which makes the problem non-convex by nature and it becomes non-smooth when using the problem formulation in the space of controller parameters [1]. It is shown in [2] that these problems are generally NP-hard to solve.

Solutions to the SOF problem can be categorized in three main groups: BMI solvers, Lyapunov methods, and non-Lyapunov methods. BMI solvers that try to solve BMI's directly are heavily dependent on initial conditions, and even then, most often do not find solutions [3]. This paper will therefore only investigate the latter two groups of methods. Examples of Lyapunov methods include iterative Linear Matrix Inequality (LMI) methods [4–8], iterative rank minimization methods [9–11], and direct search methods [12–14]. Most often, iterative LMI (ILMI) methods aim to render the original BMI into a convex LMI by fixing one of the decision variable matrices inside terms containing a product between two decision variable matrices. The LMI problem can then be described as a Semi-Definite Programming (SDP) optimization problem, which can be solved efficiently by interior point optimizers, as described in [15, Section 1.4.4]; examples of such optimizers are MOSEK or SDPT3. A disadvantage of iterative Lyapunov methods is that their computational performance is limited by the large amount of decision variables in their problem formulation when compared to non-Lyapunov methods. In contrast, direct search methods are non-iterative and are thus more numerically tractable. However, they mostly describe sufficient conditions and are generally more conservative due to additional assumptions made about the plant to make the problem convex. Iterative methods are often derived from these sufficient conditions though, to make the method less conservative by optimizing parts of the conditions in an iterative manner. For example, [16] is based on conditions in [12], and similarly, [17] is based on conditions in [13]*.

Examples of non-Lyapunov methods are the non-smooth optimization methods Hinfstruct and HIFOO, which are described in [18, 19], respectively. A disadvantage of these methods is that they cannot guarantee robustness to uncertainties in the plant, at least not in the same way as the Lyapunov methods can [3]. Furthermore, they do not allow for much flexibility and transparency in their approach as they rely on rather sophisticated theoretical tools (non-smooth optimization), without much control interpretation in said approach. They are, however, extremely efficient and provide exceptional performance in terms of performance index γ, and have therefore grown into the most well-established methods. For example, Hinfstruct has been implemented in MATLAB's Robust Control Toolbox.

Further extensive comparisons of the aforementioned group of methods can be found in [3, 20]. However, these papers compared the methods mostly qualitatively, explaining the different advantages and disadvantages of each method. Decisive conclusions on the competitiveness of the methods could be drawn with more confidence when these conclusions are accompanied by quantitative results to support their claims. There is thus a clear gap to exploit in research, which is defined to be a lack in assessment of the competitiveness of Lyapunov-based SOF algorithms to the well-established non-smooth optimization methods.

The main contribution of this paper is two-fold. The first aspect is the investigation and implementation of Lyapunov-based SOF synthesis methods that could potentially compete with the well-established non-smooth optimization methods Hinfstruct and HIFOO. In particular, three Lyapunov methods were deemed most promising; these are T-K iteration described in [16, 21], and two S-variable approaches described in [22, Section 6.3]. These methods are bundled into a toolbox, which will be called SOFHi for the remainder of the paper. Besides comparing these algorithms to the non-smooth methods, they will also be used to design a gain-scheduled flight controller for the longitudinal dynamics of a highly maneuverable aircraft. The second aspect of the contribution is the presentation of a variant to the original implementation of the algorithms, which aims to significantly improve the original implementation in terms of computational efficiency. This variant will be called SOFHi$_{\mathrm{EVO}}$ for the remainder of the paper.

The paper is organized as follows. In section III, the SOF problem for H_∞ performance is described, after which the SOF algorithms that try to tackle this problem are presented and elaborated on in section IV. The results of comparing the algorithms are shown in section V and lastly, an elaborate flight control example is given in section VI to showcase the effectiveness of SOFHi.

Notation: $X^\top$ for a matrix X denotes the transpose, $\mathrm{Sym}\{X\}$ denotes $X + X^\top$, $X^\perp$ denotes the orthogonal complement, X^+ denotes the Moore-Penrose pseudo-inverse, $||X||_\infty$ denotes the H_∞-norm of X, and lastly, $*$ denotes the symmetric term in a block matrix.

*The conditions in [17] are also made less conservative by considering a triangular part of the Lyapunov matrix, instead of a diagonal structure.

III. $\mathbf{H_\infty}$ Static Output Feedback Problem

Consider the following state-space system,

$$\begin{aligned} \dot{x} &= Ax + B_w w + B_u u \\ z &= C_z x + D_{zw} w + D_{zu} u \\ y &= C_y y + D_{yw} w + D_{yu} u \end{aligned} \tag{1}$$

where the dependence on time is neglected (e.g. $y \leftarrow y(t)$), $x \in \mathbb{R}^n$ is the state vector, $w \in \mathbb{R}^q$ the exogenous input vector, $u \in \mathbb{R}^r$ the input vector, $z \in \mathbb{R}^p$ the evaluated output vector, and $y \in \mathbb{R}^l$ the measured output vector. Without loss of generality, it is assumed that the input and output matrices are of full rank ($\text{rank}(B_u) = r$ and $\text{rank}(C_y) = l$, respectively).

The SOF problem defines the control input to be related to the measured output as

$$u = Ky, \tag{2}$$

where K is the SOF gain matrix. Applying Eq. (2) to the system in Eqs. (1) leads to the following closed-loop system:

$$\begin{aligned} \dot{x} &= A_{cl} x + B_w w \\ z &= C_{cl} x + D_{zw} w \end{aligned} \tag{3}$$

where

$$A_{cl} = A + B_u K C_y, \quad C_{cl} = C_z + D_{zu} K C_y \tag{4}$$

In [23], it was shown that the system in Eqs. (3) is internally stable and the H_∞-norm of the transfer function from w to z is smaller than a positive scalar γ (i.e. $\|T_{zw}\|_\infty < \gamma$) if and only if there exist a positive-definite matrix $P \in \mathbb{S}^n$ and a controller matrix $K \in \mathbb{R}^{r \times l}$, such that

$$\begin{bmatrix} \text{Sym}\{A_{cl} P\} & * & * \\ C_{cl} P & -\gamma I & * \\ B_w^\top & D_{zw}^\top & -\gamma I \end{bmatrix} < 0 \tag{5}$$

Eq. (5) is a BMI due to the product between the decision variables P and K, which makes the optimization problem non-convex. The following section will cover algorithms that try to tackle this problem.

IV. Static Output Feedback Algorithms

The conditions provided in this section for H_∞ SOF are derived from the BMI in Eq. (5) and provide sufficient conditions for said BMI. The algorithms described in this section then utilize these conditions to monotonically decrease γ by alternatively fixing decision variables to render the BMI conditions as an LMI, the latter of which is convex by nature and easy to solve using interior point methods, such as MOSEK or SDPT3 [15, Section 1.4.4].

Three algorithms were found to be most promising, namely T-K iteration [16, 21], and two S-variable approaches described in [22, 24]. For full details on the theorems and corresponding proofs, the reader is referred to the aforementioned literature, but the main procedures of the algorithms will nevertheless be described in this section.

A. T-K Iteration

It was explained before that ILMI methods often use sufficient conditions derived in other direct search methods, and then opt to reduce conservatism by optimizing parts of the conditions for H_∞ performance. T-K iteration is no exception and is based on the following conditions.

Lemma 4.1 *from [16]: The closed-loop system in Eqs. (3) is internally stable and $\|T_{zw}\|_\infty < \gamma$ if there exist a positive-definite matrix $P \in \mathbb{S}^n_{++}$ and control gain matrix $K \in \mathbb{R}^{r \times l}$ s.t.*

$$\begin{bmatrix} \text{Sym}\left\{(\bar{A} + \bar{B}_u K \bar{C}_y) P\right\} & * & * \\ (\bar{C}_z + D_{zu} K \bar{C}_y) P & -\gamma I & * \\ \bar{B}_w^\top & D_{zw}^\top & -\gamma I \end{bmatrix} < 0, \tag{6}$$

where

$$\bar{A} = T_y A T_y^{-1}, \qquad \bar{B}_w = T_y B_w, \qquad \bar{B}_u = T_y B_u$$
$$\bar{C}_z T_y^{-1} = C_z T_y^{-1} \qquad \bar{C}_y = C_y T_y^{-1} = [I_l \quad 0]$$

and T_y is a non-singular matrix such that $C_y T_y^{-1} = [I_l \quad 0]$

***Lemma 4.2** from [25]: The closed-loop system in Eqs. (3) is internally stable and $\|T_{zw}\|_\infty < \gamma$ if there exist a positive-definite matrix $P_d = \begin{bmatrix} P_1 & 0 \\ 0 & P_2 \end{bmatrix}$ with $P_1 \in \mathbb{S}^l_{++}$, $P_2 \in \mathbb{S}^{(n-l)\times(n-l)}_{++}$, and matrix $Y \in \mathbb{R}^{r\times l}$ s.t.*

$$\begin{bmatrix} Sym\{\bar{A}P_d + \bar{B}_u Y \bar{C}_y\} & * & * \\ \bar{C}_z P_d + D_{zu} Y \bar{C}_y & -\gamma I & * \\ \bar{B}_w^\top & D_{zw}^\top & -\gamma I \end{bmatrix} < 0 \tag{7}$$

*where $\bar{A}$,$\bar{B}_u$, $\bar{B}_w$, $\bar{C}_z$, and $\bar{C}_y$ are defined in **Lemma 4.1**. When Eq. (7) holds, the optimal H_∞ SOF controller gain matrix can be obtained through:*

$$K = Y P_1^{-1}$$

The conditions described in Lemma's 4.1 and 4.2 are generally conservative, due to the fact that the Coordinate Transformation Matrix (CTM) T_y is chosen to be a constant value in [26] (i.e. $T_y = [C_y^\top \quad C_y^\perp]^\top$). T-K iteration opts to reduce this conservatism by using a parametrisation form of T_y to perform a coordinate transformation on the original system. This form is given by

$$T_y^{-1} = \begin{bmatrix} C_y^+ + C_y^\perp T_{c1} & C_y^\perp \end{bmatrix} \tag{8}$$

Using this form allows one to optimize the CTM through T_{c1}, which ultimately reduces conservatism of Eqs. (6). The first step of this optimization is to find an initial T_{c1} to ensure feasibility of Eqs. (7), which is done as follows: first, an initial stabilizing SOF gain matrix K is obtained through Cone Complementarity Linearization (CCL)†, which is described in [11]. Second, Eqs. (6) are solved using K to obtain P, after which an initial CTM $T_{c1} = P_{21} P_{11}^{-1}$ can now be obtained. This process of finding an initial T_{c1} is described below [16].

***Part 1 of T-K iteration** from [16]:*

1) *Check if (A, B_u) is stabilizable. If it is not, stop. If it is, continue to step 2).*
2) *Randomly generate P_0 and Q_0 from a uniform distribution $U \sim (0, 1)$.*
3) *Obtain a stabilizing K through CCL(P_0, Q_0).*
4) *Use $T_y^{-1} = [C_y^+ \quad C_y^\perp]$ to solve the following SDP problem with a fixed K:*

$$\min_P \gamma \quad s.t. \quad Eqs. \ (6).$$

An initial $T_{c1} = P_{21} P_{11}^{-1}$ is found, stop.

The second step, as mentioned above, is to optimize the choice of T_{c1} in an iterative procedure to reduce conservatism of the solution and reach a locally optimal solution.

***Part 2 of T-K iteration** from [16]:*

1) *Set $k = 0$ and $T_{c1}^{(0)}$ as the initial T_{c1} produced by **Part 1 of T-K iteration**.*
2) *Use $T_y^{-1} = [C_y^+ + C_y^\perp T_{c1}^{(k)} \quad C_y^\perp]$ to solve the following SDP problem:*

$$\min_{P_1, P_2, Y} \gamma_1 \quad s.t. \quad Eqs. \ (7).$$

Calculate $K = Y P_1^{-1}$.

†The algorithm is not recalled here for practical purposes, the reader is referred to [11] for more details on CCL.

3) *Use* $T_y^{-1} = [C_y^{+} + C_y^{\perp} T_{c1}^{(k)} \quad C_y^{\perp}]$ *and the obtained* K *to solve the following SDP problem:*

$$\min_{P} \gamma_2 \quad s.t. \quad Eqs.\ (6).$$

Calculate $N = P_{11}^{-1} P_{12}$.

4) *Let* $\epsilon << 1$ *be the prescribed tolerance. If* $||N|| < \epsilon$, $||\gamma_1 - \gamma_2|| < \epsilon$, *or* $k > k_{max}$, *a given maximum iteration number,* K *is a locally optimal* H_∞ *SOF gain matrix, stop. Otherwise, set* $T_{c1}^{(k+1)} \leftarrow T_{c1}^{(k)} + N^\top$ *and* $k \leftarrow k + 1$, *and go to* 2).

The procedure of fixing decision variables to render the BMI conditions into LMI conditions is a common theme among ILMI methods. In fact, the two algorithms described in subsection IV.B will use this philosophy as well.

Remark 1: When the variable $N = P_{11}^{-1} P_{12}$ approaches zero, the algorithm has approached a local solution as the CTM has converged by then. N thus serves as an evaluation index of the obtained minimum, which is a rare feature among ILMI methods, that almost always provide no conclusive characterisation of the local minimum and just stop when no better solution is found.
Remark 2: T-K iteration is extended in SOFHi to incorporate *structured* SOF. A zero-nonzero structure of K_{SOF} can be imposed through the SDP variable Y in Lemma 4.2, while keeping P_1 proportional to the identity matrix.
Remark 3: $K = Y P_1^{-1}$ in Lemma 4.2 is replaced in SOFHi by $K = Y P_1^{+}$ when P_1 is singular or near-singular, to avoid inaccuracies in the results. Singularity is checked through the condition number $\kappa(P_1) = ||P_1|| \cdot ||P_1^{-1}||$. When $\kappa(P_1) \to \infty$, P_1 is considered singular. Ultimately, this increases the robustness of the algorithm in at least obtaining an adequate result.

B. S-variable Approaches

This section describes two approaches that aim to render variations to Eq. (5) as LMI's, by alternatively fixing the decision variables in the conditions. The name of the S-variable methods lends itself from the use of slack variables in the Lyapunov conditions for H_∞ SOF performance. By introducing slack variables in the conditions, a new parametrization form can be constructed, where the Lyapunov matrix P is de-coupled from the other SDP variables. This de-coupling allows one to optimize over P with less constraints and consequently less conservatism in the solution. This section describes two approaches that stem from this philosophy, but which are based on different parametrizations. The difference lies in that Approach I is based on an initial state-feedback gain, whereas Approach II is based on an initial SOF gain. The parametrizations on which Approaches I and II are based, respectively, are given as follows [22, Section 6.3.3]:

Lemma 4.3 *from [22, Section 6.3.3]: The* H_∞ *optimal SOF gain is given by* $K = -F^{-1}Z$, *where the triplet* $(P_\infty, Z, F) \in \mathbb{S}^n_{++} \times \mathbb{R}^{r \times l} \times \mathbb{R}^{r \times r}$ *is the global optimal solution of the following non-convex optimization problem:*

$$\min_{P_\infty, Z, F, K_{SF_\infty}, K_{w_\infty}} \gamma^2 \quad s.t.$$
$$N_\infty(P_\infty) + Sym\left\{\begin{bmatrix} K_{SF_\infty}^\top \\ K_{w_\infty}^\top \\ -I \end{bmatrix} \begin{bmatrix} ZC_y & ZD_{yw} & F \end{bmatrix}\right\} < 0, \tag{9}$$

where

$$N_\infty(P_\infty) = \begin{bmatrix} I & 0 & 0 \\ A & B_w & B_u \end{bmatrix}^\top \begin{bmatrix} 0 & P_\infty \\ P_\infty & 0 \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ A & B_w & B_u \end{bmatrix} + \begin{bmatrix} C_z & D_{zw} & D_{zu} \\ 0 & I & 0 \end{bmatrix}^\top \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \begin{bmatrix} C_z & D_{zw} & D_{zu} \\ 0 & I & 0 \end{bmatrix} \tag{10}$$

Lemma 4.4 *from [22, Section 6.3.3]: Given the quadruple* $(P_\infty, K_1, K_2, F) \in \mathbb{S}^n_{++} \times \mathbb{R}^{n\times r} \times \mathbb{R}^{q\times r} \times \mathbb{R}^{r\times r}$*, the optimal* H_∞ *SOF gain matrix* $K = K_{SOF}$ *is the global optimal solution to the following non-convex optimization problem:*

$$\begin{aligned}
&\min_{P_\infty, K_1, K_2, F, K_{SOF}} \gamma^2 \quad s.t. \\
&\qquad N_\infty(P_\infty) + Sym\left\{\begin{bmatrix} K_1 \\ K_2 \\ F \end{bmatrix} \begin{bmatrix} K_{SOF}C_y & K_{SOF}D_{yw} & -I \end{bmatrix}\right\} < 0
\end{aligned} \tag{11}$$

with $N_\infty(P_\infty)$ *defined in Eq. (10).*

Eqs. (9) and (11) are BMI due to the product terms between the different SDP variables. Approaches I and II are both coordinate-descent cross-decomposition algorithms, which means that SDP variables are alternatively fixed to render the BMI's into LMI's, while monotonically decreasing the performance value γ.

1. S-variable Approach I

The first approach needs an initial guess of the SDP variables K_{SF_∞} and K_{w_∞} in Eqs. (9) before it starts iterating. These can be chosen at random, but a good guess for the BMI in Eqs. (9) can be found by assuming full-information feedback:

Lemma 4.5 *from [22, Section 6.3.3]: Given the quadruple* $(X_\infty, R, Y, K_w) \in \mathbb{S}^n_{++} \times \mathbb{R}^{r\times n} \times \mathbb{R}^{r\times n} \times \mathbb{R}^{r\times n}$*, solve the following SDP problem:*

$$\begin{aligned}
&\min_{X_\infty, R, Y, K_w} \gamma^2 \quad s.t. \\
&\qquad \begin{bmatrix} Sym\{AX_\infty + B_uY\} & * & * \\ (B_w + B_uK_w)^\top & -\gamma^2 I & * \\ (C_z + D_{zu}R) & (D_{zw} + D_{zu}K_w) & -I \end{bmatrix} < 0 \\
&\qquad K_{SF_\infty} = YX_\infty^{-1} \\
&\qquad K_{w_\infty} = K_w
\end{aligned} \tag{12}$$

where $C_y = [I \quad 0]^\top$ *and* $D_{yw} = [0 \quad I]^\top$.

In [22, Section 6.3], K_{SF_∞} and K_{w_∞} are used as a good initial guess for the subsequent coordinate-descent algorithm. A Hit-and-Run (H.R.) algorithm[‡] in [24] is applied to generate a set of state-feedback gains based on the initial guess K_{SF_∞}. By using several initial guesses, instead of only K_{SF_∞}, better results were obtained, since the results of the algorithms are heavily dependent on their initial conditions. Thus, having many initial guesses instead of just one, increases the likelihood of converging to a local solution.

Approach I *from [22, Chapter 6]:*

1) *Solve the following SDP problem:*

$$\min_{X_\infty, R, Y, K_w} \gamma^2 \quad \text{s.t.} \quad \text{Eqs. (12)}$$

Let the solutions be K_{SF_∞} *and* K_{w_∞}

2) *Choose* k_{max} *as a positive integer and apply H.R.*(K_{SF_∞}) *to generate* $\mathfrak{K}_{SF}$*:*

$$\mathfrak{K}_{SF} = \left\{K_{SF}^{(1)}, K_{SF}^{(2)}, ..., K_{SF}^{(k_{max})} \in \mathbb{R}^{r\times n} : \quad \lambda(A + B_uK_{SF}) \in \mathbb{C}_{--}\right\},$$

where $K_{\text{SF}}^{(1)} := K_{SF_\infty}$

[‡]H.R. is able to construct a set of gains of any size, starting from one gain, using a randomized approach. This method is found to be very efficient in generating many stabilizing SOF gains. For practical purposes, the reader is referred to [24] for the full description of the algorithm.

3) *Set $k = 1$ and choose $K_{SF}^{(1)} \in \mathfrak{K}_{SF}$.*
4) *For fixed $K_{SF}^{(k)}$ and $K_{w_\infty}^{(k)}$, solve the following SDP problem:*

$$\min_{P_\infty, Z, F} \gamma_1^2 \quad s.t. \quad Eqs. \ (9)$$

Let the solutions be $Z^{(k)} = Z$, $F^{(k)} = F$.
5) *For fixed $Z^{(k)}$ and $F^{(k)}$, solve the following SDP problem:*

$$\min_{P_\infty, K} \gamma_2^2 \quad s.t. \quad Eqs. \ (9)$$

Let the solutions be $K^{(k)} = K$.
6) *Let $\epsilon << 1$ be the prescribed tolerance. If $\left\| \gamma_1^{(k)} - \gamma_2^{(k)} \right\| < \epsilon$, stop, the algorithm has converged. Else if $k = k_{max}$, stop, the maximum amount of iterations has been reached. Set $K_{SOF} = K_{SOF}^{(k)}$. Else, $k \leftarrow k + 1$ and go to step 4.*

A similar procedure to T-K iteration can be observed, where SDP variables are alternatively fixed to render BMI's into LMI's. This procedure can also be observed in the subsequent section.

2. S-variable Approach II

The general procedure for Approach II of the S-variable methods is of similar nature to that of Approach I, where both algorithms are of the coordinate-descent cross-decomposition type. However, where Approach I is based on an initial $K_{\text{SF}} \in \mathfrak{K}_{\text{SF}}$, Approach II is in fact based on a stabilizing $K_{\text{SOF}} \in \mathfrak{K}_{\text{SOF}}$, where $\mathfrak{K}_{\text{SOF}}$ of size $j_{\max}$ is defined by

$$\mathfrak{K}_{\text{SOF}} = \left\{ K_{\text{SOF}}^{(1)}, K_{\text{SOF}}^{(2)}, \ldots, K_{\text{SOF}}^{(j_{\max})} \in \mathbb{R}^{r \times l} : \quad \lambda(A + B_u K_{\text{SOF}} C_y) \in \mathbb{C}_{--} \right\} \tag{13}$$

Approach II uses the following lemma from [22, Section 6.2] to obtain K_{SOF} from an initial K_{SF}:

Lemma 4.5 *from [22, Chapter 6]: There exists $K_{SOF} \in \mathfrak{K}_{SOF}$ for the closed-loop system in Eqs. (3) if and only if there exist a stabilizing state-feedback matrix $K_{SF} \in \mathfrak{K}_{SF}$, a matrix $P \in \mathbb{S}_{++}^n$, and matrices $F \in \mathbb{R}^{r \times r}$, $Z \in \mathbb{R}^{r \times l}$, s.t.*

$$\begin{aligned} & Trace(P) > 0 \\ & M(P) + Sym \left\{ \begin{bmatrix} I \\ -K_{SF}^\top \end{bmatrix} \begin{bmatrix} F & ZC_y \end{bmatrix} \right\} < 0 \end{aligned} \tag{14}$$

where $M(P) = \begin{bmatrix} B_u^\top & 0 \\ A^\top & I \end{bmatrix} \begin{bmatrix} 0 & P \\ P & 0 \end{bmatrix} \begin{bmatrix} B_u & A \\ 0 & I \end{bmatrix}$ and $K_{SOF} = F^{-1} Z$.

Approach II can then be described as follows, where steps 1) to 4) generate a set of SOF gains and steps 5) to 8) deal with obtaining the optimum SOF gain from the set of gains.

Approach II *from [22, Chapter 6]:*
1) *Choose k_{max} and generate $\mathfrak{K}_{SF}$ of size k_{max}, in a similar way to **Approach I** using H.R..*
2) *Let $k = 1$.*
3) *Choose a $K_{SF}^{(k)} \in \mathfrak{K}_{SF}$ and solve the following SDP problem with a fixed $K_{SF}^{(k)}$:*

$$\min_{P, F, Z} Trace(P) \quad s.t. \quad Eqs. \ (14).$$

Let the solution be $K_{SOF}^{(k)} = -F^{-1} Z$. If $K_{SOF}^{(k)} \in \mathfrak{K}_{SOF}$, $K_{SOF} = K_{SOF}^{(k)}$ and continue to step 4). Else, if $k = k_{max}$, stop, the algorithm was not able to find a stabilizing K_{SOF}. Else, $k \leftarrow k + 1$ and repeat step 3).
4) *Choose j_{max} and apply H.R.(K_{SOF})* to generate $\mathfrak{K}_{\text{SOF}}$ of size $j_{\max}$ defined in Eq. 13, where $K_{\text{SOF}}^{(1)} := K_{\text{SOF}}$.
5) *Let $j = 1$.*

6) *For fixed $K_{SOF}^{(j)} = K_{SOF}$, solve the following SDP optimization problem:*

$$\min_{P_\infty, K_1, K_2, F} \gamma_1^2 \quad s.t. \quad Eqs.\ (11)$$

Let the solutions be $K_1^{(j)} = K_1$, $K_2^{(j)} = K_2$, and $F^{(j)} = F$.

7) *For fixed $K_1^{(j)}$, $K_2^{(j)}$, and $F^{(j)}$, solve the following SDP optimization problem:*

$$\min_{P_\infty, K_{SOF}} \gamma_2^2 \quad s.t. \quad Eqs.\ (11)$$

Let the solution be $K_{SOF}^{(j)} = K_{SOF}$.

8) *Let $\epsilon << 1$ be the prescribed tolerance. If $\left\|\gamma_1^{(j)} - \gamma_2^{(j)}\right\| < \epsilon$, stop, the algorithm has converged. Set $K = K_{SOF}^{(j)}$. Else, if $j = j_{max}$, stop, the maximum number of iterations has been reached. Else, $j \leftarrow j + 1$ and go to step 6).*

Remark 1: Approaches I and II are, like most iterative LMI methods, heavily dependent on their initial conditions. Unfortunately, there exists no dead-point criterion for the local minima, as was the case for T-K iteration with N, the algorithm just stops whenever γ has converged.

Remark 2: Approaches I and II are easily extended in SOFHi to incorporate *structured* SOF. A zero-nonzero structure of K_{SOF} can be imposed through Z in Lemma 4.3 (keeping F proportional to the identity matrix), and directly through K_{SOF} in Lemma 4.4.

Remark 3: $K = -F^{-1}Z$ in Lemma 4.3 is replaced in SOFHi by $K = -F^{+}Z$ when F is singular or near-singular, to avoid inaccurate results. Singularity is checked through the condition number $\kappa(F) = ||F|| \cdot ||F^{-1}||$. When $\kappa(F) \to \infty$, F is considered singular. Ultimately, this increases the robustness of the algorithm in at least obtaining an adequate result.

C. SOFHi$_{\text{EVO}}$

The algorithms described in subsection IV.A-IV.B are implemented and bundled into one toolbox named SOFHi. A downside that is intrinsic to Lyapunov methods is their lack of computational efficiency when compared to non-smooth optimization methods, since the latter do not have Lyapunov matrices in their problem formulation. Let n be the #states, r the #inputs, and l the #outputs. Lyapunov methods are then of computational complexity $O(n^2)$, while non-Lyapunov methods are of complexity $O(r \cdot l)$, the latter of which is generally much smaller. Consequently, this leads to non-smooth methods being more computationally efficient than Lyapunov methods, even more so for systems that include many states.

Additionally, the methods in SOFHi are prone to converge to a local plateau, since they are probabilistic by nature and highly dependent on the initial conditions: the initial CTM T_y for T-K iteration and the initial stabilizing gain matrices for the S-variable approaches. To reduce conservatism and likely improve the solution to a local optimum, multiple runs starting from different initial conditions are beneficial, which ultimately increases the likelihood of the algorithms to converge to- or near a local optimum. However, the need to run multiple times from different initial points emphasizes the need for more computational efficiency in the Lyapunov methods, since the required computational power is directly proportional to the amount of runs performed.

For those reasons, SOFHi$_{\text{EVO}}$ is introduced in this paper to serve as an additional "fast setting" for SOFHi. It is based on the idea of filtering out unfavorable runs early and proceeding with the set of promising runs, which is a subset of the original set of runs. The concept is visualized in Fig. 1, where the three parameters of SOFHi$_{\text{EVO}}$ are shown, namely the number of starts N_{starts}, the number of samples N_{samples}, and the number of candidates $N_{\text{candidates}}$. The idea is to start from multiple random initial conditions, evaluate γ of each run after N_{samples}, and discard the unfavorable runs, except for the candidates, i.e. the runs that are optimized further. Let $\mathfrak{L}$ be the set of gains at the iteration index k that is equal to N_{samples}:

$$\mathfrak{L} = \left\{K^{(1)}, K^{(2)}, ..., K^{(N_{\text{starts}})} : \quad k = N_{\text{samples}}\right\} \tag{15}$$

The set of candidates is then described as $\mathfrak{C} \subseteq \mathfrak{L}$:

$$\mathfrak{C} = \left\{K^{(1)}, K^{(2)}, ..., K^{(N_{\text{candidates}})} : \quad N_{\text{candidates}} < N_{\text{starts}}\right\}, \tag{16}$$

where $K \in \mathfrak{C}$ are the gain matrices that lead to the set with lowest γ's of size $N_{\text{candidates}}$. $\mathfrak{C}$ then serves as the new set of initial gains which is optimized further. The motivation for using SOFHi$_{\text{EVO}}$ and its computational efficiency with respect to SOFHi will become apparent from the results in section V.

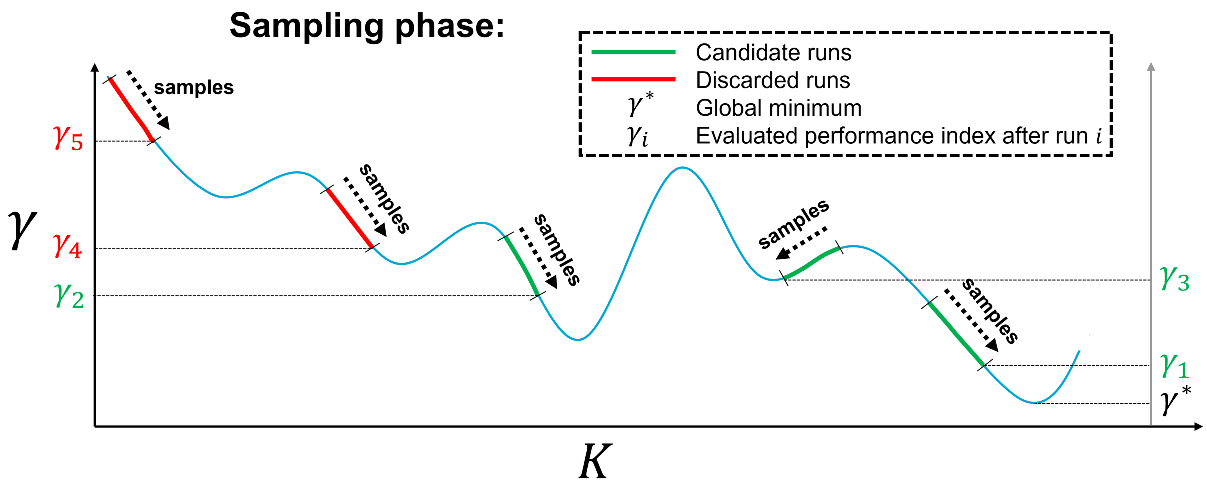


Fig. 1 Visualization of the solution space for a fictional case to explain the sampling phase in SOFHi$_{\text{EVO}}$, where the gain matrix K is assumed to be the only decision variable and is a single scalar gain to be able to have a two-dimensional visualization of the optimization procedure. The following user-defined settings are emphasized: in this example, $N_{\text{starts}} = 5$ and $N_{\text{candidates}} = 3$, meaning that 2 of the 5 runs are discarded after N_{samples} iterations, based on the fact that their corresponding performance indices (γ_4 and γ_5) are worse than (γ_1, γ_2, and γ_3). The runs corresponding to $(\gamma_1, \gamma_2, \gamma_3)$ are then iterated further until convergence.

V. Results

This section will cover the results from implementing the SOF algorithms described in section IV. The algorithms are applied on 54 low-medium sized benchmark models from the *Compl$_e$ib* library in [27]. Comparisons are made between SOFHi, SOFHi$_{\text{EVO}}$, and the two most well-established fixed-structure synthesis algorithms that optimize the H_∞-norm of the complete block, these algorithms are HIFOO and Hinfstruct. The latter comparisons are made only on γ, not on computational times, since the Lyapunov methods are limited by the amount of decision variables in the problem formulation, as explained in subsection IV.C, and further improvements could be made by writing in Fortran.

Quad-core computations in the subsequent sections have been performed on hardware with the following specifications:

Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz 2.81 GHz
16.0 GB RAM

A. SOFHi comparison

A full overview of the benchmark models and results can be found in Table 8 and Table 9. A summary of the comparisons between standard SOFHi, Hinfstruct, and HIFOO is presented in Table 2, though. It can be seen that SOFHi outperforms HIFOO in terms of performance index γ, where the difference becomes more apparent when considering a margin of significance. When comparing SOFHi to Hinfstruct, however, Hinfstruct can be seen to outperform SOFHi when no margin of significance is considered. When a margin is considered, however, the results are very similar and SOFHi is able to slightly outperform Hinfstruct.

Table 2 Summary for both 30 and 100 starts. Results represent the % of times, from all the 54 example models, method A has a more optimal γ than B. When a margin of e.g. 1% is included, the results are considered equal when $0.99\gamma(B) < \gamma(A) < 1.01\gamma(B)$. Superior results are placed in bold.

	30 starts				100 starts			
	SOFHi	HIFOO	SOFHi	Hinfstruct	SOFHi	HIFOO	SOFHi	Hinfstruct
No margin	**64.81%**	33.33%	40.74%	**53.70%**	**62.96%**	35.19%	44.44%	**51.85%**
Margin = 1%	**14.81%**	0.00%	**9.26%**	0.00%	**12.96%**	0.00%	**9.26%**	1.85%

B. $\text{SOFHi}_{\text{EVO}}$ comparison

The comparison between SOFHi and $\text{SOFHi}_{\text{EVO}}$ in terms of performance index γ is presented in Table 3. It can be seen that even for significantly less candidates than the number of starts of SOFHi, $\text{SOFHi}_{\text{EVO}}$ is able to outperform SOFHi and takes up less computational time. The most extreme case is emphasized, when $\text{SOFHi}_{\text{EVO}}$ and SOFHi have the same N_{starts} and $\text{SOFHi}_{\text{EVO}}$ only proceeds with 8 of the original 30 starts after 30 sample iterations. It is expected that SOFHi will outperform $\text{SOFHi}_{\text{EVO}}$, since $\text{SOFHi}_{\text{EVO}}$ discards 22 runs, without the added benefit of starting out with more runs. The difference is marginal though, highlighting the motivation for creating $\text{SOFHi}_{\text{EVO}}$: the loss in accuracy after discarding the unfavorable runs is marginal, and more often than not, the optimal run is present among the number of candidates, given that N_{samples} and $N_{\text{candidates}}$ are chosen appropriately.

Table 3 Comparisons between $\text{SOFHi}_{\text{EVO}}$ and SOFHi on computational performance and γ. Parameters are denoted as "$(N_{\text{starts}}, N_{\text{samples}}, N_{\text{candidates}})$" for $\text{SOFHi}_{\text{EVO}}$ and as "N_{starts}" for SOFHi. Relative computational time denotes the time it takes for $\text{SOFHi}_{\text{EVO}}$ to compute for all 54 benchmark models relative to SOFHi, e.g. 100% means it has taken exactly the same amount of (real) time as SOFHi. Further results represent the % of times, from all the 54 example models, method A has a more optimal γ than method B. When a margin of e.g. 1% is included, the results are considered equal when $0.99\gamma(B) < \gamma(A) < 1.01\gamma(B)$.

	$\text{SOFHi}_{\text{EVO}}$	SOFHi	$\text{SOFHi}_{\text{EVO}}$	SOFHi	$\text{SOFHi}_{\text{EVO}}$	SOFHi	$\text{SOFHi}_{\text{EVO}}$	SOFHi
Parameters	(250, 50, 16)	30	(150, 50, 16)	30	(300, 20, 8)	30	(30, 30, 8)	30
Rel. comp. time	93.29%	-	60.45%	-	48.07%	-	22.76%	-
No Margin	**62.96%**	31.48%	**57.41%**	37.04%	**55.56%**	40.74%	42.59%	**53.70%**
Margin = 1%	**3.70%**	0.00%	**3.70%**	0.00%	**1.85%**	0.00%	**3.70%**	1.85%

The results of $\text{SOFHi}_{\text{EVO}}$ when compared to HIFOO and Hinfstruct are presented in Table 4. It can be seen that $\text{SOFHi}_{\text{EVO}}$ scores significantly better than HIFOO, with and without a margin. Compared to Hinfstruct, it can be seen that $\text{SOFHi}_{\text{EVO}}$ is able to outperform with and without a margin for the (250, 50, 16) run where a gradual decrease in results can be observed when choosing runs with faster computational times, relative to SOFHi. Even then, $\text{SOFHi}_{\text{EVO}}$ is at least competitive to Hinfstruct, slightly outperforming when a significance margin is included.

Table 4 Comparison between $\text{SOFHi}_{\text{EVO}}$, HIFOO, and Hinfstruct on γ. Parameters are denoted as "$(N_{\text{starts}}, N_{\text{samples}}, N_{\text{candidates}})$" for $\text{SOFHi}_{\text{EVO}}$ and as "N_{starts}" for HIFOO and Hinfstruct. Furthermore, results represent the % of times, from all the 54 example models, method A has a more optimal γ than B. When a margin of e.g. 1% is included, the results are considered equal when $0.99\gamma(B) < \gamma(A) < 1.01\gamma(B)$. Superior results are placed in bold.

	$\text{SOFHi}_{\text{EVO}}$	HIFOO	$\text{SOFHi}_{\text{EVO}}$	HIFOO	$\text{SOFHi}_{\text{EVO}}$	HIFOO
Parameters	(250, 50, 16)	30	(150, 50, 16)	30	(300, 20, 8)	30
No Margin	**70.37%**	27.78%	**68.52%**	29.63%	**59.26%**	38.89%
Margin = 1%	**14.81%**	0.00%	**14.81%**	0.00%	**14.81%**	0.00%
	$\text{SOFHi}_{\text{EVO}}$	Hinfstruct	$\text{SOFHi}_{\text{EVO}}$	Hinfstruct	$\text{SOFHi}_{\text{EVO}}$	Hinfstruct
Parameters	(250, 50, 16)	30	(150, 50, 16)	30	(300, 20, 8)	30
No Margin	**51.85%**	44.44%	48.15%	48.15%	42.59%	**55.56%**
Margin = 1%	**9.26%**	1.85%	**9.26%**	0.00%	**9.26%**	1.85%

Remark: It is important to acknowledge that the comparisons in Table 3 and Table 4 are akin to "comparing apples to pears", since $\text{SOFHi}_{\text{EVO}}$ has many more starts than 30. However, most of these runs are discarded after the sampling phase and $\text{SOFHi}_{\text{EVO}}$ ends up with much less runs, i.e. the candidates. The point of this comparison is merely to show that within the same time-frame of the original algorithms (see "Rel. comp. time." in Table 3), $\text{SOFHi}_{\text{EVO}}$ is able to be *more* competitive to HIFOO and Hinfstruct than the original algorithms in SOFHi were, to the point that $\text{SOFHi}_{\text{EVO}}$ is even able to outperform Hinfstruct for one of its runs. It is thus shown that the original algorithms were improved upon.

VI. Fighter Aircraft Example

To showcase the effectiveness of SOFHi in tuning H_∞ SOF controllers, a classical PI-controller with an inner-loop pitch-rate feedback is designed for a normal acceleration Control Augmentation System (CAS) for the F-16 Fighting Falcon. The non-linear equations of motion stem from [28], and the aerodynamic model is taken from [29, p. 714-723].

A Two-Degree-of-Freedom (2-DoF) approach is implemented, where the first stage is based on a classical Four-Block design, similar to [30, Section 5.1]. Furthermore in the second stage, a second-order feedforward function F is tuned using an additional H_∞-block. The full configuration of the 2-DoF design can be seen in Fig. 2, where F is added outside the loop to improve the transient time response without affecting the robustness and disturbance rejection characteristics of the Four-Block design. Furthermore, the proportional part of the PI-controller is chosen to be in the feedback path. This is done to reduce the effect of proportional kicks on the output and control input, which occur in presence of step reference input [31, Section II.F]. This ultimately favors soft-starts and smooth responses in tracking applications.

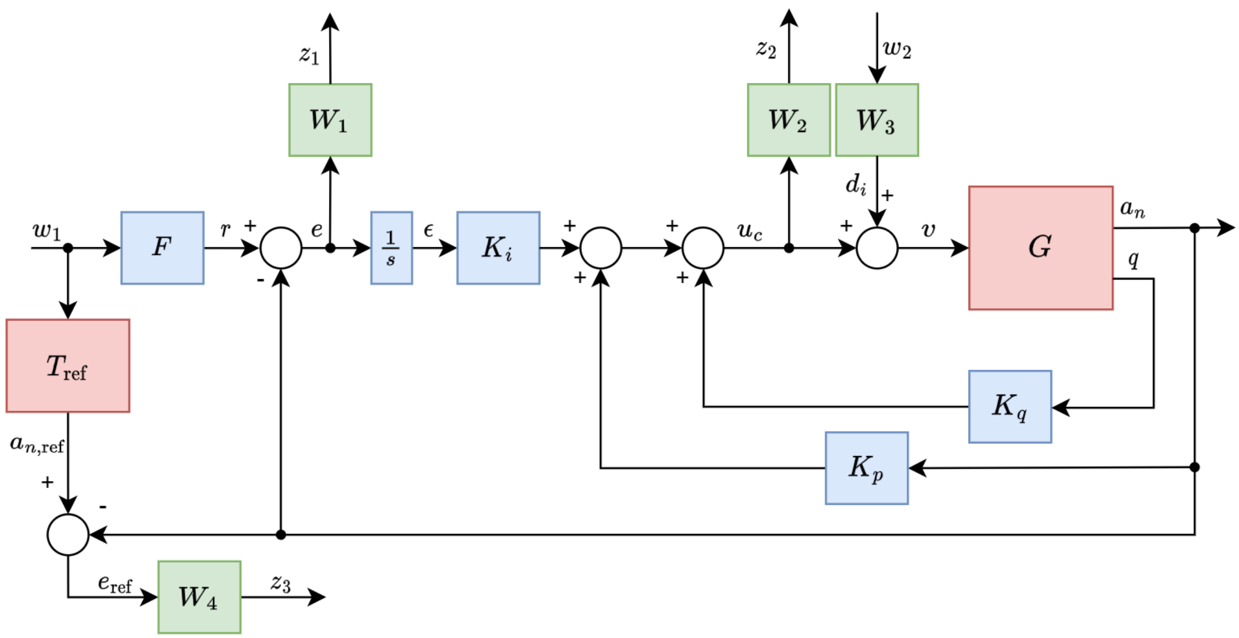


Fig. 2 2-DoF design for the normal acceleration CAS. G denotes the Linear Time Invariant (LTI) plant, which originates from trimming and linearizing the non-linear model around each flight point. W_i denotes the weighting functions that shape the closed-loop sensitivity functions. (K_i, K_p, K_q) are the SOF gains in the Four-Block design. For the second stage in the 2-DoF design, F denotes the feedforward transfer function, and T_{ref} denotes the reference model that is to be followed.

A. Design Objectives

One can formulate design objectives in terms of the closed-loop sensitivity functions, which describe the desired performance and robustness of the closed-loop system. An overview of the sensitivity functions in Fig. 2 can be found in Table 5. The design objectives to the sensitivity functions in the frequency domain for this 2-DoF design are as follows [32].

- To attenuate output disturbance signals at the plant output and to reduce the steady-state tracking error, $\bar{\sigma}(S_{a_n})$ is to be minimized, where S_{a_n} is the output sensitivity function of the normal acceleration loop a_n.
- To attenuate control input, $\bar{\sigma}(KS_{a_n})$ is desired to be minimized, where K is the controller block. Additionally, high-frequency measurement noise at the plant input is to be attenuated through KS_{a_n}.
- To attenuate input disturbance signals at the plant output, $\bar{\sigma}(S_{a_n}G)$ is to be minimized, where G is the plant.
- To attenuate high-frequency unmodelled dynamics at the plant input, $\bar{\sigma}(T_i)$ is to be minimized, where T_i is the complementary input sensitivity function.

- In the second stage of the 2-DoF design, to accurately follow the reference model T_{ref}, $\bar{\sigma}(M)$ is to be minimized, where M is the model-matching sensitivity.
- In the second stage of the 2-DoF design, to limit the actuator effort from pilot input, $\bar{\sigma}(KS_{a_n}F)$ is to be minimized.

Table 5 Overview of the closed-loop sensitivity functions in Fig. 2.

From	To			
	e	a_n	u_c	e_{ref}
w_1	$S_{a_n}F$	$T_{a_n}F$	$KS_{a_n}F$	M
r	S_{a_n}	T_{a_n}	KS_{a_n}	$-T_{a_n}$
d_i	$-S_{a_n}G$	$S_{a_n}G$	T_i	$-S_{a_n}G$

It is not possible to minimize all the sensitivity functions above over all frequencies. Luckily, the design objectives stated above are relevant over different frequency ranges, so one can therefore use weighting functions (W_1, W_2, W_3, and W_4 in Fig. 2) to shape the closed-loop sensitivity functions, in order to meet the design objectives to the best of its abilities[§]. This often involves certain trade-offs between performance and robustness. Other requirements are stated as follows, which stem from [29, 33, 34].

- The closed-loop system shall be internally stable.
- The overshoot of the normal acceleration time responses shall not exceed 10% [34].
- The settling times of the normal acceleration time responses shall not exceed 3 [s] [34].
- The damping ratio of the normal acceleration time responses shall be at least 0.3 [-] [33, 34].
- The control actuator deflections [deg] and deflection rates [deg/s] shall not exceed the physical limits, i.e. 25 deg and 60 deg/s, respectively [29].
- The minimum gain and phase margins shall be at least 6 [dB] and 40 [deg], respectively [33].
- The Control Anticipation Parameter (CAP) requirements shall be met [33].

B. H$_\infty$ SOF Design

This section describes the two design stages for a 2-DoF design. The first stage tunes the feedback gains, whereas the second stage tunes the parameters of the feedforward function F.

1. First Stage of the 2-DoF Design

The first stage of the 2-DoF design aims at providing disturbance rejection and robustness to the system. From the design objectives defined in subsection VI.A and Fig. 2, the transfer function from the exogenous inputs to the evaluated outputs T_{zw} is described by:

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} W_1 & 0 \\ 0 & W_2 \end{bmatrix} \begin{bmatrix} S_{a_n} & -S_{a_n}G \\ KS_{a_n} & -T_i \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & W_3 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{17}$$

The H_∞-performance function to which the SOF algorithms in section IV are applied can now be defined as:

$$||T_{zw}||_\infty = \left\| \begin{bmatrix} W_1 S_{a_n} & W_1 S_{a_n} G W_3 \\ W_2 K S_{a_n} & W_2 T_i W_3 \end{bmatrix} \right\|_\infty < \gamma_1 \tag{18}$$

Eq. 18 is tuned using SOFHi to obtain the feedback gains.

[§]These weighting functions must be stable and proper due to assumptions made about the generalized plant [32, Page 98].

2. Second Stage of the 2-DoF Design

The second stage aims at improving the transient response by tuning the second-order feedforward function F, in order for the output of the model to follow a reference model T_{ref}, with T_{ref} being defined as:

$$T_{\text{ref}} = \frac{\omega_{\text{sp,ref}}^2}{s^2 + 2\zeta_{\text{sp,ref}}\omega_{\text{sp,ref}}s + \omega_{\text{sp,ref}}^2}, \tag{19}$$

where $\omega_{\text{sp,ref}}$ and $\zeta_{\text{sp,ref}}$ are the desired natural frequency and damping ratio of the short period dynamics, respectively. T_{zw} is now the transfer function from $w_1 \longrightarrow [z_2 \quad z_3]^\top$ and so $\|T_{zw}\|_\infty$ becomes

$$\|T_{zw}\|_\infty = \left\| \begin{bmatrix} W_2 K S_{a_n} F \\ W_4 M \end{bmatrix} \right\|_\infty < \gamma_2, \tag{20}$$

where the bandwidth of W_2 is now set to a higher value than for the first stage to allow for more transient tracking performance. The parameters of the transfer function F are then tuned using Hinfstruct instead of SOFHi, since the matrices B_u and C_y are not of full rank in that case, which is an assumption made by the algorithms in SOFHi.

C. Gain-Scheduling

Flight dynamics change significantly with varying Mach and altitude through dynamic pressure. To make the designed CAS more adaptable to variations in speeds and altitudes, 27 flight points are designed throughout the flight envelope, between which the controller parameters are linearly interpolated. The shape of the flight envelope is based on [34], where 15 flight points are added to the original 12 in the aforementioned literature to make the design grid more dense, and ultimately make the controller perform better in between the original 12 flight points. The 2-DoF design procedure described in subsection VI.B is applied around all the design points shown in Fig. 3a.

The (sub)-optimal H_∞-norm of each flight point is presented in Table 6 for both the first and the second stage of the 2-DoF design, i.e. γ_1 and γ_2, respectively. The consequent gain surfaces can be seen in Figs. 3b-3d, where it can be seen that gains with higher magnitude are observed at lower speeds and higher altitudes; the system needs more actuator effort at these points in the flight envelope to meet the design objectives described in subsection VI.A.

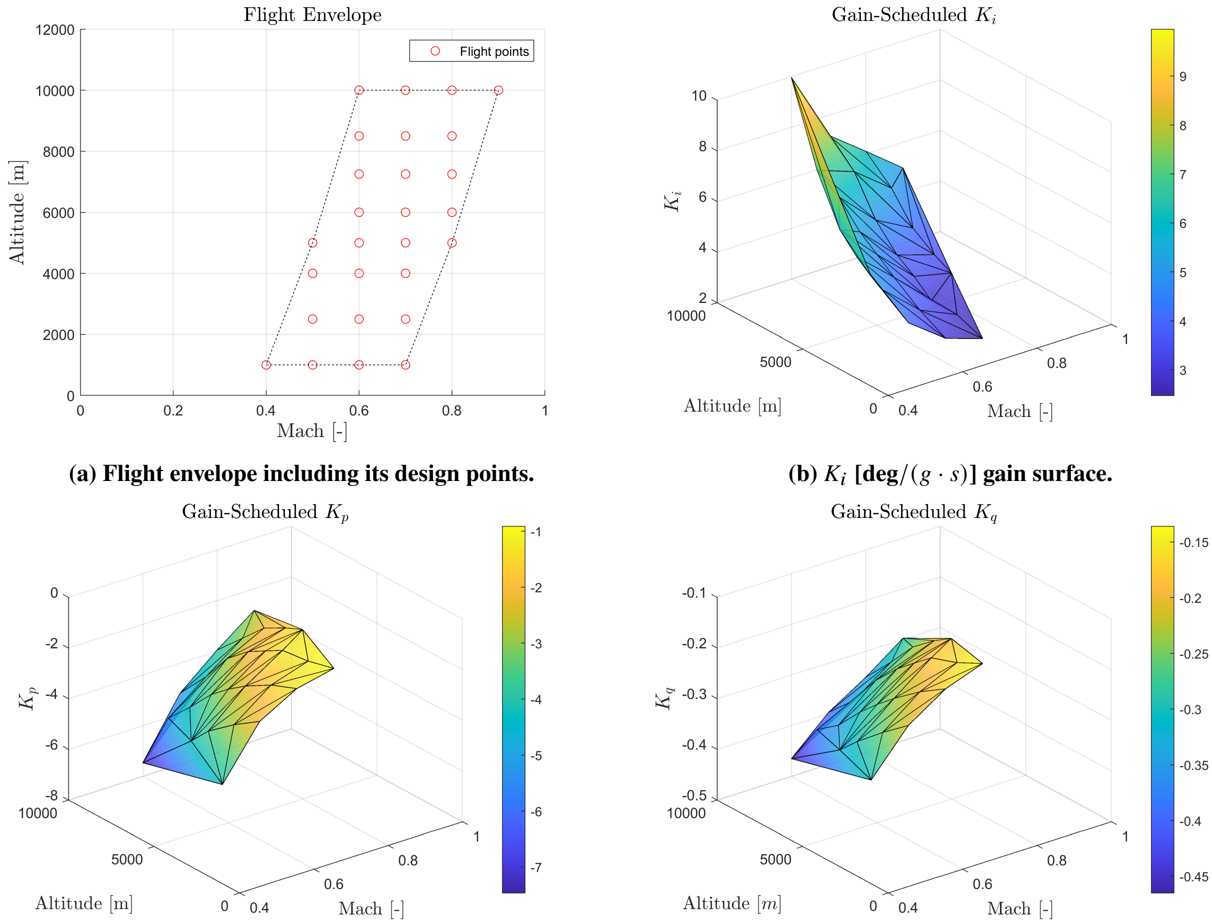
Table 6 H_∞ norms for each flight point in both stages of the 2-DoF design.

(a) γ_1 [-] in Eq. 18 of the first stage in the 2-DoF design.

h [m] \ Mach [-]	0.4	0.5	0.6	0.7	0.8	0.9
1000	1.08	1.18	1.23	1.07		
2500		1.15	1.20	1.08		
4000		1.16	1.17	1.23		
5000		1.11	1.15	1.19	1.16	
6000			1.13	1.18	1.10	
7250			1.11	1.17	1.11	
8500			1.13	1.14	1.07	
10000			1.14	1.13	1.06	1.08

(b) γ_2 [-] in Eq. 20 of the second stage in the 2-DoF design.

h [m] \ Mach [-]	0.4	0.5	0.6	0.7	0.8	0.9
1000	1.03	1.09	1.07	1.04		
2500		1.06	1.04	1.02		
4000		1.06	1.02	1.08		
5000		1.07	1.03	1.08	1.06	
6000			1.03	1.10	1.09	
7250			1.03	1.10	1.02	
8500			1.06	1.09	1.04	
10000			1.06	1.11	1.08	1.06



(a) Flight envelope including its design points.

(b) K_i [deg/$(g \cdot s)$] gain surface.

(c) K_p [deg/g] gain surface.

(d) K_q [s] gain surface.

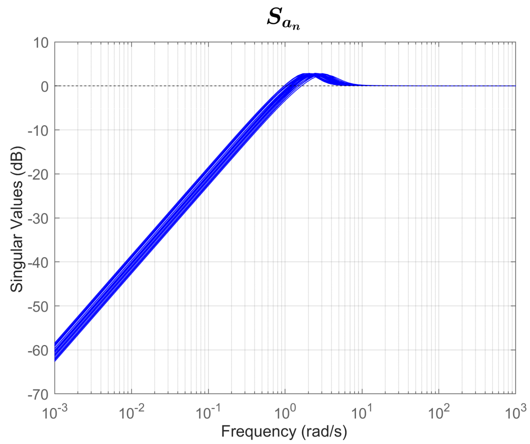
Fig. 3 Gain-scheduling results: Flight envelope and the gain surfaces, tuned by SOFHi.

D. Linear Analysis

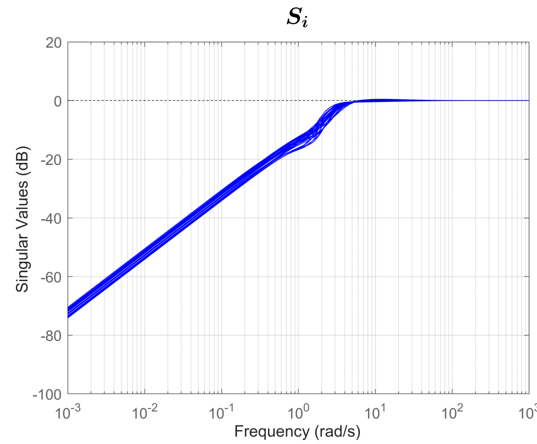
Now that the gain surfaces have been tuned in subsection VI.B and subsection VI.C, the closed-loop system can be analyzed in both frequency- and time-domain. This will be done in the subsequent subsections.

1. Sensitivity Functions

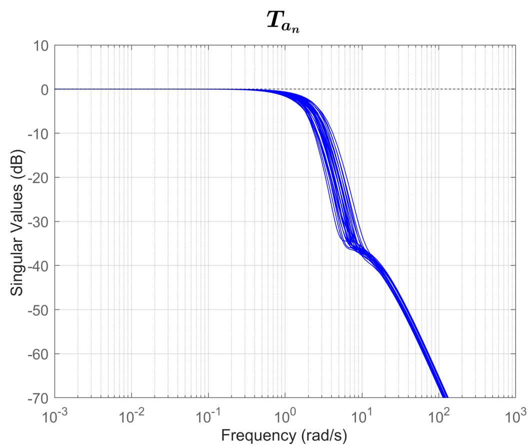
The six most relevant sensitivity functions to assess the frequency-domain characteristics of the closed-loop system are shown in Fig. 4 for the first stage in the 2-DoF design. To attenuate input- and output disturbances acting on both the plant input and output, S_{a_n}, $S_{a_n}G$, and S_i are minimized at low frequencies. Furthermore, it can be seen that the KS_{a_n}, T_{a_n}, and T_i have adequate roll-off at higher frequencies to attenuate the higher frequency measurement noise and unmodelled dynamics at the plant input.



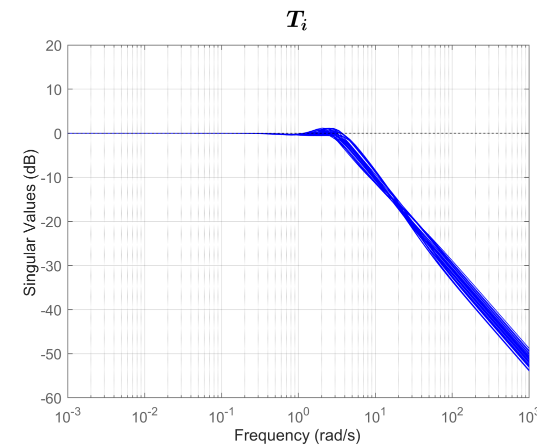
(a) Singular value plot of S_{a_n}.



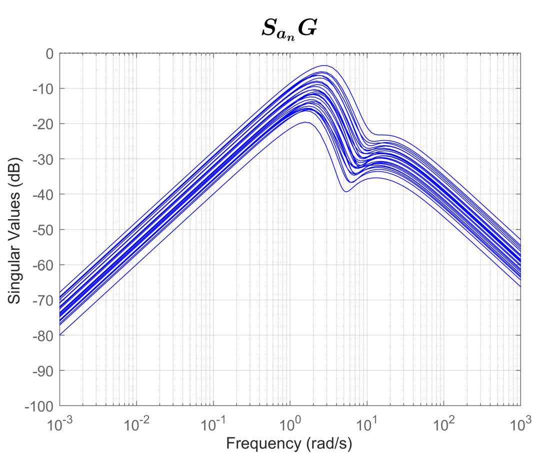
(b) Singular value plot of S_i.



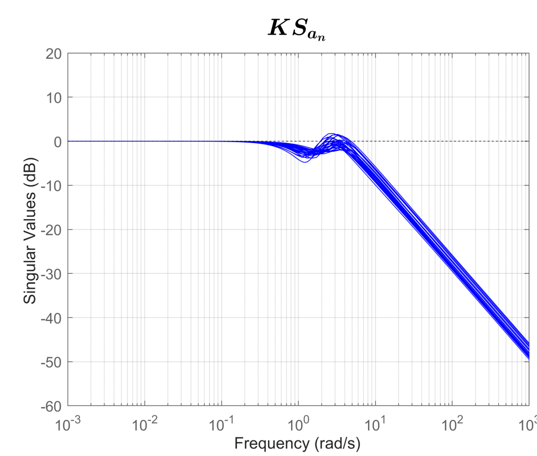
(c) Singular value plot of T_{a_n}.



(d) Singular value plot of T_i.



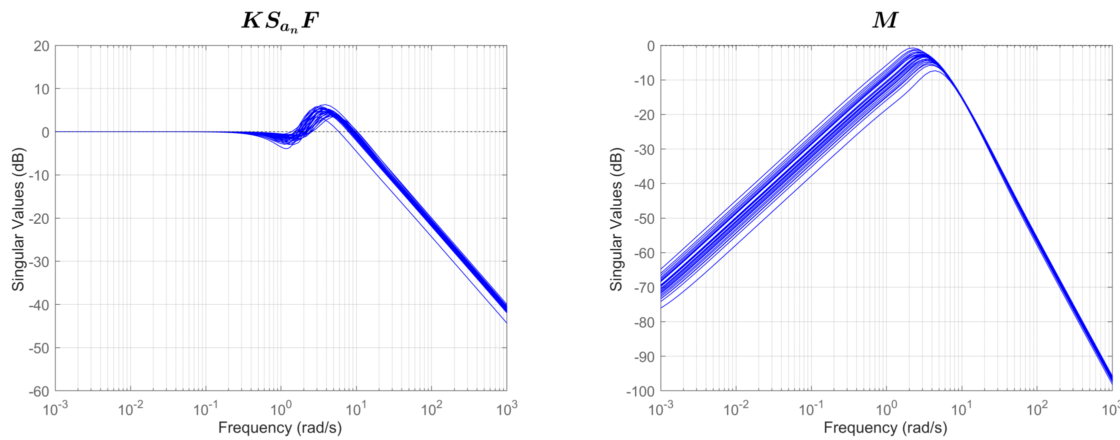
(e) Singular value plot of $S_{a_n}G$.



(f) Singular value plot of KS_{a_n}.

Fig. 4 Sensitivity functions of the first stage in the 2-DoF design described in subsection VI.B.

The sensitivity functions of the second stage in the 2-DoF design are shown in Fig. 5. In Fig. 5a, it can be seen that $KS_{a_n}F$ has a higher peak than KS_{a_n} in the mid-frequency range, which is due to the addition of the feedforward function F, which leads to the controlled system requiring more control effort in this frequency range. Furthermore, the function M in Fig. 5b is seen to be small at lower frequencies to reduce the error between the reference model and the output of the system at lower- to mid-frequencies. A clear trade-off takes place, where the peak of M is limited as much as possible in the mid-frequency range, while minimizing $KS_{a_n}F$, i.e. the required control effort from pilot input.

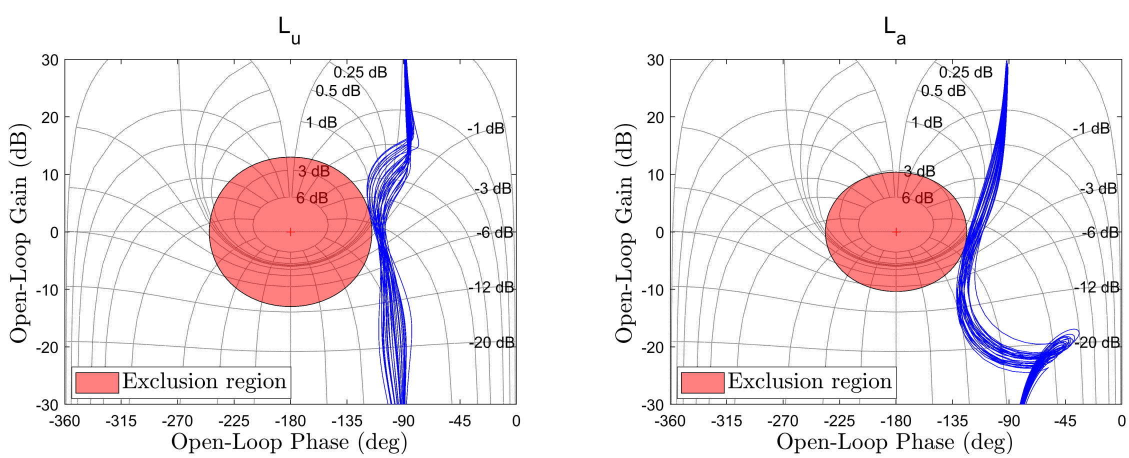


(a) Singular value plot of $KS_{a_n}F$. **(b) Singular value plot of M.**

Fig. 5 Singular values of the sensitivity functions of the second stage in the 2-DoF design described in subsection VI.B.

2. Stability Margins

Classical stability margins can give an optimistic perspective of the stability margins, since they do not take into account simultaneous gain and phase. To assess the stability of the closed-loop system more conservatively, the Nichols plots of each flight point including the worse-case exclusion region are shown in Fig. 6 at plant input and output. The smallest disk margin can be seen in Fig. 6b and consists of a 10 [dB] gain margin and 56 [deg] phase margin. The minimum requirements of 6 [dB] and 40 [deg], as defined in subsection VI.A, are thus considered to be met.



(a) At plant input. **(b) At plant output.**

Fig. 6 Nichols plots of all flight points [blue] at both plant input as well as normal acceleration output. Exclusion region [red] is based on the worst-case disk margin of all the flight points.

3. Linear Simulations

To assess the performance in controlling the normal acceleration output, time-domain simulations were performed on linear models around each of the 27 flight points in Fig. 3a. These are presented in Fig. 7. The results show good reference tracking and disturbance rejection, with no steady-state error, very small percentages in overshoot and a settling time under the required 3 seconds.

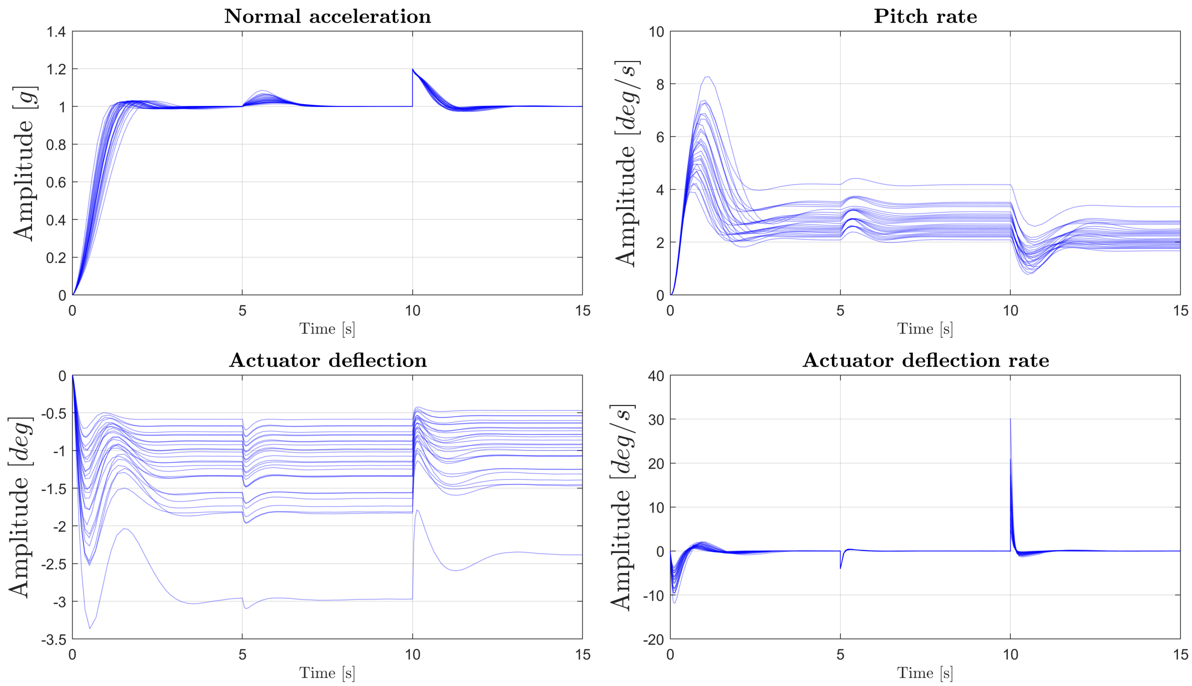


Fig. 7 Linear simulations of all the flight points including an input disturbance of 0.2 [deg] at 5 seconds and an output disturbance of 0.2 [g] at 10 seconds.

The specific values for the response parameters are presented in Table 7. In Table 7b, it can be seen that the damping ratios of the short-period dynamics are all above the minimum requirement of 0.3 [-] set in subsection VI.A. Furthermore, the settling times in Table 7c and the percentages overshoot in Table 7e meet the requirements of respectively 3 seconds and 10% for all of the flight points.

4. Handling Qualities

Handling qualities are analyzed to assess the short-period dynamics of the controlled system. To do this, first a Low-Order Equivalent System (LOES) is fit to the Higher-Order System (HOS). Since the LOES is a second-order transfer function, this allows to extract important short-period characteristics such as the natural frequency, damping ratio and time constant. Ultimately, this allows one to obtain the longitudinal handling qualities, which are assessed through the Control Anticipation Parameter (CAP) [35], one of the design requirements defined in subsection VI.A:

$$\mathrm{CAP} = \frac{\omega_{\mathrm{sp}}^2}{n/\alpha}, \tag{21}$$

The result of this is visualized in Fig. 8. It can be seen that the gain-scheduled controller meets the requirements for CAP. Naturally, there is a clear trade-off between improving CAP requirements and reducing control effort, so care was taken to ensure as little control effort as possible, while still meeting the CAP requirements.

E. Non-Linear Simulations

To showcase the robustness of the gain-scheduled controller, Monte-Carlo simulations are performed with varying uncertain parameters, which are chosen to be the longitudinal position of the center of gravity X_{COG} and the aircraft

Table 7 **Linear response characteristics for all designed flight points.**

(a) Natural frequency ω_{sp} [rad/s].

Mach [-] / h [m]	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1000	2.28	2.77	3.23	3.79		
2500		2.52	2.92	3.38		
4000		2.29	2.65	3.10		
5000		2.15	2.49	2.87	3.23	
6000			2.46	2.67	3.03	
7250			2.12	2.47	2.75	
8500			1.93	2.24	2.55	
10000			1.75	2.01	2.26	2.54

(b) Damping ratio ζ_{sp} [-].

Mach [-] / h [m]	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1000	0.47	0.49	0.51	0.54		
2500		0.47	0.50	0.52		
4000		0.46	0.48	0.51		
5000		0.46	0.48	0.49	0.52	
6000			0.48	0.49	0.50	
7250			0.46	0.48	0.49	
8500			0.45	0.47	0.49	
10000			0.44	0.46	0.47	0.49

(c) Settling times [s] with a 5% threshold.

Mach [-] / h [m]	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1000	2.21	1.85	1.56	1.01		
2500		2.03	1.75	1.45		
4000		2.21	1.92	1.65		
5000		2.34	2.02	1.79	1.57	
6000			2.04	1.92	1.69	
7250			2.35	2.06	1.85	
8500			2.57	2.23	1.98	
10000			2.81	2.46	2.22	1.99

(d) Rise times [s] from 10% to 90% of the steady-state value.

Mach [-] / h [m]	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1000	1.08	0.88	0.77	0.68		
2500		0.96	0.84	0.73		
4000		1.04	0.91	0.79		
5000		1.11	0.97	0.84	0.75	
6000			0.97	0.89	0.79	
7250			1.11	0.95	0.86	
8500			1.20	1.04	0.92	
10000			1.32	1.15	1.03	0.92

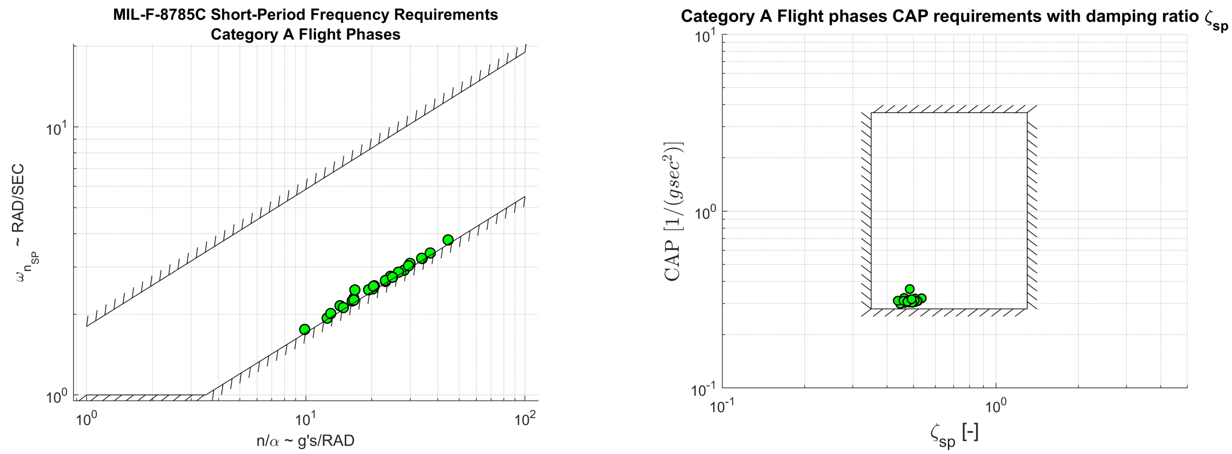
(e) Percentages in overshoot.

Mach [-] / h [m]	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1000	2.31	2.53	2.22	1.45		
2500		2.69	2.54	2.08		
4000		2.80	2.70	2.60		
5000		2.80	2.65	2.79	2.42	
6000			2.93	3.02	2.76	
7250			2.85	3.12	2.86	
8500			2.94	3.10	2.99	
10000			2.91	3.18	3.19	3.15

(f) Bandwidth ω_{BW} [rad/s] of the loop gain L_{a_n}.

Mach [-] / h [m]	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1000	2.23	2.49	2.78	3.28		
2500		2.40	2.68	3.11		
4000		2.23	2.58	2.68		
5000		2.15	2.49	2.68	3.11	
6000			2.36	2.58	3.00	
7250			2.27	2.54	2.89	
8500			2.11	2.49	2.78	
10000			2.03	2.27	2.58	2.78

mass m. Variations in these parameters range in $X_{\mathrm{COG}} \pm 4.35\%$ and $m \pm 1000$ kg, similarly to [34]. The simulations were performed on the non-linear 6-DoF model from [28, 29] for high-g maneuvers, for which the results are shown in Fig. 9. It can be seen that even for very high g's, the control effort is small and the response accurately follows the reference signal. Furthermore, although uncertainties can be seen to have an effect on the response, the controller is still able to accurately track the reference signal with a gradual degradation in performance for increasing uncertainty. The 6-DoF position and attitude trajectory of the aircraft for the nominal run can be seen in Fig. 10 in the Appendix.



(a) CAP requirement.

(b) Damping ratio requirement.

Fig. 8 Level 1 CAP requirements for Category A flight phases based on [35]. Dots indicate the result for each of the 27 flight points. Green indicates that the dot is within the desired region.

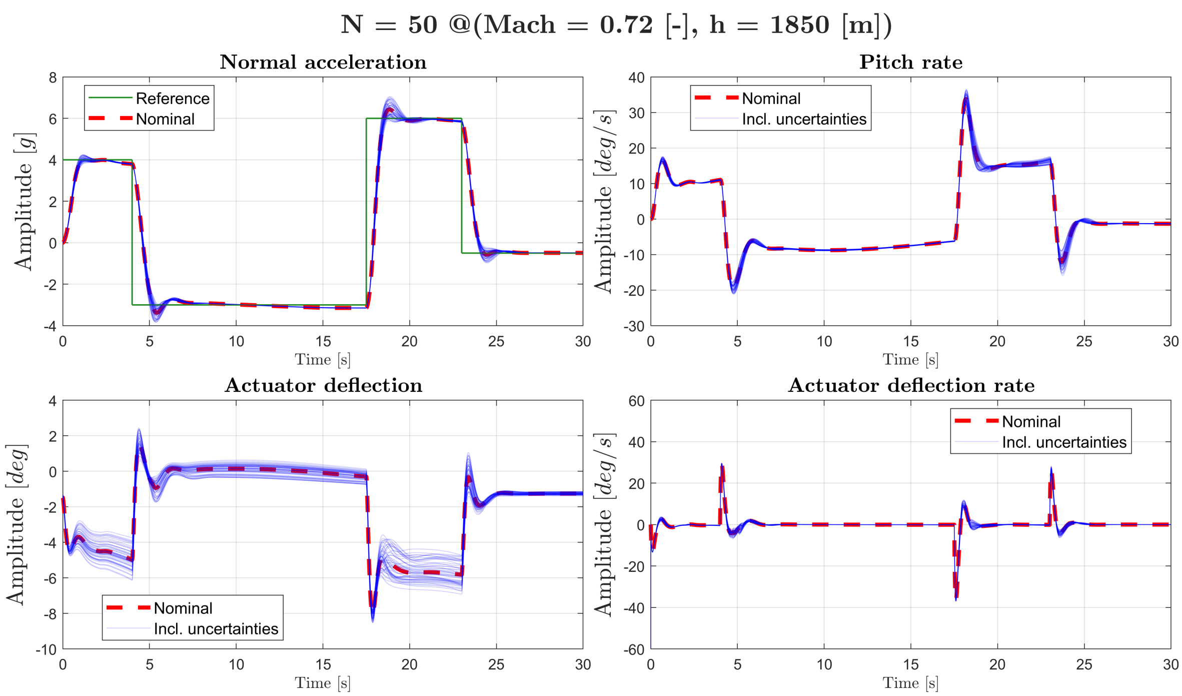


Fig. 9 Monte-Carlo simulations applied on the non-linear model with 50 runs to showcase the robustness of the controller at high-g's maneuvers.

VII. Conclusion

The purpose of this paper was to investigate Lyapunov-based SOF synthesis methods that could potentially compete with the well-established non-smooth optimization methods, Hinfstruct and HIFOO. This would ultimately lead to more insight into the performance of SOF controller synthesis methods in the H_∞ framework.

Three algorithms were deemed most promising, i.e. T-K iteration and two S-variable approaches; these were implemented and bundled into a toolbox named SOFHi. Results showed that SOFHi was able to significantly outperform HIFOO in terms of H_∞ performance index. However, comparing SOFHi to Hinfstruct showed that Hinfstruct outperformed SOFHi when no significance margin was considered, but SOFHi was able to compete and even slightly outperform Hinfstruct when this margin was in fact considered. It was thus shown that Lyapunov-based methods are indeed competitive to the non-smooth optimization methods, although the Lyapunov methods are limited by the amount of decision variables in the Lyapunov matrices, so the methods are mostly applicable to low-medium sized plants. In the end, the Lyapunov methods serve as an alternative to the well-established non-smooth methods.

Structured SOF was achieved in the implementation and a "fast setting" of SOFHi was proposed, $\text{SOFHi}_{\text{EVO}}$, which was able to significantly improve on SOFHi in terms of computational efficiency. This allows one to start many more runs from different initial conditions within the same time-frame. Doing this, $\text{SOFHi}_{\text{EVO}}$ was able to significantly outperform HIFOO and was able to be competitive to Hinfstruct, even outperforming it for one of its runs.

Finally, to showcase the effectiveness of the algorithms, SOFHi was applied to design a robust gain-scheduled PI-controller for a normal acceleration CAS of the F-16 Fighting Falcon for a large portion of the flight envelope. Performance was achieved through meeting the handling qualities and robustness to uncertainties was demonstrated through Monte-Carlo simulations.

VIII. Appendix

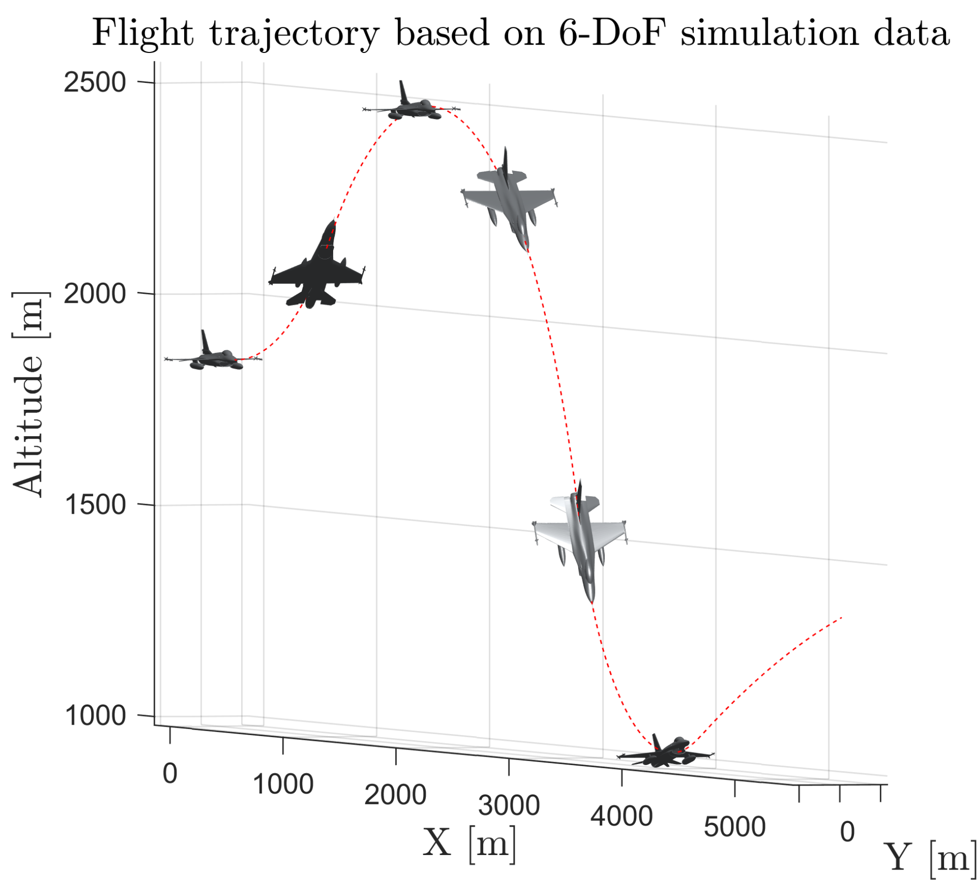


Fig. 10 Position and attitude trajectory of the aircraft, based on 6-DoF data from the non-linear model [36].

Table 8 $\|T_{zw}(P,K)\|_\infty$ **for 30 random starts applied on benchmark models from the *Compl$_e$ib* library. 'X' denotes either that the algorithm has failed or that the algorithm is not applicable to that model. Superior results are placed in bold.**

Model	T-K	S-approach I	S-approach II	SOFHi	Hinfstruct	HIFOO
AC2	**1.114893E-01**	1.114946E-01	1.114893E-01	**1.114893E-01**	1.114893E-01	1.114893E-01
AC3	3.429719E+00	3.444490E+00	3.523945E+00	3.429719E+00	3.606264E+00	**3.424724E+00**
AC5	6.640498E+02	**6.610349E+02**	6.636103E+02	**6.610349E+02**	6.644574E+02	6.652504E+02
AC6	4.113974E+00	**4.113943E+00**	4.113965E+00	**4.113943E+00**	4.113956E+00	4.113944E+00
AC7	X	6.509075E-02	6.509076E-02	6.509075E-02	6.509066E-02	**6.509060E-02**
AC8	X	2.005012E+00	2.005012E+00	2.005012E+00	2.005012E+00	**2.005012E+00**
AC11	2.834266E+00	2.814323E+00	2.827427E+00	2.814323E+00	**2.812956E+00**	2.831976E+00
AC15	1.516871E+01	1.602499E+01	1.589971E+01	1.516871E+01	**1.516871E+01**	1.516884E+01
AC17	6.612428E+00	6.612428E+00	6.612428E+00	6.612428E+00	6.612428E+00	6.612428E+00
AC18	1.074842E+01	X	1.069786E+01	1.069786E+01	**1.069716E+01**	1.255910E+01
HE1	1.538209E-01	1.538620E-01	1.539639E-01	1.538209E-01	1.538209E-01	**1.537869E-01**
HE2	4.258125E+00	3.936859E+00	3.900153E+00	3.900153E+00	**3.896131E+00**	3.921615E+00
HE4	2.304968E+01	2.413298E+01	2.283837E+01	2.283837E+01	**2.283817E+01**	2.283867E+01
HE6	X	1.923842E+02	1.923599E+02	1.923599E+02	**1.923531E+02**	1.923736E+02
HE7	X	1.923880E+02	1.923882E+02	1.923880E+02	**1.923862E+02**	1.923913E+02
REA1	8.665937E-01	8.670439E-01	8.656412E-01	8.656412E-01	**8.654988E-01**	8.665155E-01
REA2	1.149412E+00	1.149488E+00	1.154296E+00	1.149412E+00	1.148838E+00	**1.148353E+00**
DIS1	**4.159406E+00**	4.162835E+00	4.160295E+00	**4.159406E+00**	4.159706E+00	4.163724E+00
DIS2	1.022484E+00	1.040235E+00	1.037782E+00	1.022484E+00	1.054774E+00	**1.022421E+00**
DIS3	1.062568E+00	1.299316E+00	1.063933E+00	1.062568E+00	**1.061249E+00**	1.066600E+00
WEC1	4.055088E+00	4.074383E+00	4.050097E+00	4.050097E+00	**4.050014E+00**	4.050143E+00
MFP	3.277044E+01	4.273129E+01	3.159155E+01	3.159155E+01	3.158987E+01	**3.158987E+01**
EB1	X	**3.122517E+00**	3.122520E+00	**3.122517E+00**	3.122525E+00	3.122521E+00
EB2	X	**2.020041E+00**	2.020041E+00	**2.020041E+00**	2.020102E+00	2.020041E+00
EB3	X	**2.057463E+00**	2.057463E+00	**2.057463E+00**	2.057527E+00	2.057463E+00
EB4	X	2.056323E+00	2.056323E+00	2.056323E+00	2.056387E+00	**2.056323E+00**
PAS	**5.232251E-01**	1.980016E+01	X	**5.232251E-01**	X	7.046017E-01
PSM	9.202430E-01	9.202430E-01	9.202430E-01	9.202430E-01	9.202430E-01	**9.202430E-01**
NN1	**1.376257E+01**	1.630724E+01	1.438951E+01	**1.376257E+01**	1.376959E+01	1.382931E+01
NN2	**2.221556E+00**	2.221556E+00	2.221556E+00	**2.221556E+00**	2.221583E+00	2.221556E+00
NN4	1.362297E+00	1.360271E+00	1.358669E+00	1.358669E+00	**1.358663E+00**	1.359066E+00
NN5	2.665445E+02	X	X	2.665445E+02	2.665445E+02	**2.665445E+02**
NN6	5.812074E+03	5.738319E+03	**3.688675E+03**	**3.688675E+03**	5.602556E+03	5.602552E+03
NN7	7.410449E+01	7.407439E+01	X	7.407439E+01	7.407568E+01	**7.407439E+01**
NN8	2.890486E+00	2.891936E+00	2.914111E+00	2.890486E+00	**2.884880E+00**	2.885378E+00
NN11	X	8.383050E-02	**8.335992E-02**	**8.335992E-02**	9.116151E-02	9.261801E-02
NN12	1.618333E+01	1.730641E+01	1.682266E+01	1.618333E+01	**1.611875E+01**	1.683929E+01
NN13	X	1.405944E+01	1.406420E+01	1.405944E+01	**1.405794E+01**	1.405795E+01
NN15	9.808961E-02	9.809005E-02	8.322924E+00	9.808961E-02	**9.808961E-02**	9.809001E-02
NN16	**9.555152E-01**	9.568741E-01	9.559968E-01	**9.555152E-01**	9.555567E-01	9.555658E-01
NN17	1.121821E+01	1.121821E+01	1.121821E+01	1.121821E+01	1.121821E+01	**1.121821E+01**
HF2D10	7.979732E+04	X	X	7.979732E+04	**7.978836E+04**	7.989110E+04
HF2D11	**7.689651E+04**	X	X	**7.689651E+04**	7.723728E+04	7.700805E+04
HF2D13	1.015485E+05	X	X	1.015485E+05	1.015485E+05	**1.015485E+05**
HF2D14	5.308371E+05	X	X	5.308371E+05	**5.263977E+05**	5.270185E+05
HF2D15	1.749107E+05	X	X	1.749107E+05	**1.733239E+05**	1.749827E+05
HF2D16	**4.441442E+05**	X	X	**4.441442E+05**	4.441442E+05	4.441442E+05
HF2D17	3.002366E+05	X	X	3.002366E+05	3.002366E+05	**3.002366E+05**
HF2D18	1.196238E+02	1.227614E+02	1.196883E+02	1.196238E+02	**1.195602E+02**	1.236557E+02
BDT1	2.662119E-01	2.680469E-01	2.662325E-01	2.662119E-01	**2.662119E-01**	2.662119E-01
TMD	X	2.180880E+00	2.157190E+00	2.157190E+00	**2.136572E+00**	2.523276E+00
FS	**8.548169E+04**	X	X	**8.548169E+04**	8.551230E+04	X
DLR1	X	X	2.779558E+00	2.779558E+00	2.777289E+00	**2.777289E+00**
ROC7	1.121820E+00	**1.117512E+00**	1.119708E+00	**1.117512E+00**	1.120327E+00	1.122203E+00

Table 9 $\|T_{zw}(P,K)\|_\infty$ **for 100 random starts applied on benchmark models from the** ***Compl$_e$ib*** **library. 'X' denotes either that the algorithm has failed or that the algorithm is not applicable to that model. Superior results are placed in bold.**

Model	T-K	S-approach I	S-approach II	SOFHi	Hinfstruct	HIFOO
AC2	1.114893E-01	1.114895E-01	1.114893E-01	1.114893E-01	1.114893E-01	**1.114893E-01**
AC3	3.417253E+00	3.434777E+00	3.432525E+00	3.417253E+00	**3.403622E+00**	3.435171E+00
AC5	**6.602662E+02**	6.616976E+02	6.636129E+02	**6.602662E+02**	6.640303E+02	6.653228E+02
AC6	4.113973E+00	**4.113941E+00**	4.113955E+00	**4.113941E+00**	4.113942E+00	4.113956E+00
AC7	X	6.509075E-02	6.509076E-02	6.509075E-02	**6.509060E-02**	6.509066E-02
AC8	X	2.005012E+00	2.005012E+00	2.005012E+00	2.005012E+00	**2.005012E+00**
AC11	2.819620E+00	2.815367E+00	2.826457E+00	2.815367E+00	**2.812758E+00**	2.826595E+00
AC15	1.516871E+01	1.599973E+01	1.579698E+01	1.516871E+01	**1.516871E+01**	1.516876E+01
AC17	6.612428E+00	6.612428E+00	6.612428E+00	6.612428E+00	6.612428E+00	6.612428E+00
AC18	1.074677E+01	1.069754E+01	1.069794E+01	1.069754E+01	**1.069715E+01**	1.191095E+01
HE1	**1.538209E-01**	1.538620E-01	1.539636E-01	**1.538209E-01**	1.538209E-01	1.538309E-01
HE2	3.922528E+00	3.904555E+00	3.901241E+00	3.901241E+00	**3.895785E+00**	3.908621E+00
HE4	2.286146E+01	2.377527E+01	2.283858E+01	2.283858E+01	**2.283817E+01**	2.283817E+01
HE6	X	1.923592E+02	1.923570E+02	1.923570E+02	**1.923529E+02**	1.923544E+02
HE7	X	1.923879E+02	1.923882E+02	1.923879E+02	**1.923860E+02**	1.923902E+02
REA1	8.666032E-01	8.659882E-01	8.657506E-01	8.657506E-01	**8.654989E-01**	8.660901E-01
REA2	1.148926E+00	1.148471E+00	1.154289E+00	1.148471E+00	**1.148223E+00**	1.148925E+00
DIS1	**4.159374E+00**	4.162344E+00	4.160670E+00	**4.159374E+00**	4.159532E+00	4.159992E+00
DIS2	**1.022326E+00**	1.022920E+00	1.022776E+00	**1.022326E+00**	1.054774E+00	1.023024E+00
DIS3	1.063862E+00	1.107925E+00	1.068321E+00	1.063862E+00	**1.061295E+00**	1.062616E+00
WEC1	4.053096E+00	4.050243E+00	4.050419E+00	4.050243E+00	**4.050012E+00**	4.050237E+00
MFP	3.167343E+01	3.159272E+01	3.159157E+01	3.159157E+01	**3.158987E+01**	3.158987E+01
EB1	X	**3.122517E+00**	3.122520E+00	**3.122517E+00**	3.122521E+00	3.122525E+00
EB2	X	**2.020041E+00**	2.020041E+00	**2.020041E+00**	2.020041E+00	2.020102E+00
EB3	X	**2.057463E+00**	2.057463E+00	**2.057463E+00**	2.057463E+00	2.057527E+00
EB4	X	**2.056323E+00**	2.056323E+00	**2.056323E+00**	2.056323E+00	2.056387E+00
PAS	**5.068844E-01**	3.146420E+01	X	**5.068844E-01**	X	7.965271E-01
PSM	9.202430E-01	9.202430E-01	9.202430E-01	9.202430E-01	9.202430E-01	9.202430E-01
NN1	**1.374446E+01**	1.383531E+01	1.388575E+01	**1.374446E+01**	1.377347E+01	1.384751E+01
NN2	2.221556E+00	**2.221556E+00**	2.221556E+00	**2.221556E+00**	2.221556E+00	2.221583E+00
NN4	1.362235E+00	1.360414E+00	1.359962E+00	1.359962E+00	**1.358776E+00**	1.359355E+00
NN5	2.665445E+02	X	2.665445E+02	2.665445E+02	2.665445E+02	**2.665445E+02**
NN6	5.777714E+03	5.602552E+03	**3.688675E+03**	**3.688675E+03**	5.602552E+03	5.602553E+03
NN7	7.411120E+01	**7.407439E+01**	7.407439E+01	**7.407439E+01**	7.407439E+01	7.407568E+01
NN8	2.886156E+00	2.887578E+00	2.891828E+00	2.886156E+00	**2.884885E+00**	2.884980E+00
NN11	X	8.494687E-02	**8.329533E-02**	**8.329533E-02**	9.112113E-02	9.298774E-02
NN12	**1.573776E+01**	X	2.035457E+01	**1.573776E+01**	1.635851E+01	1.668808E+01
NN13	X	1.405835E+01	1.405809E+01	1.405809E+01	**1.405794E+01**	1.405795E+01
NN15	9.808961E-02	9.808962E-02	9.809066E-02	9.808961E-02	**9.808961E-02**	9.808988E-02
NN16	**9.552605E-01**	9.578570E-01	9.559619E-01	**9.552605E-01**	9.555631E-01	9.556033E-01
NN17	1.121821E+01	1.121821E+01	1.121821E+01	1.121821E+01	1.121821E+01	**1.121821E+01**
HF2D10	7.979057E+04	X	X	7.979057E+04	**7.978000E+04**	7.990348E+04
HF2D11	**7.689588E+04**	X	X	**7.689588E+04**	7.712759E+04	7.702937E+04
HF2D13	1.015485E+05	X	X	1.015485E+05	1.015485E+05	**1.015485E+05**
HF2D14	5.270742E+05	X	X	5.270742E+05	**5.263977E+05**	5.273724E+05
HF2D15	1.749006E+05	X	X	1.749006E+05	**1.733193E+05**	1.737786E+05
HF2D16	**4.441442E+05**	X	X	**4.441442E+05**	4.441442E+05	4.442765E+05
HF2D17	3.002366E+05	X	X	3.002366E+05	3.002366E+05	**3.002366E+05**
HF2D18	1.196188E+02	1.247764E+02	1.237381E+02	1.196188E+02	**1.195597E+02**	1.195751E+02
BDT1	**2.662119E-01**	2.662240E-01	2.662323E-01	**2.662119E-01**	2.662119E-01	2.662119E-01
TMD	X	2.160220E+00	2.150294E+00	2.150294E+00	**2.128761E+00**	2.294840E+00
FS	**8.548154E+04**	X	X	**8.548154E+04**	8.548257E+04	X
DLR1	X	X	2.777886E+00	2.777886E+00	**2.777289E+00**	2.777289E+00
ROC7	1.121657E+00	**1.116740E+00**	1.119300E+00	**1.116740E+00**	1.120098E+00	1.121316E+00

References

[1] Toscano, R., *Structured controllers for uncertain systems*, Springer, 2013.

[2] Toker, O., and Ozbay, H., "On the NP-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback," *Proceedings of 1995 American Control Conference-ACC'95*, Vol. 4, IEEE, 1995, pp. 2525–2526.

[3] Sadabadi, M. S., and Peaucelle, D., "From static output feedback to structured robust static output feedback: A survey," *Annual Reviews in Control*, Vol. 42, 2016, pp. 11–26. https://doi.org/10.1016/J.ARCONTROL.2016.09.014.

[4] Cao, Y.-Y., and Sun, Y.-X., "Static output feedback simultaneous stabilization: ILMI approach," *International journal of control*, Vol. 70, No. 5, 1998, pp. 803–814.

[5] Tran Dinh, Q., Gumussoy, S., Michiels, W., and Diehl, M., "Combining Convex–Concave Decompositions and Linearization Approaches for Solving BMIs, With Application to Static Output Feedback," *IEEE Transactions on Automatic Control*, Vol. 57, No. 6, 2012, pp. 1377–1390. https://doi.org/10.1109/TAC.2011.2176154.

[6] Gadewadikar, J., Lewis, F., Xie, L., Kucera, V., and Abu-Khalaf, M., "Parameterization of all stabilizing H∞ static state-feedback gains: Application to output-feedback design," *Proceedings of the IEEE Conference on Decision and Control*, 2006, pp. 3566–3571. https://doi.org/10.1109/CDC.2006.377280.

[7] El Ghaoui, L., and Balakrishnan, V., "Synthesis of fixed-structure controllers via numerical optimization," Vol. 3, 1994, pp. 2678–2683 vol.3. https://doi.org/10.1109/CDC.1994.411398.

[8] Iwasaki, T., "The dual iteration for fixed-order control," *IEEE transactions on automatic control*, Vol. 44, No. 4, 1999, pp. 783–788.

[9] Iwasaki, T., and Skelton, R., "The XY-centring algorithm for the dual LMI problem: a new approach to fixed-order control design," *International Journal of Control*, Vol. 62, No. 6, 1995, pp. 1257–1272.

[10] Apkarian, P., Noll, D., and Tuan, H. D., "Fixed-order Hinf control design via a partially augmented Lagrangian method," *IFAC Proceedings Volumes*, Vol. 36, No. 8, 2003, pp. 69–74.

[11] Ghaoui, L. E., Oustry, F., and Aitrami, M., "A Cone Complementarity Linearization Algorithm for Static Output-Feedback and Related Problems," *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, Vol. 42, 1997.

[12] Prempain, E., and Postlethwaite, I., "Static output feedback stabilisation with Hinf performance for a class of plants," *Systems & Control Letters*, Vol. 43, No. 3, 2001, pp. 159–166.

[13] Rubió-Massegú, J., Rossell, J. M., Karimi, H. R., and Palacios-Quiñonero, F., "Static output-feedback control under information structure constraints," *Automatica*, Vol. 49, 2013, pp. 313–316. https://doi.org/10.1016/J.AUTOMATICA.2012.10.012.

[14] Sahoo, P. R., Goyal, J. K., Ghosh, S., and Naskar, A. K., "New results on restricted static output feedback H-8 controller design with regional pole placement," *IET Control Theory & Applications*, 2019. https://doi.org/10.1049/iet-cta.2018.6138, URL www.ietdl.org.

[15] Scherer, C., and Weiland, S., "Linear matrix inequalities in control," *Lecture Notes, Dutch Institute for Systems and Control, Delft, The Netherlands*, Vol. 3, No. 2, 2000.

[16] Feng, Z. Y., She, J., and Xu, L., "A brief review and insights into matrix inequalities for H_∞ static-output-feedback control and a local optimal solution," *International Journal of Systems Science*, Vol. 50, 2019, pp. 2292–2305. https://doi.org/10.1080/00207721.2019.1654008.

[17] Gopmandal, F., and Ghosh, A., "A hybrid search based synthesis of static output feedback controllers for uncertain systems with application to multivariable PID control," *International Journal of Robust and Nonlinear Control*, Vol. 31, 2021, pp. 6069–6090. https://doi.org/10.1002/RNC.5581.

[18] Apkarian, P., and Noll, D., "Nonsmooth Hinf synthesis," *IEEE Transactions on Automatic Control*, Vol. 51, No. 1, 2006, pp. 71–86.

[19] Burke, J. V., Henrion, D., Lewis, A. S., and Overton, M. L., "HIFOO-a MATLAB package for fixed-order controller design and Hinf optimization," *IFAC Proceedings Volumes*, Vol. 39, No. 9, 2006, pp. 339–344.

[20] Syrmos, V. L., Abdallah, C. T., Dorato, P., and Grigoriadis, K., "Static output feedback—A survey," *Automatica*, Vol. 33, 1997, pp. 125–137. https://doi.org/10.1016/S0005-1098(96)00141-0.

[21] Feng, Z. Y., Guo, H., She, J., and Xu, L., "Weighted sensitivity design of multivariable PID controllers via a new iterative LMI approach," *Journal of Process Control*, Vol. 110, 2022, pp. 24–34. https://doi.org/10.1016/J.JPROCONT.2021.11.016.

[22] Ebihara, Y., Peaucelle, D., and Arzelier, D., *S-variable approach to LMI-based robust control*, Springer, 2015.

[23] Gahinet, P., and Apkarian, P., "A linear matrix inequality approach to Hinf control," *International journal of robust and nonlinear control*, Vol. 4, No. 4, 1994, pp. 421–448.

[24] Arzelier, D., Gryazina, E. N., Peaucelle, D., and Polyak, B. T., "Mixed LMI/randomized methods for static output feedback control design," *Proceedings of the 2010 American Control Conference*, 2010, pp. 4683–4688. https://doi.org/10.1109/ACC.2010.5531075.

[25] Mattei, M., "Sufficient conditions for the synthesis of H-infinity fixed-order controllers," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, Vol. 10, No. 15, 2000, pp. 1237–1248.

[26] Feng, Z., and Allen, R., "Comments on 'Sufficient conditions for the synthesis of Hinf fixed-order controllers'," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, Vol. 14, No. 1, 2004, pp. 79–81.

[27] Leibfritz, F., "COMPleib: Constrained matrix optimization problem library," , 2006.

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

[29] Stevens, B. L., Lewis, F. L., and Johnson, E. N., *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems, 3rd Edition*, Wiley-Blackwell, 2016.

[30] Bérard, C., Biannic, J.-M., and Saussié, D., *La commande multivariable: Application au pilotage d'un avion*, Dunod, 2012.

[31] Ang, K. H., Chong, G., and Li, Y., "PID control system analysis, design, and technology," *IEEE transactions on control systems technology*, Vol. 13, No. 4, 2005, pp. 559–576.

[32] Bates, D., and Postlethwaite, I., *Robust multivariable control of aerospace systems*, Vol. 8, IOS Press, 2002.

[33] "MIL-STD-1797A - Flying Qualities of Piloted Aircraft," 1990. URL www.dodssp.daps.mil.

[34] Lhachemi, H., Saussié, D., and Zhu, G., "A robust and self-scheduled longitudinal flight control system: A multi-model and structured H_∞ approach," *AIAA Guidance, Navigation, and Control Conference*, 2014. https://doi.org/10.2514/6.2014-0601.

[35] "MIL-F-8785C," 1980.

[36] Scordamaglia, V., "Trajectory and Attitude Plot Version 3," , 2023. URL https://www.mathworks.com/matlabcentral/fileexchange/5656-trajectory-and-attitude-plot-version-3.

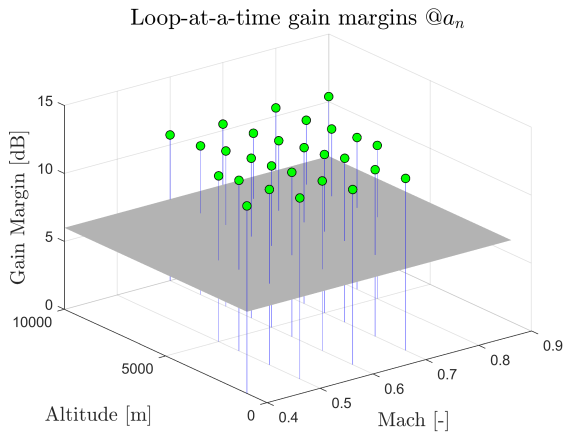
3

Additional Results and V&V

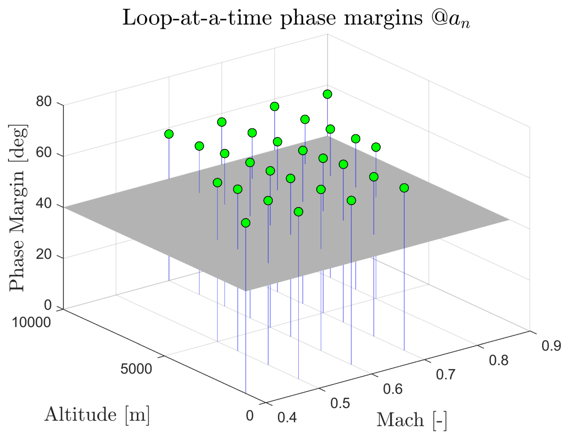
In this chapter any additional results to those of presented in the paper in chapter 2 are shown and elaborated on. Furthermore, verification and validation of the SOF algorithms and their corresponding results will be covered.

3.1. Loop-at-a-time Disk Margins

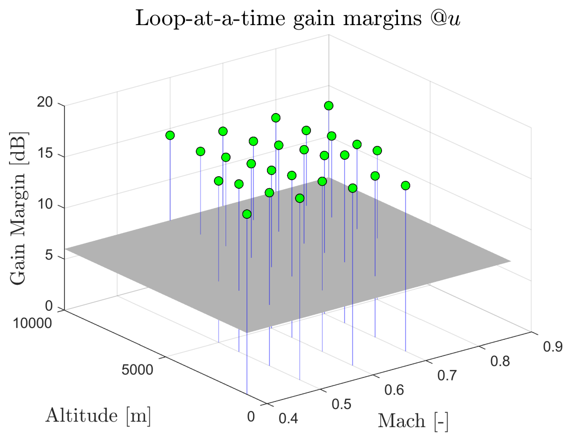
The worst-case disk margin was already presented through the Nichols plots in chapter 2, but an overview of the disk margins for each flight point is presented in Figure 3.1. It can be seen that the margins are all far above the minimum required values of 6 [dB] and 40 [deg] for the gain and phase margins, respectively, as defined in the design objectives in chapter 2.



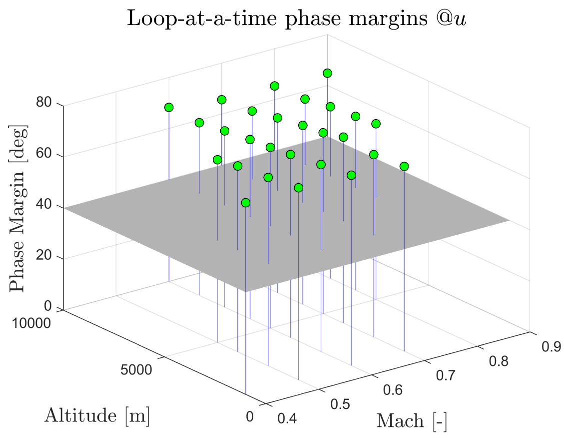
(a) Gain Margin at normal acceleration output.



(b) Phase Margin at normal acceleration output.



(c) Gain Margin at plant input.



(d) Phase Margin at plant input.

Figure 3.1: Loop-at-a-time disk margins at different locations in the controlled system. Green dots indicate significant margin to the minimum requirement.

3.2. Comparison Within SOFHi

The main purpose of this research is to assess the competitiveness of LMI-based SOF algorithms to the well-established methods, Hinfstruct and HIFOO; this has been covered in chapter 2. Additionally, it is worth it to compare the algorithms to themselves, to get a complete comparison. The result of this is presented in Table 3.1. It can be seen that the results do not change significantly with number of starts and that T-K iteration performs the best out of the three LMI algorithms.

Table 3.1: Comparison of the three algorithms in SOFHi for both 30 and 100 starts, applied on 54 benchmark models from *Compl$_e$ib*. Results represent the percentage of times the algorithm obtained the most optimal γ from all the algorithms. Superior results are placed in bold.

	T-K Iteration	S-variable App. I	S-variable App. II
30 starts	**53.70%**	24.07%	22.22%
100 starts	**50.00%**	27.78&	22.22%

3.3. Verification

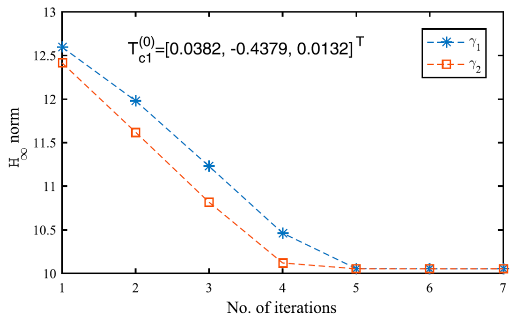
In this section, the verification process and results of the SOF algorithms are presented and elaborated on in subsection 3.3.1. Furthermore, verification of correct calculation of all stability margins are presented and explained in subsection 3.3.2.

3.3.1. Verification of the SOF algorithms

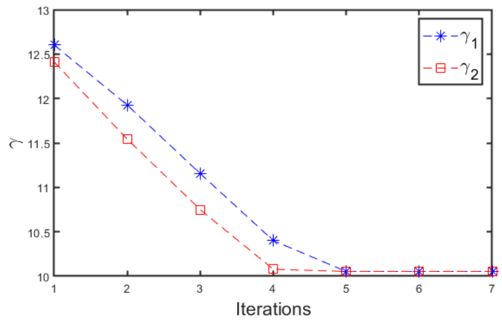
To obtain confidence in the output of the SOF algorithms, the results of the implementation are compared to the examples in the paper. Analysis on the closed-loop system with the obtained gain matrix should yield exactly similar values as the output of the algorithms.

Verifying T-K iteration

Verifying correct implementation of T-K iteration algorithm is done by comparing the results from the implementation to the results that are presented in the paper. Fortunately, the authors provide a reproducible example. Any probabilistic factors, such as determining the initial CTM, are circumvented because this CTM is already provided for Example 1 in (Feng, She, and Xu 2019). The results of this are presented in Figure 3.2. When the same LMI solver as the paper is selected, the exact same results are found as those presented in (Feng, She, and Xu 2019). T-K iteration is therefore considered verified.



(a) From (Feng, She, and Xu 2019).



(b) From SOFHi implementation with the *LMI Toolbox* as LMI solver.

Figure 3.2: Comparison in iteration procedure through γ for $T_{c1}^{(0)} = \begin{bmatrix} 0.0382 & -0.4379 & 0.0132 \end{bmatrix}^\top$ for Example 1 in (Feng, She, and Xu 2019).

Verifying Hit-and-Run

(Ebihara, Dimitri Peaucelle, and Denis Arzelier 2015) unfortunately provides no reproducible example for the S-variable approaches. So, the same procedure as subsubsection 3.3.1 cannot be applied. Luckily, results were shown for the H.R. algorithm, so to verify the Hit-and-Run algorithm, results shown in the paper are exactly reproduced to ensure correct implementation. The results of this are shown in Figure 3.3, where 1000 stabilizing SOF gain matrices $K_{SOF} \in \mathbb{R}^{1\times 2}$ are obtained. It can be seen that the solution space for SOF gains obtained by the SOFHi implementation, corresponds to the results shown in (Ebihara, Dimitri Peaucelle, and Denis Arzelier 2015) for both AC7 and NN5 benchmark models. The H.R. algorithm is therefore considered verified.[1]

Verifying the peak gains

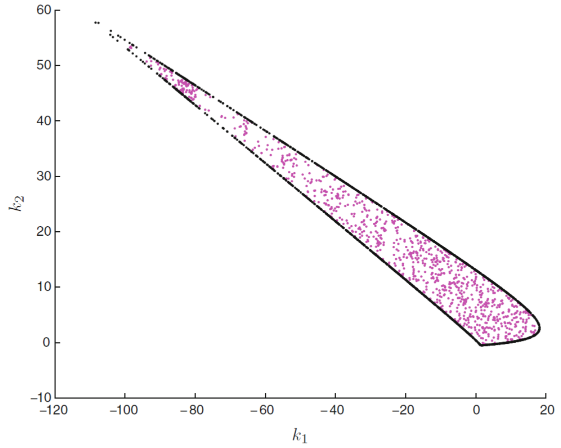
A major part of the verification process of the SOF algorithms is verifying that the γ's obtained from the SOF algorithms correspond to $\|T_{zw}(P,K)\|_\infty$, where $T_{zw}(P,K)$ is the interconnection between P and K in Linear Fractional Transformation (LFT) form. Consider P to be defined as follows:

$$P := \left(\begin{array}{c|cc} A & B_w & B_u \\ \hline C_z & D_{zw} & D_{zu} \\ C_y & D_{yw} & D_{yu} \end{array}\right) \tag{3.1}$$

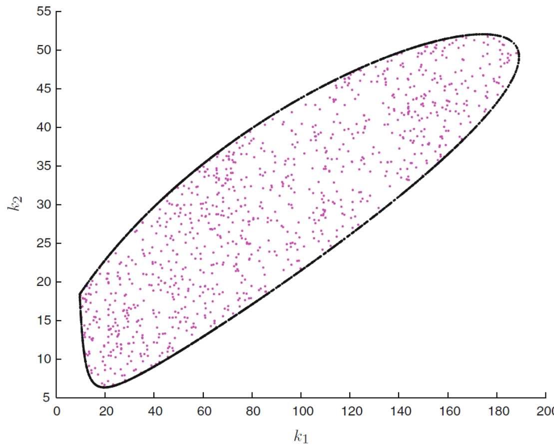
T_{zw} is then obtained using the MATLAB function `lft` and the SOF gain matrix K synthesized by SOFHi to make the lower LFT $F_l(P,K) = P_{zw} + P_{zu}K(I - P_{yu}K)^{-1}P_{yw} = T_{zw}$.

To verify with full confidence, the γ's are compared to two values: the output of the MATLAB function `getPeakGain`(T_{zw}), which outputs the peak gain of the frequency response, and the maximum of the upper

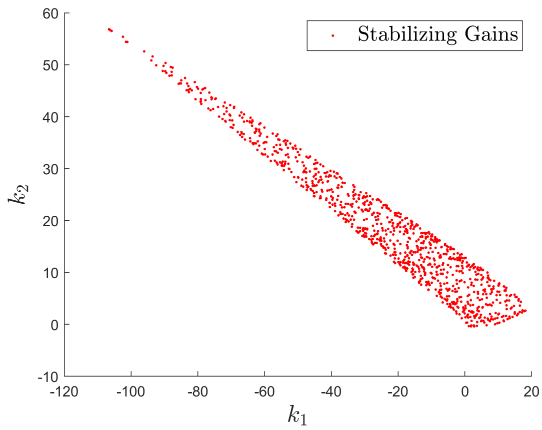
[1]The exact same gains do not have to be found since it is a probabilistic algorithm. 1000 runs are therefore done, so the same solution space can be observed.



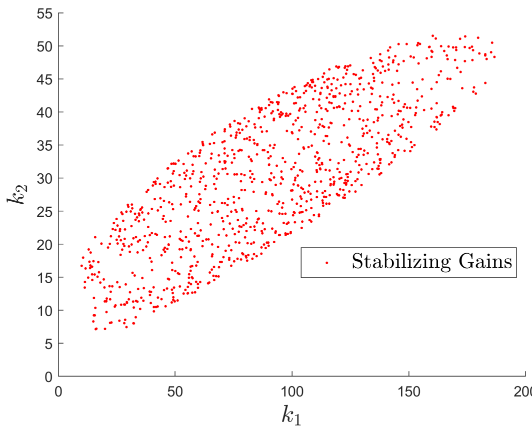
(a) Results of H.R. for *AC7*, extracted from (Ebihara, Dimitri Peaucelle, and Denis Arzelier 2015, Page 193).



(b) Results of H.R. for *NN5*, extracted from (Ebihara, Dimitri Peaucelle, and Denis Arzelier 2015, Page 194).



(c) Results of H.R. for *AC7*, from SOFHi implementation.



(d) Results of H.R. for *NN5*, from SOFHi implementation.

Figure 3.3: Comparisons in stabilizing SOF gain matrices between (Ebihara, Dimitri Peaucelle, and Denis Arzelier 2015) and the SOFHi implementation of the H.R. algorithm. $K_{SOF} \in \mathbb{R}^{1\times 2}$ for both *AC7* and *NN5*.

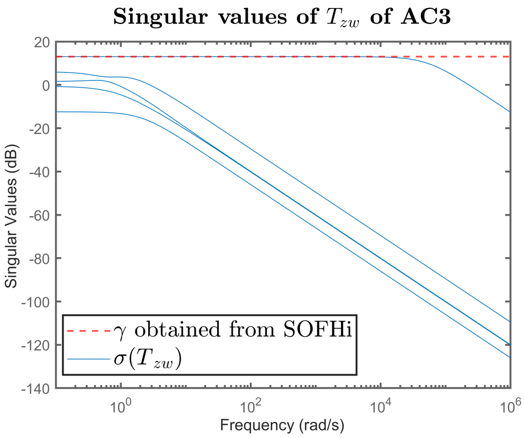
singular values over all frequencies $\max_\omega \bar{\sigma}(T_{zw})$. The results of this verification can be found in Table 3.2 and Figure 3.4. It can be seen that besides marginal numerical errors, the three results correspond to eachother for all three algorithms. It is thus verified that the synthesized gain matrices lead to closed-loop systems with correct H_∞ norms corresponding to the γ's obtained from the algorithms. To ensure that the peak gain is always correct, this process is checked for each γ that SOFHi outputs at each iteration. If it does not correspond, that run is considered failed.

Table 3.2: Verification of the obtained γ's of each algorithm in SOFHi for one random run on model *AC3* of the *Compl$_e$ib* library in (Friedemann Leibfritz 2004).

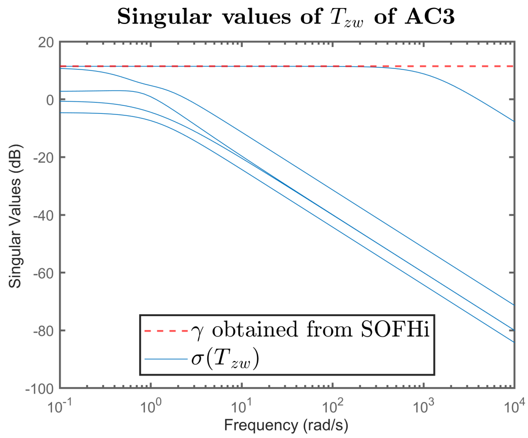
Algorithm	γ(SOFHi) [-]	`getPeakGain`(T_{zw}) [-]	$\max_\omega \bar{\sigma}(T_{zw})$ [-]
T-K	4.4855	4.4852	4.4852
S1	3.7277	3.7277	3.7277
S2	3.5095	3.5094	3.5095

3.3.2. Verification of the stability margins

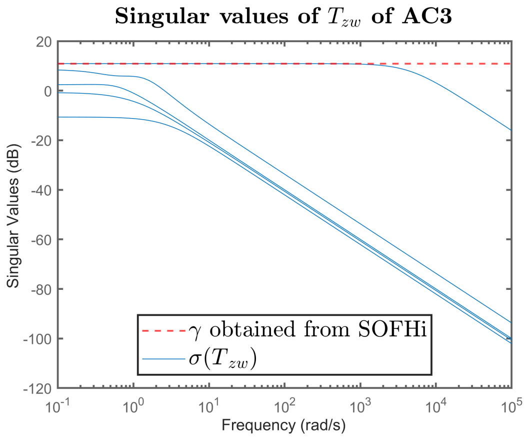
Verification of the loop transfer functions is a straightforward process; however, it should be done with care to ensure sufficient confidence in the resulting stability margins. The idea is to apply a multiplicative factor F to the system at the relevant signals. Either pure gain or pure phase is added to the system through F at the relevant signals (in this case at u_c, a_n, and q, to analyze both plant input as plant output). This process is implemented by building a simplified closed-loop system with the MATLAB function `connect` including the feedback gains and the model G. Multiplicative factors F are then added to the system at strategic locations.



(a) T-K iteration.



(b) S-variable Approach I.



(c) S-variable Approach II.

Figure 3.4: Comparison of γ with $\max_\omega \bar{\sigma}(T_{zw})$.

The locations of F can be seen in Figure 3.5. To verify the gain margin, F would be a static gain with the obtained gain margin as its value. To verify the phase margin, F would be a time-delay block with the calculated delay that corresponds to the phase margin value.[2] Consequently, analysis of the poles of the system should indicate that a pole is crossing the right-hand plane at the frequency of the gain- or phase margin, which would mean the system is on the verge of instability at this frequency and for this multiplicative factor F. That way, one verifies that the correct loop transfer functions are generated, since the consequent stability margins from `allmargin` and `diskmargin` are correct.

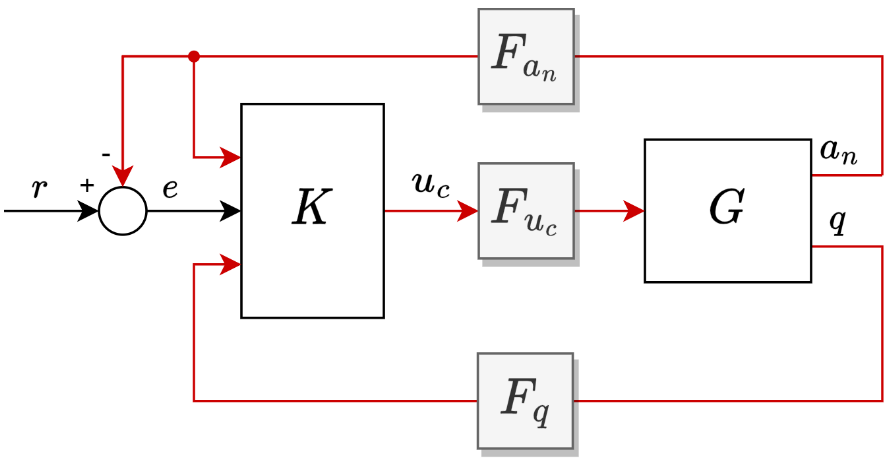


Figure 3.5: Locations of the multiplicative factors F_i, where i indicates the signal name.

[2]Delay T in seconds is calculated by $T = \frac{\pi}{180}\frac{\text{Phase [deg]}}{w\text{[rad/s]}}$, where w is the frequency at the phase margin.

3.4. Validation

In this section the validation of the designed gain-scheduled controller is described. Validation is achieved by ensuring that the controlled system behaves as desired throughout each point in the flight envelope when applied on the non-linear model. For this, the CAS is tested at points at the very edge of the flight envelope (at points with the highest velocities). Then, using a repetitive reference signal, the flight envelope is "discovered". This validation process serves as an addition to the non-linear model simulations described in chapter 2, which already validates that the flight controller behaves as desired on the 6-DoF model.

The results of this are presented in Figure 3.6 and Figure 3.7. It can be seen that the entirety of the flight envelope is covered with this method and no significant outliers are observed in the responses. The percentages in overshoot are small for each of the step responses with fast rise times. Significantly more overshoot in normal acceleration can be observed for the first step input, though, since these are the responses in the neighbourhood around the most difficult flight point, i.e. (Mach $= 0.4$, $h = 1000$m).

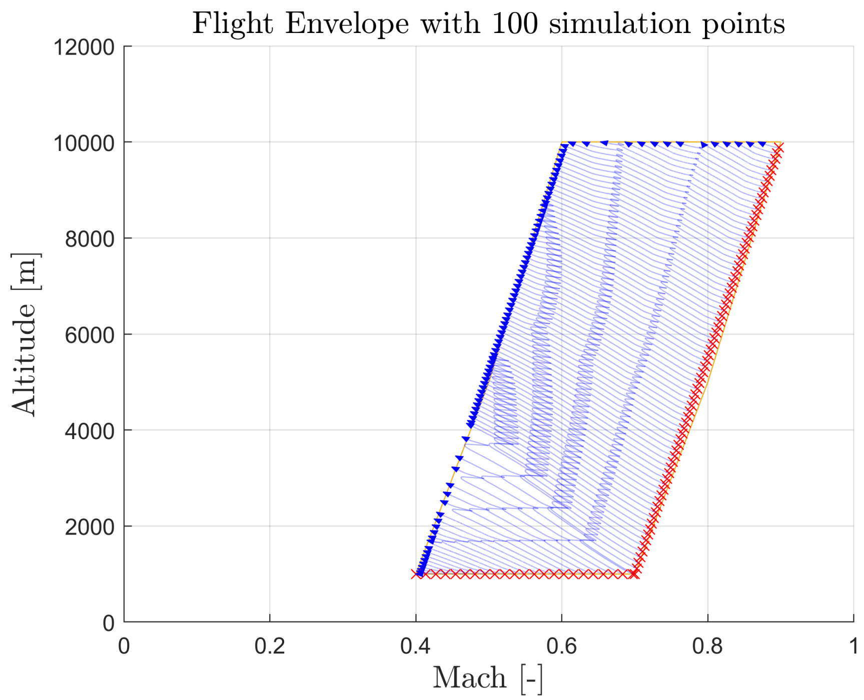
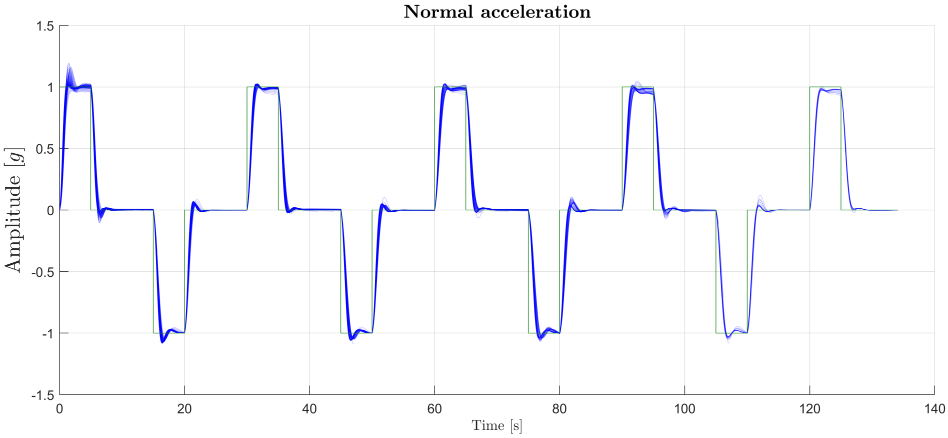
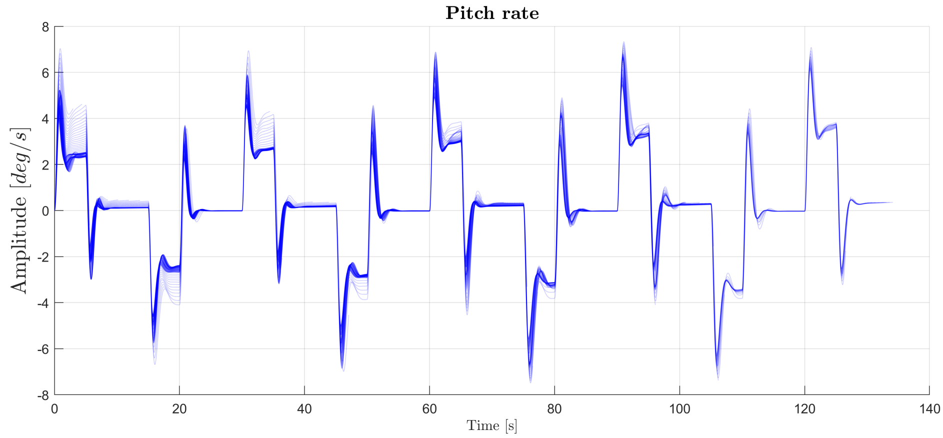


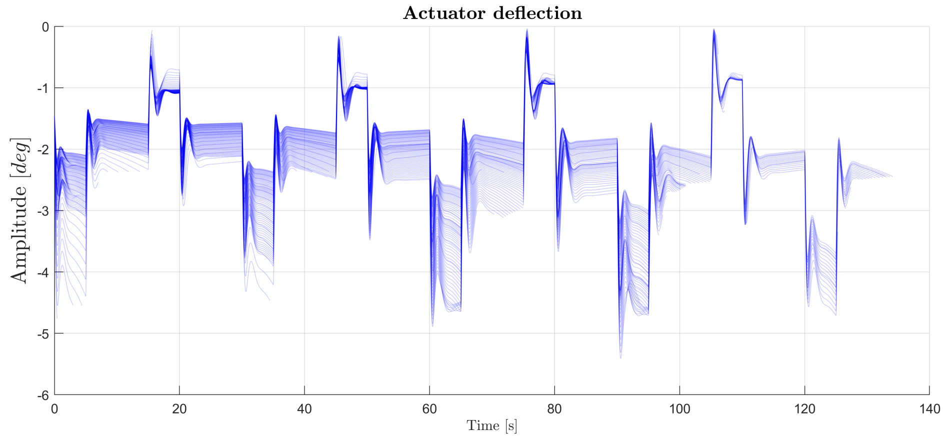
Figure 3.6: Mach-altitude trajectories for each of the 100 simulations throughout the flight envelope. Red crosses indicate the starting points of the simulations and the blue arrows indicate the direction of the trajectory. A stopping criterion is used for when the trajectory moves outside of the designed flight envelope.



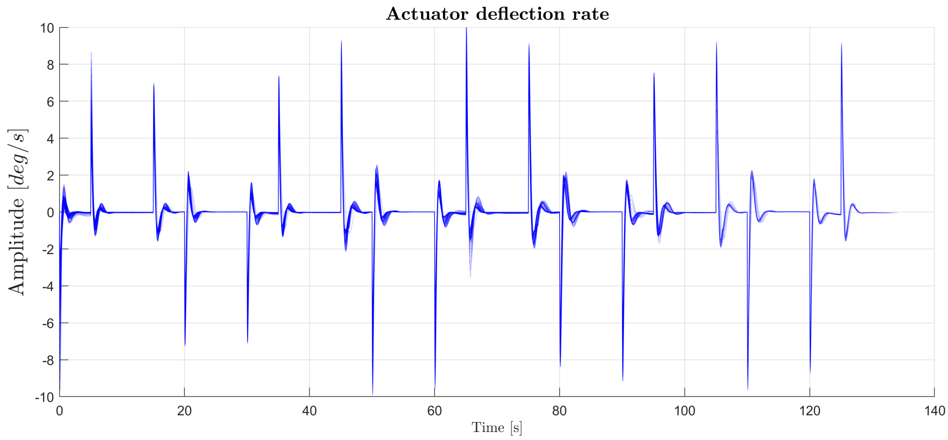
(a) Normal acceleration response.



(b) Pitch rate response.



(c) Actuator deflection.



(d) Actuator deflection rate.

Figure 3.7: Responses to the simulations of the validation procedure shown in Figure 3.6.

4
Conclusion & Outlook

4.1. Conclusion

The objective of this thesis was to obtain more insight into the performance of Lyapunov-based SOF controller synthesis methods in the H_∞ framework.

Research questions **Q1** and **Q2** were answered in the literature study, by comparing different mixed-sensitivity methods and SOF synthesis methods, and commenting on the strengths and limitations of each approach. The most promising SOF methods were implemented and bundled into a toolbox named SOFHi, and a variant of SOFHi, i.e. $\text{SOFHi}_{\text{EVO}}$, was proposed to greatly improve upon computational efficiency. This allowed one to answer **Q3** of the research questions, by comparing the performance of the methods to Hinfstruct and HIFOO on 54 benchmark models. Results showed that SOFHi was able to significantly outperform HIFOO in terms of the performance index γ and was at least able to be competitive with Hinfstruct. Furthermore, $\text{SOFHi}_{\text{EVO}}$ was observed to significantly outperform HIFOO and to even slightly outperform Hinfstruct for one of its runs.

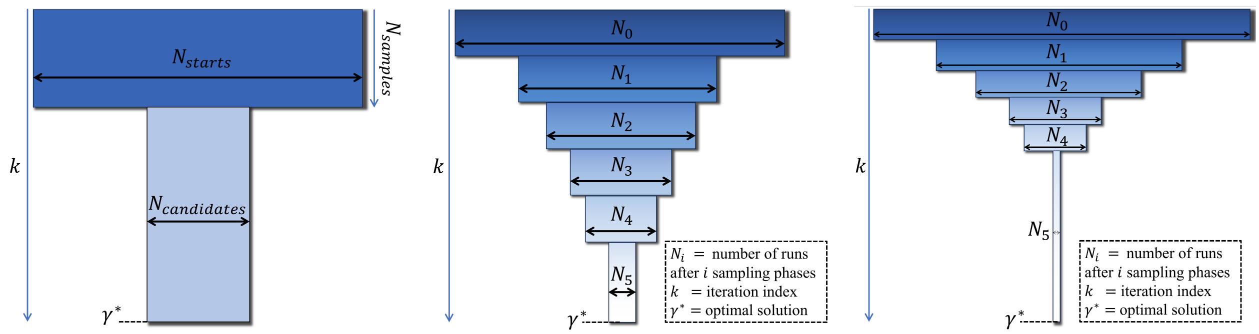
Lastly, to showcase the effectiveness of SOFHi and to provide an answer to research questions **Q4** and **Q5**, an elaborate flight example is performed in which SOFHi was able to robustly tune the controllers around 27 flight points to have good H_∞ performance and robustness, while meeting the design objectives and handling qualities requirements.

Finally, to answer the main research question **Q**, it was shown that the Lyapunov methods are at least competitive to the non-smooth optimization methods in terms of performance index γ, although their usage is limited to low-medium sized plants. In the end, SOFHi provides an alternative to the well-established non-smooth methods.

4.2. Outlook

Further research on SOFHi would include more testing of the existing methods. For e.g. $\text{SOFHi}_{\text{EVO}}$, more testing on the settings of the algorithm would potentially yield more insight into the optimal performance and competitiveness to Hinfstruct. For example, would choosing more candidates be more optimal, or would increasing the sample size be more preferable? Optimal settings might also be dependent on the sizes of the plants and the rate of convergence of the algorithms, so a more automated approach to determining the settings based on model characteristics such as plant size might lead to better results. These questions could be answered more accurately when more testing on $\text{SOFHi}_{\text{EVO}}$ is performed in addition to the runs that have been performed already.

Consider Figure 4.1. $\text{SOFHi}_{\text{EVO}}$ now performs a single sampling procedure to discard unfavourable runs once, as shown in Figure 4.1a. It might be beneficial to investigate the use of multiple or many sampling phases to have a "funnel-like" type of filtering where runs are discarded gradually, as shown in Figure 4.1b and Figure 4.1c. This might limit the risk of discarding the optimal run from the original set of runs, since more evaluations take place. In the end, many types of funnel shapes are possible, so time could be spent on finding the optimal settings of $\text{SOFHi}_{\text{EVO}}$ to manage the trade-off between computational efficiency and optimal γ.



(a) Current "T-like" procedure in $\text{SOFHi}_{\text{EVO}}$, where one sampling phase takes place.

(b) Proposed "funnel-like" procedure where multiple sampling phases take place.

(c) Second proposed "funnel-like" procedure where more runs are discarded at start.

Figure 4.1: Visualization of different filtering procedures.

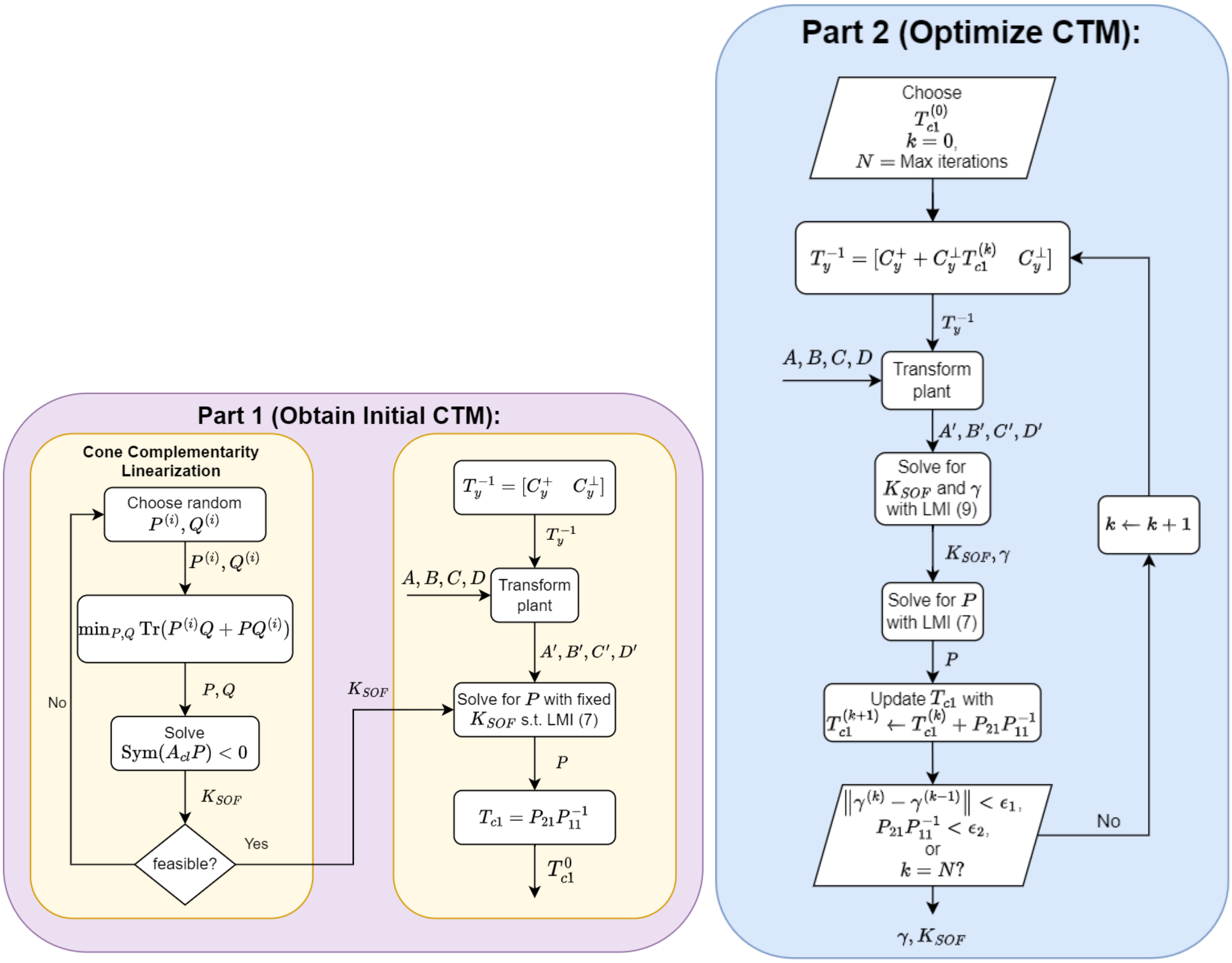
Furthermore, SOFHi could potentially be optimized for computational efficiency to make it even more competitive to the non-smooth optimization methods in that regard. Although the efficiency of Lyapunov methods is limited by the amount of decision variables, as explained before, care could be taken to optimize the code with the aim of improving the performance of the algorithms. This could, for example, be done by a group of experts in the field of computer science, if needed.

Although three algorithms were deemed most promising to implement into SOFHi, more algorithms could be added in a later stage. For example, T-K iteration as described in (Feng, She, and Xu 2019) assumes that D_{yw} is null. (Feng, Guo, et al. 2022) does not have this assumption, but in turn assumes that D_{zu} is null. T-K iteration could therefore be extended to include (Feng, Guo, et al. 2022) in cases where $D_{yw} \neq 0$ and $D_{zu} = 0$. This was not done in this implementation yet due to practical considerations and the fact that D_{zu} is more often non-zero than D_{yw} is, e.g. in an S/KS configuration.

Additionally, for the aircraft model, μ-analysis would achieve more confidence in the robustness of the controller. This was considered beyond the scope of this current research though, where robustness was demonstrated through a Monte-Carlo approach. Further improvements could also be made in the aircraft design part by tuning more flight points than the current 27 flight points to obtain more accurate gain surfaces, which would ultimately lead to better performance between the flight points.

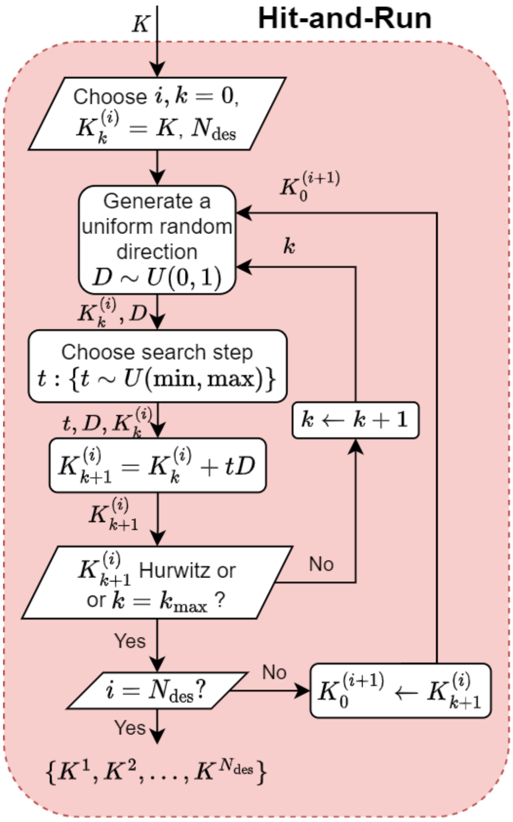
A

Flowcharts

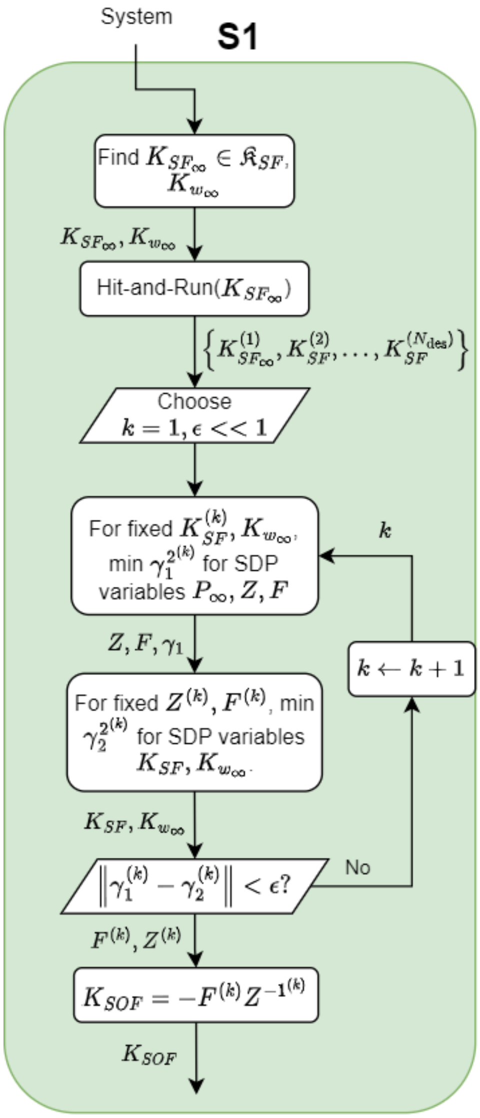


(a) Part 1. (b) Part 2.

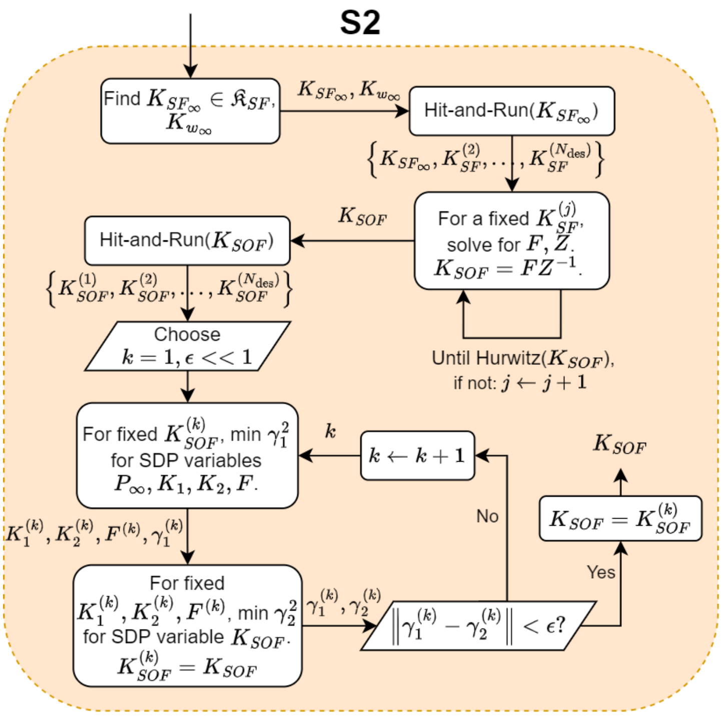
Figure A.1: Part 1 and 2 of T-K iteration. LMI (7) and LMI (9) refer to equations 7 and 9 in (Feng, She, and Xu 2019).



(a) Hit-and-Run algorithm procedure.



(b) S-variable Approach I.



(c) S-variable Approach II.

Figure A.2: S-variable methods procedure.

B

SOFHi User Guide

```
%% Made by:
    % A.D.P. Schoon
    % M.Sc. student at Delft University of Technology, Control & Simulation
    % department in the Aerospace Engineering faculty.
    % As part of my thesis "Advancements in H-infinity Static Output Feedback Synthesis for
        Fighter Aircraft Control".
%% SOFHi serves as an implementation of the following algorithms:
% ---------------------------------------------------------------------------------
% T-K iteration:
% [1].  Feng, Zhi Yong, Jinhua She, and Li Xu (Sept. 2019). "A brief review and insights into
    matrix inequalities for H-infinity static-output-feedback control and a local optimal
    "solution. In: International Journal of Systems Science 50 (12), pp. -22922305. issn:
    14645319. doi: 10.1080/00207721.2019.1654008.

% S-variable methods:
% [2].  Ebihara, Yoshio, Dimitri Peaucelle, and Denis Arzelier (2015). S-variable approach to LMI-
    based robust control. Springer
% ---------------------------------------------------------------------------------
%% Note!
% Make sure to have Yalmip and the desired solvers installed properly and
% added to the path before running.

%% Inputs:
% P
% class:        struct
% P defines the plant and must be a struct class with the following fields
% containing your system matrices:
% A, Bu, Bw, Cz, Cy, Dzw, Dzu, Dyw, Dyu

% options.choice
% class:        double
% default:      123 [all algorithms]
% Defines which algorithms to run.
% 1 defines T-K iteration, 2 the S-variable Approach I, 3 defines
% S-variable Approach II. E.g., 31 will run S-variable Approach II and T-K
% iteration.

% options.parallel
% class:        binary
% default:      1
% Defines whether to use parallel computing (highly advised).
% !! This requires the Parallel Computing Toolbox. !!

% options.starts
% class:        double
% default:      10
% Defines the number of tries for the algorithm. Since the
% algorithm produces local solutions and these are dependent on the initial
% conditions, different results occur. Therefore, to increase the
% chance of a local solution being a local optimum, it is advised to have
% multiple random starts at the cost of computational time.

% options.evo
% class:        binary
```

```
% default:      0
% Selects whether to use the "fast" option. Algorithm pre-selects from a
% pool of runs (#starts) the most promosing runs: the candidates.
% The selection takes place after a certain amount of iterations: the
% number of samples. These candidates are then continued to run.

% options.iterSample
% class:        double
% default:      10
% Related to options.evo. Defines the number of iterations after which a
% selection takes place to define the most promising runs

% options.struc
% class:        double
% default:      ones(#inputs,#outputs)
% Allows to define zero elements of Ksof. The ones indicate non-zero
% elements, the zeros in options.struc define the zeros in Ksof

% options.itermaxTK | options.itermaxS1 | options.itermaxS2
% class:        double
% default:      3000
% Defines the maximum amount of iterations for each algorithm

% options.display defines if the level of user display
% class:        string
% default:      'iter'
%   1. 'final'  will display only the final gamma
%   2. 'iter'   will display the gamma after every start
%   3. 'off'    will display nothing

% options.candidates
% class:        double
% default:      3
% Defines the number of candidates which are further optimized, after the
% sampling phase.

% options.tolerance
% class:        double
% default:      1e-6
% Defines tolerance level for convergence of gamma. Does not define the
% tolerance of the actual LMI solvers, which are at default values.

%% Outputs:
% K is the optimal SOF gain matrix.

% gamma is the best (sub)-optimal ∞H-norm found by the algorithm(s).

% T is the time in seconds it has taken to obtain the best gamma by the
% corresponding algorithm.
```

References

Adams, Richard J. et al. (1994). *Robust Multivariable Flight Control.*

Ali, HI et al. (2011). "Design of H-inf Controller with Tuning of Weights Using Particle Swarm Optimization Method." In: *IAENG International Journal of Computer Science* 38.2.

Alvarez, Daniel J and Bei Lu (2011). "Piloted simulation study comparing classical, H-infinity, and linear parameter-varying control methods". In: *Journal of guidance, control, and dynamics* 34.1, pp. 164–176.

Anderson, Mark R., Abbas Emami-Naeini, and James H. Vincent (1991). "Robust Control Law Development for a Hypersonic Cruise Aircraft". In: *1991 American Control Conference*, pp. 839–845. DOI: 10.23919/ACC.1991.4791490.

Ankelhed, Daniel, Anders Helmersson, and Anders Hansson (2012). "A Partially Augmented Lagrangian Method for Low Order H-Infinity Controller Synthesis Using Rational Constraints". In: *IEEE transactions on automatic control* 57.11, pp. 2901–2905.

Apkarian, Pierre, Minh Ngoc Dao, and Dominikus Noll (2015). "Parametric Robust Structured Control Design". In: *IEEE Transactions on Automatic Control* 60.7, pp. 1857–1869. DOI: 10.1109/TAC.2015.2396644.

Apkarian, Pierre and Dominikus Noll (2006a). "Controller Design via Nonsmooth Multidirectional Search". In: *SIAM Journal on Control and Optimization* 44.6, pp. 1923–1949. DOI: 10.1137/S0363012904441684.

— (2006b). "Nonsmooth Hinf synthesis". In: *IEEE Transactions on Automatic Control* 51.1, pp. 71–86.

Apkarian, Pierre, Dominikus Noll, and Hoang Duong Tuan (2003). "Fixed-order Hinf control design via a partially augmented Lagrangian method". In: *IFAC Proceedings Volumes* 36.8, pp. 69–74.

Arfiadi, Y and MNS Hadi (2001). "Optimal direct (static) output feedback controller using real coded genetic algorithms". In: *Computers & Structures* 79.17, pp. 1625–1634.

Arzelier, D, D Peaucelle, and S Salhi (2003). "Robust static output feedback stabilization for polytopic uncertain systems: improving the guaranteed performance bound". In: *IFAC Proceedings Volumes* 36.11, pp. 425–430.

Arzelier, D. et al. (2018). "Robust static output feedback design with deterministic and probabilistic certificates". In: *Uncertainty in Complex Networked Systems: In Honor of Roberto Tempo*, pp. 121–148.

Arzelier, Denis et al. (2010). "Mixed LMI/randomized methods for static output feedback control design". In: *Proceedings of the 2010 American Control Conference, ACC 2010*, pp. 4683–4688. DOI: 10.1109/ACC.2010.5531075.

Asali, Modestus Oliver, Toto Indriyanto, Bambang Riyanto Trilaksono, et al. (2019). "Visual Servoing using Mixed Sensitivity H_∞ Control for Yaw-Pitch Camera Platform". In: *2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT)*. IEEE, pp. 48–53.

Audet, Charles and John E Dennis Jr (2006). "Mesh adaptive direct search algorithms for constrained optimization". In: *SIAM Journal on optimization* 17.1, pp. 188–217.

Bates, Declan and Ian Postlethwaite (2002a). *Robust Multivariable Control of Aerospace Systems.* Control and Simulation. Delft, NL: Delft University Press.

— (2002b). *Robust multivariable control of aerospace systems.* Vol. 8. IOS Press.

Beran, Eric, Lieven Vandenberghe, and Stephen Boyd (1997). "A global BMI algorithm based on the generalized Benders decomposition". In: *1997 European Control Conference (ECC)*. IEEE, pp. 3741–3746.

Box, George E. P. (1957). "Evolutionary Operation: a Method for Increasing Industrial Productivity". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 6.2, pp. 81–101. DOI: https://doi.org/10.2307/2985505.

Burke, James V et al. (2006). "HIFOO-a MATLAB package for fixed-order controller design and Hinf optimization". In: *IFAC Proceedings Volumes* 39.9, pp. 339–344.

Cao, Yong-Yan and You-Xian Sun (1998). "Static output feedback simultaneous stabilization: ILMI approach". In: *International journal of control* 70.5, pp. 803–814.

Cao, Yong-Yan, You-Xian Sun, and Wei-Jie Mao (1998). "Output feedback decentralized stabilization: ILMI approach". In: *Systems & Control Letters* 35.3, pp. 183–194.

Chanekar, Prasad Vilas, Nikhil Chopra, and Shapour Azarm (2017). "Optimal structured static output feedback design using generalized benders decomposition". In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 4819–4824. DOI: 10.1109/CDC.2017.8264371.

Cheng, Zilong et al. (2022). "Second-Order NonConvex Optimization for Constrained Fixed-Structure Static Output Feedback Controller Synthesis". In: *IEEE Transactions on Automatic Control* 67.9, pp. 4854–4861. DOI: 10.1109/TAC.2022.3164839.

Chesi, Graziano (2013). "Robust static output feedback controllers via robust stabilizability functions". In: *IEEE Transactions on Automatic Control* 59.6, pp. 1618–1623.

Chiang, R.Y. et al. (1993). "A fixed H-infinity controller for a supermaneuverable fighter performing the herbst maneuver". In: *Automatica* 29.1, pp. 111–127. ISSN: 0005-1098. DOI: https://doi.org/10.1016/0005-1098(93)90176-T.

Chou, Yung Shan and Sheng Fong Wei (Apr. 2020). "On optimal static output feedback control synthesis". In: *IEEE Control Systems Letters* 4 (2), pp. 468–473. ISSN: 24751456. DOI: 10.1109/LCSYS.2019.2951286.

Clarke, Frank H (1983). *Optimization and nonsmooth analysis*. SIAM.

Crusius, C.A.R. and A. Trofino (1999). "Sufficient LMI conditions for output feedback control problems". In: *IEEE Transactions on Automatic Control* 44.5, pp. 1053–1057. DOI: 10.1109/9.763227.

Debrouwere, Frederik et al. (2013). "Time-Optimal Path Following for Robots With Convex–Concave Constraints Using Sequential Convex Programming". In: *IEEE Transactions on Robotics* 29.6, pp. 1485–1495. DOI: 10.1109/TRO.2013.2277565.

Dingchao Ren Junlin Xiong, Daniel W. C. Ho (2022). "Static output feedback negative imaginary controller design with H2 performance". In: *International Journal of Control*. DOI: 10.1080/00207179.2022.2118176.

Doyle, John, Kemin Zhou, and Bobby Bodenheimer (1989). "Optimal control with mixed H2 and H8 performance objectives". In: *1989 American Control Conference*, pp. 2065–2070. DOI: 10.23919/ACC.1989.4790529.

Ebihara, Yoshio, Dimitri Peaucelle, and Denis Arzelier (2014). "On the structure of generalized plant convexifying static Hinf control problems". In: *Automatica* 50.6, pp. 1706–1714. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2014.04.027. URL: https://www.sciencedirect.com/science/article/pii/S000510981400171X.

— (2015). *S-variable approach to LMI-based robust control*. Springer.

Ebihara, Yoshio, Kazuma Tokuyama, and Tomomichi Hagiwara (2004). "Structured controller synthesis using LMI and alternating projection method". In: *International Journal of Control* 77.12, pp. 1137–1147. DOI: 10.1080/002071704123312989 47.

El Ghaoui, L. and V. Balakrishnan (1994). "Synthesis of fixed-structure controllers via numerical optimization". In: 3, 2678–2683 vol.3. DOI: 10.1109/CDC.1994.411398.

Felipe, A. and R. Oliveira (2021). "An LMI-Based Algorithm to Compute Robust Stabilizing Feedback Gains Directly as Optimization Variables". In: *IEEE Transactions on Automatic Control* 66.9, pp. 4365–4370. DOI: 10.1109/TAC.2020.3038359.

Feng, Zhi Yong, Huiru Guo, et al. (Feb. 2022). "Weighted sensitivity design of multivariable PID controllers via a new iterative LMI approach". In: *Journal of Process Control* 110, pp. 24–34. ISSN: 0959-1524. DOI: 10.1016/J.JPROCONT.2021.11.016.

Feng, Zhi Yong, Jinhua She, and Li Xu (Sept. 2019). "A brief review and insights into matrix inequalities for H_∞ static-output-feedback control and a local optimal solution". In: *International Journal of Systems Science* 50 (12), pp. 2292–2305. ISSN: 14645319. DOI: 10.1080/00207721.2019.1654008.

Francis, Bruce A (1987). "A course in H_∞ control theory". In: *Lecture notes in control and information sciences* 88, R5.

Gadewadikar, J and FL Lewis (2006). "Aircraft flight controller tracking design using H-infinity static output-feedback". In: *Transactions of the Institute of Measurement and Control* 28.5, pp. 429–440.

Gadewadikar, J. et al. (2006). "Parameterization of all stabilizing H∞ static state-feedback gains: Application to output-feedback design". In: *Proceedings of the IEEE Conference on Decision and Control*, pp. 3566–3571. ISSN: 25762370. DOI: 10.1109/CDC.2006.377280.

Gadewadikar, Jyotirmay, Ajay Bhilegaonkar, and F. L. Lewis (Oct. 2007). "Bounded L2 gain static output feedback: Controller design and implementation on an electromechanical system". In: *IEEE Transactions on Industrial Electronics* 54 (5), pp. 2593–2599. ISSN: 02780046. DOI: 10.1109/TIE.2007.899883.

Gadewadikar, Jyotirmay, Frank L. Lewis, and Murad Abu-Khalaf (May 2012). "Necessary and Sufficient Conditions for H-Infinity Static Output-Feedback Control". In: *https://doi.org/10.2514/1.16794* 29 (4), pp. 915–920. ISSN: 15333884. DOI: 10.2514/1.16794. URL: https://arc.aiaa.org/doi/10.2514/1.16794.

Gadewadikar, Jyotirmay, Frank L. Lewis, Kamesh Subbarao, et al. (Apr. 2009). "H-infinity static output-feedback control for rotorcraft". In: *Journal of Intelligent and Robotic Systems: Theory and Applications* 54 (4), pp. 629–646. ISSN: 09210296. DOI: 10.1007/S10846-008-9279-5.

Gahinet, Pascal and Pierre Apkarian (1994). "A linear matrix inequality approach to Hinf control". In: *International journal of robust and nonlinear control* 4.4, pp. 421–448.

Garg, Sanjay (1993). “Robust integrated flight/propulsion control design for a STOVL aircraft using H-infinity control design techniques”. In: *Automatica* 29.1, pp. 129–145. ISSN: 0005-1098. DOI: https://doi.org/10.1016/0005-1098(93)90177-U.

Garg, Sanjay and Peter J. Ouzts (1991). “Integrated Flight/Propulsion Control Design for a STOVL Aircraft using H-Infinity Control Design Techniques”. In: pp. 568–576. DOI: 10.23919/ACC.1991.4791435.

Geromel, Jose, Mauricio C de Oliveira, and Liu Hsu (1998). “LMI characterization of structural and robust stability”. In: *Linear Algebra and its Applications* 285.1-3, pp. 69–80.

Geromel, José Claudio, Cid C de Souza, and Robert E Skelton (1998). “Static output feedback controllers: Stability and convexity”. In: *IEEE Transactions on Automatic Control* 43.1, pp. 120–125.

Ghaoui, Laurent El, Francois Oustry, and Mustapha Aitrami (1997). “A Cone Complementarity Linearization Algorithm for Static Output-Feedback and Related Problems”. In: *IEEE TRANSACTIONS ON AUTOMATIC CONTROL* 42 (8).

Glover, Keith and John C Doyle (1988). “State-space formulae for all stabilizing controllers that satisfy an Hinf-norm bound and relations to relations to risk sensitivity”. In: *Systems & control letters* 11.3, pp. 167–172.

Gopmandal, Falguni and Arun Ghosh (Aug. 2021). “A hybrid search based synthesis of static output feedback controllers for uncertain systems with application to multivariable PID control”. In: *International Journal of Robust and Nonlinear Control* 31 (12), pp. 6069–6090. ISSN: 1099-1239. DOI: 10.1002/RNC.5581.

Grigoriadis, Karolos M and Eric B Beran (2000). “Alternating projection algorithms for linear matrix inequalities problems with rank constraints”. In: *Advances in linear matrix inequality methods in control*. SIAM, pp. 251–267.

Grigoriadis, Karolos M and Robert E Skelton (1996). “Low-order control design for LMI problems using alternating projection methods”. In: *Automatica* 32.8, pp. 1117–1125.

Harikumar, Kandath, Sidhant Dhall, and Seetharama Bhat (Jan. 2019). “Design and Experimental Validation of a Robust Output Feedback Control for the Coupled Dynamics of a Micro Air Vehicle”. In: *International Journal of Control, Automation and Systems* 17 (1), pp. 155–167. ISSN: 20054092. DOI: 10.1007/S12555-017-0799-2/METRICS. URL: https://link.springer.com/article/10.1007/s12555-017-0799-2.

Hassibi, Arash, Jonathan P. How, and Stephen P. Boyd (1999). “A path-following method for solving BMI problems in control”. In: *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)* 2, 1385–1389 vol.2.

Helmersson, Anders (2009). *On polynomial coefficients and rank constraints*. Linköping University Electronic Press.

Henrion, Didier (2006). “Solving static output feedback problems by direct search optimization”. In: *2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*, pp. 1534–1537. DOI: 10.1109/CACSD-CCA-ISIC.2006.4776869.

Ho, D.W.C., J. Lam, and T.W.K. Chan (Sept. 1994). “Mixed H_∞ optimisation with model-matching”. English. In: *IEE Proceedings - Control Theory and Applications* 141 (5), 329–340(11). ISSN: 1350-2379.

Ho, Daniel WC, James Lam, and TWK Chan (1992). “An application of Hinf design to model-following”. In: *International Journal of Control* 55.2, pp. 483–509.

Holicki, Tobias and Carsten W Scherer (2021). “Revisiting and generalizing the dual iteration for static and robust output-feedback synthesis”. In: *International Journal of Robust and Nonlinear Control* 31.11, pp. 5427–5459.

Hooke, Robert and Terry A Jeeves (1961). “Direct Search Solution of Numerical and Statistical Problems”. In: *Journal of the ACM (JACM)* 8.2, pp. 212–229.

Hsiao, Ying-Tung, Cheng-Long Chuang, and Cheng-Chih Chien (2004). “Ant colony optimization for designing of PID controllers”. In: *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No. 04CH37508)*. IEEE, pp. 321–326.

Huang, Yu et al. (2010). “Control of a two-load-path trimmable horizontal stabilizer actuator of an aircraft - Comparison of Hinf design approaches”. In: *Proceedings of the IEEE Conference on Decision and Control*, pp. 4863–4868. ISSN: 25762370. DOI: 10.1109/CDC.2010.5716948.

Ikeda, Y., P.H. McDowell, and R.L. Hargis (1990). “Robust controller design for an advanced fighter aircraft”. In: 2600–2606 vol.5. DOI: 10.1109/CDC.1990.203342.

Iwasaki, T and RE Skelton (1995). “The XY-centring algorithm for the dual LMI problem: a new approach to fixed-order control design”. In: *International Journal of Control* 62.6, pp. 1257–1272.

Iwasaki, Tetsuya (1999). “The dual iteration for fixed-order control”. In: *IEEE transactions on automatic control* 44.4, pp. 783–788.

Iwasaki, Tetsuya and Shinji Hara (2005). “Generalized KYP lemma: Unified frequency domain inequalities with design applications”. In: *IEEE Transactions on Automatic Control* 50.1, pp. 41–59.

Ji, Shiyong and Aiguo Wu (2011). "Study on dual-loop controller of helicopter based on the robust H-infinite loop shaping and mixed sensitivity". In: pp. 1291–1294. DOI: 10.1109/ICECENG.2011.6057783.

Keviczky, Tamás and Gary J Balas (2006). "Receding horizon control of an F-16 aircraft: A comparative study". In: *Control Engineering Practice* 14.9, pp. 1023–1033.

Kim, Jong Han and Ick Ho Whang (Sept. 2018). "Augmented three-loop autopilot structure based on mixed-sensitivity H-oo optimization". In: *Journal of Guidance, Control, and Dynamics* 41 (3), pp. 748–753. ISSN: 15333884. DOI: 10.2514/1.G003119/ASSET/IMAGES/LARGE/FIGURE11.JPEG. URL: https://arc.aiaa.org/doi/10.2514/1.G003119.

Kim, Seog-Joo, Young-Hyun Moon, and Soonman Kwon (2007). "Solving rank-constrained LMI problems with application to reduced-order output feedback stabilization". In: *IEEE Transactions on Automatic Control* 52.9, pp. 1737–1741.

Kumar, Parvesh and Shiv Narayan (2016). "Optimal design of robust FOPID for the aircraft pitch control system using multi-objective GA". In: pp. 1–6. DOI: 10.1109/SCEECS.2016.7509267.

Kwakernaak, Huibert (2002a). *Mixed sensitivity design*. This reference is perfect for general rules on mixed-sensitivity design.

— (2002b). *Mixed sensitivity design: An aerospace study*. URL: www.elsevier.com/locate/ifac.

Leibfritz, F and Siam J Control Optim (July 2001). "An LMI-Based Algorithm for Designing Suboptimal Static $\mathcal{H}_{\in}/\mathcal{H}_{\infty}$ Output Feedback Controllers". In: *https://doi-org.tudelft.idm.oclc.org/10.1137/S0363012999349553* 39 (6), pp. 1711–1735. ISSN: 03630129. DOI: 10.1137/S0363012999349553. URL: https://epubs-siam-org.tudelft.idm.oclc.org/doi/10.1137/S0363012999349553.

Leibfritz, Friedemann (2004). "COMPLeIB, COnstraint Matrix-optimization Problem LIbrary-a collection of test examples for nonlinear semidefinite programs, control system design and related problems". In: *Dept. Math., Univ. Trier, Trier, Germany, Tech. Rep* 2004.

Li, Pengxu et al. (2021). "Robust gain-scheduling static output-feedback Hinf control of vehicle lateral stability with heuristic approach". In: *Information Sciences* 546, pp. 220–233. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2020.08.023. URL: https://www.sciencedirect.com/science/article/pii/S0020025520307817.

Li, Xianwei and Huijun Gao (2014). "A Heuristic Approach to Static Output-Feedback Controller Synthesis With Restricted Frequency-Domain Specifications". In: *IEEE Transactions on Automatic Control* 59.4, pp. 1008–1014. DOI: 10.1109/TAC.2013.2281472.

Lu, Bei and Fen Wu (2005). "Probabilistic robust control design for an f-16 aircraft". In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*, p. 6080.

Lu, Bei, Fen Wu, and SungWan Kim (2006). "Switching LPV control of an F-16 aircraft via controller state reset". In: *IEEE Transactions on Control Systems Technology* 14.2, pp. 267–277. DOI: 10.1109/TCST.2005.863656.

Ma, Sike et al. (2021). "Robust mixed-sensitivity Hinf control of weight on bit in geological drilling process with parameter uncertainty". In: *Journal of the Franklin Institute* 358.13, pp. 6433–6461. ISSN: 0016-0032. DOI: https://doi.org/10.1016/j.jfranklin.2021.06.005. URL: https://www.sciencedirect.com/science/article/pii/S0016003221003380.

Marrison, Christopher I and Robert F Stengel (1997). "Robust control system design using random search and genetic algorithms". In: *IEEE Transactions on Automatic Control* 42.6, pp. 835–839.

McFarlane, D. and K. Glover (1992). "A loop-shaping design procedure using H/sub infinity / synthesis". In: *IEEE Transactions on Automatic Control* 37.6, pp. 759–769. DOI: 10.1109/9.256330.

McFarlane, Duncan C. and Keith Glover (1990). *Robust Controller Design Using Normalized Coprime Factor Plant Descriptions*. Vol. 138. Lecture Notes in Control and Information Sciences.

Mehdi, D., E. K. Boukas, and O. Bachelier (2004). "Static output feedback design for uncertain linear discrete time systems". In: *IMA Journal of Mathematical Control and Information* 21.1, pp. 1–13. DOI: 10.1093/imamci/21.1.1.

"MIL-F-8785C" (1980). In.

Nag, Anirban et al. (2013). "A robust H-infinity based depth control of an autonomous underwater vehicle". In: *2013 International Conference on Advanced Electronic Systems (ICAES)*, pp. 68–73. DOI: 10.1109/ICAES.2013.6659363.

Nelder, J.A. and R. Mead (1965). "Downhill simplex method". In: *Computer journal* 7, pp. 308–313.

Nguyen, Vi H and Thanh T Tran (2020). "A novel hybrid robust control design method for f-16 aircraft longitudinal dynamics". In: *Mathematical Problems in Engineering* 2020, pp. 1–10.

Noll, Dominikus, Mounir Torki, and Pierre Apkarian (2004). "Partially augmented Lagrangian method for matrix inequality constraints". In: *SIAM Journal on Optimization* 15.1, pp. 161–184.

Peaucelle, D. and D. Arzelier (2001). "An efficient numerical solution for H2 static output feedback synthesis". In: pp. 3800–3805. DOI: 10.23919/ECC.2001.7076526.

Peaucelle, D. and D. Arzelier (2005). “Ellipsoidal sets for resilient and robust static output-feedback”. In: *IEEE Transactions on Automatic Control* 50.6, pp. 899–904. DOI: 10.1109/TAC.2005.849256.

Peaucelle, D., D. Arzelier, and C. Farges (2004). “Lmi results for resilient state-feedback with h/sub/spl infin//performance”. In: *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*. Vol. 1. IEEE, pp. 400–405.

Peretz, Y, O Merzbach, and S Moyall (2020). “On a new generalised LMI condition and randomised algorithm for robust stabilisation via static-output-feedback”. In: *International Journal of Systems Science* 51.1, pp. 61–71.

Peretz, Y. (2017). “On applications of the ray-shooting method for structured and structured-sparse static-output-feedbacks”. In: *International Journal of Systems Science* 48.9, pp. 1902–1913. DOI: 10.1080/00207721.2017.1290300.

Peretz, Yossi (2016). “A randomized approximation algorithm for the minimal-norm static-output-feedback problem”. In: *Automatica* 63, pp. 221–234. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2015.10.001.

— (2018). “A Randomized Algorithm for Optimal PID Controllers”. In: *Algorithms* 11.6. ISSN: 1999-4893. DOI: 10.3390/a11060081. URL: https://www.mdpi.com/1999-4893/11/6/81.

Pérez, Carlos Alcolea et al. (Jan. 2022). “Automatic weighting filter tuning for robust flight control law design”. In: *IFAC-PapersOnLine* 55 (16), pp. 400–405. ISSN: 2405-8963. DOI: 10.1016/J.IFACOL.2022.09.057.

Pirat, Camille et al. (May 2020). “Hinf and μ-Synthesis for nanosatellites rendezvous and docking”. In: *IEEE Transactions on Control Systems Technology* 28 (3), pp. 1050–1057. ISSN: 15580865. DOI: 10.1109/TCST.2019.2892923.

Postlethwaite, I., Mi-Ching Tsai, and Da-Wei Gu (1990). “Weighting Function Selection in H-infinity Design”. In: *IFAC Proceedings Volumes* 23.8, Part 3. 11th IFAC World Congress on Automatic Control, Tallinn, 1990 - Volume 3, Tallinn, Finland, pp. 127–132. ISSN: 1474-6670. DOI: https://doi.org/10.1016/S1474-6670(17)51904-7.

Prempain, Emmanuel and Ian Postlethwaite (2001). “Static output feedback stabilisation with Hinf performance for a class of plants”. In: *Systems & Control Letters* 43.3, pp. 159–166.

Reigelsperger, William C, Kelly D Hammett, and Siva S Banda (1997). “Robust control law design for lateral-directional modes of an F-16/MATV using μ-synthesis and dynamic inversion”. In: *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal* 7.8, pp. 777–795.

Ren, Dingchao, Junlin Xiong, and Daniel W.C. Ho (Apr. 2021). “Static output feedback negative imaginary controller synthesis with an H∞ norm bound”. In: *Automatica* 126. ISSN: 00051098. DOI: 10.1016/J.AUTOMATICA.2020.109157.

Ren, Yingying et al. (2021). “A successive convex optimization method for bilinear matrix inequality problems and its application to static output-feedback control”. In: *International Journal of Robust and Nonlinear Control* 31.18, pp. 9709–9730. DOI: https://doi.org/10.1002/rnc.5796.

Rubió-Massegú, J. et al. (Jan. 2013). “Static output-feedback control under information structure constraints”. In: *Automatica* 49 (1), pp. 313–316. ISSN: 0005-1098. DOI: 10.1016/J.AUTOMATICA.2012.10.012.

Sadabadi, Mahdieh S. and Dimitri Peaucelle (Jan. 2016). “From static output feedback to structured robust static output feedback: A survey”. In: *Annual Reviews in Control* 42, pp. 11–26. ISSN: 1367-5788. DOI: 10.1016/J.ARCONTROL.2016.09.014.

Saeki, Masami and Correspondence Masami Saeki (Mar. 2018). “L1 synthesis of a static output controller for positive systems by LMI iteration”. In: *International Journal of Robust and Nonlinear Control* 28 (4), pp. 1319–1333. ISSN: 1099-1239. DOI: 10.1002/RNC.3956. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/rnc.3956%20https://onlinelibrary.wiley.com/doi/abs/10.1002/rnc.3956%20https://onlinelibrary.wiley.com/doi/10.1002/rnc.3956.

Safonov, MG, RY Chiang, and H Flashner (1988). “H-infinity robust control synthesis for a large space structure”. In: *1988 American Control Conference*. IEEE, pp. 2038–2045.

Sahoo, Pradosh Ranjan et al. (2019). “New results on restricted static output feedback H-8 controller design with regional pole placement”. In: ISSN: 1751-8644. DOI: 10.1049/iet-cta.2018.6138. URL: www.ietdl.org.

Scherer, C., P. Gahinet, and M. Chilali (1997). “Multiobjective output-feedback control via LMI optimization”. In: *IEEE Transactions on Automatic Control* 42.7, pp. 896–911. DOI: 10.1109/9.599969.

Sefton, J. and K. Glover (1990). “Pole/zero cancellations in the general Hinf problem with reference to a two block design”. In: *Systems & Control Letters* 14.4, pp. 295–306. ISSN: 0167-6911. DOI: https://doi.org/10.1016/0167-6911(90)90050-5. URL: https://www.sciencedirect.com/science/article/pii/0167691190900505.

Sekaj, Ivan and Vojtech Veselý (2002). “ROBUST OUTPUT FEEDBACK CONTROLLER DESIGN VIA GENETIC ALGORITHM”. In: *IFAC Proceedings Volumes* 35.1. 15th IFAC World Congress, pp. 413–

418. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20020721-6-ES-1901.00402. URL: https://www.sciencedirect.com/science/article/pii/S1474667015388236.

Sereni, B et al. (2018). "Relaxed LMI conditions for the design of robust static output feedback controllers". In: *IFAC-PapersOnLine* 51.25, pp. 428–433.

Sereni, B. (Jan. 2023). "Two-stage static output feedback controller design for linear systems under LMI pole placement and H2/Hinf guaranteed cost constraints". In.

Seshagiri, Sridhar and Ekprasit Promtun (2008). "Sliding mode control of F-16 longitudinal dynamics". In: *2008 American control conference*. IEEE, pp. 1770–1775.

Sève, F et al. (2017). "Design and Validation of Aerospace Control Systems Gain-Scheduled Hinf Loop-Shaping Autopilot Design for Spin-Stabilized Canard-Guided Projectiles". In: DOI: 10.12762/2017.AL13-03.

Sève, Florian et al. (2014). "Pitch/Yaw Channels Control Design for a 155mm Projectile with Rotating Canards, using a H-infinity Loop-Shaping Design Procedure". In: DOI: 10.2514/6.2014-1474.

Shin, Jong-Yeob, Gary J Balas, and M Alpay Kaya (2002). "Blending methodology of linear parameter varying control synthesis of F-16 aircraft system". In: *Journal of Guidance, Control, and Dynamics* 25.6, pp. 1040–1048.

Simon, E. (2011). "Optimal static output feedback design through direct search". In: pp. 296–301. DOI: 10.1109/CDC.2011.6160195.

Simon, Emile (2011). "Optimal static output feedback design through direct search". In: *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, pp. 296–301.

Skogestad, Sigurd and Ian Postlethwaite (2005). *Multivariable Feedback Control: Analysis and design*.

Sonneveldt, L. et al. (2009). "Nonlinear adaptive flight control law design and handling qualities evaluation". In: pp. 7333–7338. DOI: 10.1109/CDC.2009.5400209.

Sonneveldt, Lars, QP Chu, and JA Mulder (2006). "Constrained adaptive backstepping flight control: application to a nonlinear F-16/MATV model". In: *AIAA guidance, navigation, and control conference and exhibit*, p. 6413.

Stein, G. (2003). "Respect the unstable". In: *IEEE Control Systems Magazine* 23.4, pp. 12–25. DOI: 10.1109/MCS.2003.1213600.

Stevens, Brian L., Frank L. Lewis, and Eric N. Johnson (2016). *Aircraft Control and Simulation*.

Syrmos, V. L. et al. (Feb. 1997). "Static output feedback—A survey". In: *Automatica* 33 (2), pp. 125–137. ISSN: 0005-1098. DOI: 10.1016/S0005-1098(96)00141-0.

Tian, Yongxiao et al. (2020). "Resilient static output feedback control of linear semi-Markov jump systems with incomplete semi-Markov kernel". In: *IEEE Transactions on Automatic Control* 66.9, pp. 4274–4281.

Toker, O. and H. Ozbay (1995). "On the NP-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback". In: 4, 2525–2526 vol.4. DOI: 10.1109/ACC.1995.532300.

Tol, HJ et al. (2014). "Nonlinear multivariate spline-based control allocation for high-performance aircraft". In: *Journal of Guidance, Control, and Dynamics* 37.6, pp. 1840–1862.

Torczon, Virginia Joanne (1989). *Multidirectional search: A direct search algorithm for parallel machines*. Rice University.

Tran Dinh, Quoc et al. (2012). "Combining Convex–Concave Decompositions and Linearization Approaches for Solving BMIs, With Application to Static Output Feedback". In: *IEEE Transactions on Automatic Control* 57.6, pp. 1377–1390. DOI: 10.1109/TAC.2011.2176154.

Tsai, Mi-Ching, EJM Geddes, and Ian Postlethwaite (1992). "Pole-zero cancellations and closed-loop properties of an H-infinity mixed sensitivity design problem". In: *Automatica* 28.3, pp. 519–530.

Varsek, Alen, Tanja Urbancic, and Bogdan Filipic (1993). "Genetic algorithms in controller design and tuning". In: *IEEE transactions on Systems, Man, and Cybernetics* 23.5, pp. 1330–1339.

Vo, Hoa and Sridhar Seshagiri (2008). "Robust control of F-16 lateral dynamics". In: *2008 34th Annual Conference of IEEE Industrial Electronics*, pp. 343–348. DOI: 10.1109/IECON.2008.4757977.

Voulgaris, Petros and Lena Valavani (1991). "High performance linear-quadratic and H-infinity designs for a 'supermaneuverable' aircraft". In: *Journal of Guidance, Control, and Dynamics* 14.1, pp. 157–165. DOI: 10.2514/3.20617.

Yang, Ciann-Dong, Chia-Yuan Chang, and Yun-Ping Sun (1996). "Synthesis of optimal H/sub /spl infin// controllers via LQG-based loop shaping design". In: 1, 739–744 vol.1. DOI: 10.1109/CDC.1996.574441.

Zames, G. (1981). "Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses". In: *IEEE Transactions on Automatic Control* 26.2, pp. 301–320. DOI: 10.1109/TAC.1981.1102603.

Zhang, Hui et al. (June 2014). "Robust finite frequency H8 static-output-feedback control with application to vibration active control of structural systems". In: *Mechatronics* 24 (4), pp. 354–366. ISSN: 0957-4158. DOI: 10.1016/J.MECHATRONICS.2013.07.013.

Zhang, Meng et al. (2021). "Optimization of Aero-engine H-infinity Robust Controller Based on Quantum Genetic Algorithm". In: pp. 225–231. DOI: 10.1109/ICMAE52228.2021.9522430.

Zhao, Xingcheng et al. (2019). "Robust H_∞ /S-plane Controller of Longitudinal Control for UAVs". In: *IEEE Access* 7, pp. 91367–91374. DOI: 10.1109/ACCESS.2019.2927000.

Zipfel, Peter H. (2007). *Modeling and simulation of aerospace vehicle dynamics*. American Institute of Aeronautics and Astronautics, p. 567. ISBN: 9781563478758.