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

Schoon, A.D.P.; Theodoulis, S.T.

DOI
10.2514/6.2025-2241

Publication date
2025

Document Version
Final published version

Published in
Proceedings of the AIAA SCITECH 2025 Forum

AIAA SciTech Forum
6-10 January 2025, Orlando, FL
AIAA SCITECH 2025 Forum

10.2514/6.2025-2241

Review of $\mathcal{H}_\infty$ Static Output Feedback Controller Synthesis Methods for Fighter Aircraft Control

A.D.P. Schoon* and S. Theodoulis†
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 $\mathcal{H}_\infty$-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 the F-16 Fighting Falcon.

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	performance output matrix	
D_{yu}	input to measured output	
D_{yw}	exogenous input to measured output	
D_{zu}	input to performance output	
D_{zw}	exogenous input to performance 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	sensitivity function	
T	complementary sensitivity function	
T_{c1}	constant matrix to build T_y	
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	
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	$\mathcal{H}_\infty$ performance index	[-]
ϵ	integrated control error	$g\cdot$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.
†Associate Professor, Control & Simulation division, Faculty of Aerospace Engineering, P.O. Box 5058, 2600GB Delft, The Netherlands; S.Theodoulis@tudelft.nl. Associate Fellow AIAA.

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 the F-16 Fighting Falcon. 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$_{\text{EVO}}$ 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, 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. $\mathcal{H}_\infty$ 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 $\mathcal{H}_\infty$-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 $\mathcal{H}_\infty$ 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* $\mathcal{H}_\infty$ *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: $K = YP_1^{-1}$ in Lemma 4.2 is replaced in SOFHi by $K = YP_1^+$ 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 $\mathcal{H}_\infty$ 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* $\mathcal{H}_\infty$ *SOF gain matrix* $K = K_{SOF}$ *is the global optimal solution to the following non-convex optimization problem:*

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

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:*

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

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 $K_{\mathrm{SF}} \in \mathfrak{K}_{\mathrm{SF}}$, Approach II is in fact based on a stabilizing $K_{\mathrm{SOF}} \in \mathfrak{K}_{\mathrm{SOF}}$, where $\mathfrak{K}_{\mathrm{SOF}}$ of size $j_{\max}$ is defined by

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

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 $\mathfrak{K}_{\mathrm{SOF}}$ of size $j_{\max}$ defined in Eq. 13, where $K_{\mathrm{SOF}}^{(1)} := K_{\mathrm{SOF}}$.
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 number of states, r the number of inputs, and l the number of 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 $\mathfrak{Q}$ be the set of gains at the iteration index k that is equal to N_{samples}:

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

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

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

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.

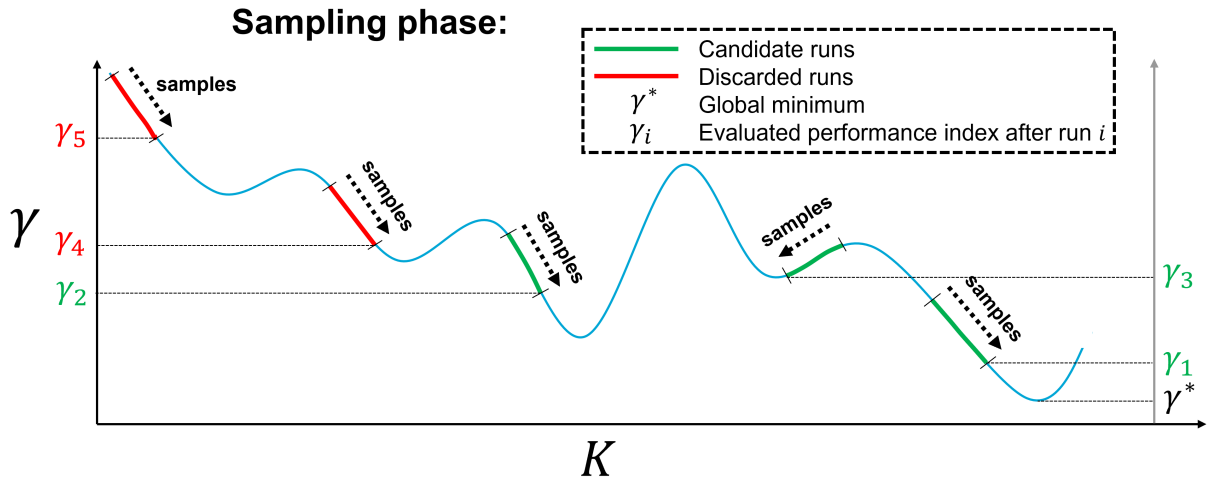


Fig. 1 Visualization of the solution space for a fictional case to explain the sampling phase in SOFHi$_{\text{EVO}}$, where the gain matrix K is assumed to be the only decision variable and is a single scalar gain to be able to have a two-dimensional visualization of the optimization procedure. The 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 $\mathcal{H}_\infty$-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 9 and Table 10. 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. SOFHi$_{\text{EVO}}$ comparison

The comparison between SOFHi and 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, SOFHi$_{\text{EVO}}$ is able to outperform SOFHi and uses less computational time. The most extreme case is emphasized, when SOFHi$_{\text{EVO}}$ and SOFHi have the same N_{starts} and SOFHi$_{\text{EVO}}$ only proceeds with 8 of the original 30 starts after 30 sample iterations. It is expected that SOFHi will outperform SOFHi$_{\text{EVO}}$, since 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 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 SOFHi$_{\text{EVO}}$ and SOFHi on computational performance and γ. Parameters are denoted as "(N_{starts}, N_{samples}, $N_{\text{candidates}}$)" for SOFHi$_{\text{EVO}}$ and as "N_{starts}" for SOFHi. Relative computational time denotes the time it takes for 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 $0.99\gamma(B) < \gamma(A) < 1.01\gamma(B)$.

	SOFHi$_{\text{EVO}}$	SOFHi	SOFHi$_{\text{EVO}}$	SOFHi	SOFHi$_{\text{EVO}}$	SOFHi	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 SOFHi$_{\text{EVO}}$ when compared to HIFOO and hinfstruct are presented in Table 4. It can be seen that SOFHi$_{\text{EVO}}$ scores significantly better than HIFOO, with and without a margin. Compared to hinfstruct, it can be seen that 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, SOFHi$_{\text{EVO}}$ is at least competitive to hinfstruct, slightly outperforming when a significance margin is included.

Table 4 Comparison between SOFHi$_{\text{EVO}}$, HIFOO, and hinfstruct on γ. Parameters are denoted as "(N_{starts}, N_{samples}, $N_{\text{candidates}}$)" for 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.

	SOFHi$_{\text{EVO}}$	HIFOO	SOFHi$_{\text{EVO}}$	HIFOO	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%
	SOFHi$_{\text{EVO}}$	hinfstruct	SOFHi$_{\text{EVO}}$	hinfstruct	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 SOFHi$_{\text{EVO}}$ has many more starts than 30. However, most of these runs are discarded after the sampling phase and 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), SOFHi$_{\text{EVO}}$ is able to be *more* competitive to HIFOO and hinfstruct than the original algorithms in SOFHi were, to the point that 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.

Remark 2: The percentages in Table 3 and Table 4 do not always add up to 100% for "No Margin", since some results are exactly equal up to MATLAB's numerical precision.

C. Structured SOF

In section IV, it was described that the algorithms in SOFHi are easily extended to structured SOF, where a zero-nonzero structure is imposed on K_{SOF}. The effectiveness of this is assessed by comparing SOFHi to other research on structured SOF described in [13, 14, 17, 28–30], and to HIFOO and hinfstruct. The algorithms are applied on the example given in [13], and the results of this are shown in Table 5. It can be seen that SOFHi is able to obtain solutions with less conservatism than other methods when imposing different zero-nonzero structures on K_{SOF}. Furthermore, in terms of computational times, SOFHi is able to be at least competitive to the other algorithms.

Table 5 Structured SOF results on γ. (MD1-MS5) are respectively described in [13, 14, 17, 28–30]. Their results on this example are extracted from [17].

	Full structure	**Tri-diagonal structure**	**Diagonal structure**
Algorithm	γ (time)	γ (time)	γ (time)
MD1	1.2084 (0.015s)	1.2358 (0.013s)	1.2859 (0.01s)
MS1	1.0543 (11s)	1.2223 (5.9s)	1.2759 (5.3s)
MS2	0.7891 (6.2s)	0.8856 (5.9s)	0.8933 (5.3s)
MS3	0.8411 (17s)	0.9338 (15s)	0.9680 (12s)
MS4	0.7720 (2.3s)	0.7730 (2.1s)	0.8400 (2.1s)
MS5	0.7689 (1.9s)	0.7841 (1.8s)	0.8925 (1.8s)
HIFOO	0.7357 (0.85s)	0.7416 (2.5s)	0.8221 (16s)
hinfstruct	0.7357 (0.49s)	0.7493 (0.47s)	0.8032 (0.33s)
T-K iteration	**0.7356** (1.6s)	0.7712 (1.7s)	1.0239 (1.5s)
S-variable App. I	0.7357 (2.1s)	0.7424 (2.5s)	0.7993 (1.6s)
S-variable App. II	0.7357 (1.8s)	**0.7411** (1.5s)	**0.7992** (2.3s)

VI. F-16 Example

To showcase the effectiveness of SOFHi in tuning $\mathcal{H}_\infty$ 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 [31], and the aerodynamic model is taken from [32, 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 [33, Section 5.1]. Furthermore in the second stage, a second-order feedforward function F is tuned using an additional $\mathcal{H}_\infty$-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 [34, Section II.F]. This ultimately favors soft-starts and smooth responses in tracking applications.

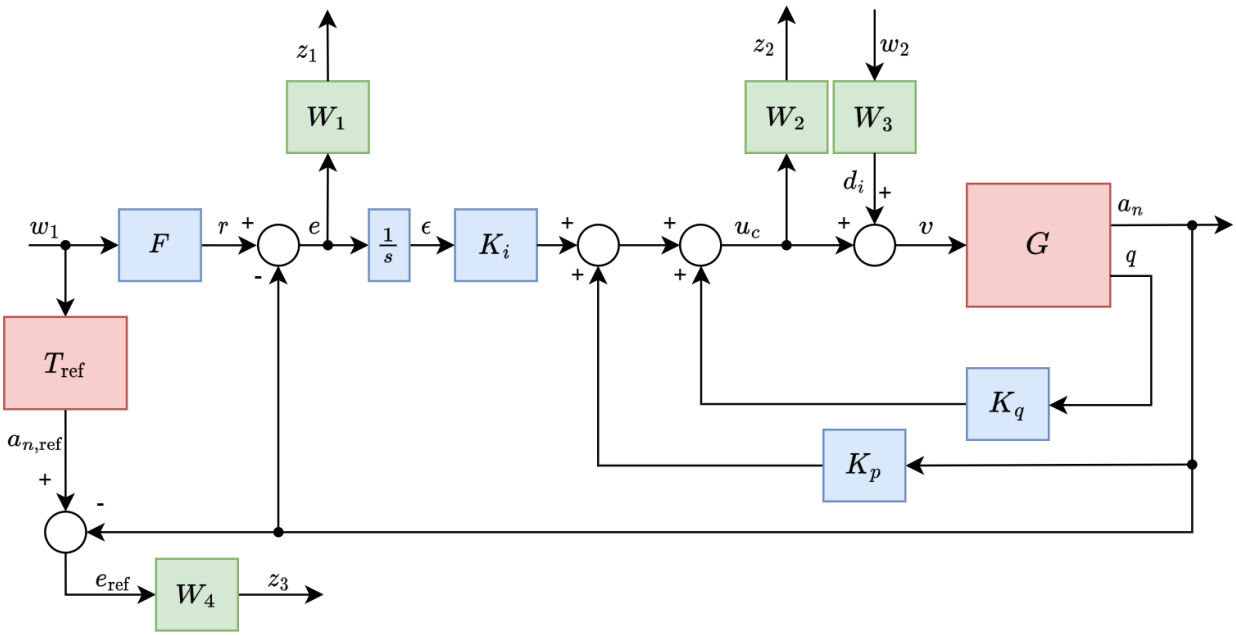


Fig. 2 2-DoF design for the normal acceleration CAS. G denotes the Linear Time Invariant (LTI) plant, which originates from trimming and linearizing the non-linear model around each flight point. W_i denotes the weighting functions that shape the closed-loop sensitivity functions. (K_i, K_p, K_q) are 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 6. The design objectives to the sensitivity functions in the frequency domain for this 2-DoF design are as follows [35].

- 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 6 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 [32, 36, 37].

- The closed-loop system shall be internally stable.
- The overshoot of the normal acceleration time responses shall not exceed 10% [37].
- The settling times of the normal acceleration time responses shall not exceed 3 [s] [37].
- The damping ratio of the normal acceleration time responses shall be at least 0.3 [-] [36, 37].
- 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 [32].
- The minimum gain and phase margins shall be at least 6 [dB] and 40 [deg], respectively [36].
- The Control Anticipation Parameter (CAP) requirements shall be met [36].

B. $\mathcal{H}_\infty$ 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:

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

The $\mathcal{H}_\infty$-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.

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 $w_1 \rightarrow [z_2 \quad z_3]^\top$ 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.

[§]These weighting functions must be stable and proper due to assumptions made about the generalized plant [35, Page 98].

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 [37], 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 $\mathcal{H}_\infty$-norm of each flight point is presented in Table 7. 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 7 $\mathcal{H}_\infty$ 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.

Mach [-] / h [m]	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.

Mach [-] / h [m]	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

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.

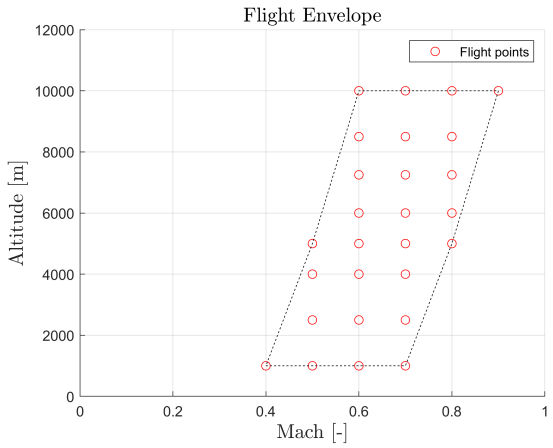
1. Sensitivity Functions

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.

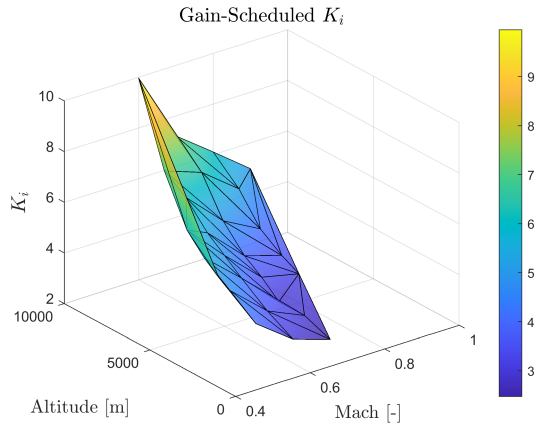
The sensitivity functions of the second stage in the 2-DoF design are shown in Fig. 5. In Fig. 5a, it can be seen that $KS_{a_n}F$ has a higher peak than KS_{a_n} in the mid-frequency range, which is due to the addition of the feedforward function F, which leads to the controlled system requiring more 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.

2. Stability Margins

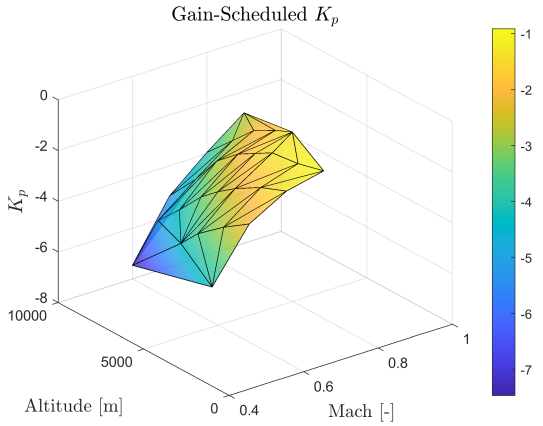
Classical stability margins can give an optimistic perspective of the stability margins, since they do not take into account simultaneous gain and phase. To assess the stability of the closed-loop system more conservatively, the Nichols plots of each flight point including the worse-case exclusion region 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.



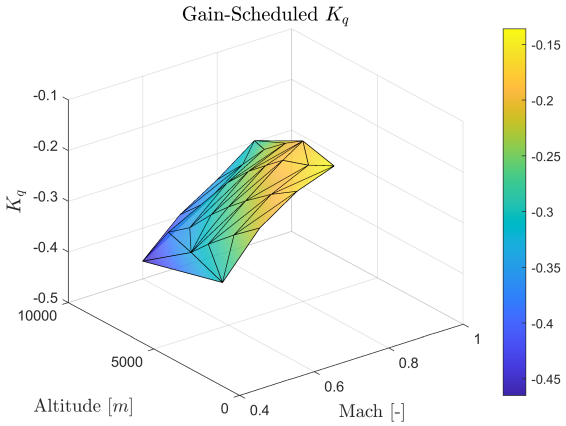
(a) Flight envelope including its design points.



(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.

3. Linear Simulations

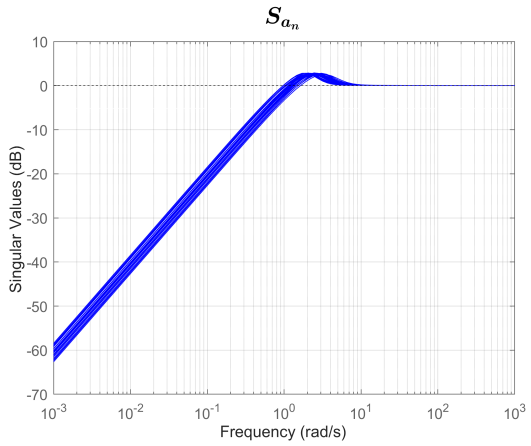
To assess the performance in controlling the normal acceleration output, time-domain simulations were performed on linear models around each of the 27 flight points in Fig. 3a. These are presented in Fig. 7. The results show good reference tracking and disturbance rejection, with no steady-state error,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

The specific values for the response parameters are presented in Table 8. In Table 8b, 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 8c and the percentages overshoot in Table 8e meet the requirements of respectively 3 seconds and 10% for all of the flight points.

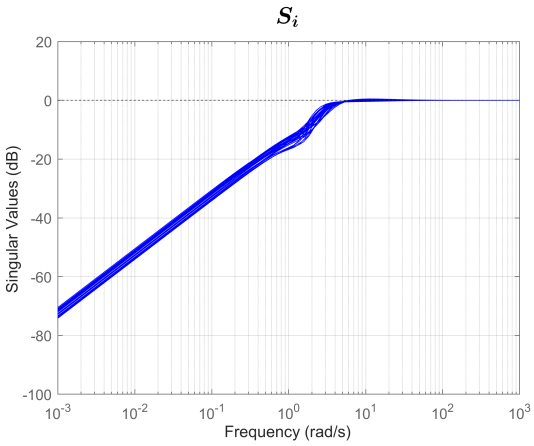
4. Handling Qualities

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) [38], one of the design requirements defined in subsection VI.A.

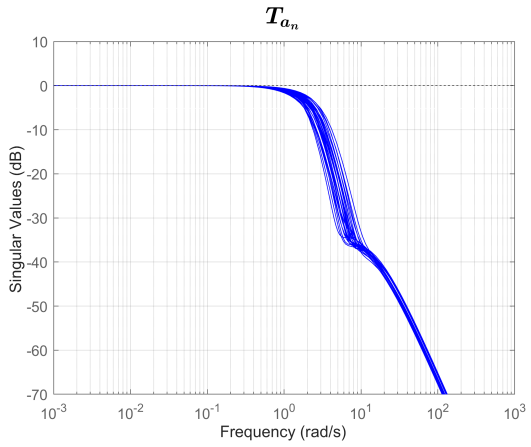
The result of this is visualized in Fig. 8. It can be seen that the gain-scheduled controller meets the requirements for CAP. Naturally, there is a clear trade-off between improving CAP requirements and reducing control effort, so care was taken to ensure as little control effort as possible, while still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



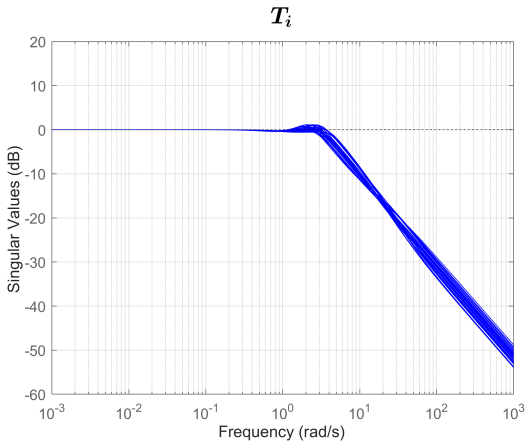
(a) Singular value plot of S_{a_n}.



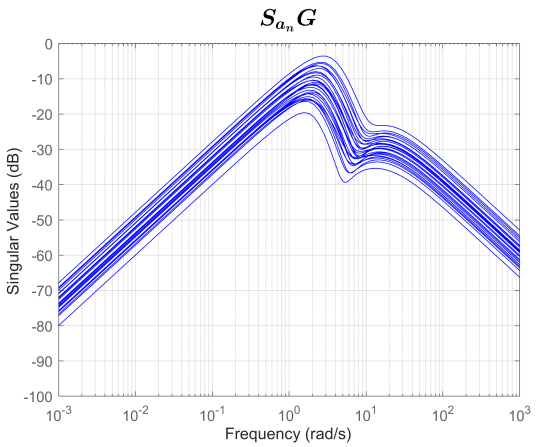
(b) Singular value plot of S_i.



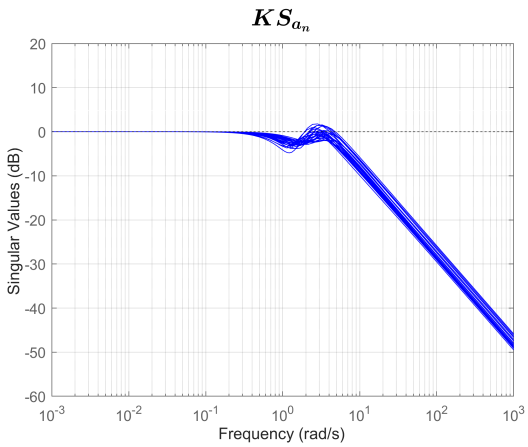
(c) Singular value plot of T_{a_n}.



(d) Singular value plot of T_i.

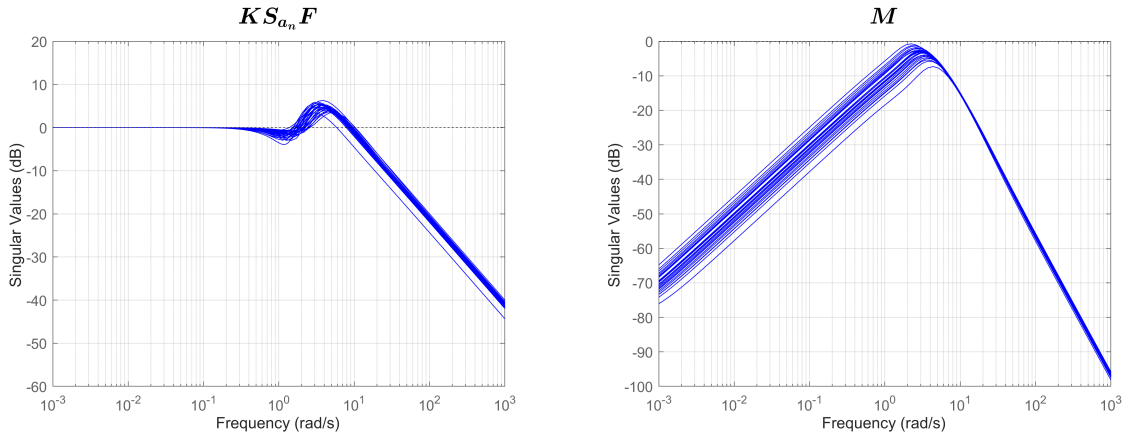


(e) Singular value plot of $S_{a_n}G$.



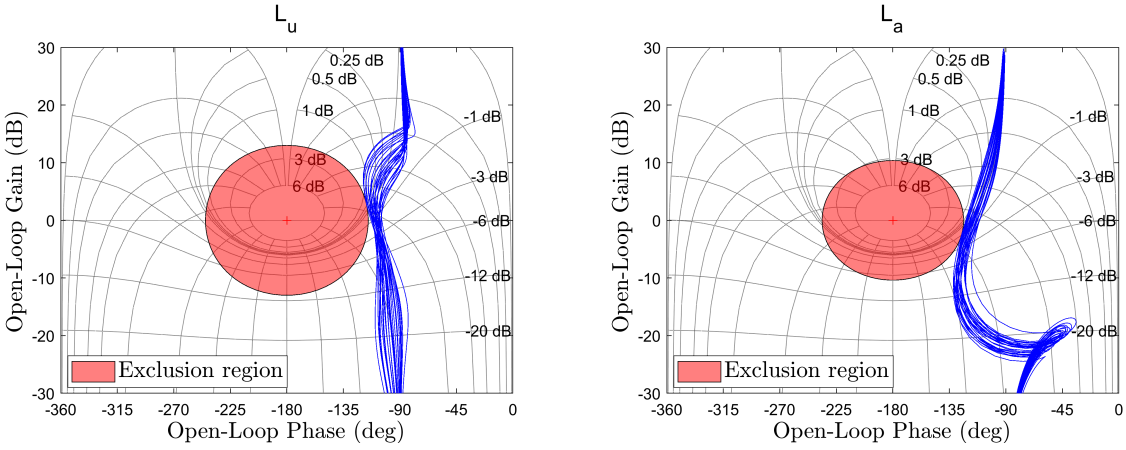
(f) Singular value plot of KS_{a_n}.

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



(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.



(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.

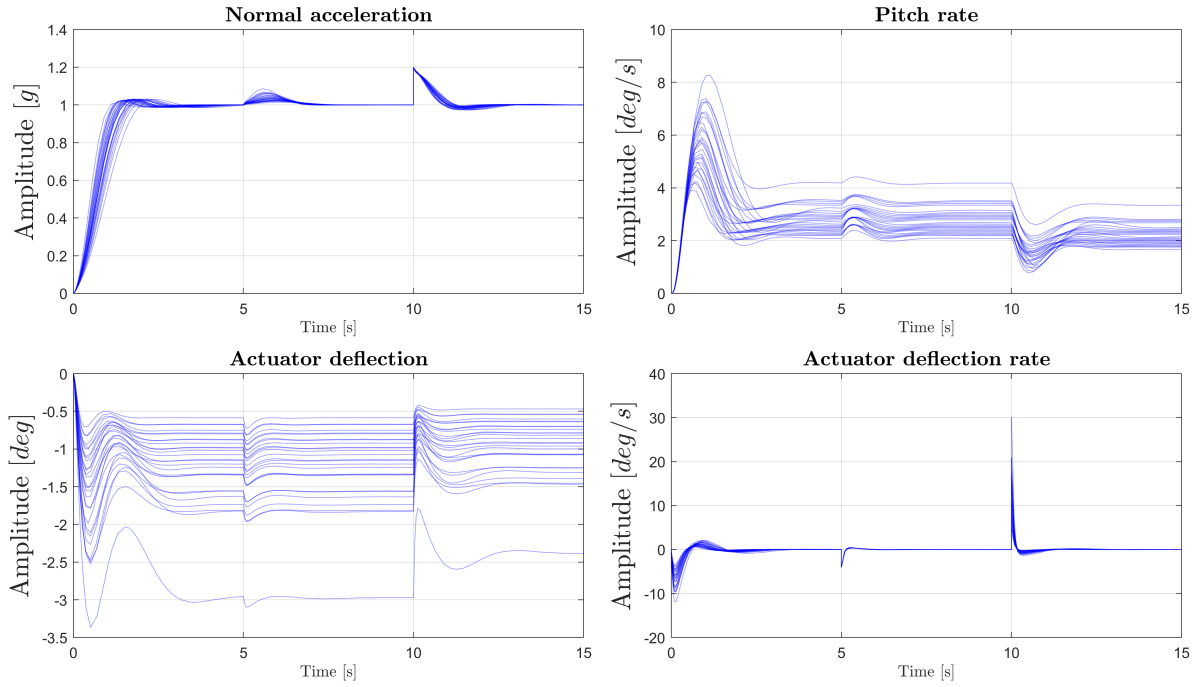
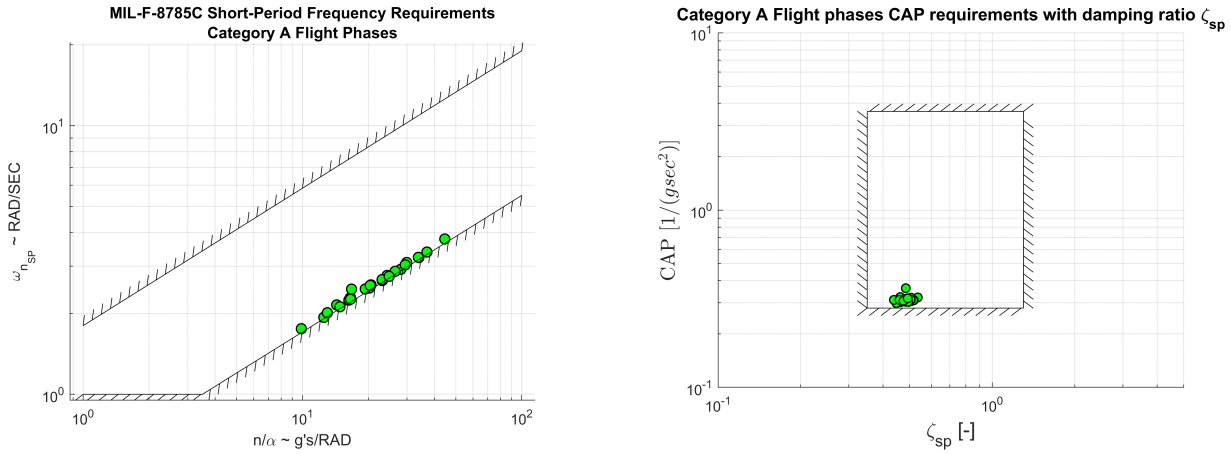


Fig. 7 Linear simulations of all the flight points including an input disturbance of 0.2 [deg] at 5 seconds and an output disturbance of 0.2 [g] at 10 seconds.



(a) CAP requirement. **(b) Damping ratio requirement.**

Fig. 8 Level 1 CAP requirements for Category A flight phases based on [38]. Dots indicate the result for each of the 27 flight points. Green indicates that the dot is within the desired region.

Table 8 Linear response characteristics for all designed flight points.

(a) Natural frequency ω_{sp} [rad/s].

h [m] \ Mach [-]	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} [-].

h [m] \ Mach [-]	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.

h [m] \ Mach [-]	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.

h [m] \ Mach [-]	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.

h [m] \ Mach [-]	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}.

h [m] \ Mach [-]	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

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 mass m. Variations in these parameters range in $X_{\text{COG}} \pm 4.35\%$ and $m \pm 1000$ kg, similarly to [37]. The simulations were performed on the non-linear 6-DoF model from [31, 32] 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.

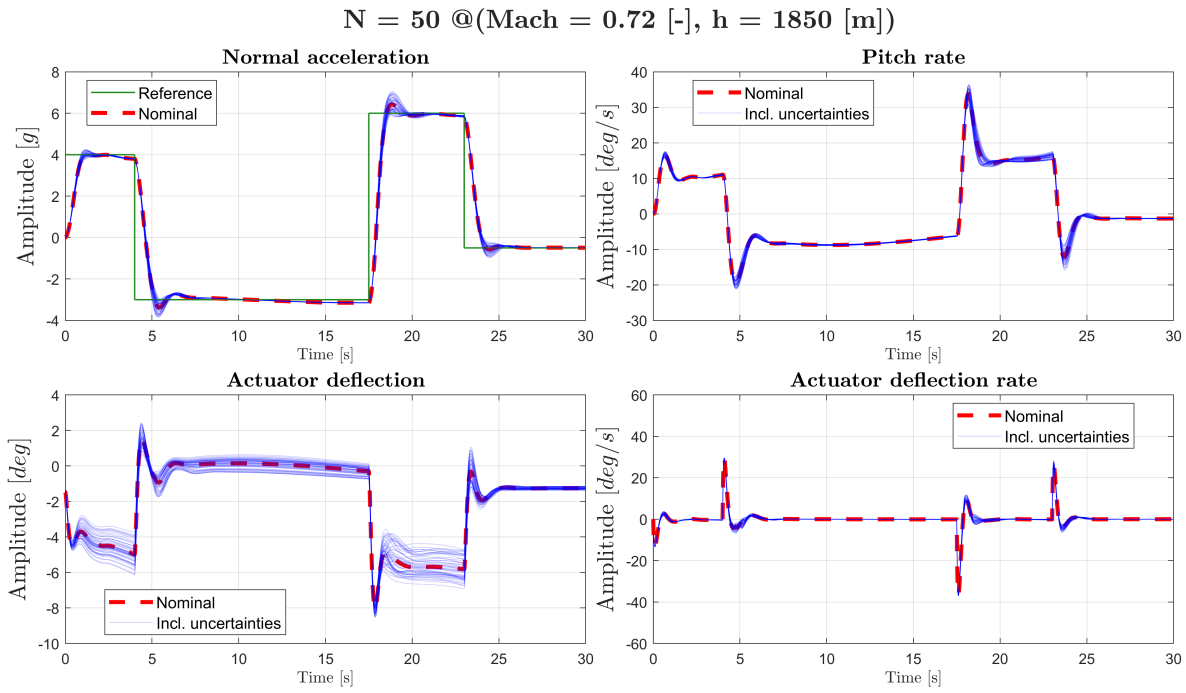


Fig. 9 Monte-Carlo simulations applied on the non-linear model with 50 runs to showcase the robustness of the controller at high-g's maneuvers.

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 $\mathcal{H}_\infty$ 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, 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, 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

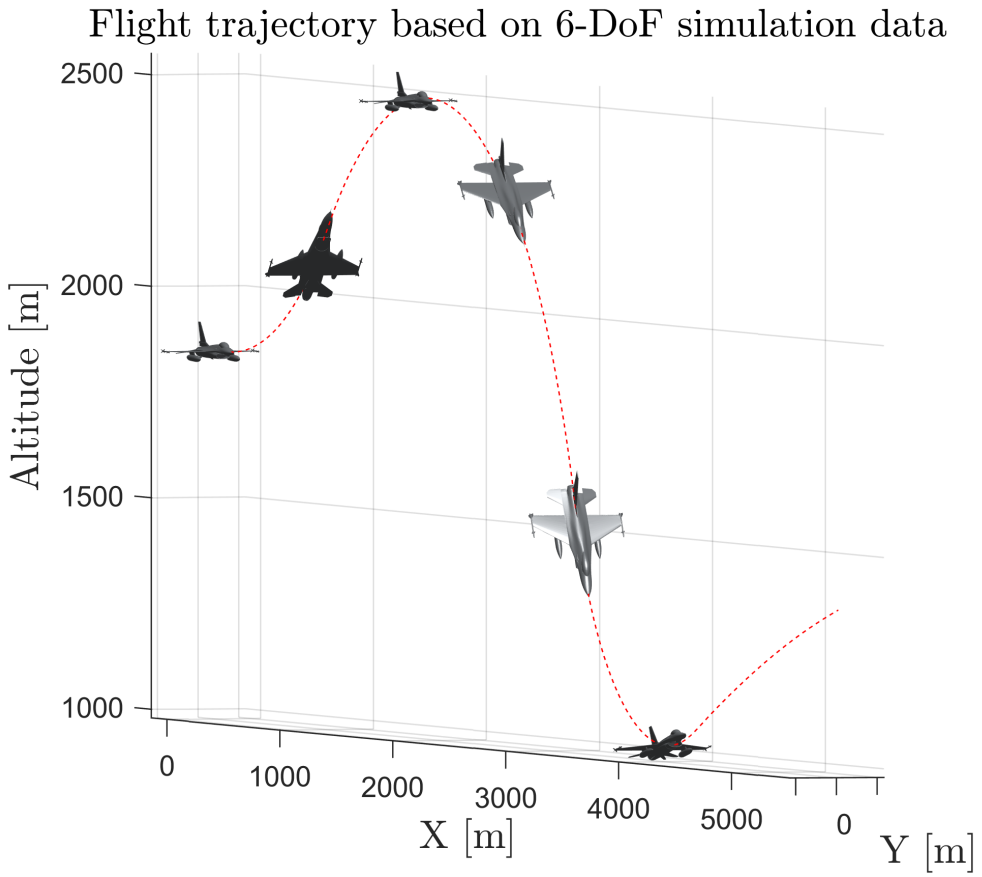


Fig. 10 Position and attitude trajectory of the aircraft, based on 6-DoF data from the non-linear model [39].

Table 9 $\|T_{zw}(P,K)\|_\infty$ **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 10 $\|T_{zw}(P,K)\|_\infty$ **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] Shaked, U., "An LPD approach to robust H2 and H infinity static output-feedback design," *IEEE Transactions on Automatic Control*, Vol. 48, No. 5, 2003, pp. 866–872.

[29] Dong, J., and Yang, G.-H., "Robust static output feedback control synthesis for linear continuous systems with polytopic uncertainties," *Automatica*, Vol. 49, No. 6, 2013, pp. 1821–1829.

[30] Chang, X.-H., Park, J. H., and Zhou, J., "Robust static output feedback H infinity control design for linear systems with polytopic uncertainties," *Systems & Control Letters*, Vol. 85, 2015, pp. 23–32.

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

[32] 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.

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

[34] 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.

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

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

[37] 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.

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

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