Advanced control laws design and validation - A set of methods and tools to bridge the gap between theory and practice

Clément Roos

ONERA
THE FRENCH AEROSPACE LAB

HABILITATION A DIRIGER DES RECHERCHES

UNIVERSITÉ TOULOUSE III - PAUL SABATIER

SPÉCIALITÉ AUTOMATIQUE

par

CLÉMENT ROOS

Advanced control laws design and validation

A set of methods and tools to bridge the gap between theory and practice

Habilitation présentée le 16 avril 2018 devant le jury composé de :

Michel Basset	Rapporteur	Professeur Université de Haute Alsace
Jean-Marc Biannic	Examinateur	Directeur de recherche ONERA
Patrick Danès	Examinateur	Professeur Université de Toulouse
Andrés Marcos	Examinateur	Professeur Université de Bristol
Isabelle Queinnec	Directrice de HDR	Directrice de recherche LAAS-CNRS
Gérard Scorletti	Rapporteur	Professeur Ecole Centrale de Lyon
Peter Seiler	Examinateur	Professeur Université du Minnesota
Matthew Turner	Rapporteur	Professeur Université de Leicester

Contents

Acknowledgments

Je remercie en premier lieu Jean-Marc Biannic, que je côtoie depuis bientôt 14 ans à l'ONERA. Il a été mon professeur puis mon directeur de thèse avant de devenir mon plus proche collègue, précieux par ses conseils, toujours présent à mes côtés, et généreux dans son aide et son soutien. Si j'achève la rédaction de ce manuscrit aujourd'hui, c'est pour beaucoup grâce à lui.

Je remercie ensuite tous les autres chercheurs qui ont accepté de faire partie de mon jury : Michel Basset de l'Université de Haute-Alsace, Patrick Danès de l'Université de Toulouse, Andrés Marcos de l'Université de Bristol, Isabelle Queinnec du LAAS-CNRS, Gérard Scorletti de l'Ecole Centrale de Lyon, Peter Seiler de l'Université du Minnesota et Matthew Turner de l'Université de Leicester. Merci aux rapporteurs pour leur lecture attentive de ce manuscrit et leurs remarques constructives. Merci à ceux qui sont venus de loin. Et merci à tous d'avoir consacré du temps à mon habilitation, notamment Isabelle pour avoir accepté de jouer le rôle de directrice de HDR et pour m'avoir guidé dans ma préparation.

Je remercie également tous les chercheurs, ingénieurs, doctorants et stagiaires avec qui j'ai pu mener à bien les travaux présentés dans ce manuscrit. Je m'abstiendrai de les nommer de manière exhaustive de peur d'en oublier, mais j'espère que je continuerai à collaborer avec eux et avec d'autres à l'avenir.

Je remercie enfin Christelle Cumer, responsable de l'équipe de recherche Commande et Intégration, devenue récemment Automatique, Expérimentation, Intégration, qui n'a jamais ménagé ses efforts pour me rendre la vie plus facile pendant toutes ces années.

Un clin d'œil pour finir à Sophie qui m'a vivement encouragé (et c'est un euphémisme) à me lancer dans la préparation de ce manuscrit, et à Victor qui du haut de ses 9 ans m'a proposé de m'aider à le rédiger.

Part I

Summary of research activities

Introduction

It all started 13 years ago. I was a young PhD student with no precise idea what direction I was heading to. I had few other options than carefully listening to my supervisor singing the praises of robustness analysis in general, and of μ-analysis in particular. I also saw my close colleagues working on the same topic. And I noticed the growing interest among control engineers from the aerospace industry. So I naturally took this path. At that time, it was already known for a long time that computing the exact value of the structured singular value μ is NP hard. So people were trying to develop polynomial-time algorithms to compute tight lower and upper bounds instead. And my colleagues were playing a key role with significant contributions that remain references even today [Ferreres and Biannic, 2001; Ferreres *et al.*, 2003], as evidenced by the Skew-Mu Toolbox for Matlab which was widely used during a whole decade [Ferreres *et al.*, 2004]. It was in this environment that I took my first steps.

Then I started thinking a little bit on my own. The first major project I was involved in dealt with the clearance of flight control laws. Before an aircraft can be tested in flight, it has to be proven to the authorities that the flight control system is reliable, *i.e.* it has to go through a certification and qualification process. It must notably be shown that the control laws provide sufficient stability margins to guarantee a safe operation of the aircraft over the entire flight envelope and for all admissible parametric variations. In the aeronautical industry, this is usually achieved using intensive Monte-Carlo simulations. A major drawback of this strategy is that clearance is restricted to a finite number of samples and nothing can in principle be assessed for the rest of the parametric domain. Moreover, significant time and money is frequently spent on this task. Fortunately, many stability, handling, loads and performance criteria can be formulated as robustness analysis problems, which can then be solved using modern analysis techniques, such as μ-analysis, IQC-based analysis or Lyapunov-based analysis. At that point, I realized that these techniques were not just research topics tailored to control theorists, but that they could be of significant practical importance to improve the efficiency and the reliability of the certification process, and more generally to enhance the validation of any feedback system. My conviction was reinforced when I noticed that the aerospace industry was following the subject closely: Airbus [Puyou *et al.*, 2012], Boeing [Dailey, 1990], Astrium Satellites [Beugnon *et al.*, 2003], Astrium Space Transportation [Ganet-Schoeller *et al.*, 2009], Deimos & ESA [Pulecchi *et al.*, 2012], Thales Alenia Space [Charbonnel, 2010] and others were all evaluating the potential benefits of μ-analysis on real-world applications. But they were also very critical. As I had already invested quite a lot of time on this subject, I was cut to the quick, and I decided to try making my own contribution to address the issues raised by our industrial partners.

A first recurrent criticism was that the gap between the bounds on μ was often too large. The resulting robust stability margin was thus too pessimistic, and it was sometimes impossible to conclude about stability even though the considered system was indeed stable. In this context, my coworkers and I performed a thorough comparison of all pratical algorithms to compute lower bounds on μ, and we proposed some strategies to take the most out of them (Section A.2).

Noting that the μ upper bound was generally responsible for the conservatism, we tried to improve the accuracy of existing algorithms, while maintaining a reasonable computational time (Section A.3). And as it was sometimes not sufficient, we proposed some enhanced branch-and-bound algorithms to further tighten the gap up to a user-defined precision (Section A.4). Evaluation on a wide set of real-world benchmarks available in the literature shows that non-conservative robust stability margins can be obtained in almost all cases. So we believe now that computing the exact value of μ is no longer an issue from a practical point of view.

Another criticism was that very few criteria could be evaluated with μ-analysis compared to other techniques such as Monte-Carlo simulations. It was true, and despite our efforts, it will remain partially. We proposed several algorithms to compute tight bounds on the skewed robust stability margin, the worst-case H_∞ performance level, the worst-case gain/phase/modulus/delay margins (Section A.5)... So μ-analysis clearly allows to go beyond stability and to address many other practical issues. Nevertheless, it must be kept in mind that only parametric uncertainties and neglected dynamics can be considered (although a few applications to nonlinear systems have been reported in the literature). Other techniques must be used to go further, such as IQC-based analysis or evolutionary algorithms, which allow uncertainties, varying parameters and various kinds of nonlinearities to be considered at the same time. But even so, the size of the considered models is limited to avoid high conservatism and prohibitive computational time. μ-analysis and similar methods thus prove useful in the early validation stages, but there comes a time when simulations are inevitable at the moment. It seems therefore obvious that there is much to be gained from combining both approaches. And this is the message we are trying to convey to control engineers.

μ-analysis was finally criticized for being difficult to apply by non-expert users, which I can understand. I believe good theories only reveal their true potential when they become applicable to realistic problems. That is why I spend a lot of time developing tools to help both researchers and engineers get the most out of the methods I am working on. The SMART Library of the Systems Modeling Analysis and Control (SMAC) Toolbox for Matlab is one of them (Section A.6). It implements most of the μ-analysis based algorithms developed at ONERA during the last two decades, and it symbolizes the way I position myself as a research engineer. I am not a pure control theorist, others do that much better than me. And I am not an engineer either. I am somehere between the two, trying to bridge the gap between these two worlds. This is how my contribution should be understood. I sometimes propose heuristics without any theoretical proof. But I prefer a heuristic which works on most real-world applications, than a more rigorous approach which can only be applied to simple examples.

As can be seen in the literature, much has been done in the field of μ-analysis, and this technique is now quite mature. Unfortunately, the same observation cannot be made for all robustness analysis techniques, and I could be criticized for having been primarily interested in μ-analysis at the expense of other methods. But many projects during and after my PhD thesis dealt with this subject, so I naturally continued along this path. And anyway, I am still young and I have quite a lot of time left to investigate other approaches! More seriously, I began considering time-varying parameters and uncertainties as well a long time ago because most aerospace vehicles are characterized by parameters that vary more or less rapidly. We first proposed generalizations of the aforementioned μ-analysis tools to evaluate the stability and the performance properties of linear systems in the presence of both LTI and arbitrarily fast varying uncertainties. Two complementary solutions were developed: the first one (in the frequency domain) is suitable for high-order models with few uncertainties (Section B.1), while the second one (in the time domain using an extended version of the KYP lemma) is better for low-order models with many uncertainties (not detailed in this manuscript due to space constraints). It

should be noted that beyond analysis, these techniques also allow to design a robust feedforward controller in order to improve performance. Nevertheless, in many practical applications, it is desirable to consider the rate of variation of time-varying uncertainties and parameters to get more accurate results. In this context, we proposed later a completely different strategy based on parameter-dependent Lyapunov functions (Section B.2). Several kinds of dependence can be considered, but the resulting optimization problem always boils down to solving an infinite number of linear matrix inequalities. Rather than introducing some relaxation variables and working on sufficient conditions, we proposed to solve the problem on a finite parametric grid and to check the result on the whole parametric domain using a fast and reliable μ-analysis based test. This strategy is symbolic of our work. Most of the aforementioned methods were not new, but the previously existing algorithms often had two major drawbacks when high-order systems were considered, namely a high computational time and a high conservatism. The central concern of our work and our main contribution have been to provide solutions to these problems.

Another issue that makes it difficult to convince control engineers to use modern robustness analysis techniques is that the considered systems must often be modeled using Linear Fractional Representations (LFR). Unfortunately in most industrial applications, physical systems are described using a mix of nonlinear analytical expressions and tabulated data, which are far from being in linear fractional form... But as no LFR means no μ-analysis, we had no other choice than developing dedicated tools that can be used by non-expert users. In general, a two-step procedure has to be implemented to obtain a suitable LFR: a linear model with a polynomial or a rational dependence on the system parameters is first generated, and then converted into a linear fractional form. Several techniques already existed to perform the latter transformation and to overcome complexity. Following the same motivations as for the SMART Library, we implemented state-of-the-art algorithms in the GSS Library and proposed an intuitive yet very general way to describe LFR, as well as a user-friendly Simulink interface (Section C.1.4). This is not what we might call research, but sometimes you have to be pragmatic. We will have taken a major step forward if engineers are no longer afraid to turn their models into LFRs. On the other hand, the preliminary issue of converting the tabulated or irrational data into simple yet accurate polynomial or rational expressions had been paid much less attention, although it is of significant practical importance. Most of the methods that existed at that time have two drawbacks: they only produce polynomial approximations, and all admissible monomials are usually nonzero regardless of their real ability to model the data. Yet, the additional degrees of freedom offered by rational expressions can lead to simpler expressions, and thus to avoid unnecessarily complex LFRs, which are usually source of conservatism and can even lead to numerical intractability. Moreover, computing approximations with sparse structure is beneficial, since it is a natural way to prevent data overfitting and to ensure a smooth behavior of the model between the points used for approximation. In this context, we developed several methods to fill this gap (Section C.1.3), and in the spirit of what we did before, we implemented the whole set of tools in the APRICOT Library, well organized for non-expert users (Section C.1.4). Finally, we proposed extensions to generate a reduced-order LFR from a set of large-scale MIMO state-space representations, representing for example an aerospace vehicle at various operating conditions (Section C.2).

While we are at it, let us take a closer look at aerospace vehicles. They depend on many parameters and are affected by plenty of uncertainties, varying or not, for which we now have a bunch of analysis tools at our disposal. But we are not out of the woods yet: they are also full of nonlinearities, among which saturations figure prominently. And this is a major issue. By limiting both the amplitude and the rate of the control signals, saturations induce particularly disturbing nonlinear phenomena (such as limit cycles), that can impair the closed loop performance or even jeopardise its stability. We tried to model saturations with time-

varying gains to apply the aforementioned tools, but the results obtained in practice are most often disappointing. Thus, we decided to focus on more appropriate analysis tools. Based on the Lyapunov theory, they allow to compute a stability domain and to quantify performance degradation. But beyond analysis, there is also the problem of taking saturations into account during the design step. Two strategies are possible. Either stay in the linear domain at the price of a lower level of performance, by computing unsaturating control laws. Or be more adventurous by allowing saturations and adapting control signals as soon as one of them is active, so as to return as quickly as possible to the linear domain guaranteeing a well-controlled nominal behavior of the system. The second approach, known as anti-windup, had already proved successful in the past and we decided to go on in that direction. Indeed, aerospace applications cannot tolerate an unnecessary loss of performance in the nominal case. Our contribution was again pragmatic and was mainly twofold. First, we developed a methodology to design a reduced-order anti-windup controller of low complexity (Section D.2). Then we introduced modal constraints during the design to avoid too slow or too fast dynamics likely to lower performance or to cause implementation issues (Section D.3). And unsurprisingly, the resulting algorithms were grouped together with a Simulink interface in the AWAS Toolbox for Matlab, which was later updated and became the SAW Library of the SMAC Toolbox (Section D.4).

Plenty of efficient tools are now available to analyze robustness and to validate control laws, and those we have developed so far are just a drop in the ocean. But in most cases, they are mainly applied once the controllers have been computed. Yet it could be advisable to better integrate them into the design process, in the hope of reducing a little bit the number of iterations between design and validation in an industrial context. And this is what we are trying to do now. We have proposed several methodologies, among which one is inspired by dynamic inversion techniques and looks promising (Chapter E). It combines partially linearizing inner-loops with structured and robust outer-loops, which are designed using a non-smooth multi-model H_∞ optimization approach. It also includes a robustness analysis scheme providing worst-case configurations, which are then used to enrich the bank of design models and thus to iteratively improve the robustness properties of the designed outer-loops. And it seems simple enough to be implemented in a flight computer. But for the moment, it is more a prospect than an achievement, and we will talk about it again later.

This manuscript is deliberately concise and mainly contains personal contributions. It is not exhaustive in the sense that I decided to highlight only certain methods, either because they are important to me or because they are little known although they may have a practical interest. Chapter A is significantly longer than the others. Robustness analysis, and μ-analysis in particular, has indeed been central to my work since day one. I do not always spend a lot of time on it, but I come back to it regularly. Moreover, I decided to focus on methods and tools rather than on their application. My philosophy is to propose general methods that can be applied to different kinds of systems. One could maybe do better by developing a specific technique for each application. But my experience is that control engineers do not have much time, so it is better to have few well-used methods than many ignored ones.

Remark: *Chapters 1 and 3 present the topics I spent most of my time on. I cannot go into too much detail because of space constraints, so I simply present a summary with the main results. Chapters 2 and 4, on the other hand, deal with issues that I addressed several years ago, and which I have now put aside (but maybe not forever!). Finally, Chapter 5 illustrates my current desire to go further towards the development of advanced design methods and their implementation on fixed wing UAVs.*

Chapter A

μ-ANALYSIS BASED ROBUSTNESS TOOLS

Despite evermore powerful design tools and increasing computer power, control laws are still designed based on mathematical models, which significantly simplify reality. A necessary but costly task consists in thoroughly validating them before they can be implemented. This involves numerous simulations of generally nonlinear and nonstationary models, which are as representative as possible of the real behavior of the considered physical system. However, it is certainly possible to significantly reduce the number of simulations by directing them towards worst-case parametric configurations detected through a series of simplified robust stability and performance tests.

Several approaches can be considered in this perspective: μ-analysis [Ferreres, 1999], integral quadratic constraints [Megretski and Rantzer, 1997], Lyapunov-based techniques [Garulli *et al.*, 2012], sum of squares [Chesi, 2010], evolutionary methods [Menon *et al.*, 2009], randomized algorithms [Tempo *et al.*, 2013]... The objective of this chapter is not to perform an extensive comparison of their advantages and drawbacks, but to focus on one of them, namely μ-analysis. This is the approach I have been working on most for a certain number of reasons already discussed in the introduction. And it is commonly accepted that this is the most suitable way to compute stability margins and performance levels for LTI systems affected by structured time-invariant uncertainties. The exact computation of the structured singular value μ being NP-hard [Braatz *et al.*, 1994], only lower and upper bounds on these quantities can usually be obtained with a reasonnable computational effort. Until recently, it was quite common with the available software to obtain a non-zero gap between the bounds, which could even be large in some cases (typically flexible systems affected by real parametric uncertainties). This is undesirable, as it may be impossible to conclude whether the required level of robustness is guaranteed or not.

We have invested a great deal of effort in proposing solutions to this problem, as well as to the various recurrent criticisms mentioned in the introduction. This chapter summarizes our contributions. A complete strategy is first described to compute accurate bounds on μ in Sections A.2 and A.3, and to bring the resulting gap to (almost) zero with a reasonable computational cost in Section A.4. A brief overview of what can be done with μ is then presented in Section A.5, where a series of pratical methods and computational tools are described to compute the (almost) exact value of the (skewed) robust stability margin, the worst-case H_∞ performance level, as well as the worst-case gain, phase, modulus and time-delay margins. Finally, Section A.6 briefly presents the SMART Library for Matlab, which implements most of the μ-analysis based tools we developed at ONERA during the last two decades.

1 A brief introduction to μ-analysis

Let us consider the standard interconnection of Figure A.1. $M(s)$ is a continuous-time stable and proper real-rational transfer function representing the nominal closed-loop system. $\Delta(s)$ is a continuous-time block-diagonal operator:

$$\Delta(s) = \text{diag}(\Delta_1(s), \ldots, \Delta_N(s)) \tag{A.1}$$

which gathers all model uncertainties. Each $\Delta_i(s)$ can be:

- either a time-invariant diagonal matrix $\Delta_i(s) = \delta_i I_{n_i}$, where δ_i is a real or a complex parametric uncertainty, and I_{n_i} is the $n_i \times n_i$ identity matrix,
- or a stable and proper real-rational unstructured transfer function of size $n_i \times n_i$ usually representing neglected dynamics.

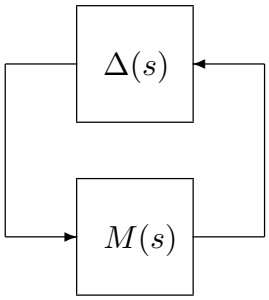


Figure A.1: Standard interconnection for robustness analysis

Let $n = \sum_{i=1}^{N} n_i$. The set of all $n \times n$ matrices with the same block-diagonal structure and the same nature (real or complex) as $\Delta(j\omega)$ is denoted by $\mathbf{\Delta}$. Let then $k\mathcal{B}_{\mathbf{\Delta}} = \{\Delta \in \mathbf{\Delta} : \overline{\sigma}(\Delta) < k\}$, where $\overline{\sigma}(\Delta)$ denotes the largest singular value of Δ. To simplify notation, $\Delta(s) \in \mathbf{\Delta}$ is used to specify that $\Delta(j\omega) \in \mathbf{\Delta}$ for all $\omega \in \Omega$, where Ω denotes the frequency range of interest (equal to $\mathbb{R}_+$ in the sequel without loss of generality). $\Delta(s) \in k\mathcal{B}_{\mathbf{\Delta}}$ means in addition that $\|\Delta(s)\|_\infty < k$. In other words, $\mathbf{\Delta}$ is the set of all admissible uncertainties, whose size is measured in terms of the H_∞ norm. In the sequel, the expression *real problem* is used when $\Delta(s)$ is only composed of *real blocks* corresponding to real parametric uncertainties. Conversely, the expression *complex problem* is used when $\Delta(s)$ is only composed of unstructured transfer functions, which become *complex blocks* when evaluated at a particular frequency.

The most efficient technique to analyze the stability of the interconnection of Figure A.1 in the H_∞ framework is certainly **μ-analysis** [Doyle, 1982], especially when high-dimensional systems are considered. The underlying theory [Ferreres, 1999; Zhou *et al.*, 1996] is not broached as such in this document due to space limitations, but a few useful definitions and results are recalled below.

Definition A.1 (structured singular value) *Let $\omega \in \mathbb{R}_+$ be a given frequency. If no matrix $\Delta \in \mathbf{\Delta}$ makes $I - M(j\omega)\Delta$ singular, then the structured singular value $\mu_{\mathbf{\Delta}}(M(j\omega))$ is equal to zero. Otherwise:*

$$\mu_{\mathbf{\Delta}}(M(j\omega)) = \left[\min_{\Delta \in \mathbf{\Delta}} \{\overline{\sigma}(\Delta),\, \det(I - M(j\omega)\Delta) = 0\} \right]^{-1} \tag{A.2}$$

Theorem A.1 (small gain theorem for structured uncertainties) *The interconnection of Figure A.1 is stable for all uncertainties $\Delta(s) \in k\mathcal{B}_{\mathbf{\Delta}}$ if and only if:*

$$\sup_{\omega \in \mathbb{R}_+} \mu_{\mathbf{\Delta}}(M(j\omega)) \leq \frac{1}{k} \tag{A.3}$$

Definition A.2 (robust stability margin) *The robust stability margin k_r is defined as the largest value of k for which equation (A.3) holds. In other words, k_r is the inverse of the largest value of $\mu_{\mathbf{\Delta}}(M(j\omega))$ over the whole frequency range:*

$$k_r = \Big[\sup_{\omega \in \mathbb{R}_+} \mu_{\mathbf{\Delta}}(M(j\omega)) \Big]^{-1} \tag{A.4}$$

The following conclusions can be drawn from the small gain theorem:

- the interconnection is stable for all admissible uncertainties $\Delta(s)$ of size less than k_r,
- there exists at least one admissible uncertainty $\tilde{\Delta}(s)$ of size k_r for which the interconnection is unstable.

Problem A.1 (robust stability margin) *With reference to Figure A.1, compute the robust stability margin k_r defined in equation (A.4) for a given block structure $\mathbf{\Delta}$.*

The exact computation of k_r is known to be NP hard in the general case [Braatz *et al.*, 1994], so both lower and upper bounds $\underline{k_r}$ and $\overline{k_r}$ are computed instead. But even computing these bounds is a challenging problem with an infinite number of frequency-domain constraints, since it requires to compute lower and upper bounds $\underline{\mu}_{\mathbf{\Delta}}(M(j\omega))$ and $\overline{\mu}_{\mathbf{\Delta}}(M(j\omega))$ on $\mu_{\mathbf{\Delta}}(M(j\omega))$ for each $\omega \in \mathbb{R}_+$. On the one hand, a wide number of very different approaches exist to compute upper bounds on k_r (*i.e.* μ lower bounds). They are summarized and compared in Section A.2 on a wide set of real-world benchmarks. A strategy is also proposed to combine them, so as to improve accuracy while keeping a reasonable computational time. On the other hand, most of the computationally tractable methods to compute lower bounds on k_r (*i.e.* μ upper bounds) rely on the so-called D and G scaling matrices. The problem is usually solved on a finite frequency grid with the risk to miss a critical frequency and to over-evaluate k_r. An algorithm is presented in Section A.3 to address this issue, and applied to the aforementioned benchmarks. Unfortunately, the gap between the bounds on k_r is sometimes large and concluding about robust stability is not always straightforward. Assume, for instance, that the uncertainties are normalized. Stability has to be investigated $\forall \Delta(s) \in \mathcal{B}_{\mathbf{\Delta}}$ and is guaranteed if and only if $k_r > 1$. But if $\underline{k_r} < 1$ and $\overline{k_r} > 1$, no conclusion can be drawn. Strategies can then be implemented to tighten the gap between the bounds, among which branch-and-bound seems to be the most effective one, as shown in Section A.4.

2 μ lower bound computation

2.1 Survey of existing methods

This section is exhaustive in the sense that all methods that can reasonably be applied to real-world benchmarks are mentioned. Only a few exponential-time algorithms are omitted, as well as techniques with a very limited range of application. In all cases, both μ lower bounds and associated destabilizing values of the uncertainties are obtained. For most methods, frequency

is fixed during optimization. In this case, $M(j\omega)$ and $\Delta(j\omega)$ are constant matrices, which are simply denoted by M and Δ.

Exponential-time methods

The first methods to compute μ lower bounds were proposed in the 1980's and most of them are exponential-time. One of the best known was introduced in [De Gaston and Safonov, 1988] for non-repeated ($n_i = 1$) real uncertainties and later generalized in [Sideris and Sanchez Pena, 1990]. It uses the mapping theorem of [Zadeh and Desoer, 1963]: the image of $k\mathcal{B}_{\mathbf{\Delta}}$ by the operator $\Delta \to \det(I - \Delta M)$ is included into the convex hull of the images of the 2^N vertices of $k\mathcal{B}_{\mathbf{\Delta}}$. A μ lower bound is obtained by considering the position of these images with respect to the origin in the complex plane. Like the mapping theorem, the Kharitonov's theorem [Barmish and Kang, 1993] and the edge theorem [Bartlett *et al.*, 1988] can also be interpreted in a frequency domain approach and used to derive μ lower bound algorithms. These are not detailed here, but a few interesting references can be found in [Sideris and Sanchez Pena, 1990].

Then, the algebraic method proposed in [Dailey, 1990] for non-repeated real uncertainties searches for a destabilizing uncertainty $\Delta \in k\mathcal{B}_{\mathbf{\Delta}}$ such that all the δ_i except 2 attain the maximal magnitude k, which can be achieved by simple matrix algebra operations. A one-dimensional halving search on k is performed to compute a μ lower bound. There are $N(N-1)/2$ ways to choose 2 uncertainties among N and 2^{N-2} ways to set each of the $N-2$ others to $-k$ or k, so $N(N-1)2^{N-3}$ searches are performed for each value of k. The method presented in [Elgersma *et al.*, 1996] is quite similar. It is based on the computation of the smallest common real root of a system of linear polynomials using the Sylvester's theorem [van der Waerden, 1991]. The main difference with [Dailey, 1990] is that both k and the two δ_i which do not attain the maximal magnitude are computed in a single step. The optimal value of k is thus determined directly and no halving search is required.

More recently, a method was presented in [Matsuda *et al.*, 2009] for (possibly repeated) real uncertainties. It consists of introducing $b = \prod_{i=1}^{N}(n_i + 1) - 1$ fictitious uncertainties $\overline{\delta}_1, \ldots, \overline{\delta}_b$ and rewriting the numerator of $\det(I - M(s)\Delta)$ as $f(s) = \hat{p}(s) + \left[\overline{\delta}_1 \ldots \overline{\delta}_b\right] [p_1(s) \ldots p_b(s)]^T$, where $\hat{p}(s), p_1(s), \ldots, p_b(s)$ are fixed real polynomials. The stability of $f(s)$ is then evaluated using the stability feeler introduced in [Matsuda and Mori, 2009], which leads to μ upper and lower bounds.

The algorithms mentioned in this section have different theoretical foundations. But they are all exponential-time, which means that they cannot be applied to most real-world benchmarks with a reasonable computational time. Thus, only polynomial-time algorithms are considered in the sequel.

Power algorithm

This technique can be used for all kinds of uncertainties. It is based on the following characterization [Young and Doyle, 1997]:

$$\mu_{\mathbf{\Delta}}(M) = \max_{Q \in \mathcal{Q}} \rho_R(QM) \tag{A.5}$$

where $\rho_R(QM)$ is the largest magnitude of the real eigenvalues of QM, while $\mathcal{Q}$ is the set of all $\Delta = \mathrm{diag}(\Delta_1, \ldots, \Delta_N) \in \mathbf{\Delta}$ such that $\delta_i \in [-1, 1]$ if $\Delta_i = \delta_i I_{n_i}$ is real and $\Delta_i^* \Delta_i = I_{n_i}$ if Δ_i is complex. Equation (A.5) defines a non-concave optimization problem, which is solved in [Young and Doyle, 1997] using a fixed point iteration usually referred to as the power algorithm. No theoretical convergence guarantee exists. But if the algorithm converges, a local maximum, *i.e.* a μ lower bound, is obtained. In practice, convergence is usually ensured for purely complex

and mixed real/complex uncertainties. But in the purely real case, limit cycles often appear and no μ lower bound can be obtained. Nevertheless, a technique is proposed in [Young *et al.*, 1995] to get a nontrivial μ lower bound when the power algorithm does not converge but returns nonzero values. Further improvements are also reported in [Tierno and Young, 1992; Cheng and De Moor, 1994; Newlin and Glavaski, 1995].

A different way to solve problem (A.5) is proposed in [Dehaene *et al.*, 1997] in the particular case of purely complex uncertainties (ρ_R is then replaced with the spectral radius ρ). The steepest ascent and the conjugate gradient algorithms are used instead of a fixed-point iteration. Unlike the power algorithm, which only provides a μ lower bound in case of convergence, a nontrivial bound is obtained at each iteration at the price of a higher computational cost.

Gain-based algorithm

This method was introduced in [Seiler *et al.*, 2010] to address the convergence issues of the power algorithm when purely real uncertainties are considered. The initial μ problem is replaced with a series of worst-case H_∞ performance problems, which are efficiently solved using the approach of [Packard *et al.*, 2000]. In the mixed real/complex case, the real blocks of the destabilizing uncertainties are obtained using the aforementioned technique, while the complex blocks are computed using the power algorithm described above.

Poles migration techniques

These techniques are mainly dedicated to purely real problems, although they can in some cases be extended to handle mixed and complex uncertainties. The idea is to move an eigenvalue of the interconnection between $M(s)$ and Δ towards the imaginary axis, which requires to work in the time domain. The state matrix of the interconnection of Figure A.1 is $A_0 = A + B\Delta(I - D\Delta)^{-1}C$, where (A, B, C, D) is a state-space representation of $M(s)$. An equivalent definition of the robust stability margin (A.4) can thus be derived:

$$k_r = \min_{\Delta \in \mathbf{\Delta}} \{\overline{\sigma}(\Delta),\, \lambda_{max}(A_0) = 0\} \tag{A.6}$$

where $\lambda_{max}(A_0) = \max_i \Re(\lambda_i(A_0))$ and $\lambda_i(A_0)$ denotes the ith eigenvalue of A_0.

A first approach is to use a first-order characterization of the variation $d\lambda_i$ of $\lambda_i(A_0)$ caused by a small variation $d\Delta$ of Δ. This can be expressed as $d\lambda_i = (uB+tD)d\Delta(Cv+Dw)$, where u/v are the left/right eigenvectors associated to $\lambda_i(A_0)$, $t = uB\Delta(I-D\Delta)^{-1}$ and $w = \Delta(I-D\Delta)^{-1}Cv$. A series of perturbations $d\Delta$ are computed, which progressively move the eigenvalues of A_0 towards the imaginary axis. Two algorithms exist. On the one hand, a series of quadratic programming problems are solved in [Magni *et al.*, 1999] for each $\lambda_i(A_0)$. The uncertainty with minimum Froebenius norm which brings $\lambda_i(A_0)$ on the imaginary axis is first determined, and then the one with minimum $\overline{\sigma}$ norm such that the interconnection remains at the limit of stability. On the other hand, [Ferreres and Biannic, 2001] notes that the power algorithm of [Young and Doyle, 1997] is quite fast, but often suffers from convergence problems when purely real uncertainties are considered. A three-step procedure is proposed in this case. The initial problem is first regularized by adding a small amount $\epsilon \ll 1$ of complex uncertainty, *i.e.* Δ is replaced with $\Delta + \epsilon\Delta_C$, where Δ_C has the same structure as Δ but can take on complex values [Packard and Pandey, 1993]. The power algorithm is then run at each point of a rough frequency grid, usually with good convergence properties. Using the resulting uncertainties as an initialization, a series or linear programming problems are finally solved, so as to find the value $\hat{\Delta}$ of Δ with minimum $\overline{\sigma}$ norm for which one of the eigenvalues of the initial interconnection becomes unstable. The resulting μ lower bound is equal to $\overline{\sigma}(\hat{\Delta})^{-1}$.

More recently, a direct approach has been proposed by [Iordanov and Halton, 2015; Iordanov *et al.*, 2003]. It considers the optimization problem:

$$\min_{\Delta \in \mathbf{\Delta}} \overline{\sigma}(\Delta) \quad \text{such that} \quad \lambda_{max}(A_0) = 0 \tag{A.7}$$

which can be recast as a sequential quadratic programming problem and solved for example using the Matlab Optimization Toolbox [The Mathworks, 2017a]. In practice, a relaxed condition is considered to avoid convergence issues: the equality constraint is replaced with $|\lambda_{max}(A_0)| \leq \epsilon$, where ϵ is a small user-defined threshold. Problem (A.7) is non-convex and the result thus strongly depends on the initial value of Δ. Several strategies are combined in [Iordanov and Halton, 2015] to improve the resulting μ lower bounds.

Even more recently, problem (A.7) was considered again in [Apkarian *et al.*, 2016] and solved this time with a nonsmooth bundle trust-region algorithm.

Direct optimization-based techniques

These algorithms are directly inspired by the definition of μ given in (A.2). A first approach consists of directly solving the following non-convex optimization problem using standard nonlinear optimization tools such as the `fmincon` function of the Matlab Optimization Toolbox:

$$\min_{\Delta \in \mathbf{\Delta}} \overline{\sigma}(\Delta) \quad \text{such that} \quad \det(I - M\Delta) = 0 \tag{A.8}$$

In practice, a relaxed condition is considered to avoid convergence issues: the equality constraint is replaced with $\underline{\sigma}(I - M\Delta) \leq \epsilon$ in [Hayes *et al.*, 2001; Bates and Mannchen, 2004], and with $|\det(I - M\Delta)| \leq \epsilon$ in [Halton *et al.*, 2008], where ϵ is a small user-defined threshold. Any kind of uncertainties can be considered, but this approach is especially relevant for purely real problems, since the number of decision variables significantly increases when complex uncertainties are considered.

A formulation similar to (A.8) is considered in [Yazıcı *et al.*, 2011] in the case of (possibly repeated) real uncertainties:

$$\min_{\substack{\Delta \in \mathbf{\Delta} \\ \lambda_1, \lambda_2 \in \mathbb{R}}} \overline{\sigma}(\Delta) + (\lambda_1^q + \lambda_2^q)\lambda \quad \text{such that} \quad \begin{cases} \Re(\det(I - M\Delta)) = \lambda_1^p \\ \Im(\det(I - M\Delta)) = \lambda_2^p \end{cases} \tag{A.9}$$

where p and q are odd and even positive integers respectively, and λ is a large penalty parameter. This optimization problem is solved using the modified subgradient algorithm based on feasible values (F-MSG) introduced in [Kasimbeyli *et al.*, 2009].

Another variation is proposed in [Brito and Kim, 2010]. The following optimization problem is first solved for different values of k using the `fmincon` function of the Optimization Toolbox:

$$g(k) = \min_{\Delta \in k\mathcal{B}_{\mathbf{\Delta}}} \Re\left(\det(I - M\Delta)\right) \quad \text{such that} \quad \left|\Im\left(\det(I - M\Delta)\right)\right| \leq \epsilon \tag{A.10}$$

where ϵ is a user-defined threshold. A function g is thus defined. A modification of the Newton-Raphson method called the secant method is then applied to determine the smallest value $\underline{k}$ of k such that $g(k) = 0$. A μ lower bound is finally obtained as the inverse of $\underline{k}$.

A common feature of the previous optimization-based algorithms is that they all use standard nonlinear optimization tools, although the objective to be minimized is a nonsmooth function. Breakdowns might thus be encountered at points that are not local optima, because the latter are typically nonsmooth points in practice. In this context, [Lemos *et al.*, 2014] proposes a nonsmooth optimization technique to solve (A.8), whose convergence to a local minimum is ensured.

Geometrical approach

This method dedicated to (possibly repeated) real uncertainties combines randomization and nonlinear optimization [Kim *et al.*, 2009]. The signs of the real and the imaginary parts of $\det(I - M\Delta)$ are computed for randomly selected points on the surface of a given hyperbox in the uncertainty space $\mathbb{R}^N$. This hyperbox is enlarged until the four possible sign combinations are found, which means that it might contain values of $\delta_1, \ldots, \delta_N$ such that $\det(I - M\Delta) = 0$. A series of contractions and expansions are then performed to approach the singular region until the size of the box becomes smaller than some tolerance value.

Remark A.1 *Except for the poles migration techniques, a classical strategy is to compute μ lower bounds on a predefined* ***frequency grid****. An alternative is to compute bounds on a set of* ***frequency intervals*** *whose union covers the whole frequency range. This can be achieved by considering frequency as an additional uncertainty δ_ω. An augmented interconnection $\tilde{M} - \tilde{\Delta}$ is obtained, where $\tilde{\Delta} = diag(\delta_\omega I_m, \Delta)$, m is the state dimension of $M(s)$, and the frequency interval of interest is covered when $\delta_\omega \in \mathbb{R}$ varies between -1 and 1 (see e.g. [Halton* et al.*, 2008; Lemos* et al.*, 2014]). A skew-μ problem (see Section A.5.2) is then to be solved, since δ_ω is bounded.*

2.2 Testing framework

Considered algorithms

The most relevant μ lower bound algorithms are summarized in Table A.1 and compared in Section A.2.3. The first two are implemented in the function `mussv` of the Robust Control Toolbox for Matlab [The Mathworks, 2017b]. The fourth one is implemented in the SMAC Toolbox for Matlab (see [Roos, 2013] and Section A.6). For the other five algorithms, Matlab functions provided by the respective authors are used. It is worth being emphasized that each of the eight considered algorithms is called in a similar fashion for all considered benchmarks.

Algorithm	Description	Reference	Admissible uncertainties
1	Power algorithm	[Young and Doyle, 1997]	all
2	Gain-based algorithm	[Seiler *et al.*, 2010]	real & mixed real/complex
3	Poles migration technique	[Magni *et al.*, 1999]	all
4	Poles migration technique	[Ferreres and Biannic, 2001]	real
5	Poles migration technique	[Iordanov and Halton, 2015]	all
6	Direct nonlinear optimization	[Halton *et al.*, 2008]	all
7	Direct nonsmooth optimization	[Lemos *et al.*, 2014]	all
8	Geometrical approach	[Kim *et al.*, 2009]	real

Table A.1: Considered μ lower bound algorithms

To compute the best possible upper bounds on k_r, grid-based methods (1-2-6-7-8) are applied at each point of a 100-point frequency grid, composed of 50 logarithmically-spaced points within the system bandwidth and 50 additional points used to refine the grid in some frequency regions corresponding to weakly damped modes. Interval-based implementations are available for methods 6 and 7 (see Remark A.1). For these algorithms, the system bandwidth is divided into 6 and 4 frequency intervals of equal size on a logarithmic scale respectively. A coarse 10-point frequency grid is used for method 4. Finally, methods 3 and 5 do not require any frequency grid.

List of benchmarks

As shown in Table A.2, 36 challenging benchmarks are considered here, corresponding to various fields of application, system dimensions and structures of the uncertainties. Some of them contain poorly damped modes, which often produce extremely sharp peaks on the μ plot, while others are characterized by large state vectors as well as numerous and/or highly repeated uncertainties. Purely real uncertainties are majority (benchmarks 1-29), since they are more common in engineering problems than complex or mixed ones (benchmarks 30-36). Moreover, the presence of complex uncertainties usually simplifies the computation of μ upper and lower bounds, making them of less interest in the present study. All these benchmarks are described in the litterature and the associated references are provided in [Roos and Biannic, 2015]. A complete Matlab implementation is also available at `http://w3.onera.fr/smac/smart_bench`.

Benchmark	**Description**	**Number of states**	**Uncertainty block Δ**	
			Size	**Structure**
1	Academic example	5	1	1×1
2	Academic example	4	3	3×1
3	Academic example	4	4	2×2
4	Inverted pendulum	4	3	3×1
5	Anti-aliasing filter	2	5	3×1 + 1×2
6	DC motor	4	5	3×1 + 1×2
7	Bus steering system	9	5	1×2 + 1×3
8	Satellite	9	4	2×1 + 1×2
9	Bank-to-turn missile	6	4	4×1
10	Aeronautical vehicle	8	4	4×1
11	Four-tank system	10	4	4×1
12	Re-entry vehicle	6	8	1×2 + 2×3
13	Missile	14	4	4×1
14	Cassini spacecraft	17	4	4×1
15	Mass-spring-damper	7	6	6×1
16	Spark ignition engine	4	7	7×1
17	Hydraulic servo system	8	8	8×1
18	Academic example	41	5	3×1 + 1×2
19	Drive-by-wire vehicle	4	16	2×1 + 7×2
20	Re-entry vehicle	7	13	3×1 + 1×4 + 1×6
21	Space shuttle	34	9	9×1
22	Rigid aircraft	9	14	14×1
23	Fighter aircraft	10	27	7×1 + 1×2 + 1×3 + 1×15
24	Flexible aircraft	46	20	20×1
25	Telescope mockup	70	20	20×1
26	Hard disk drive	29	27	19×1 + 4×2
27	Launcher	30	45	16×1 + 10×2 + 1×3 + 1×6
28	Helicopter	12	120	4×30
29	Biochemical network	7	507	13×39
30	Himat fighter aircraft	16	4	2×2(c)
31	F14 fighter aircraft	52	8	1×2(c) + 1×6(c)
32	DC motor	4	6	3×1 + 1×2 + 1×1(c)
33	Four-tank system	12	6	4×1 + 1×2(c)
34	Missile	19	6	4×1 + 2×1(c)
35	Hydraulic servo system	9	9	8×1 + 1×1(c)
36	Space shuttle	46	18	9×1 + 1×9(c)

The notation m×p in the last column means that Δ contains m blocks of size p×p.
All blocks are real unless (c) is specified. All real/complex blocks are diagonal/full.

Table A.2: List of benchmarks

2.3 Numerical results

Computations were performed in 2014 using Matlab R2010b on a Windows 7 Workstation with a CPU Intel Xenon W3530 running at 2.8 GHz and 6 GB of RAM. Computational time would certainly be lower with a state-of-the-art computer, but this is not an issue here. Indeed, the main objective is to compare the algorithms, and it should be emphasized that all tests have been performed with the same computer. More detailed numerical results can be found in [Roos and Biannic, 2015].

Purely real problems

All algorithms are applied to benchmarks 1-29 and results are summarized in Table A.3, where $\underset{\sim}{\mu}_{\boldsymbol{\Delta}}$ and $\underline{\mu}_{\boldsymbol{\Delta}}$ denote the highest μ lower bound (*i.e.* the lowest upper bound on k_r) for the considered algorithm and the highest μ lower bound out of all algorithms respectively. The notations (g) and (i) denote the grid-based and the interval-based implementations of the algorithms (see Remark A.1). The mean computational time is computed for each algorithm over all benchmarks except the biochemical network (benchmark 29), which is significantly more complicated than the others and requires a high computational effort.

Algorithm	**Number of benchs for which $(\underline{\mu}_{\Delta}-\underset{\sim}{\mu}_{\Delta})/\underline{\mu}_{\Delta}$ is**					**Mean value of**	**Mean CPU time**
	=0%	**≤1%**	**≤5%**	**≤25%**	**<100%**	**$(\underline{\mu}_{\Delta}-\underset{\sim}{\mu}_{\Delta})/\underline{\mu}_{\Delta}$**	
1	6	6	9	13	25	44.10%	1.7 s
2	2	8	19	24	29	11.87%	27.1 s
3	5	6	16	22	27	18.81%	1.0 s
4	26	26	27	29	29	0.88%	0.9 s
5	22	25	26	26	29	8.95%	22.8 s
6 (g)	5	13	18	24	29	11.09%	448.8 s
6 (i)	9	18	20	23	29	15.72%	97.8 s
7 (g)	8	15	17	23	27	17.44%	1694.9 s
7 (i)	24	25	25	25	29	9.36%	124.3 s
8	0	4	8	17	28	24.38%	749.4 s

Table A.3: Synthetic results for purely real problems (benchmarks 1-29)

The poles migration technique of [Ferreres and Biannic, 2001] is the most efficient algorithm, with the highest accuracy (less than 1% on average for all 29 benchmarks) and also the lowest computational time (less than 1 second on average for benchmarks 1-28 and 47 seconds for the biochemical network). The poles migration technique of [Iordanov and Halton, 2015] gives very good results too for all except 3 benchmarks, and the computational time remains reasonable (but still 20 times higher). This shows that poles migration is probably the most efficient way to compute accurate μ lower bounds for purely real problems. Moreover, an attractive feature of this approach is that frequency is not fixed, which allows to detect critical frequencies corresponding to peak values of μ more easily. The gain-based algorithm of [Seiler *et al.*, 2010], the direct nonlinear optimization technique (grid-based version) of [Halton *et al.*, 2008] and the direct nonsmooth optimization technique (interval-based version) of [Lemos *et al.*, 2014] also give quite satisfactory results in most cases, since the accuracy is about 10% on average. Nevertheless, these algorithms are much slower (especially the last two). This is partly explained by the fact that they compute a μ lower bound as a function of frequency, whereas the poles migration techniques focus on the peak values. Note also that the nonsmooth technique gives the best results in most cases but sometimes proves very conservative, whereas the nonlinear one usually produces bounds which are not the best ones but yet good ones. Moreover,

interval-based versions allow to drastically reduce the computational time, but do not bring any significant improvement in terms of accuracy. Then, the poles migration technique of [Magni *et al.*, 1999] exhibits both a reasonable accuracy (less than 20% on average for all 29 benchmarks) and a very low computational time (about 1.0 seconds on average for benchmarks 1-28 and only 63 seconds for the biochemical network). Finally, the power algorithm of [Young and Doyle, 1997] often suffers from convergence problems, especially for frequencies corresponding to peak values of $\mu_{\mathbf{\Delta}}(M(j\omega))$. This is in stark contrast to the complex and mixed cases for which the convergence properties are excellent (see below).

Purely complex and mixed real/complex problems

The poles migration technique of [Ferreres and Biannic, 2001] and the geometrical approach of [Kim *et al.*, 2009] cannot be applied in the presence of complex uncertainties. Moreover, the gain-based algorithm of [Seiler *et al.*, 2010] is not relevant when purely complex uncertainties are considered, since it reduces to the power algorithm of [Young and Doyle, 1997], and no improvement over the power algorithm is observed for the considered mixed real/complex problems. Therefore, these methods are not considered in Table A.4.

Algorithm	**Number of benchs for which $(\underline{\mu}_{\mathbf{\Delta}} - \underline{\tilde{\mu}}_{\mathbf{\Delta}})/\underline{\mu}_{\mathbf{\Delta}}$ is**					**Mean value of $(\underline{\mu}_{\mathbf{\Delta}} - \underline{\tilde{\mu}}_{\mathbf{\Delta}})/\underline{\mu}_{\mathbf{\Delta}}$**	**Mean CPU time**
	=0%	**≤1%**	**≤5%**	**≤25%**	**<100%**		
1	4	7	7	7	7	0.10%	1.1 s
3	0	2	4	6	7	16.60%	0.9 s
5	2	2	2	3	4	57.12%	140.8 s
6 (g)	1	6	7	7	7	0.34%	2648.8 s
6 (i)	0	4	5	7	7	4.26%	874.2 s
7 (g)	4	4	4	6	7	10.63%	3972.6 s
7 (i)	2	4	5	6	7	7.91%	249.5 s

Table A.4: Synthetic results for purely complex & mixed real/complex problems (benchmarks 30-36)

The power algorithm of [Young and Doyle, 1997] is the most efficient algorithm, with the highest accuracy and almost the lowest computational time. This confirms its good convergence properties when mixed or complex uncertaintes are considered. The direct nonlinear optimization technique (grid-based version) of [Halton *et al.*, 2008] also gives very accurate results, but the mean computational time is 2400 times higher. More generally, optimization-based techniques (algorithms 5-6-7) are not attractive in terms of computational time, since full complex blocks generate a large number of optimization variables ($2n_i^2$ variables for a $n_i \times n_i$ block).

2.4 Results improvement

Purely real problems

The results presented in Section A.2.3 show that the best μ lower bounds are obtained for 26 purely real benchmarks out of 29 with the poles migration technique of [Ferreres and Biannic, 2001]. Moreover, the bounds computed for the other 3 benchmarks (20/26/29) are not very far from the highest values over all algorithms. Improving the poles migration technique to obtain the best results in all cases does not seem to be a trivial issue. A more promising idea is to combine different algorithms to get the most out of them. Apart from the poles migration techniques, the most efficient methods are the gain-based algorithm of [Seiler *et al.*, 2010] and the global optimization tools of [Halton *et al.*, 2008] and [Lemos *et al.*, 2014]. The following three-step strategy is thus proposed in [Roos and Biannic, 2015]:

1. The poles migration technique of [Ferreres and Biannic, 2001] (algorithm 4) is executed first as explained in Section A.2.1.

2. The gain-based algorithm of [Seiler *et al.*, 2010] (algorithm 2) is then executed for a few selected frequencies only using the previous results as an initialization. These frequencies can be those for which the uncertain system becomes unstable after algorithm 4 is applied or if a single uncertainty is considered. The frequency for which the lower bound on k_r (*i.e.* the highest μ upper bound) is obtained is also a good candidate.

3. A global optimization tool is finally applied using the previous results as an initialization. Particle swarm optimization is implemented here, since it offers a bunch of tuning parameters (numbers of swarms, of particles in each swarm, of topologies...), which allows to easily handle the trade-off between accuracy and computational time.

The improved μ lower bounds so obtained for benchmarks 20/26/29 are shown in Table A.5. They are compared with the bounds computed in Section A.2.3 by independently applying all algorithms one at a time.

Benchmark	Algorithm 4 only		Other algorithms one at a time			Proposed combination	
	value	time	best value	time	algorithm	value	time
20	0.9380	0.5 s	0.9947	41.6 s	7	0.9947	5.5 s
26	0.9881	2.2 s	1.2134	184.9 s	7	1.2144	18.0 s
29	724.15	46.9 s	733.86	2864.8 s	2	753.10	580.0 s

Table A.5: Improved μ lower bounds for purely real benchmarks

The best upper bound on k_r is obtained in all cases with the proposed combination. The computational time is of course larger than if algorithm 4 is applied alone. Nevertheless, it remains much smaller than with algorithms 2 and 7, which produced the best results so far for benchmarks 20/26/29. The bound is even improved for benchmarks 26 and 29, for which a single application of any algorithm cannot bring the best value. The proposed strategy is implemented in the SMART Library of the SMAC Toolbox (see [Roos, 2013] and Section A.6).

Remark A.2 *The method of [Apkarian* et al.*, 2016] was published after our comparative study was conducted. It is therefore not evaluated in this work. Nevertheless, it was applied by the authors to benchmarks 1-29 and results are very consistent with the strategy proposed in [Roos and Biannic, 2015]. It confirms once again that solving problem (A.7) is probably the best way to compute accurate upper bounds on k_r, i.e. μ lower bounds.*

Purely complex and mixed real/complex problems

The results presented in Section A.2.3 show that the power algorithm of [Young and Doyle, 1997] is the most relevant technique for purely complex and mixed real/complex problems. Nevertheless, the resulting upper bounds on k_r are conservative in some cases: the power algorithm is slightly outperformed by other techniques for benchmarks 33/34/35. A careful investigation of several benchmarks shows that the main reasons for this are threefold:

- The power algorithm is applied at each point of a fixed frequency grid. Unless by some stroke of luck, the critical frequency corresponding to the exact value of k_r is usually not part of this grid, even if the latter is very tight.

- The power algorithm almost always converges in the presence of complex uncertainties, but it sometimes requires a quite large number of iterations.
- Optimization problem (A.5) is non-concave. Thus, the accuracy of the resulting μ lower bound strongly depends on the way the power algorithm is initialized.

In this context, the following strategy introduced in [Roos and Biannic, 2015] can be implemented to improve the results of Section A.2.3. First, the power algorithm of [Young and Doyle, 1997] is classically applied at each point of a fixed frequency grid. In most cases, the μ plot does not exhibit very sharp peaks in the presence of complex uncertainties. Thus, a reasonably rough grid (*e.g.* 20 frequency points) is usually enough to capture approximately the main peaks. Then, the grid is gradually tightened around the frequencies of these peaks until improvement in the μ lower bound becomes marginal. This strategy allows to address the first two issues highlighted before. Indeed, the grid becomes very tight in the most critical frequency regions and the frequency corresponding to the exact value of k_r can be detected with a very good accuracy. Moreover, repeatedly applying the power algorithm with a limited number of iterations (*e.g.* 100 iterations) at very close frequencies is roughly equivalent to applying it once with a large number of iterations, provided that it is initialized every time with the result obtained at the previous frequency. Finally, in order to address the third issue, the power algorithm is not only applied with the latter initialization, but also with one or more random initializations. The improved μ lower bounds so obtained for benchmarks 33/34/35 are shown in Table A.6. They are compared with the ones computed in Section A.2.3 using the power algorithm of `mussv` on a fixed frequency grid, but also the other available algorithms.

Benchmark	Standard power algo		Other algorithms one at a time			Improved power algo	
	value	time	best value	time	algorithm	value	time
33	0.4346	0.7 s	0.4362	1380.1	7 (g)	0.4362	0.6 s
34	0.9587	1.1 s	0.9604	565.8	7 (g)	0.9606	1.3 s
35	0.9910	1.7 s	0.9927	12.3	5	0.9927	2.1 s

Table A.6: Improved μ lower bounds for purely complex & mixed real/complex benchmarks

The best upper bound on k_r is obtained in all cases with the proposed strategy. The mean computational time over all purely complex & mixed real/complex benchmarks is 1.1 s. It is exactly the same as with the standard call to `mussv` (see Table A.4). Indeed, the initial frequency grid is quite rough and it is only tightened in a few frequency intervals. Moreover, the number of power iterations performed at each frequency is quite low. The total computational time thus remains very reasonable. The enhanced call to the power algorithm is implemented in the SMART Library of the SMAC Toolbox (see [Roos, 2013] and Section A.6).

Remark A.3 *Several techniques have been proposed to improve the convergence properties of the power algorithm in case of purely real problems [Tierno and Young, 1992; Cheng and De Moor, 1994; Newlin and Glavaski, 1995]. But convergence is usually not an issue in the presence of complex uncertainties. These techniques are thus of little help in the present context.*

2.5 Conclusion

The main contribution of this section is to provide a thorough comparative analysis of all practical methods to compute upper bounds on the robust stability margin k_r, *i.e.* lower bounds on the structured singular value μ. It appears that the tradeoff between accuracy and computational time is best handled by the poles migration technique of [Ferreres and Biannic, 2001] for

purely real problems and by the power algorithm of [Young and Doyle, 1997] for purely complex and mixed real/complex problems. Based on these conclusions, simple improvements and combinations are proposed to further improve the bounds with a reasonable computational effort. The resulting Matlab tools are evaluated on a wide set of 36 real-world benchmarks available in the literature. Results are conclusive, since the best bounds are obtained in all cases. The next step is to compute lower bounds on k_r, *i.e.* guaranteed μ upper bounds on the whole frequency range.

3 μ upper bound computation

3.1 Classical (D, G)-scalings formulation

Most of the algorithms which can reasonably be applied to real-world benchmarks are based on the following result [Young *et al.*, 1995; Fan *et al.*, 1991]:

Theorem A.2 *Let M be a complex matrix and $\beta > 0$. If there exist matrices $D \in \mathcal{D}$ and $G \in \mathcal{G}$ which satisfy one of the following relations:*

$$M^*DM + j(GM - M^*G) \leq \beta^2 D \tag{A.11}$$

$$\overline{\sigma}\left((I+G^2)^{-\frac{1}{4}}\left(\frac{DMD^{-1}}{\beta} - jG\right)(I+G^2)^{-\frac{1}{4}}\right) \leq 1 \tag{A.12}$$

where $\mathcal{D} = \{D = D^* > 0 : \forall \Delta \in \mathbf{\Delta},\ D\Delta = \Delta D\}$ and $\mathcal{G} = \{G = G^* : \forall \Delta \in \mathbf{\Delta},\ G\Delta = \Delta^* G\}$, then $\mu_{\mathbf{\Delta}}(M) \leq \beta$. The problem of minimizing β can be solved either optimally using an LMI solver or faster but suboptimally using a gradient descent algorithm.

Computing a lower bound on k_r, *i.e.* an upper bound on $\mu_{\mathbf{\Delta}}(M(j\omega))$ on the whole frequency range $\mathbb{R}_+$, is a challenging problem with an infinite number of frequency-domain constraints and optimization variables, since Theorem A.2 must be applied for each $\omega \in \mathbb{R}_+$. In practice, it is usually solved on a finite frequency grid only. However, a crucial problem appears in this procedure: the grid must contain the frequency for which the maximal value of $\mu_{\mathbf{\Delta}}(M(j\omega))$ is reached. If not, the resulting lower bound on k_r can be over-evaluated, *i.e.* be larger than the real value of k_r. Unfortunately, the aforementioned critical frequency is usually unknown!

To overcome this issue, several approaches have been proposed to compute μ upper bounds, which are **guaranteed** on the whole frequency range. One of the first solutions was introduced in [Sideris, 1992] and then further investigated in [Helmersson, 1995; Ferreres and M'Saad, 1994; Ferreres *et al.*, 1996; Ferreres and Fromion, 1997; Young, 2001; Halton *et al.*, 2008]. It consists of considering the frequency as an additional parametric uncertainty δ_ω. An augmented operator $\tilde{\Delta}(s) = diag(\delta_\omega I_m,\ \Delta(s))$ is considered, where m is the order of $M(s)$. A recursive application of Theorem A.2 then provides a guaranteed lower bound on k_r. Alternatively, the problem can be solved by computing a single skew-μ upper bound, which is computationally cheaper and can be achieved using a rather straightforward generalization of Theorem A.2 (see Section A.5.2). Unfortunately, this approach is intractable for high-order systems. Indeed, the number of decision variables associated to $\delta_\omega I_m$ in the D and G matrices increases quadratically with m, and it can be quite large in real-world applications.

Another approach was proposed in [Ferreres *et al.*, 2003] and improved in [Biannic and Ferreres, 2005; Roos and Biannic, 2010]. A μ upper bound β and associated matrices D and G are first computed for a given frequency $\tilde{\omega}$ by setting $M = M(j\tilde{\omega})$ in Theorem A.2. It is then slightly increased, *i.e.* $\beta \leftarrow (1+\epsilon)\beta$, so as to enforce a strict inequality:

$$\overline{\sigma}\left((I+G^2)^{-\frac{1}{4}}\left(\frac{DM(j\tilde{\omega})D^{-1}}{\beta} - jG\right)(I+G^2)^{-\frac{1}{4}}\right) < 1 \tag{A.13}$$

The objective is then to compute the largest frequency interval I_v containing $\tilde{\omega}$, for which the increased upper bound β and the associated matrices D and G remain valid, *i.e.* such that $\forall \omega \in I_v$:

$$\overline{\sigma}\left((I+G^2)^{-\frac{1}{4}}\left(\frac{DM(j\omega)D^{-1}}{\beta}-jG\right)(I+G^2)^{-\frac{1}{4}}\right)\leq 1 \tag{A.14}$$

Proposition A.1 shows that the determination of I_v boils down to an eigenvalues computation.

Proposition A.1 *Let (A_M, B_M, C_M, D_M) be a state-space representation of $M(s)$. Build the Hamiltonian-like matrix:*

$$\mathcal{H}=\begin{bmatrix} A_H & 0 \\ -C_H{}^*C_H & -A_H{}^* \end{bmatrix}+\begin{bmatrix} B_H \\ -C_H{}^*D_H \end{bmatrix} X \begin{bmatrix} D_H{}^*C_H & B_H{}^* \end{bmatrix} \tag{A.15}$$

where $X=(I-D_H{}^*D_H)^{-1}$ and:

$$\begin{bmatrix} A_H & B_H \\ C_H & D_H \end{bmatrix}=\begin{bmatrix} I & 0 \\ 0 & \dfrac{(I+G^2)^{-1/4}}{\sqrt{\beta}} \end{bmatrix}\begin{bmatrix} A_M-j\tilde{\omega}I & B_MD^{-1} \\ DC_M & DD_MD^{-1}-j\beta G \end{bmatrix}\begin{bmatrix} I & 0 \\ 0 & \dfrac{(I+G^2)^{-1/4}}{\sqrt{\beta}} \end{bmatrix}$$

Define $\delta_{\tilde{\omega}_-}$ and $\delta_{\tilde{\omega}_+}$ as follows:

$$\begin{aligned} \delta_{\tilde{\omega}_-} &= \max\{\lambda\in\mathbb{R}_-:\det(\lambda I+j\mathcal{H})=0\} \\ &= -\tilde{\omega} \text{ if } j\mathcal{H} \text{ has no positive real eigenvalue} \end{aligned}$$

$$\begin{aligned} \delta_{\tilde{\omega}_+} &= \min\{\lambda\in\mathbb{R}_+:\det(\lambda I+j\mathcal{H})=0\} \\ &= \infty \text{ if } j\mathcal{H} \text{ has no negative real eigenvalue} \end{aligned}$$

Then inequality (A.14) holds $\forall \omega \in I_v$ where:

$$I_v=[\tilde{\omega}+\delta_{\tilde{\omega}_-}\ ,\ \tilde{\omega}+\delta_{\tilde{\omega}_+}] \tag{A.16}$$

An iterative algorithm is finally implemented, which mainly consists of repeatedly applying the previous strategy to a list of intervals. A guaranteed μ upper bound, *i.e.* a lower bound on k_r, is obtained as soon as the union of all intervals covers the whole frequency range.

Algorithm A.1 (computation of a lower bound on k_r)

1. *Initialization:*
   (a) *Compute a μ lower bound β_{max} (see Section A.2.4).*
   (b) *Let $\mathcal{I}=\{\mathbb{R}_+\}$ be the initial list of frequency intervals to be investigated.*
2. *While $\mathcal{I}\neq\emptyset$, repeat:*
   (a) *Choose an interval $I\in\mathcal{I}$ and a frequency $\tilde{\omega}\in I$.*
   (b) *Compute the minimum value of β and the associated matrices D and G such that inequality (A.12) holds with $M=M(j\tilde{\omega})$.*
   (c) *Set $\beta\leftarrow\max((1+\epsilon)\beta,\beta_{max})$. Apply Proposition A.1 to compute I_v.*
   (d) *Set $\beta_{max}\leftarrow\beta$. Update the intervals in $\mathcal{I}$ by eliminating the frequencies contained in I_v.*
3. *A guaranteed lower bound on k_r is given by $1/\beta_{max}$.*

The proposed algorithm is implemented in the SMART Library of SMAC Toolbox (see [Roos, 2013] and Section A.6). It is not based on a frequency grid to be defined a priori, with the risk of missing a critical frequency. On the contrary, the list of frequency intervals is updated automatically during the iterations, and no tricky initialization is required.

Remark A.4 *A fairly similar approach is proposed in [Lawrence* et al.*, 2000] and implemented in the function* `munorm` *of the Subroutine Library in Systems and Control Theory (SLICOT) for Matlab.*

3.2 Numerical results

A lower bound $\underline{k_r}$ on k_r is computed using Algorithm A.1 for each of the 36 benchmarks listed in Table A.2. The same computer is used as in Section A.2.3. For the moment, the objective is to get the most accurate results, *i.e.* to ensure that the gap with respect to the best upper bound $\overline{k_r}$ on k_r computed in Section A.2.3 is as low as possible. Computational time is not an issue here and an LMI solver is used in step 2(b) of Algorithm A.1 whenever possible, *i.e.* when the number of repetitions of the uncertanties is not too large.

Purely real problems

For 19 benchmarks out of 29, the gap between $\underline{k_r}$ and $\overline{k_r}$ is less than 0.01%, which means that the exact value of k_r is obtained. For the other 10 benchmarks, the gap is less than 10% in 4 cases, between 20% and 40% in 5 cases, and up to 215% for benchmark 27 (which has 28 uncertainties, 12 of them being repeated). The mean value of the gap over all benchmarks is 12.71%.

Purely complex and mixed real/complex problems

For 6 benchmarks out of 7, the gap between $\underline{k_r}$ and $\overline{k_r}$ is less than 0.01%, which means that the exact value of k_r is obtained. For the last one, the gap is 2.68%. The mean value of the gap over all benchmarks is 0.39%.

3.3 Use of the μ-sensitivities

The gap between the bounds on k_r can have several causes. The most obvious ones are mentioned below:

1. The μ lower bound algorithms are conservative and they only provide an upper bound on k_r,
2. A suboptimal result is obtained when the minimization of β in step 2(b) of Algorithm A.1 is performed with a gradient descent algorithm instead of an LMI solver,
3. The sufficient characterizations of Theorem A.2 are not necessary, and they only provide a lower bound on k_r.

In other words, is it the lower or the upper bound on k_r which is responsible for this gap? Although there is no theoretical proof, it is observed in practice that the best k_r upper bound obtained using the strategies proposed in Section A.2.4 almost always equals the exact value of k_r. This will be highlighted in the next sections. So item 1 above is not really an issue. In contrast, two solutions are proposed in Sections A.3.4 and A.3.5 to address items 2 and 3 respectively, and to improve the accuracy of the μ upper bound. They both use the μ-sensitivities. The latter were introduced in [Braatz and Morari, 1991], and they allow to quantify the extent to which a μ upper bound $\overline{\mu}_{\mathbf{\Delta}}$ changes when a single uncertainty slightly varies in $\mathbf{\Delta}$.

Definition A.3 *Let $M = M(j\omega)$ denote the frequency response of $M(s)$ at a given frequency ω. Pre- and post-multiply Δ by a diagonal matrix $A(\epsilon_j) = \mathrm{diag}(I_{n_1}, \ldots, (1-\epsilon_j)I_{n_j}, \ldots, I_{n_N})$, so that the jth uncertainty is multiplied by $(1-\epsilon_j)^2$ while the others stay at their nominal values. Let then $M_{\epsilon_j} = A(\epsilon_j)MA(\epsilon_j)$ (note that $M_0 = M$). The jth μ-sensitivity is defined as:*

$$S_j^\mu = \frac{\partial}{\partial \epsilon_j} \overline{\mu}_{\mathbf{\Delta}}(M_{\epsilon_j}) = \lim_{\epsilon_j \to 0^+} \frac{|\overline{\mu}_{\mathbf{\Delta}}(M) - \overline{\mu}_{\mathbf{\Delta}}(M_{\epsilon_j})|}{\epsilon_j}$$

Using the above formulation is numerically sensitive and the algorithm proposed in [Douglas and Athans, 1995] is preferred here. The μ-sensitivities can be very useful to identify which uncertainties have the greatest influence on stability. This can be interesting when the μ upper bound computation is based on solving large LMI problems. In such a case, only selected parts of the LMIs corresponding to the uncertainties with the largest μ-sensitivities are solved with a dedicated solver, while the others are solved using suboptimal and faster methods. This is illustrated for the (D, G)-scalings based formulation of Theorem A.2 in Section A.3.4, and for the more general and less conservative multiplier based formulation of [Fu and Barabanov, 1997] in Section A.3.5. Branch-and-bound techniques can also benefit from this approach by iteratively cutting the uncertainty domain along the edges with the largest μ-sensitivities only, as described in Section A.4.

3.4 Partial LMI optimization of the scaling matrices

As already emphasized, the μ upper bound of [Young *et al.*, 1995; Fan *et al.*, 1991] can be computed either with an LMI solver or with a gradient descent algorithm. The first approach allows to compute the minimum value of β such that (A.12) is satisfied, at the price of a large (and sometimes dissuasive) computational effort in case of highly repeated uncertainties ($n_i \gg 1$). In contrast, the second approach only gives a suboptimal result in most cases, but it is much faster. In order to master the numerical complexity while preserving the accuracy of the LMI-based technique, step 2(b) of Algorithm A.1 is adapted as follows. Initial values of D and G are first obtained using the gradient descent algorithm. Selected parts of these matrices are then re-optimized using an LMI solver, based on the μ-sensitivities computation and on a user-defined maximum number of decision variables n_{opt}, as described in [Lesprier *et al.*, 2015c].

Remark A.5 *The parameter n_{opt} allows to handle the tradeoff between accuracy and computational time at step 2(b) of Algorithm A.1. A small value makes the algorithm faster but does not significantly increase the lower bound $\underline{k_r}^{\mathrm{GD}}$ on k_r obtained with the gradient descent method. On the contrary, a large value increases the computational time but makes the lower bound on k_r tend to the optimal value $\underline{k_r}^{\mathrm{LMI}}$ obtained with the full LMI optimization approach.*

This technique is especially useful when both the gap between $\underline{k_r}^{\mathrm{LMI}}$ and $\underline{k_r}^{\mathrm{GD}}$ and the computational time required to compute $\underline{k_r}^{\mathrm{LMI}}$ are large. The results obtained on selected benchmarks satisfying these conditions are summarized in Table A.7, where $\underline{k_r}^{\mathrm{pLMI}}$ denotes the lower bound on k_r computed using the aforementioned procedure. The maximum number of decision variables is set to n_{opt} =80, so that the computation time to apply Algorithm A.1 is reasonable (ideally around 30 seconds). It can be observed that the gap between $\underline{k_r}^{\mathrm{LMI}}$ and $\underline{k_r}^{\mathrm{pLMI}}$ is usually quite low (third column) compared to the initial gap between $\underline{k_r}^{\mathrm{LMI}}$ and $\underline{k_r}^{\mathrm{GD}}$ (fifth column), and that the computational time remains quite reasonable. Note that the last benchmark is not part of the list of Table A.2. It corresponds to a flexible satellite model with 95 states and a 34×34 Δ operator composed of 15 parametric uncertainties with the following structure: 3×1+8×2+1×3+3×4. For this benchmark, the gap is still high, but the improvement is nonetheless substantial since

the initial gap between $\underline{k_r}^{\text{LMI}}$ and $\underline{k_r}^{\text{GD}}$ is very large. These results prove the capability of the method to compute an upper bound close to $\underline{k_r}^{\text{LMI}}$ with a computation time close to the gradient descent algorithm.

Benchmark	$\underline{k_r}^{\text{LMI}}$	$\underline{k_r}^{\text{pLMI}}$		$\underline{k_r}^{\text{GD}}$	
	time	**gap w.r.t $\underline{k_r}^{\text{LMI}}$**	**time**	**gap w.r.t $\underline{k_r}^{\text{LMI}}$**	**time**
19	26s	0.43%	17 s	12.56%	5 s
20	122 s	14.60%	30 s	66.52%	4 s
25	63 s	0.01%	20 s	20.41%	9 s
26	306 s	0.57%	30 s	20.47%	6 s
27	3546 s	26.58%	100 s	63.37%	36 s
-	2555 s	97.34%	49 s	230.48%	22 s

Table A.7: Results for partial LMI optimization

3.5 Multiplier-based μ upper bound

The previous section deals with the case where the gap between $\underline{k_r}^{\text{LMI}}$ and $\underline{k_r}^{\text{GD}}$ is large. The case where the gap between k_r and $\underline{k_r}^{\text{LMI}}$ is large is now considered. In this perspective, the multiplier based μ upper bound introduced in [Fu and Barabanov, 1997] is first reviewed, which is a generalization of the (D,G)-scalings based μ upper bound of [Young *et al.*, 1995]. The main results are based on the following lemma.

Lemma A.1 *A matrix family $\mathcal{A}$ is nonsingular if there exists another matrix C (called a multiplier) such that* He $CA < 0$ *for all $A \in \mathcal{A}$, where* He X *is the hermitian part of the matrix X.*

In this section, it is assumed for simplicity reasons but without loss of generality that Δ is only composed of parametric uncertainties, *i.e.* $\Delta = \text{diag}(\delta_1 I_{n_1}, \ldots, \delta_N I_{n_N})$. This assumption is realistic anyway, since the gap between k_r and $\underline{k_r}^{\text{LMI}}$ is usually quite low as soon as there is at least one complex block in Δ. The following theorem derives from Lemma A.1 and Definition A.1.

Theorem A.3 *[Fu and Barabanov, 1997] Let M be a complex matrix and $\alpha > 0$. If there exists a multiplier $C \in \mathbb{C}^{n\times n}$ such that:*

$$\text{He } C^* (I_n - \Delta M) < 0 \tag{A.17}$$

for all $\Delta \in \frac{1}{\alpha}\mathcal{B}_{\mathbf{\Delta}}$, then $\mu_{\mathbf{\Delta}}(M) \leq \alpha$.

Equation (A.17) is linear in δ_i, so $\mathcal{B}_{\mathbf{\Delta}}$ can be equivalently replaced with $\overline{\mathcal{B}}_{\mathbf{\Delta}} = \{\Delta \in \mathbf{\Delta} : \delta_i = \pm 1\}$. The aim is thus to find the minimum value of $\alpha > 0$ such that (A.17) holds for each of the 2^N matrices $\Delta \in (1/\alpha)\overline{\mathcal{B}}_{\mathbf{\Delta}}$, which can be formulated as a GEVP (Generalized EigenValue minimization Problem):

$$\begin{aligned} \nu_{\mathbf{\Delta}}(M) = & \min_{\alpha\in\mathbb{R}_+, C\in\mathbb{C}^{n\times n}} \alpha \\ \text{subject to } & \text{He } C^* (\alpha I_n - \Delta M) < 0, \ \ \forall \Delta \in \overline{\mathcal{B}}_{\mathbf{\Delta}} \end{aligned} \tag{A.18}$$

where $\nu_{\mathbf{\Delta}}(M)$ is called the multiplier based μ upper bound. If M has the following structure:

$$M = AB^* \tag{A.19}$$

where $A, B \in \mathbb{C}^{n\times q}$ and $q = \text{rank}(M) \leq n$, then (A.17) can be equivalently replaced with:

$$\text{He } \hat{C}^* (I_q - B^* \Delta A) < 0 \tag{A.20}$$

where $\hat{C} \in \mathbb{C}^{q\times q}$. Hence, if M does not have full rank, solving (A.20) is obviously advantageous since the size of the multiplier is $q \times q$, rather than $n \times n$ in (A.17). The following theorem then shows that $\nu_{\mathbf{\Delta}}(M)$ is less conservative than the (D,G)-scalings based upper bound $\overline{\mu}^{LMI}_{\mathbf{\Delta}}(M)$.

Theorem A.4 *(adapted from [Fu and Barabanov, 1997]) For every structure* $\mathbf{\Delta}$ *and matrix* M, $\nu_{\mathbf{\Delta}}(M) \leq \overline{\mu}^{LMI}_{\mathbf{\Delta}}(M)$.

The multiplier-based method is thus attractive, since it allows to get a more accurate μ upper bound. Nevertheless, it suffers from two main drawbacks:

1. This is an exponential-time method. Indeed, the GEVP (A.18) involves 2^N LMIs, which can be computationally very costly if the number of uncertainties is large.
2. It only provides a μ upper bound at a given frequency. Thus, a lower bound on the robust stability margin k_r is usually computed on a frequency grid and can be over-evaluated, as already emphasized in Section A.3.1.

A way to address the first issue is to solve (A.18) only for a limited number of uncertainties $r < N$ and to use the (D,G)-scalings based formulation for the $N-r$ others. The obtained upper bound $\nu^r_{\mathbf{\Delta}}(M)$ lies between $\nu_{\mathbf{\Delta}}(M)$ and $\overline{\mu}^{LMI}_{\mathbf{\Delta}}(M)$, yielding a slightly looser but more computationally-friendly bound than $\nu_{\mathbf{\Delta}}(M)$. Moreover, the μ-sensitivities can be used to select the r uncertainties of Δ with the greatest influence on stability, as explained in Section A.3.3. M and Δ are thus reordered, so that $\Delta = \mathrm{diag}(\Delta_1, \Delta_2)$, where $\Delta_1 \in \mathbf{\Delta_1} \subset \mathbb{R}^{p\times p}$ contains the selected r blocks ($p = \sum_{i=1}^{r} n_i$) and Δ_2 the other $N-r$ blocks. Let $D \in \mathcal{D}_2$ and $G \in \mathcal{G}_2$ be the scaling matrices associated to Δ_2, as defined in Theorem A.2. A rank decomposition $M = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \begin{bmatrix} B_1^* & B_2^* \end{bmatrix}$ is also performed to reduce the number of decision variables, where $A_1, B_1 \in \mathbb{C}^{p\times q}$, $A_2, B_2 \in \mathbb{C}^{(n-p)\times q}$ and $q = \mathrm{rank}(M)$. The following problem must now be solved:

$$\nu^r_{\mathbf{\Delta}}(M) = \min_{\alpha\in\mathbb{R}_+, \hat{C}\in\mathbb{C}^{q\times q}, D\in\mathcal{D}_2, G\in\mathcal{G}_2} \alpha$$

$$\text{subject to } \begin{bmatrix} L_{11} & \hat{C}^* B_2^* + j A_2^* G \\ B_2\hat{C} - jGA_2 & -\alpha^2 D \end{bmatrix} < 0, \ \ \forall \Delta_1 \in \frac{1}{\alpha}\overline{\mathcal{B}}_{\mathbf{\Delta_1}} \tag{A.21}$$

$$\text{where } L_{11} = 2\mathrm{He}\, \hat{C}^*(I_q - B_1^*\Delta_1 A_1) + A_2^* D A_2$$

With this new formulation there are only 2^r LMIs to solve instead of 2^N, and thanks to the μ-sensitivities, only the most relevant ones are considered. Note that the solution to problem (A.21) is also a solution to the initial problem (A.20), as shown below.

Theorem A.5 *(adapted from [Fu and Barabanov, 1997]) Suppose that there exist* $\alpha > 0$ *and* $\hat{C} \in \mathbb{C}^{q\times q}$ *such that (A.21) is satisfied. Then* $\mathrm{He}\, \hat{C}^*(I_q - B^*\Delta A) < 0$ *for all* $\Delta \in (1/\alpha)\overline{\mathcal{B}}_{\mathbf{\Delta}}$.

Let us now focus on the second issue. A method is proposed to determine on which frequency intervals the multiplier C (or $\hat{C}$) computed at one particular frequency remains valid. It is based on the following theorem.

Theorem A.6 *[Lesprier* et al.*, 2015b] Let* (A_M, B_M, C_M, D_M) *be a state-space representation of* $M(s)$. *Let* $\alpha > 0$, $\Delta \in \frac{1}{\alpha}\overline{\mathcal{B}}_{\mathbf{\Delta}}$ *and* $C \in \mathbb{C}^{n\times n}$ *such that* $X = 2\mathrm{He}\, C^*(I_n - \Delta D_M)$ *is nonsingular. Then* $\mathrm{He}\, C^*(I_n - \Delta M(j\omega))$ *is singular if and only if* $j\omega$ *is an eigenvalue of the following Hamiltonian matrix:*

$$\mathcal{H} = \begin{bmatrix} A_M & 0 \\ 0 & -A_M^* \end{bmatrix} + \begin{bmatrix} B_M \\ C_M^* \Delta C \end{bmatrix} X^{-1} \begin{bmatrix} C^*\Delta C_M & -B_M^* \end{bmatrix}$$

Theorem A.6 can be used to compute the list of frequency intervals $\mathcal{I}_v$ on which:

$$\text{He } C^*(I_n - \Delta M(j\omega)) < 0 \tag{A.22}$$

i.e. on which the multiplier C remains valid for a given uncertainty matrix $\Delta \in \frac{1}{\alpha}\overline{\mathcal{B}}_{\mathbf{\Delta}}$. Let $j\omega_1, \ldots, j\omega_k$ be the purely imaginary eigenvalues of H such that $0 < \omega_1 < \ldots < \omega_k$. It is obvious that $[0\ \omega_1] \cup [\omega_1\ \omega_2] \cup \ldots \cup [\omega_{k-1}\ \omega_k] \cup [\omega_k\ +\infty[= \mathbb{R}_+$. So it suffices to pick a frequency ω_t inside each of these $k+1$ intervals and to test whether He $C^*(I_n - \Delta M(j\omega_t)) < 0$. $\mathcal{I}_v$ is composed of all the invervals for which this condition is satisfied.

Remark A.6 *The nonsingularity condition on X in Theorem A.6 is not restrictive in practice. Indeed, the feedthrough matrix D_M is generally a null matrix when real systems are considered. In this case, X is always nonsingular, since $X = 2\text{He } C^* < 0$. Moreover, several benchmarks have been studied, for which D_M was not zero, and the condition on X was always satisfied.*

Theorem A.6 is only valid for a multiplier C with full rank n. For the case where a multiplier $\hat{C} \in \mathbb{C}^{q\times q}$, $q < n$, is computed (e.g. using (A.20) or (A.21)), the following theorem is applied first to derive a valid full-rank multiplier $C \in \mathbb{C}^{n\times n}$ so that Theorem A.6 can then be used.

Theorem A.7 *[Lesprier* et al.*, 2015b] Let $M = M(j\omega)$ denote the frequency response of $M(s)$ at a given frequency ω. Consider the reduced-rank structure $M = AB^*$, where $A, B \in \mathbb{C}^{n\times q}$ and $q = \text{rank}(M)$. Assume that there exist $\hat{C} \in \mathbb{C}^{q\times q}$ and $\alpha > 0$ such that:*

$$\text{He } \hat{C}^*(I_q - B^*\Delta A) < 0, \ \ \forall \Delta \in \frac{1}{\alpha}\overline{\mathcal{B}}_{\mathbf{\Delta}} \tag{A.23}$$

Let $C = B\hat{C}B^* - \epsilon EE^* \in \mathbb{C}^{n\times n}$, where $\epsilon > 0$ is small enough and $E \in \mathbb{C}^{n\times(n-q)}$ is such that $\begin{bmatrix} B & E \end{bmatrix}$ has full rank. Then:

$$\text{He } C^*(I_n - \Delta M) < 0, \ \ \forall \Delta \in \frac{1}{\alpha}\overline{\mathcal{B}}_{\mathbf{\Delta}} \tag{A.24}$$

Remark A.7 *Numerically speaking, a full-rank multiplier C is thus quickly computed from $\hat{C}$ using the above theorem. Indeed it suffices to choose a small enough ϵ and a random complex matrix E such that $\begin{bmatrix} B & E \end{bmatrix}$ has full rank. Note that the latter assumption is not restrictive and that a suitable E can be easily obtained.*

The following algorithm is now proposed to compute a guaranteed multiplier based μ upper bound on the whole frequency range $\mathbb{R}_+$, *i.e.* a lower bound on k_r.

Algorithm A.2 (computation of a lower bound on k_r)

1. *Initialization:*
   (a) *Compute a μ lower bound α_{max} (see Section A.2.4).*
   (b) *Let $\mathcal{I} = \{\mathbb{R}_+\}$ be the initial list of frequency intervals to be investigated.*
   (c) *Define the number r of uncertainties with the highest μ-sensitivities to be put in Δ_1.*
2. *While $\mathcal{I} \neq \emptyset$, repeat:*
   (a) *Choose an interval $I \in \mathcal{I}$ and a frequency $\tilde{\omega} \in I$.*
   (b) *Compute the minimum value of α and the associated matrices $\hat{C} \in \mathbb{C}^{q\times q}$, D and G such that inequality (A.21) holds with $M = M(j\tilde{\omega})$.*

(c) *If $q < n$, apply Theorem A.7 to find the corresponding full-rank multiplier $C \in \mathbb{C}^{n\times n}$. Otherwise, set $C = \hat{C}$.*

(d) *Set $\alpha \leftarrow \max(\alpha, \alpha_{max})$. Apply Theorem A.6 for each uncertainty matrix $\Delta_j \in (1/\alpha)\overline{\mathcal{B}_{\mathbf{\Delta}}}$ to compute the list of frequency intervals $\mathcal{I}_{v_j}$ on which (A.22) holds.*

(e) *Set $\alpha_{max} \leftarrow \alpha$. Update the intervals in $\mathcal{I}$ by eliminating the frequencies contained in $\bigcap_j \mathcal{I}_{v_j}$.*

3. *A guaranteed lower bound on k_r is given by $1/\alpha_{max}$.*

The results obtained on several benchmarks are summarized in Table A.8. These benchmarks are interesting, since the (D,G)-scalings based approach of [Young *et al.*, 1995] does not give accurate enough μ upper bounds. Algorithm A.2 is thus applied to reduce conservatism. $\underline{k_r}^{\text{LMI}}$ and $\underline{k_r}^{\text{MULT}}$ denote the lower bounds on k_r obtained with Algorithm A.1 (using an LMI solver to compute μ upper bounds at step 2(b)) and Algorithm A.2 respectively. $\overline{k_r}$ is the best upper bound on k_r computed in Section A.2.4 among all existing algorithms.

Benchmark	$\underline{k_r}^{\text{LMI}}$ **(Algorithm A.1)**		$\underline{k_r}^{\text{MULT}}$ **(Algorithm A.2)**			
	gap w.r.t. $\overline{k_r}$	**time**	r	**gap w.r.t.** $\overline{k_r}$	**time**	**iterations**
3	41.03%	1 s	2	1.23%	42 s	27
10	27.59%	6 s	2	14.20%	120 s	50
			4	0.09%	477 s	89
22	24.46%	11 s	5	12.88%	133 s	4
			9	5.34%	1190 s	6

Table A.8: Comparison between the (D,G)-scalings and the multiplier based approaches

Conservatism is drastically reduced with a reasonable computation time. Note that for benchmark 22, the use of the reduced-rank structure of M greatly decreases the computational burden. This property can be exploited for many realistic systems. It is also interesting to notice that the number of iterations to validate the multiplier based μ upper bound on the whole frequency range is reasonable, and even very small for benchmark 22. On the contrary, solving the problem on a frequency grid would require to consider a significantly larger number of frequencies to get reasonably accurate results. This would imply a much larger computation time in addition to an impossibility to ensure that no critical frequency has been missed.

Finally, 50 runs of Algorithm A.2 are performed on benchmarks 10 and 22. At each iteration, the r uncertainty blocks handled by the multiplier-based μ upper bound are randomly chosen at step 2(b) instead of being selected using the μ-sensitivities. The corresponding results are presented in Table A.9. The latter displays the minimum, the mean and the maximum values of the gap between $\underline{k_r}^{\text{MULT}}$ and $\overline{k_r}$, and compares them with the gap previously obtained in Table A.8. This clearly shows the relevance of using the μ-sensitivities.

Benchmark	r	**Gap between** $\underline{k_r}^{\text{MULT}}$ **and** $\overline{k_r}$	
		using the μ-sensitivities	**min/mean/max values (50 runs)**
10	2	14.20%	20.55% / 20.84% / 20.93%
22	5	12.88%	10.35% / 18.27% / 22.89%
	9	5.34%	5.34% / 11.49% / 17.88%

Table A.9: Results showing the relevance of using the μ-sensitivities

Remark A.8 *The method of [Fu and Barabanov, 1997] was extended in [Chen and Sugie, 1997] by considering a parameter dependent multiplier $C(\Delta)$. The resulting μ upper bounds are necessarily less conservative than the ones computed using a constant multiplier C, but the LMI problem to be solved becomes more demanding. This extension is thus restricted to small-size problems and has not been considered in this work.*

3.6 Conclusion

A practical method is first proposed to compute lower bounds on the robust stability margin k_r, *i.e.* upper bounds on the structured singular value μ. Unlike classical grid-based methods, the results are guaranteed on the whole frequency range, and there is no risk of missing a critical frequency. Moreover, the algorithm is computationally efficient and can be applied to high-order models with highly repeated parametric uncertainties. The resulting Matlab tools are evaluated on the same set of benchmarks as in Section A.2. The exact value of k_r is obtained in 25 cases out of 36, and the mean gap over all benchmarks between the bounds on k_r is only 10%. Some strategies are then proposed to improve the accuracy of the lower bounds on k_r, while keeping a reasonable computational time. Firstly, the μ-sensitivities are used to identify the uncertainties, which have the greatest influence on stability. When the classical (D, G)-scalings based μ upper bound is computed, the corresponding blocks of the matrices D and G are optimized with an LMI solver, while the other ones are determined using a suboptimal and faster method. Secondly, the multiplier-based μ upper bound of [Fu and Barabanov, 1997] is revisited. An algorithm is proposed to compute a guaranteed robust stability margin on the whole frequency range, and not only on a finite frequency grid. To alleviate the computational cost, the μ-sensitivities are also used to decide whether each uncertainty is treated with the more accurate multiplier based characterization or the cheaper (D, G)-scalings based one. Significantly less conservative μ upper bounds are obtained for many benchmarks, but computational time becomes prohibitive for the most complicated ones. The next step is to further tighten the gap between lower and upper bounds on k_r using branch-and-bound techniques, so as to approach the exact value in all cases and with a moderate computational effort.

Remark A.9 *Several other techniques exist to compute μ upper bounds (see e.g. [De Gaston and Safonov, 1988; Ly* et al.*, 1998; Lee and Edgar, 2003; Scorletti* et al.*, 2007]). It could be worthwhile to perform a detailed comparison of all existing algorithms and to propose efficient combinations to get the most out of them. This would probably further improve the results reported in Section A.3.*

4 Branch-and-bound to reduce conservatism

4.1 Standard algorithm

Conservatism is defined here as the relative gap η between the lower and the upper bounds on the robust stability margin:

$$\eta = \frac{\overline{k_r} - \underline{k_r}}{\underline{k_r}} \tag{A.25}$$

Despite the efforts reported in Sections A.2 and A.3 to improve the bounds on k_r, η sometimes reaches unacceptable values, notably in the presence of highly repeated real parametric uncertainties (see Section A.3.2). To overcome this issue, a branch-and-bound algorithm can be used to ensure that conservatism remains below a specified threshold η_{tol} [Balemi *et al.*, 1991; Newlin and Young, 1997]. The idea is to partition the real parametric domain in more and more

subsets until the relative gap between the highest lower bound and the highest upper bound computed on all subsets becomes less than η_{tol}. This algorithm is known to converge for uncertain systems with only real parametric uncertainties [Newlin and Young, 1997], *i.e.* η can be reduced to an arbitrarily small value. However, it suffers from an exponential growth of computational complexity as a function of the number of real uncertainties. The choice of η_{tol} thus allows to handle the tradeoff between accuracy and computational time.

A standard version of this branch-and-bound algorithm is first applied to the benchmarks listed in Table A.2. Upper bounds on k_r are computed using the strategies described in Section A.2.4, and lower bounds on k_r are obtained using Algorithm A.1. Algorithm A.2 could have been used for some benchmarks to compute better lower bounds, but it has not been considered here, since it cannot be applied to all benchmarks with a reasonable computational time.

Purely real problems

For 19 benchmarks out of 29, the gap between $\underline{k_r}$ and $\overline{k_r}$ is already less than 0.01%, so no further improvement is needed (see Section A.3.2). For the other 10 benchmarks, branch-and-bound allows to increase $\underline{k_r}$, leading to a gap less than 0.01% in 7 cases, and of 0.12%, 1.99% and 10.00% in the last 3 cases. The mean value of the gap over all 29 benchmarks is reduced from 12.71% to only 0.42%. And the exact value of k_r is (almost) obtained in all but one case.

Purely complex and mixed real/complex problems

For 6 benchmarks out of 7, the gap between $\underline{k_r}$ and $\overline{k_r}$ is already less than 0.01%, so no further improvement is needed (see Section A.3.2). For the last one, branch-and-bound allows to increase $\underline{k_r}$, leading to a gap less than 0.01%. The mean value of the gap over all 29 benchmarks is reduced from 0.39% to less than 0.01%. And the exact value of k_r is obtained in all cases.

Remark A.10 *In all cases where branch-and-bound is applied, $\underline{k_r}$ is improved while $\overline{k_r}$ is not. Although there is no theoretical proof, this shows that the k_r upper bounds obtained in Section A.2.4 almost always equal the exact value of k_r, i.e. that the best μ lower bound algorithms are almost always non-conservative. This justifies the claim of Section A.3.3.*

However, despite these good results, computational complexity grows exponentially as a function of the number N of uncertainties. Thus, computing tight lower bounds on k_r can be extremely long (it can take hours and even days for some benchmarks).

4.2 Improved algorithm

Two solutions are proposed in this section to improve the computational time, while maintaining the same level of accuracy. A strategy based on the progressive validation of the frequency range is first proposed in [Roos *et al.*, 2011]. Assume for example that a subset $\mathcal{S}$ and a frequency domain Ω are considered at step k of the branch-and-bound scheme. Algorithm A.1 is applied to compute a validated frequency domain $\Omega_v \subset \Omega$ such that $\det(I - M(j\omega)\Delta) \neq 0$ holds $\forall \Delta \in \mathcal{S}$ and $\forall \omega \in \Omega_v$. During the next step, the analysis performed on each subset of $\mathcal{S}$ only considers the frequencies $\Omega_{nv} = \Omega \setminus \Omega_v$ which have not been validated at step k. Consequently, after a few steps, the analysis is restricted to very narrow frequency intervals corresponding to critical frequencies. This results in a drastic reduction of the computational load induced by a classical branch-and-bound procedure.

It is then proposed to use the μ-sensitivities to cut the uncertainty domain in a relevant way, so as to reduce the number of iterations and to decrease the computational time. The following algorithm is introduced in [Lesprier *et al.*, 2015c]:

Algorithm A.3 (branch-and-bound using the μ-sensitivities)

1. *Initialization:*
   (a) *Choose the maximum allowable gap η between the lower and the upper bounds on k_r.*
   (b) *Compute an upper bound $\overline{k_r}$ on k_r (see Section A.2.4). Let the candidate lower bound k_r^{test} on k_r be defined as $\overline{k_r} = (1+\eta)k_r^{\text{test}}$.*
   (c) *Let $\mathcal{E} = \{(\mathcal{S}_0, \Omega_0)\}$ be the initial list of elements to be tested, where $\mathcal{S}_0 = k_r^{\text{test}}\mathcal{B}_{\mathbf{\Delta}}$ and $\Omega_0 = \mathbb{R}_+$.*
2. *While $\mathcal{E} \neq \emptyset$:*
   (a) *Take an element $(\mathcal{S}, \Omega)$ from $\mathcal{E}$.*
   (b) *Compute $\tilde{M}(s)$ such that the interconnections $(M(s), \Delta \in \mathcal{S})$ and $(\tilde{M}(s),\ \Delta \in \mathcal{B}_{\mathbf{\Delta}})$ are equivalent.*
   (c) *Apply Algorithm A.1 to determine the frequencies $\Omega_v \subset \Omega$ such that the μ upper bound is less than 1 for the normalized interconnection $(\tilde{M}(s), \Delta \in \mathcal{B}_{\mathbf{\Delta}})$ computed at step 2(b).*
   (d) *If $\Omega_v \neq \Omega$, compute the μ-sensitivities for a frequency $\omega \in \Omega_{nv} = \Omega \setminus \Omega_v$. Partition $\mathcal{S}$ along the edge with the largest μ-sensitivity to get two subdomains $\mathcal{S}_1$ and $\mathcal{S}_2$. Add $(\mathcal{S}_1, \Omega_{nv})$ and $(\mathcal{S}_2, \Omega_{nv})$ to $\mathcal{E}$.*

Remark A.11 *A new upper bound on k_r, i.e. a μ lower bound, is also computed occasionally, for example every 10 or 20 iterations. This is enough, since upper bounds on k_r are usually very close or even equal to k_r, as highlighted in Section A.4.1. If the new upper bound is smaller, k_r^{test} is updated as described in step 1(b) of Algorithm A.3 and the domain $\mathcal{S}$ associated to each element of $\mathcal{E}$ is adjusted accordingly.*

4.3 Numerical results

Algorithm A.3 is applied to all benchmarks for which the gap between the lower and the upper bounds on k_r is larger than 0.1%. The final results obtained for all 36 benchmarks are summarized below:

- For 26 benchmarks out of 36, the computational time is negligible: less than 5 s for a gap less than 0.1%.
- For 8 benchmarks, the computational time is low: between 8 s and 80 s for a gap less than 1%, and between 10 s and 158 s for a gap less than 0.1%.
- Only 2 benchmarks (among the most complex ones with many uncertainties, some of them being highly repeated) require a larger but still reasonable computational effort:
  - benchmark 27 (launcher model with 30 states and a 45×45 Δ matrix): 9 min 30 s for a gap less than 1%, much longer for a gap less than 0.1%.
  - benchmark 29 (biochemical network with 7 states and a 507×507 Δ matrix): 22 min 30 s for a gap less than 10%, 3 h 04 min for a gap less than 5%, 14 h 54 min for a gap less than 1%, much longer for a gap less than 0.1%.

Therefore, **computing the exact value of the robust stability margin k_r, *i.e.* the exact value of the structured singular value μ, is no longer an issue.** Beyond this rather provocative claim stands a conclusion supported by 20 years of research, software development and practical validation. With very few exceptions, k_r can be computed for all real-world applications with both an excellent accuracy (less than 1%) and a low computational time (less than 5 minutes and usually just a few seconds) [Roos and Biannic, 2018]. This is illustrated in Figure A.2, which must be interpreted as follows. Let (x, y) be the coordinates of any star sign. Then x corresponds to the number of benchmarks for which the computational time is lower than y. Note that benchmarks 27 and 29 are not represented, since it is not possible with the current tools to reduce the gap to 0.1% with a reasonable computational time.

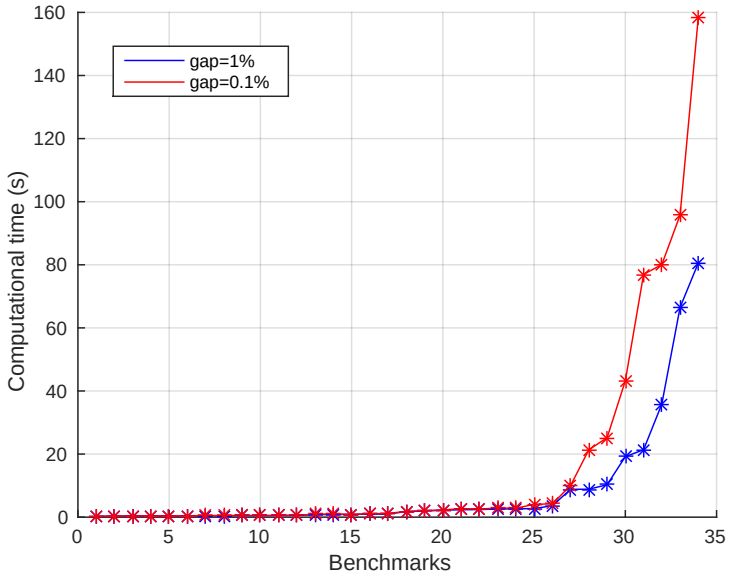


Figure A.2: Computational time to compute tight bounds on the robust stability margin

5 What can be done with μ

The most common use of μ-analysis is to compute the robust stability margin k_r. But other criteria of practical importance can be evaluated. Modal performance is first considered in Section A.5.1. The skewed structured singular value is then introduced. It serves as a basis to compute the skewed robust stability margin in Section A.5.2, the worst-case H_∞ performance level in Section A.5.3, and the worst-case input-output margins in Section A.5.4. All the associated algorithms are implemented in the SMART Library of SMAC Toolbox for Matlab, which is finally presented in Section A.6.

5.1 Modal performance

It is often desirable to quantify the performance degradations, which are induced by model uncertainties and appear before instability. In this context, the objective is not only to check that the poles of the uncertain system are stable, but also that they are sufficiently fast and well damped. In other words, the largest size of the uncertainties is to be computed, for which it can be guaranteed that the poles of the interconnection between $M(s)$ and Δ are located to the left of the truncated sector depicted in Figure A.3. The structured singular value μ is thus

computed along the boundary of this truncated sector instead of the imaginary axis, *i.e.* along the two following segments:

- segment 1: $s = -\alpha + j\omega$ for $\omega \leq \omega_c$,
- segment 2: $s = j\omega z$ for $\omega \geq \omega_c$,

where $z = [1 + j\tan(\phi)]$ and $\omega_c = \alpha/\tan(\phi)$. Let $(A_M,\ B_M, C_M, D_M)$ be a state-space representation of $M(s)$. Then:

- $M(-\alpha + j\omega) = M_1(j\omega)$ on segment 1, where $(A_M + \alpha I,\ B_M, C_M, D_M)$ is a state-space representation of $M_1(s)$.
- $M(j\omega z) = M_2(j\omega)$ on segment 2, where $(A_M/z, B_M/\sqrt{z},\ C_M/\sqrt{z}, D_M)$ is a state-space representation of $M_2(s)$.

Therefore, the adaptation of Algorithm A.1 to handle modal performance is rather straightforward by considering $M_1(s)$ and $M_2(s)$ instead of $M(s)$ [Roos *et al.*, 2011]. The μ lower bound algorithm proposed in Section A.2.4 is modified in the same way. A perturbation Δ is computed, which brings an eigenvalue of the interconnection of Figure A.1 on the boundary of the truncated sector of Figure A.3.

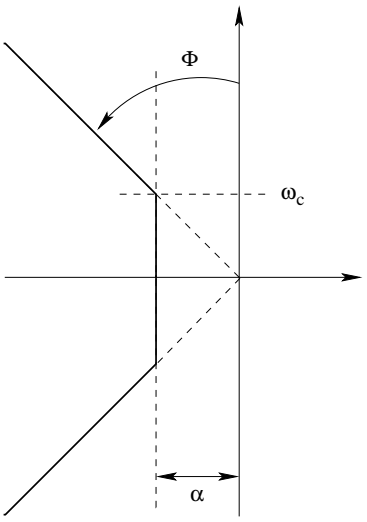


Figure A.3: Truncated sector used for modal performance assessment

5.2 Skewed robust stability margin

5.2.1 Problem statement

With the same notation as in Section A.1, assume now that $\mathbf{\Delta}$ is split into two distinct block structures, *i.e.* $\mathbf{\Delta} = \text{diag}(\mathbf{\Delta_1}, \mathbf{\Delta_2})$. Let $\mathbf{\Delta_s} = \text{diag}(\mathcal{B}_{\mathbf{\Delta_1}}, \mathbf{\Delta_2})$ and $k\mathcal{B}_{\mathbf{\Delta_s}} = \text{diag}(\mathcal{B}_{\mathbf{\Delta_1}}, k\mathcal{B}_{\mathbf{\Delta_2}})$. The skewed structured singular value is defined as follows [Fan and Tits, 1992].

Definition A.4 (skewed structured singular value) *Let $\omega \in \mathbb{R}_+$ be a given frequency. If no matrix $\Delta = \text{diag}(\Delta_1, \Delta_2) \in \mathbf{\Delta_s}$ makes $I - M(j\omega)\Delta$ singular, then the skewed structured singular value $\nu_{\mathbf{\Delta_s}}(M(j\omega))$ is equal to zero. Otherwise:*

$$\nu_{\mathbf{\Delta_s}}(M(j\omega)) = \Big[\min_{\Delta \in \mathbf{\Delta_s}} \{\overline{\sigma}(\Delta_2),\ \det(I - M(j\omega)\Delta) = 0\} \Big]^{-1} \tag{A.26}$$

Lemma A.2 (skewed robust stability margin) *The interconnection of Figure A.1 is stable* $\forall \Delta(s) \in k_s \mathcal{B}_{\mathbf{\Delta_s}}$, *where the skewed robust stability margin* k_s *is defined as the inverse of the largest value of* $\nu_{\mathbf{\Delta_s}}(M(j\omega))$ *over the whole frequency range:*

$$k_s = \left[\sup_{\omega \in \mathbb{R}_+} \nu_{\mathbf{\Delta_s}}(M(j\omega)) \right]^{-1} \tag{A.27}$$

In other words, k_s is the H_∞ norm of the smallest uncertainty $\Delta_2(s) \in \mathbf{\Delta_2}$ such that there exists $\Delta_1(s) \in \mathcal{B}_{\mathbf{\Delta_1}}$ for which the interconnection between $M(s)$ and $\Delta(s) = \text{diag}(\Delta_1(s), \Delta_2(s)) \in \mathbf{\Delta_s}$ is unstable. $\Delta_1(s)$ and $\Delta_2(s)$ thus correspond to fixed range and unbounded uncertainties respectively. ν-analysis is a more general framework than μ-analysis, since the classical structured singular value is recovered in Definition A.4 if $\mathbf{\Delta_1}$ is empty. It also proves very useful in practice. For example, it allows to compute the maximal allowable amount of parametric uncertainties in the presence of neglected dynamics [Ferreres and Fromion, 1996]. And it provides a solution to several analysis problems, such as computing the worst-case H_∞ performance level (see [Roos, 2010; Fan and Tits, 1992] and Section A.5.3) and the worst-case gain, phase, modulus and delay margins (see [Lescher and Roos, 2011] and Section A.5.4). Similarly to k_r, the exact computation of k_s is NP hard in the general case, so both lower and upper bounds are computed instead.

Problem A.2 (skewed robust stability margin) *With reference to Figure A.1, compute the skewed robust stability margin* k_s *defined in equation (A.27) for a given block structure* $\mathbf{\Delta_s}$.

5.2.2 Skew-μ lower bound computation

Most of the μ lower bound algorithms surveyed in Section A.2.1 can be extended more or less easily to handle skew uncertainties. The objective here is not to study these generalizations in details, but to show that the most relevant methods highlighted in Section A.2.4 still perform very well in the skew case.

Purely complex and mixed real/complex problems

The power algorithm of [Young and Doyle, 1997] is the most efficient way to compute μ lower bounds. It can be easily extended as explained in [Ferreres and Fromion, 1996; Glavaski and Tierno, 1998; Holland *et al.*, 2005]. Combined with the strategy detailed in Section A.2.4, it usually provides very good ν lower bounds. A significant number of tests have indeed revealed that the exact value of ν can be obtained in many cases. This improved version of the power algorithm is implemented in the SMART Library of the SMAC Toolbox (see [Roos, 2013] and Section A.6).

Purely real problems

The poles migration technique of [Ferreres and Biannic, 2001] is the most efficient μ lower bound algorithm. It can be easily extended (see [Roos and Biannic, 2015]), and the main lines are summarized below. Similarly to the original algorithm briefly described in Section A.2.1, the considered skew-μ problem is first regularized by adding a small amount ϵ of complex uncertainty. In this perspective, a new block structure $\mathbf{\Delta_{sc}} \subset \mathbb{C}^{n \times n}$ is introduced, which satisfies $\mathbf{\Delta_s} = \mathbf{\Delta_{sc}} \cap \mathbb{R}^{n \times n}$ [Packard and Pandey, 1993]. The power algorithm of [Ferreres and Fromion, 1996] is then applied at a given frequency ω_i to the augmented interconnection between $M_a(j\omega_i)$ and $\Delta_a \in \text{diag}(\mathbf{\Delta_s}, \mathbf{\Delta_{sc}})$, where $M_a(s) = \begin{bmatrix} M(s) & \sqrt{\epsilon} M(s) \\ \sqrt{\epsilon} M(s) & \epsilon M(s) \end{bmatrix}$. Good convergence properties are usually observed in practice. Nevertheless, the resulting ν lower bound is not a lower bound for the original skew-μ problem. Indeed, an uncertainty $\Delta_a^0 = \text{diag}(\Delta^0, \Delta_c^0)$ is obtained, which

renders the matrix $I - M_a(j\omega_i)\Delta_a^0$ singular, but it cannot be claimed that $I - M(j\omega_i)\Delta^0$ is itself singular. Nevertheless, if ϵ is small enough, an eigenvalue λ^0 of the interconnection of Figure A.1 is usually located near the point $j\omega_i$ of the imaginary axis when $\Delta = \Delta^0$. Starting from this good initial guess $\Delta^0 = \text{diag}(\delta_1^0 I_{n_1}, \ldots, \delta_N^0 I_{n_N}) \in \mathbf{\Delta_s}$, the idea is now to move λ^0 through the imaginary axis to obtain a destabilizing uncertainty for the initial skew-μ problem. A solution is to introduce a perturbation $d\Delta = \text{diag}(\hat{\delta}_1 I_{n_1}, \ldots, \hat{\delta}_N I_{n_N}) \in \mathbf{\Delta}$. The problem is then to find the uncertainty $\tilde{\Delta} = \Delta^0 + d\Delta = \text{diag}(\tilde{\Delta}_1, \tilde{\Delta}_2) \in \mathbf{\Delta_s}$ with minimum $\overline{\sigma}(\tilde{\Delta}_2)$, which brings λ^0 on the imaginary axis. In this perspective, the first-order relationship between the variation $d\lambda$ of λ^0 due to a variation $d\Delta$ of Δ^0 (see the subsection on poles migration techniques in Section A.2.1) is reformulated as:

$$d\lambda = \sum_{i=1}^{N} \alpha_i \hat{\delta}_i \tag{A.28}$$

where $\alpha_i = (uB + tD)\dfrac{\partial \Delta}{\partial \delta_i}(Cv + Dw)$ and $\dfrac{\partial \Delta}{\partial \delta_i} = \text{diag}\left(0_{n_1+\cdots+n_{i-1}}, I_{n_i}, 0_{n_{i+1}+\cdots+n_N}\right)$. A simple linear programming problem is finally obtained:

$$\min_{\hat{\delta}_i} \eta \quad \text{s.t.} \quad \begin{cases} -1 \leq \delta_i^0 + \hat{\delta}_i \leq 1 & \text{for fixed-range uncertainties} \\ -\eta \leq \delta_i^0 + \hat{\delta}_i \leq \eta & \text{for unbounded uncertainties} \\ \Re\left(\lambda^0 + \displaystyle\sum_{i=1}^{N} \alpha_i \hat{\delta}_i\right) = 0 \end{cases} \tag{A.29}$$

In the above formulation, the inequality constraints ensure that $\tilde{\Delta} \in \mathbf{\Delta_s}$, while the equality constraint forces $\lambda^0 + d\lambda$ to lie on the imaginary axis once the perturbation $d\Delta$ is applied (or very near since a first-order approximation is used here). Problem (A.29) can be efficiently solved using existing softwares [The Mathworks, 2017a]. The minimum value η_{opt} of η can thus be computed with a very low computational cost, as well as the corresponding values $d\lambda_{opt}$ and $d\Delta_{opt}$ of $d\lambda$ and $d\Delta$. Moreover, $\det(I - M(j\tilde{\omega})\tilde{\Delta}) \approx 0$, where $\tilde{\omega} = \Im\left(\lambda^0 + d\lambda_{opt}\right)$ and $\tilde{\Delta} = \Delta^0 + d\Delta_{opt}$. This quantity is not exactly equal to 0, since (A.29) is just a linearized version of the problem to be solved. A solution to achieve equality is to replace $\tilde{\Delta}$ with $\text{diag}(\tilde{\Delta}_1, \alpha\tilde{\Delta}_2)$ and to perform a dichotomy search on α until $|\det(I - M(j\tilde{\omega})\text{diag}(\tilde{\Delta}_1, \alpha\tilde{\Delta}_2))|$ becomes small enough. An upper bound on k_s is then given by $\alpha\eta_{opt}$, where α is usually very close to 1. The resulting ν lower bound is equal to $(\alpha\eta_{opt})^{-1}$.

Remark A.12 *If λ^0 is not sufficiently close to the imaginary axis, the first order development (A.28) may not be accurate enough to directly move the eigenvalue onto the imaginary axis. A solution consists in partitioning the real segment which separates λ^0 from the imaginary axis, and to iteratively perform the migration on each sub-segment.*

Remark A.13 *The frequency at which the interconnection of Figure A.1 becomes unstable is left free during optimization and it can be very far from the initial one. Thus, applying the aforementioned algorithm at each point of a rough initial grid is usually sufficient to capture the most critical frequencies.*

The above method is implemented in the SMART Library of the SMAC Toolbox (see [Roos, 2013] and Section A.6). It has been applied to the realistic examples proposed in [Lemos *et al.*, 2014], and compared to the nonlinear and the nonsmooth optimization techniques of [Halton *et al.*, 2008] and [Lemos *et al.*, 2014]. Results are similar as in the classical μ case: the proposed algorithm gives the best upper bounds on k_s for both examples with a very low computational time (a few seconds on average).

5.2.3 Skew-μ upper bound computation

Theorem A.2 can be extended to handle skew uncertainties. The matrices D and G are split into two distinct blocks associated to the fixed range and the unbounded uncertainties respectively, *i.e.* $D = \text{diag}(D_1, D_2)$ and $G = \text{diag}(G_1, G_2)$. If the following inequality derived from (A.11) holds:

$$M^* \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} M + j \left(\begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix} M - M^* \begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix} \right) \leq \begin{bmatrix} D_1 & 0 \\ 0 & \beta^2 D_2 \end{bmatrix} \tag{A.30}$$

then $\nu_{\mathbf{\Delta_s}}(M) < \beta$. A skew-$\mu$ upper bound can thus be computed by directly minimizing β in (A.30) using an LMI solver. To alleviate the computational cost, another solution first consists of reformulating inequality (A.12) in the same way as above. A dichotomic search on β is then applied (or equivalently a fixed-point algorithm), and at each iteration a gradient descent algorithm is used to check quickly whether it can be satisfied or not. This is not detailed here for the sake of brevity, but more details can be found in [Ferreres and Fromion, 1997]. The resulting adapted version of Algorithm A.1 is implemented in the SMART Library of the SMAC Toolbox (see [Roos, 2013] and Section A.6). It has been applied to the examples of [Lemos *et al.*, 2014]. In both cases, the lower bounds on k_s are very good, and even equal to the upper bounds computed in Section A.5.2.2. The exact value of k_s is thus obtained in both cases, which confirms the good results already obtained in the classical μ case.

5.3 Worst case H_∞ performance

Once robust stability is ensured, another question of interest is to determine whether the worst-case H_∞ performance of the system is satisfactory. Assuming the uncertainties are normalized, *i.e.* $\Delta(s) \in \mathcal{B}_{\mathbf{\Delta}}$, this can be formulated as follows:

Problem A.3 (worst-case H_∞ performance level) *With reference to Figure A.4 and assuming that $k_r > 1$, compute the highest value k_∞ of the H_∞ norm $\|T_{e\to y}(M(s), \Delta(s))\|_\infty$ of the map from e to y when $\Delta(s)$ takes all possible values in $\mathcal{B}_{\mathbf{\Delta}}$.*

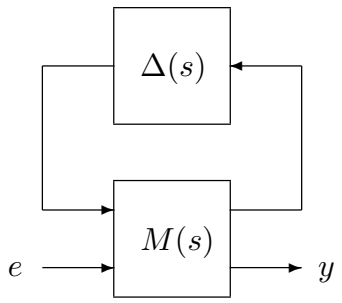


Figure A.4: Standard interconnection for worst-case H_∞ performance analysis

The following proposition is a direct consequence of the main loop theorem [Packard and Doyle, 1993] and allows to reformulate Problem A.3 as a specific skew-μ problem. It is assumed without loss of generality that both e and y are p-dimensional signals.

Proposition A.2 *The following statements are equivalent:*

1. $\max\limits_{\Delta(s)\in\mathcal{B}_{\mathbf{\Delta}}} \|T_{e\to y}(M(s), \Delta(s))\|_\infty \leq \gamma$

2. $\mu_{\mathbf{\Delta_a}}\left(\text{diag}(I_n, I_p/\sqrt{\gamma})M(j\omega)\text{diag}(I_n, I_p/\sqrt{\gamma})\right) \leq 1\ \forall\omega \in \mathbb{R}_+$, *where* $\mathbf{\Delta_a} = \text{diag}(\mathbf{\Delta}, \mathbb{C}^{p\times p})$

Therefore, it appears that checking whether k_∞ is less than a given value γ amounts to computing the robust stability margin of the interconnection between $\text{diag}(I_n, I_p/\sqrt{\gamma})M(s)\text{diag}(I_n, I_p/\sqrt{\gamma})$ and an augmented uncertainty $\Delta_a \in \mathbf{\Delta_a}$. The computation of k_∞ can be performed using a dichotomic search on γ. But a better solution is to note that a skew-μ problem can be solved instead. Indeed, statement 2. of Proposition A.2 is in turn equivalent to:

3. $\nu_{\mathbf{\Delta_a}}(M(j\omega) \leq \gamma\ \forall\omega \in \mathbb{R}_+$, *where* $\mathbf{\Delta_a} = \text{diag}(\mathbf{\Delta_1}, \mathbf{\Delta_2}) = \text{diag}(\mathbf{\Delta}, \mathbb{C}^{p\times p})$

Thus, an upper bound on k_∞ can be computed using a slightly modified version of Algorithm A.1. At step 2(b), γ is minimized under the following LMI constraint derived from (A.30):

$$M^*\begin{bmatrix} D_1 & 0 \\ 0 & I_p \end{bmatrix} M + j\left(\begin{bmatrix} G_1 & 0 \\ 0 & 0 \end{bmatrix} M - M^*\begin{bmatrix} G_1 & 0 \\ 0 & 0 \end{bmatrix}\right) \leq \begin{bmatrix} D_1 & 0 \\ 0 & \gamma^2 I_p \end{bmatrix} \tag{A.31}$$

Two strategies can be used to compute a lower bound on k_∞ depending on the nature of $\Delta(s)$:

- For purely real problems, a two-step procedure is implemented at each frequency ω_i of a rough grid, in the spirit of the skew-μ lower bound algorithm described in Section A.5.2.2:

  1. The unit ball $\mathcal{B}_{\mathbf{\Delta}}$ is investigated by iteratively:
     - computing the gradient of $\overline{\sigma}(T_{e\to y}(M(j\omega_i), \Delta))$,
     - performing a line search to maximize this quantity (which boils down to computing the eigenvalues of a Hamiltonian-like matrix),

     until the problem is roughly solved at ω_i.
  2. Using the value of Δ computed at step 1 as an initialization, a linear (if $p = 1$) or a quadratic (if $p > 1$) programming problem is repeatedly solved until convergence, so as to locally maximize $\overline{\sigma}(T_{e\to y}(M(j\omega), \Delta))$ with respect to both Δ and ω.

  A lower bound on k_∞ is finally obtained, as well as the associated value of Δ. The whole algorithm is thoroughly described in [Roos, 2010] and implemented in the SMART Library of the SMAC Toolbox (see [Roos, 2013] and Section A.6). Numerical results obtained on several high-dimensional systems show that the proposed algorithm compares very favorably with previously existing ones, and almost non-conservative bounds are obtained with a low computational time. Note that another approach was published recently in [Apkarian and Noll, 2017] after our comparative study was conducted. It is therefore not evaluated in this work, but very accurate results are reported.

- For purely complex and mixed real/complex problems, the skew version of the power algorithm presented in Section A.5.2.2 is applied.

Remark A.14 *Problem A.3 assumes that there is a single performance channel between e and y. However, it is possible to consider several channels by splitting* $e = [e_1 \dots e_q]^T$ *and* $y = [y_1 \dots y_q]^T$. *This leads to the following optimization problem:*

$$k_\infty = \max_{\Delta(s)\in\mathcal{B}_{\mathbf{\Delta}}} \max_{1\leq k\leq q} \|T_{e_k\to y_k}(M(s), \Delta(s))\|_\infty \tag{A.32}$$

It suffices to structure $\mathbf{\Delta_2}$ *and to solve a more general skew-*μ *problem using the algorithms presented in Section A.5.2.2 and A.5.2.3.*

5.4 Worst-case input-output margins

Structured robustness analysis considers that the systems uncertainties are perfectly identified and localized, which can be unrealistic. So in addition to the robust stability margin, it is often desirable to compute some worst-case gain, modulus, phase and delay margins to evaluate the effective robustness. In this perspective, the uncertain closed-loop model is first opened at the point where those margins are to be calculated. It is then rearranged to obtain the interconnection of Figure A.4, in such a way that the uncertain closed-loop model can be recovered by directly connecting y to e. The following problem can then be stated.

Problem A.4 (worst-case input-output margins) *With reference to Figure A.4 and assuming that $k_r > 1$, compute the worst-case gain, modulus, phase and delay margins k_g, k_m, k_p and k_d, i.e. the highest value of the real gain, the complex gain, the phase shift and the time delay respectively that can be inserted between y and e without destabilizing the system when $\Delta(s)$ takes all possible values in $\mathcal{B}_{\mathbf{\Delta}}$.*

To address this issue, some additional uncertainties $\Delta_m(s) = \text{diag}\,(\delta_{m,1}, \ldots, \delta_{m,p})$ are introduced between e and y, such that $e = \Delta_m(s)\, y$. The $\delta_{m,i}$ correspond to real gains, complex gains, phase shifts or time delays. Their expressions are given hereafter:

- real gain: $\delta_{m,i} = 1 + \hat{\delta}_i$, $\hat{\delta}_i \in \mathbb{R}$
- complex gain: $\delta_{m,i} = 1 + \hat{\delta}_i$, $\hat{\delta}_i \in \mathbb{C}$
- phase shift: $\delta_{m,i} = e^{j\phi_i}$, $\phi_i \in [-\pi\,,\,\pi]$
- time delay: $\delta_{m,i} = e^{-\tau_i s}$, $\tau_i \in \mathbb{R}_+$

For the gain and the modulus margins, the expressions of $\delta_{m,i}$ are polynomial and can be written directly in linear fractional form. For the phase and the delay margins, however, the non-rational elements $e^{j\phi_i}$ and $e^{-\tau_i s}$ must be transformed first. A bilinear transform is applied to replace $e^{j\phi_i}$ with $\frac{1-j\hat{\delta}_i}{1+j\hat{\delta}_i}$, where $\hat{\delta}_i \in \mathbb{R}$. Similarly to $e^{j\phi_i}$, this new expression has a unitary modulus. Moreover, its phase variation covers the whole interval $[-\pi\,,\,\pi]$ when $\hat{\delta}_i \in \mathbb{R}$. On the other hand, the case of the delay $e^{-\tau_i s}$ is more complicated because of the Laplace variable s. A solution proposed in [Lescher and Roos, 2011] consists of replacing it with a static rational complex function $f(\hat{\delta}_i)$, where the range of variation of $\hat{\delta}_i \in \mathbb{R}$ depends on ω.

The structured uncertainties $\Delta(s)$ and the additional real or complex parametric uncertainties $\hat{\delta}_i$ are finally stacked into a single block-diagonal operator $\overline{\Delta}(s)$ in feedback loop with an augmented plant $\overline{M}(s)$, leading to an interconnection similar to that in Figure A.1. In this context, Problem A.4 can be reformulated as a skew-μ problem: find the highest amplitude of the $\hat{\delta}_i$ for which the interconnection is stable for all $\Delta \in \mathcal{B}_{\mathbf{\Delta}}$. On the one hand, upper bounds on k_g, k_m, k_p or k_d are obtained using the algorithms described previously:

- the poles migration technique of Section A.5.2.2 for purely real problems where the gain, phase or delay margin is to be computed,
- the two-step procedure of Section A.5.3 for purely real problems where the modulus margin is to be computed,
- the power algorithm of Section A.5.2.2 for purely complex or mixed real/complex problems whatever the margin to be computed.

On the other hand, lower bounds on k_g, k_m or k_p can be computed using the adapted version of Algorithm A.1 presented in Section A.5.2.3. The case of the delay margin k_d must be dealt with separately, since the range of variation of $\hat{\delta}_i \in \mathbb{R}$ depends on ω. This dependence can be treated using some results from [Tits and Balakrishnan, 1998]. The whole process is quite tedious and is not presented here for the sake of brevity. All details can be found in [Lescher and Roos, 2011], where an extension of Algorithm A.1 is proposed and validated on a realistic application.

6 An insight into the SMART library

Despite numerous contributions to μ-analysis from many researchers during the past three decades, only few practical algorithms have been implemented. Some routines are available in the Robust Control Toolbox [The Mathworks, 2017b], but they cannot address all the aforementioned issues, although progress has been made recently with a new routine based on Algorithm A.1 to compute a guaranteed μ upper bound on the whole frequency range. In this context, the main purpose of the *Skew-Mu Analysis based Robustness Tools* (SMART) Library of the Systems Modeling, Analysis and Control (SMAC) Toolbox is to help bridging this gap between theory and practice. As already highlighted in the previous sections, it implements most of the μ-analysis based algorithms developed at ONERA during the last two decades [Roos, 2013; Roos *et al.*, 2011]. It provides a complete suite of routines to compute the (skewed) structured singular value, the (skewed) robust stability margin, the worst-case H_∞ performance level, as well as the worst-case gain, modulus, phase and delay margins for linear time-invariant systems affected by time-invariant uncertainties. A full version can be freely downloaded from the SMAC website `http://w3.onera.fr/smac/smart`, which also provides a complete documentation as well as applicative examples. It has been evaluated on a large set of challenging benchmarks, corresponding to various fields of application, system dimensions and structures of the uncertainties.

7 Summary of the contributions

The main contributions presented in this chapter are briefly summarized below, and a selection of related publications is given:

- **Section A.2:** detailed comparison of all existing methods to compute upper bounds on the robust stability margin k_r, *i.e.* lower bounds on the structured singular value μ ; improvements and combinations of existing algorithms to further refine the bounds with a reasonable computational effort [Roos and Biannic, 2015; Fabrizi *et al.*, 2014].

- **Section A.3:** computation of lower bounds on k_r, *i.e.* μ upper bounds, which are guaranteed on the whole frequency range ; use of the μ-sensitivities to better handle the tradeoff between computational time and accuracy ; improvement of the multiplier-based approach of Fu and Barabanov to make it reliable and computationally more attractive [Roos and Biannic, 2010; Lesprier *et al.*, 2015b; 2015c].

- **Section A.4:** branch-and-bound algorithms to further reduce the gap between the bounds up to the desired accuracy [Lesprier *et al.*, 2015c; Roos *et al.*, 2011].

- **Section A.5:** extensions to modal performance, skewed robust stability, worst-case H_∞ performance and worst-case gain/phase/modulus/delay margins [Roos and Biannic, 2015; Roos, 2010; Lescher and Roos, 2011].

- **Section A.6:** implementation of all algorithms in the SMART Library of the SMAC Toolbox [Roos, 2013; Roos *et al.*, 2011].
- **all along Chapter A:** validation of the proposed algorithms on a large set of realistic benchmarks [Roos and Biannic, 2015].

A few other issues have also been addressed over the past decade, but they have not been discussed in this chapter due to space constraints: determination of a guaranteed stability domain for (possibly uncertain) parameter dependent plants [Roos and Biannic, 2010], evaluation of specific clearance criteria using μ tools [Biannic and Roos, 2012], use of a probabilistic framework (see the future prospects at the end of this manuscript), extension of the proposed tools to discrete-time systems...

Chapter B

Extensions to time-varying uncertainties

μ-analysis is now well recognized as mature and efficient by the control community. Several practical issues can be adressed, as emphasized in Chapter A. Nevertheless, only time-invariant uncertainties can be considered, which may be restrictive in some applications, for example in the aerospace domain. It is thus desirable to develop some methods and tools to handle time-varying uncertainties and parameters as well.

In this perspective, the μ-analysis framework was extended in [Meinsma *et al.*, 2000; Paganini, 1996; Shamma, 1994]. It is shown in these papers that the D and G scaling matrices classically used to compute a lower bound on the robust stability margin in the LTI case can still be used in the presence of mixed time-invariant and arbitrarily fast time-varying uncertainties. The idea is to consider either frequency dependent or constant blocks according to the nature of the uncertainties. The resulting optimization problem is convex, but it involves an infinite number of both decision variables and constraints. Moreover, it cannot be solved independently at each frequency because of the constant blocks, and this can be computationally demanding.

Another approach has aroused a growing interest in the last twenty years. It consists of characterizing the uncertainties using integral quadratic constraints (IQC) [Megretski and Rantzer, 1997]. In addition to time-invariant and arbitrarily fast time-varying uncertainties, this technique allows to consider uncertainties with bounded rates of variation [Jonsson and Rantzer, 1996; Helmersson, 1999; Köroğlu and Scherer, 2007; Pfifer and Seiler, 2016]. Computing a robust stability margin or a worst-case performance level is usually achieved by transforming an infinite set of frequency-domain constraints into a single state-space based LMI using the Kalman-Yakubovich-Popov lemma. But a special attention must be paid to computational complexity. The resulting optimization problem may indeed become intractable when the size of the system and the number of IQC used to characterize the uncertainties increase.

The main alternative to IQC analysis is to use parameter-dependent Lyapunov (PDL) functions [Gahinet *et al.*, 1996; Haddad and Kapila, 1998; Amato *et al.*, 1997; Montagner and Peres, 2004; Chesi *et al.*, 2007]. Indeed, this approach allows to introduce bounds on the rates of variation of the uncertain parameters in a relatively simple and natural fashion. Several kinds of dependence to the parameters can be considered: affine, multi-affine, quadratic, polynomial, rational... In practice, the affine case is often preferred to avoid an excessive increase in computational complexity. But whatever the form used, the resulting optimization problem almost always boils down to solving an infinite number of LMI. Various techniques can be implemented to address this issue, but the large number of decision variables usually makes it difficult to consider high-order systems.

Most of the above techniques can be used to assess the robustness properties of a linear system in the presence of mixed time-invariant and time-varying uncertainties. Nevertheless,

when the dimension of the system increases, they usually suffer the same drawback: a dramatic increase in the computational time. In this context, two methods are proposed in this chapter to alleviate the computational effort, while keeping the same level of accuracy. The first one builds upon the (D, G)-scalings based characterization of [Meinsma *et al.*, 2000; Paganini, 1996]. A two-step procedure is described in Section B.1. Some scaling matrices are first computed on a finite frequency grid. A validation is then performed on the whole frequency range using a μ-analysis based technique inspired from Algorithm A.1. This procedure is repeated until all frequencies are validated. The second method, presented in Section B.2, is also a two-step procedure. A PDL function is first optimized on a finite parametric grid. Its validity is then checked on the whole parametric domain using a fast and reliable μ-analysis based test. In case the validation fails, a worst-case configuration is added to the grid and the whole process is repeated.

The objective of this chapter is not to make an extensive comparison of all practical algorithms to analyze LTI systems in the presence of mixed time-invariant and time-varying uncertainties. The primary concern is to show how μ-analysis can sometimes play a key role in making the computation of robust stability margins and worst-case performance levels more efficient.

1 A μ-analysis based approach

1.1 Problem statement

Let us consider the standard interconnections of Figure B.1. $M(s)$ is a continuous-time stable and proper real-rational transfer function representing the nominal closed-loop system, exactly as in Chapter A. For the sake of simplicity and because this is often sufficient in practice, Δ is restricted to be a linear time-varying memoryless multiplication operator, *i.e.* $w(t) = \Delta(t)z(t)$ for all $t \geq 0$. Moreover, it has a block-diagonal structure of the form:

$$\Delta(t) = \mathrm{diag}(\delta^{[1]}(t)I_{n_1}, \ldots, \delta^{[N]}(t)I_{n_N}) \tag{B.1}$$

where $n_v = \sum_{i=1}^{N} n_i$ and each $\delta^{[i]}$ is a real-valued scalar function corresponding to a time-varying parameter or uncertainty. The set of all such operators is denoted by $\mathbf{\Delta}$, and $k\mathcal{B}_{\mathbf{\Delta}} = \{\Delta \in \mathbf{\Delta} : \|\Delta\|_{i2} \leq k\}$, where $\| \ \|_{i2}$ denotes the induced $\mathcal{L}_2$ norm.

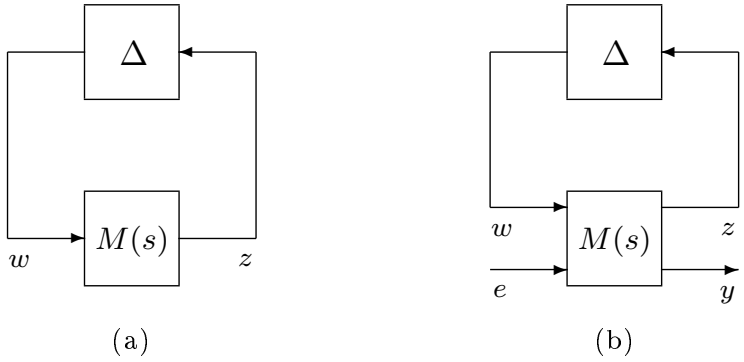


Figure B.1: Interconnections for robust stability (a) and worst-case performance (b) analysis

Definition B.1 (stability) *The interconnection of Figure B.1(a) is stable with respect to a set $\mathcal{S}$ of bounded operators Δ if there exists $\alpha > 0$ such that $\|(I - \Delta M)^{-1}\|_{i2} < \alpha$ for all $\Delta \in \mathcal{S}$.*

Two analysis problems are considered in this chapter. They are similar to Problems A.1 and A.3, except that Δ contains time-varying parameters. Note that the induced $\mathcal{L}_2$ norm reduces to the H_∞ norm when only time-invariant uncertainties are considered, and thus $k_{i2} = k_\infty$ in this case.

Definition B.2 (robust stability margin) *The robust stability margin k_r is defined as the largest value of k such that the interconnection of Figure B.1(a) is stable for all $\Delta \in k\mathcal{B}_{\mathbf{\Delta}}$.*

Problem B.1 (robust stability margin) *With reference to Figure B.1(a), compute (a lower bound on) the robust stability margin k_r for a given set of operators $\mathbf{\Delta}$.*

Problem B.2 (worst-case performance level) *With reference to Figure B.1(b) and assuming that $k_r > 1$, compute (an upper bound on) the highest value k_{i2} of the induced $\mathcal{L}_2$ norm $\|T_{e\to y}\|_{i2}$ of the map from $e \in \mathcal{L}_2^p$ to $y \in \mathcal{L}_2^p$ when Δ takes all possible values in $\mathcal{B}_{\mathbf{\Delta}}$.*

The following theorem provides a sufficient condition of stability when the $\delta^{[i]}$ are allowed to vary arbitrarily fast, *i.e.* when no assumption is made on their rates of variation [Paganini, 1996; Meinsma *et al.*, 2000].

Theorem B.1 *Let $\beta > 0$. If there exist matrices $D \in \mathcal{D}_{TV}$ and $G \in \mathcal{G}_{TV}$ such that the following relation holds for all $\omega \in \mathbb{R}_+$:*

$$M^*(j\omega)DM(j\omega) + j(GM(j\omega) - M^*(j\omega)G) \leq \beta^2 D \tag{B.2}$$

where $\mathcal{D}_{TV} = \{0 < D = D^T \in \mathbb{R}^{n_v \times n_v} : \forall \Delta \in \mathbf{\Delta},\, D\Delta = \Delta D\}$ and $\mathcal{G}_{TV} = \{G = G^* \in j\mathbb{R}^{n_v \times n_v} : \forall \Delta \in \mathbf{\Delta},\, G\Delta = \Delta^* G\}$, then the interconnection of Figure B.1(a) is stable for all time-varying uncertainties $\Delta \in \frac{1}{\beta}\mathcal{B}_{\mathbf{\Delta}}$. The inverse of the smallest value of β such that (B.2) holds is a lower bound on the robust stability margin k_r.

The characterization of Theorem B.1 is quite similar to the one dedicated to time-invariant uncertainties (see Section A.3). But a main difference is worth being mentioned: the matrices D and G are constant on the whole frequency range. This is in stark contrast to the time-invariant case, where D and G can be different at each frequency.

Computing k_r using Theorem B.1 involves an infinite number of frequency-domain constraints. There are two classical ways to address this issue. A first method consists of solving (B.2) on a frequency grid instead of the whole frequency range. But as already discussed in Chapter A, there is a risk to miss a critical frequency and to over-evaluate the robust stability margin. Another approach is to apply the KYP lemma, so as to transform the infinite number of frequency-dependent inequalities (B.2) into a unique LMI. In this case, no critical frequency can be missed, but the computational time may be prohibitive for high-order systems.

Assume now that $\Delta = \mathrm{diag}(\Delta_{TI}, \Delta_{TV}) \in \mathbf{\Delta}$ is a block-diagonal operator with time-invariant uncertainties $\Delta_{TI} \in \mathbf{\Delta_{TI}}$ as in Chapter A, *i.e.* real parametric uncertainties and/or neglected dynamics, but also real time-varying parameters or uncertainties $\Delta_{TV} \in \mathbf{\Delta_{TV}}$ as above. The associated scaling matrices are defined for all $\omega \in \mathbb{R}_+$ as $D(\omega) = \mathrm{diag}(D_{TI}(\omega), D_{TV}) \in \mathrm{diag}(\mathcal{D}_{TI}, \mathcal{D}_{TV})$ and $G(\omega) = \mathrm{diag}(G_{TI}(\omega), G_{TV}) \in \mathrm{diag}(\mathcal{G}_{TI}, \mathcal{G}_{TV})$, where $\mathcal{D}_{TI}, \mathcal{G}_{TI} \in \mathbb{C}^{n_i \times n_i}$ are defined as in Theorem A.2, and $\mathcal{D}_{TV}, \mathcal{G}_{TV} \in \mathbb{C}^{n_v \times n_v}$ are defined above. Theorem B.1 is modified as follows:

Proposition B.1 *Let $\beta > 0$. If there exist matrices $D(\omega) = \mathrm{diag}(D_{TI}(\omega), D_{TV})$ and $G(\omega) = \mathrm{diag}(G_{TI}(\omega), G_{TV})$ such that the following relation holds for all $\omega \in \mathbb{R}_+$:*

$$M^*(j\omega)D(\omega)M(j\omega) + j(G(\omega)M(j\omega) - M^*(j\omega)G(\omega)) \leq \beta^2 D(\omega) \tag{B.3}$$

then the interconnection of Figure B.1(a) is stable for all mixed time-invariant / time-varying uncertainties $\Delta \in \frac{1}{\beta}\mathcal{B}_{\mathbf{\Delta}}$. The inverse of the smallest value of β such that (B.3) holds is a lower bound on the robust stability margin k_r.

The previous result can be extended to worst-case performance analysis.

Proposition B.2 *Let $\gamma > 0$ and $M_\gamma(s) = \mathrm{diag}(I_{n_i}, I_{n_v}, \gamma^{-1} I_p) M(s)$. If there exist matrices $D(\omega) = \mathrm{diag}(D_{TI}(\omega), D_{TV}, I_p)$ and $G(\omega) = \mathrm{diag}(G_{TI}(\omega), G_{TV}, 0_p)$ such that one of the following relation holds for all $\omega \in \mathbb{R}_+$:*

$$M_\gamma^*(j\omega) D(\omega) M_\gamma(j\omega) + j\left(G(\omega) M_\gamma(j\omega) - M_\gamma^*(j\omega) G(\omega)\right) \leq D(\omega) \tag{B.4}$$

$$\overline{\sigma}\left(\left(D(\omega) M_\gamma(j\omega) D(\omega)^{-1} - jG(\omega)\right)\left(I + G(\omega)^2\right)^{-\frac{1}{2}}\right) \leq 1 \tag{B.5}$$

then $\|T_{e\to y}\|_{i2} \leq \gamma$ for all mixed time-invariant / time-varying uncertainties $\Delta \in \mathcal{B}_{\mathbf{\Delta}}$. The smallest value γ^* of γ such that (B.4) holds is an upper bound on the worst-case performance level k_{i2}.

A practical method is proposed in this section to solve the infinite dimensional optimization problems of Proposition B.2, and to avoid the disadvantages of the classical grid-based and KYP-based techniques. Section B.1.2 is first devoted to the computation of an upper bound on k_{i2} (*i.e.* a guaranteed worst-case performance level). A nontrivial extension to the design of robust feedforward controllers is then highlighted in Section B.1.3. Note that the extension to the computation of a lower bound on k_r (*i.e.* a guaranteed robust stability margin) is straightworward, so it is not detailed here. Indeed, it suffices to work with (B.3) instead of (B.4).

Remark B.1 *(B.3) and (B.4) are only sufficient conditions, that is why only a lower bound on k_r and an upper bound on k_{i2} are obtained. We do not focus on computing upper bounds on k_r and lower bounds on k_{i2} (see e.g. [Köroğlu and Scherer, 2007; Peni and Seiler, 2016]).*

Remark B.2 *The uncertainties in Δ_{TV} can vary arbitrarily fast, i.e. no bounds on their rates of variation are taken into account. A Lyapunov-based approach which addresses this issue is proposed in Section B.2.*

1.2 Performance analysis

1.2.1 Computation of a worst-case performance level

Proposition B.2 requires minimizing a linear objective γ under the LMI constraints (B.4). This problem has an infinite number of optimization variables, namely the matrices $D_{TI}(\omega)$ and $G_{TI}(\omega)$. Let γ^* be the global minimum solution. As already emphasized in Section B.1.1, there are two classical ways to compute γ^*, which is the best upper bound on k_{i2} that can be computed using Proposition B.2.

The first one is a state-space solution. It consists of applying the KYP lemma to transform the constraints (B.4) into a single LMI, so as to obtain an augmented finite dimensional optimization problem. This requires turning the frequency dependent scaling matrices $D_{TI}(\omega)$ and $G_{TI}(\omega)$ into LTI systems $D_{TI}(s) = \sum_i \alpha_i D_i(s)$ and $G_{TI}(s) = \sum_i \beta_i G_i(s)$, where the filters $D_i(s)$ and $G_i(s)$ are fixed while α_i and β_i are the optimization parameters. This approach has three drawbacks. First, only an upper bound on γ^* is computed, since finite dimensional bases of $D_{TI}(s)$ and $G_{TI}(s)$ are used, which cannot cover the whole set of possible scaling matrices $D_{TI}(\omega)$ and $G_{TI}(\omega)$. Second, the choice of the bases is not obvious. Third, the order of the augmented state-space representation resulting from the application of the KYP lemma can be very high, since it contains $M(s)$ as well as the $D_i(s)$ and the $G_i(s)$. This can lead to an excessive computational burden for the LMI solver.

The second approach consists of solving (B.4) on a frequency grid instead of the whole frequency range, which is a finite dimensional optimization problem. This was proposed in [Lind

et al., 1995; Sparks *et al.*, 1996; Paganini, 1996], but the result can be unreliable for reasons already discussed in Chapter A and in Section B.1.1. Only a lower bound $\underline{\gamma}$ on γ^* is obtained, since the optimization problem is less constrained than on the whole frequency range, and it cannot be guaranteed that $\underline{\gamma} > k_{i2}$. Moreover, the computational time can become prohibitive if a fine grid is considered to reduce the risk of missing a critical frequency. The idea here is to go one step further by first solving the problem on a rough grid and then validating the result on the whole frequency range using a fast and reliable Hamiltonian-based technique, in the spirit of Algorithm A.1. The computational time will be much lower, but still larger than in the time-invariant case because of the constant matrices D_{TV} and G_{TV}, which make it impossible to independently solve the problem at each frequency. The inequalities (B.4) to be considered on the grid must indeed be stacked into a single one before an LMI solver can be used. The following algorithm is introduced in [Ferreres and Roos, 2007] to compute γ^* with the desired accuracy.

Algorithm B.1 (optimal computation of an upper bound on k_{i2})

1. *Define an initial frequency grid $(\omega_i)_i$ composed of a few points (or even a single one).*
2. *Solve the optimization problem of Proposition B.2 on the grid with respect to the frequency dependent matrices $D_{TI}(\omega_i)$, $G_{TI}(\omega_i)$ and to the constant matrices D_{TV}, G_{TV}. Let $\underline{\gamma}$ be the minimized value of γ. It is a lower bound on γ^*.*
3. *Let $\gamma = (1+\epsilon)\underline{\gamma}$, where $\epsilon > 0$ is a user-defined threshold. For the values of D_{TV} and G_{TV} computed at step 2, check whether there exist frequency dependent matrices $D_{TI}(\omega)$ and $G_{TI}(\omega)$ such that inequality (B.4) is satisfied $\forall \omega \in \mathbb{R}_+$ (see Section B.1.2.2). If the answer is yes, stop the algorithm since $\overline{\gamma} = (1+\epsilon)\underline{\gamma}$ is an upper bound on γ^* and γ^* has been computed with a precision ϵ. Otherwise, let $\tilde{\omega}$ be a frequency where (B.4) is not satisfied. Add $\tilde{\omega}$ to the grid and go back to step 2.*

Algorithm B.1 eliminates most disadvantages of the classical grid-based techniques:

- the value of γ^* is obtained, and not only a lower bound,
- the validation performed at step 3 guarantees that no worst-case frequency is missed,
- to some extent, the size of the frequency grid is minimized, as well as the computational time.

1.2.2 Validation with a frequency elimination technique

At the end of step 2 of Algorithm B.1, the constant matrices D_{TV} and G_{TV} are **fixed**. Some matrices $D_{TI}(\omega_i)$ and $G_{TI}(\omega_i)$ are known, which satisfy inequality (B.4) at each point of the grid $(\omega_i)_i$ for $\gamma = \underline{\gamma}$. The objective is now to determine whether there exist matrices $D_{TI}(\omega)$ and $G_{TI}(\omega)$ such that inequality (B.4) holds on the whole frequency range for $\gamma = (1+\epsilon)\underline{\gamma}$. The following proposition reformulates this problem as a μ test, which can then be solved efficiently with Algorithm A.1.

Proposition B.3 *[Ferreres and Roos, 2007] Let $\tilde{D}_2 = D_2^{1/2}$ and $\tilde{G}_2 = D_2^{-1/2} G_2 D_2^{-1/2}$. Let:*

$$F(j\omega) = \left(\begin{bmatrix} I & 0 & 0 \\ 0 & \tilde{D}_2 & 0 \\ 0 & 0 & I \end{bmatrix} M_\gamma^*(j\omega) \begin{bmatrix} I & 0 & 0 \\ 0 & \tilde{D}_2^{-1} & 0 \\ 0 & 0 & I \end{bmatrix} - j \begin{bmatrix} 0 & 0 & 0 \\ 0 & \tilde{G}_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right) \begin{bmatrix} I & 0 & 0 \\ 0 & (I + \tilde{G}_2^2)^{-\frac{1}{2}} & 0 \\ 0 & 0 & I \end{bmatrix}$$

The following two assertions are equivalent:

1. *There exist matrices $D_{TI}(\omega)$ and $G_{TI}(\omega)$ satisfying inequality (B.4) for all $\omega \in \mathbb{R}_+$, where $D(\omega) = \mathrm{diag}(D_{TI}(\omega), D_{TV}, I_p)$, $G(\omega) = \mathrm{diag}(G_{TI}(\omega), G_{TV}, 0_p)$ and $\gamma = (1+\epsilon)\underline{\gamma}$.*

2. *There exist matrices $D_{TI}(\omega)$ and $G_{TI}(\omega)$ satisfying for all $\omega \in \mathbb{R}_+$:*

$$\overline{\sigma}\left(\left(\tilde{D}(\omega)F(j\omega)\tilde{D}^{-1}(\omega) - j\tilde{G}(\omega)\right)\left(I + \tilde{G}^2(\omega)\right)^{-\frac{1}{2}}\right) \leq 1 \tag{B.6}$$

where $\tilde{D}(\omega) = \mathrm{diag}(D_{TI}(\omega), I_{n_v}, I_p)$, $\tilde{G}(\omega) = \mathrm{diag}(G_{TI}(\omega), 0_{n_v}, 0_p)$ and $\gamma = (1+\epsilon)\underline{\gamma}$.

The first assertion is the one to be checked at step 3 of Algorithm B.1. The second one consists of checking whether $\mu_{\mathbf{\Delta_c}}(F(j\omega)) \leq 1$ for all $\omega \in \mathbb{R}_+$, *i.e.* whether the robust stability margin of $F(s)$ is larger than 1. The corresponding block structure is $\mathbf{\Delta_c} = \mathrm{diag}(\mathbf{\Delta_{TI}}, \mathbb{C}^{(\mathbf{n_v}+\mathbf{p})\times(\mathbf{n_v}+\mathbf{p})})$, where n_v and p are the size of Δ_{TV} and of the performance channel, *i.e.* the map between e and y, respectively. Algorithm A.1 can now be applied with $\beta_{max} = 1$ as an initialization. It is interrupted if β_{max} becomes strictly larger than 1, which means that a worst-case frequency has been found such that the test (B.6) cannot be satisfied. This frequency is added to the grid and another iteration of Algorithm B.1 is performed.

1.2.3 Convergence of the algorithm

Algorithm B.1 converges in a finite number of iterations. The proof is quite technical and it is omitted here, but all details can be found in [Ferreres and Roos, 2007]. Moreover, it has been observed on several practical applications that the number of iterations is usually quite low.

1.2.4 A suboptimal but faster algorithm

The LMI optimization performed at step 2 of Algorithm B.1 can be very demanding. Moreover, the number of decision variables and the computational time increase with the number of frequencies, since $D_{TI}(\omega)$ and $G_{TI}(\omega)$ have to be determined at each point of the grid. So instead of performing a single LMI step, it can be worthwhile to compute $D_{TI}(\omega)$ and $G_{TI}(\omega)$ using a gradient descent algorithm as it is classically done when computing μ upper bounds (see Section A.3.1). In this context, a suboptimal but faster algorithm is introduced, which consists of iteratively optimizing with respect to the frequency dependent matrices $D_{TI}(\omega)$ and $G_{TI}(\omega)$ on the one hand, and to the constant matrices D_{TV} and G_{TV} on the other hand.

Algorithm B.2 (suboptimal computation of an upper bound on k_{i2})

1. *Define an initial frequency grid $(\omega_i)_i$ as for Algorithm B.1. Let $D_{TV} = I_{n_v}$ and $G_{TV} = 0_{n_v}$.*

2a. *Let D_{TV} and G_{TV} be fixed. At each frequency ω_i, minimize γ with respect to $D_{TI}(\omega_i)$ and $G_{TI}(\omega_i)$ under the constraint (B.4). This can be achieved by performing a dichotomy search on γ and iteratively applying the gradient descent algorithm of [Young* et al.*, 1995].*

2b. *Let $D_{TI}(\omega_i)$ and $G_{TI}(\omega_i)$ be fixed. Minimize γ with respect to D_{TV} and G_{TV} such that (B.4) is satisfied at all grid points. This can be achieved by stacking all constraints into a single one and then using an LMI solver. If the decrease in γ after steps 2a and 2b have been applied is larger than a given threshold, return to step 2a to go on with the γ minimization process. Otherwise, let $\underline{\gamma}$ be the minimized value of γ, and continue to step 3.*

3. *Same as step 3 of Algorithm B.1.*

Unlike Algorithm B.1, the value of $\underline{\gamma}$ at the end of iteration 2b is not guaranteed to be a lower bound of γ^*. Thus it is not possible to quantify the accuracy of the final upper bound $\overline{\gamma}$ on γ^*. Nevertheless, Algorithms B.1 and B.2 can be combined to determine guaranteed lower and upper bounds of γ^* with a given accuracy ϵ, while keeping a reasonable computational time.

Algorithm B.3 (combination of optimal and suboptimal algorithms)

1. *Perform Algorithm B.2 to determine an upper bound $\overline{\gamma}$ of γ^*.*
2. *Perform Algorithm B.1 to compute a lower bound $\underline{\gamma}$ of γ^* and stop as soon as $\underline{\gamma} > (1-\xi)\overline{\gamma}$, where $\xi > 0$ is the desired accuracy. Algorithm B.1 can be initialized with the frequency grid and the values of D_{TV} and G_{TV} obtained at the end of step 1. To reduce the computational time, this frequency grid can be reduced by keeping only critical frequencies corresponding to peak values of the μ upper bound.*

These algorithms have been applied in [Ferreres and Roos, 2007] to a realistic missile example with 2 time-varying parameters and 8 time-invariant uncertainties, some of them being repeated up to 6 times. All algorithms perform well, and the suboptimal one allows to decrease the computational time with no loss of accuracy.

1.3 Extension to robust feedforward design

In many industrial applications, badly damped flexible modes and interactions among the various degrees of freedom of the system can result in large tracking errors. Although this can be accounted for by an inverse-based feedforward controller, flexible modes are often highly uncertain, which may result in a large degradation of performance caused by the feedforward components [Devasia, 2002]. This is a strong motivation to design robust feedforward controllers for systems that are affected by structured time-invariant and time-varying uncertainties. In this perspective, Proposition B.2 is now adapted, so that the methodology developed in Section B.1.2 can be further used.

1.3.1 Problem statement

Let us consider the interconnection of Figure B.2, where $N(s)$ is a continuous-time stable and proper real-rational transfer function representing the nominal closed-loop system. The feedback controller is fixed and included into $N(s)$. The objective is to compute a feedforward controller $H(s)$ which minimizes the induced $\mathcal{L}_2$ norm $\|T_{e\to y}\|_{i2}$ of the map from e to y when $\Delta = \mathrm{diag}(\Delta_{TI}, \Delta_{TV})$ takes all possible values in $\mathcal{B}_{\mathbf{\Delta}}$.

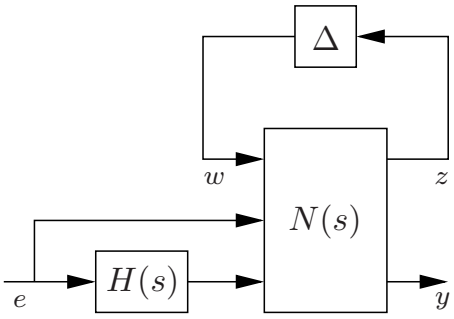


Figure B.2: Interconnection for robust feedforward design

1.3.2 Practical solution

The design scheme of Figure B.2 is first transformed into the interconnection of Figure B.1(b): if $N(s) = \begin{bmatrix} N_{11}(s) & N_{12}(s) & N_{13}(s) \\ N_{21}(s) & N_{22}(s) & N_{23}(s) \end{bmatrix}$, then $M(s) = \begin{bmatrix} N_{11}(s) & N_{12}(s) + N_{13}(s)H(s) \\ N_{21}(s) & N_{22}(s) + N_{23}(s)H(s) \end{bmatrix}$. Proposition B.2 can now be applied to $M(s)$. However, the considered feedforward design problem is more complex than the worst-case performance problem solved in Section B.1.2, since inequality (B.4) is no longer convex. The first step to recover convexity in order to apply the algorithms of Section B.1.2 is to set:

$$H(s) = \sum_{i=1}^{n_H} \theta_i H_i(s) \tag{B.7}$$

where the filters $H_i(s)$ are fixed, while the real parameters θ_i are to be optimized. The orthonormal filters basis of [Akcay and Ninness, 1999] is chosen to reduce numerical issues:

$$H_i(s) = \frac{\sqrt{2\Re(a_i)}}{s + a_i} \prod_{k=1}^{i-1} \frac{s - \overline{a_k}}{s + a_k} \tag{B.8}$$

where $\overline{a_k}$ is the conjugate of a_k. So both the order n_H and the poles $-a_k$ of the feedforward controller $H(s)$ must be fixed, but there is no constraint on the choice of n_H. Proposition B.2 can then be reformulated as follows.

Proposition B.4 *[Ferreres and Roos, 2007] Let $\gamma > 0$. If there exist real parameters $(\theta_i)_{1 \le i \le n_H}$ and matrices $D(\omega) = \mathrm{diag}(D_{TI}(\omega), D_{TV})$ and $G(\omega) = \mathrm{diag}(G_{TI}(\omega), G_{TV})$ such that the following relation holds for all $\omega \in \mathbb{R}_+$:*

$$\begin{bmatrix} D - N_{11}DN_{11}^* + j\left(N_{11}G - GN_{11}^*\right) & -(N_{11}D + jG)N_{21}^* & N_{12} + N_{13}\sum_{i=1}^{n_H}\theta_i H_i \\ \star & \gamma I - N_{21}DN_{21}^* & N_{22} + N_{23}\sum_{i=1}^{n_H}\theta_i H_i \\ \star & \star & \gamma I \end{bmatrix} \ge 0 \tag{B.9}$$

where $\star$ denotes the conjugate part of the hermitian matrix and where the dependence on ω is removed to simplify notation, then $\|T_{e\to y}\|_{i2} \le \gamma$ for all mixed time-invariant and time-varying uncertainties $\Delta \in \mathcal{B}_{\mathbf{\Delta}}$.

The proof of Proposition B.4 is omitted here for the sake of brevity, but it shows that inequalities (B.4) and (B.9) are equivalent. The latter being convex in $D(\omega)$, $G(\omega)$ and θ_i, il can be handled by an LMI solver. All the algorithms detailed in Section B.1.2 can thus be applied. The only difference concerns step 2, which now consists of solving the optimization problem of Proposition B.4 with respect to the frequency dependent matrices $D_{TI}(\omega), G_{TI}(\omega)$ and the constant matrices D_{TV}, G_{TV} as previously, but also to the real parameters θ_i of the feedforward controller.

Remark B.3 *Another method is proposed in [Giusto and Paganini, 1999], which does not require to write $H(s)$ as in (B.7)-(B.8). Frequency responses $H(j\omega_i)$ are first computed independently at each point of a frequency grid $(\omega_i)_i$. They are then approximated so as to obtain a transfer function $H(s)$. However, this last step is a delicate one, as a high-order feedforward controller may be necessary to get a good approximation. Moreover, there is no guarantee that performance is satisfactory on the whole frequency range.*

Remark B.4 *Another design method allowing to consider both time-invariant and time-varying uncertainties is proposed in [Köse and Scherer, 2009; Scorletti and Fromion, 2006]. The main advantage is that the poles of the feedforward controller do not have to be chosen a priori. The counterpart is that the scaling matrices are restricted to a set of real rational transfer functions, whose order and denominator are fixed. Moreover, the design problem is convex only in case a full-order feedforward controller is computed. The order of $H(s)$ is thus greater than or equal to the order of the closed-loop system $N(s)$, which can be quite large. A strategy could be to combine this method and the one presented in this section, in the spirit of what is proposed in Section D.2 to design anti-windup controllers. A full-order feedforward controller is first designed. Its poles are analyzed and the ones which are too slow or too fast compared to the system dynamics are eliminated. The remaining ones are then used to define the filters $H_i(s)$ in equation (B.8).*

1.4 Conclusion

A practical approach is proposed in this section to compute guaranteed stability margins and performance levels in the presence of mixed time-invariant and arbitrarily fast time-varying uncertainties. It is shown that μ-analysis can be used both to handle the time-invariant uncertainties with a low computational time, and to validate on the whole frequency range the results initially obtained on a rough frequency grid. A convex extension to robust feedforward design is also highlighted. An application to realistic missile models is reported in [Ferreres and Roos, 2007], which shows that computational time remains very reasonable despite the complexity of the problem.

Another method (not detailed in this manuscript due to space constraints) is proposed in [Roos and Biannic, 2006a] to address exactly the same issue. A state-space characterization of the robust stability margin k_r is obtained thanks to an extended version of the KYP lemma. It involves the search for an extended Lyapunov matrix $Z = R + jS$ where $R = R^*$ and $S = S^*$, and for scaling matrices D and G that are assumed to be constant but nevertheless complex in the case of time-invariant uncertainties. Applications to several benchmarks of Table A.2 show that almost non-conservative results are obtained in the purely time-invariant case. Moreover, numerical results and computational time are similar to the ones obtained with the method of Section B.1 on the aforementioned missile example. Extensions to robust feedforward design, as well as to combined robust feedback/feedforward design, are also described in [Roos and Biannic, 2006b].

Finally, it is worth being emphasized that both approaches are complementary. On the one hand, the frequency-domain technique described in Section B.1 is adapted to high-order systems with a limited number of uncertainties. On the other hand, the time-domain method of [Roos and Biannic, 2006a] is better suited to low-order systems with a large number of uncertainties.

2 A Lyapunov based approach

2.1 Problem statement

The framework is roughly the same as in Section B.1.1. Let us consider the standard interconnections of Figure B.1. $M(s)$ is a continuous-time stable and proper real-rational transfer function representing the nominal closed-loop system. Δ is a linear time-varying memoryless multiplication operator, *i.e.* $w(t) = \Delta(t)z(t)$ for all $t \geq 0$. It has a block-diagonal structure of the form:

$$\Delta(t) = \mathrm{diag}(\delta^{[1]}(t)I_{n_1}, \ldots, \delta^{[N]}(t)I_{n_N}) \tag{B.10}$$

where $n_v = \sum_{i=1}^{N} n_i$ and each $\delta^{[i]}$ is a real-valued scalar function corresponding to a time-varying parameter or uncertainty. The set of all such operators is denoted by $\mathbf{\Delta}$. It is assumed in the sequel that Δ is bounded:

$$\|\Delta\|_{i2} \leq 1 \tag{B.11}$$

which means that $\delta(t) = (\delta^{[1]}(t), \ldots, \delta^{[N]}(t))$ belongs to the unit hypercube $\mathcal{I}$ for all $t \geq 0$. Note that any bounded operator (as defined above) can be normalized. Thus, imposing $\delta^{[k]}(t) \leq 1$ for all k is not restrictive. It is also assumed that the rate of variation of each $\delta^{[k]}$ is bounded:

$$\dot{\delta}(t) \in \mathcal{J} = [\underline{\nu}^{[1]}, \overline{\nu}^{[1]}] \times \cdots \times [\underline{\nu}^{[N]}, \overline{\nu}^{[N]}] \tag{B.12}$$

for all $t \geq 0$. The set of all operators $\Delta \in \mathbf{\Delta}$ such that $\delta(t) \in \mathcal{I}$ and $\dot{\delta}(t) \in \mathcal{J}$ for all $t \geq 0$ is denoted by $\mathcal{B}_{\mathbf{\Delta}}$ (same definition as in Section B.1.1 plus the constraint on $\dot{\delta}$).

Remark B.5 *The method proposed here can be applied to a wider class of operators (including for example time-invariant uncertainties as in Section B.1.1), but extensions are not presented here for the sake of simplicity.*

Two analysis problems are considered: robust stability and worst-case performance. The latter exactly corresponds to Problem B.2. The former is a slight variation of Problem B.1:

Problem B.3 (robust stability check) *Check whether the interconnection of Figure B.1(a) is stable for all operators $\Delta \in \mathcal{B}_{\mathbf{\Delta}}$, i.e. whether the robust stability margin k_r is larger than 1.*

As already emphasized in the introduction, these problems can be addressed in different ways. The first one consists of using IQC to characterize the uncertainties. The calculations can then be carried out in the time domain using the KYP lemma, or in the frequency domain by generalizing the method of Section A.1: the time-invariant uncertainties are handled by μ-analysis based tools, while the IQC characterization of [Helmersson, 1999] is used for time-varying ones. The second way, proposed in [Roos *et al.*, 2012] and summarized here, is to use PDL functions. Rather than introducing some relaxation variables or trying to deal with an infinite dimensional problem, a two-step procedure is proposed in Section B.2.2 to solve Problem B.3 with a reasonable computational cost. The stability conditions are first formulated in Section B.2.2.1 as an LMI feasibility problem involving the search of a suitable parameter-dependent Lyapunov function. This problem is then solved in Section B.2.2.2 for a finite number of parametric configurations and the validity of the resulting Lyapunov function is checked on the whole parametric domain using a μ-analysis based test in Section B.2.2.3. In case the validation fails, a worst-case configuration is determined using particle swarm optimization in Section B.2.2.4 and the whole process is repeated using an augmented set of parametric configurations. Extension of the method to performance analysis is finally described in Section B.2.3.

2.2 Stability analysis

2.2.1 LMI-based formulation of the stability analysis problem

Let $(\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D})$ denote a state-space representation of M. The dynamics of the interconnection depicted in Figure B.1(a) are described for all $t \geq 0$ by:

$$\dot{x} = A(\delta(t))x \tag{B.13}$$

where:

$$A(\delta(t)) = \mathcal{A} + \mathcal{B}\Delta(t)(I - \mathcal{D}\Delta(t))^{-1}\mathcal{C} \tag{B.14}$$

Assumption B.1 *$A(\delta)$ is well-posed $\forall \delta \in \mathcal{I}$, i.e. $\mu_{\mathbf{\Delta}}(\mathcal{D}) < 1$.*

Lemma B.1 *Stability of system (B.13) is guaranteed for all operators $\Delta \in \mathcal{B}_{\mathbf{\Delta}}$ if there exists a parameter-dependent Lyapunov function $P(\delta)$ such that $\forall \delta \in \mathcal{I}$:*

$$P(\delta) = P(\delta)^T > 0 \tag{B.15}$$

and $\forall (\delta, \nu) \in \mathcal{I} \times \mathcal{J}$:

$$\Phi(\delta, \nu) = A(\delta)^T P(\delta) + P(\delta) A(\delta) + \sum_{k=1}^{N} \nu^{[k]} \frac{\partial P(\delta)}{\partial \delta^{[k]}} < 0 \tag{B.16}$$

Lemma B.1 is a direct consequence of the Lyapunov's stability theorem in which the Lyapunov function is defined as:

$$V(x, \delta) = x^T P(\delta) x \tag{B.17}$$

Indeed, inequality (B.16) clearly implies that $\dot{V}$ is negative along any trajectory of the plant (characterized by $\delta(t) \in \mathcal{I}$ and $\dot{\delta}(t) \in \mathcal{J}$ for all $t \geq 0$).

Unfortunately, this characterization of $P(\delta)$ is numerically intractable, since it involves an infinite number of both variables and constraints. The first problem is easily solved by restricting the parameter dependence to a polynomial expression:

$$P(\delta) = \sum_{i=1}^{x(d,N)} f_i(\delta) P_i \tag{B.18}$$

where $x(d, N)$ denotes the dimension of the vector space of all N-variate polynomial functions of degree d and $(f_i)_{1 \leq i \leq x(d,N)}$ is any associated basis, while $(P_i)_{1 \leq i \leq x(d,N)}$ are constant symmetric matrices to be determined. Using this polynomial expression, finding a PDL function $P(\delta)$ which satisfies inequalities (B.15) and (B.16) boils down to a standard LMI feasibility problem, which can be stated as follows. Do there exist symmetric matrices $(P_i)_{1 \leq i \leq x(d,N)}$ such that:

$$\Psi(\delta) < 0 \quad \forall \delta \in \mathcal{I} \tag{B.19}$$

where:

$$\Psi(\delta) = \mathrm{diag}\left(- \sum_{i=1}^{x(d,N)} f_i(\delta) P_i, \Phi(\delta, \nu_1), \ldots, \Phi(\delta, \nu_{2^N}) \right) \tag{B.20}$$

and for all $j \in [1\,,\, 2^N]$:

$$\Phi(\delta, \nu_j) = \sum_{i=0}^{x(d,N)} \left(f_i(\delta)(A(\delta)^T P_i + P_i A(\delta)) + \sum_{k=1}^{N} \nu_j^{[k]} \frac{\partial f_i(\delta)}{\partial \delta^{[k]}} P_i \right) \tag{B.21}$$

In equation (B.21), $\nu_j^{[k]}$ is equal to either $\underline{\nu}^{[k]}$ or $\overline{\nu}^{[k]}$ for each $k \in [1\,, N]$, thus leading to 2^N combinations. Indeed, equation (B.16) is linear in the variables $\nu^{[k]}$, which means that it is satisfied for all $\nu \in \mathcal{J}$ if and only if it is satisfied on the 2^N vertices of $\mathcal{J}$.

2.2.2 Grid-based resolution

Let us now focus on the second problem related to the infinite number of constraints. Two options are typically available here. The first one consists of using the Kalman-Yakubovic-Popov lemma [Iwasaki and Hara, 2005]. By introducing some additional variables, this technique allows

to convert the infinite set of parameterized inequalities (B.19) into a unique LMI. Unfortunately, as soon as either the order of the plant or the degree of the polynomial Lyapunov function increases, the latter becomes too large for existing softwares. The second option is definitely more intuitive. It simply consists in gridding the hypercube $\mathcal{I}$, so as to replace (B.19) with:

$$\Psi(\delta_l) < 0 \quad \forall\, l \in [1\,,\, M] \tag{B.22}$$

where δ_l denotes the value of δ at the l^{th} grid point. Checking whether there exists a PDL function which satisfies (B.22) is numerically tractable. Moreover, no conservatism is introduced, since (B.19) clearly implies (B.22). Of course, the converse is not necessarily true. The validity of the PDL function must then be checked *a posteriori* between the grid points.

2.2.3 Validity of the parameter-dependent Lyapunov function

At this stage, a PDL function $P(\delta)$ has been computed such that $\Psi(\delta_l) < 0$ for all $l \in [1\,,\, M]$. The objective is now to check whether this function also satisfies $\Psi(\delta) < 0$ for all $\delta \in \mathcal{I}$. By combining (B.14), (B.20) and (B.21), it can be easily checked that $\Psi(\delta)$ depends rationally on the $\delta^{[k]}$ and can thus be written in linear fractional form as follows (see Chapter C):

$$\Psi(\delta) = L_{22} + L_{21}\widehat{\Delta}(I - L_{11}\widehat{\Delta})^{-1}L_{12} \tag{B.23}$$

where $\widehat{\Delta} = \mathrm{diag}(\delta^{[1]}I_{\widehat{n}_1}, \ldots, \delta^{[N]}I_{\widehat{n}_N})$ is a structured matrix. Let us now assume that $\Psi(0) < 0$. Using a continuity argument, there exists $\epsilon > 0$ such that for all $\delta \in \mathbb{R}^N$ which satisfy $\|\delta\|_\infty < \epsilon$, then $\Psi(\delta) < 0$ and thus $\det(\Psi(\delta)) \neq 0$. The following lemma is introduced in this context.

Lemma B.2 *Let $\Psi(\delta)$ be defined by equation (B.23). Assume that $\Psi(0) = L_{22} < 0$. Let $X = L_{11} - L_{12}L_{22}^{-1}L_{21}$. The following two statements are equivalent:*

1. $\det(\Psi(\delta)) \neq 0$
2. $\det(I - X\widehat{\Delta}) \neq 0$

The largest value of ϵ for which $\det(I - X\widehat{\Delta}) \neq 0$ for all $\delta \in \mathbb{R}^N$ such that $\|\delta\|_\infty < \epsilon$ is by definition equal to $1/\mu_{\widehat{\mathbf{\Delta}}}(X)$, where $\mu_{\widehat{\mathbf{\Delta}}}(X)$ denotes the structured singular value of X. If $\mu_{\widehat{\mathbf{\Delta}}}(X) < 1$, then (B.19) is satisfied and the interconnection of Figure B.1(a) is stable for all $\Delta \in \mathbf{\Delta}$, *i.e.* $k_r > 1$. The validity of the PDL function can thus be checked using a single structured singular value computation. In practice, computing the exact value of $\mu_{\widehat{\mathbf{\Delta}}}(X)$ is known to be NP-hard in the general case (see Chapter A), but both upper [Young *et al.*, 1995] and lower [Seiler *et al.*, 2010] bounds $\overline{\mu}_{\widehat{\mathbf{\Delta}}}(X)$ and $\underline{\mu}_{\widehat{\mathbf{\Delta}}}(X)$ can be determined using polynomial-time algorithms. The following strategy can thus be implemented:

1. If $\overline{\mu}_{\widehat{\mathbf{\Delta}}}(X) < 1$, then $\mu_{\widehat{\mathbf{\Delta}}}(X) < 1$ and stability is proved by the current PDL function.
2. If $\underline{\mu}_{\widehat{\mathbf{\Delta}}}(X) > 1$, then $\mu_{\widehat{\mathbf{\Delta}}}(X) > 1$ and stability cannot be proved by the current PDL function, since there exists at least one value $\tilde{\delta_0} \in \mathcal{I}$ such that $\Psi(\tilde{\delta_0})$ is not negative definite.
3. If $\underline{\mu}_{\widehat{\mathbf{\Delta}}}(X) < 1$ and $\overline{\mu}_{\widehat{\mathbf{\Delta}}}(X) > 1$, no conclusion can be drawn. In such a case, a recursive algorithm is implemented. It consists in splitting the unit hypercube $\mathcal{I}$ in more and more subsets until either the μ lower bound computed on one of the subsets becomes larger than 1 (stability cannot be proved) or the highest μ upper bound computed on all subsets becomes lower than 1 (stability is proved). Such an algorithm is described in [Roos and Biannic, 2010] and in Section A.4. It is guaranteed to converge if the size of the subsets is allowed to become arbitrarily small.

2.2.4 Determination of a worst-case configuration

In case the PDL function determined in Section B.2.2.2 does not satisfy $\Psi(\delta) < 0$ for all $\delta \in \mathcal{I}$, a worst-case parametric configuration $\tilde{\delta}$ is determined, for which $\Psi(\tilde{\delta})$ is not negative definite. Intuitively, a relevant choice is to compute the value of δ for which the constraint $\Psi(\delta) < 0$ is the most violated, which can be formulated as follows:

$$\tilde{\delta} = \arg\max_{\delta \in \mathcal{I}} \overline{\lambda}(\Psi(\delta)) \tag{B.24}$$

where $\overline{\lambda}(\Psi(\delta))$ denotes the maximum eigenvalue of the real symmetric matrix $\Psi(\delta)$. Problem (B.24) is nonconvex, which makes it difficult to compute the global maximum. Nevertheless, the objective here is not to capture exactly this maximum, but to quickly explore the parametric domain $\mathcal{I}$ to find a configuration $\tilde{\delta}$ for which $\overline{\lambda}(\Psi(\tilde{\delta}))$ is as positive as possible. Particle swarm optimization [Clerc, 2006] is well adapted to address this issue. It is based on the swarm behavior of birds or fishes around food, and consists in moving a set of particles in the continuous parametric domain $\mathcal{I}$ according to simple mathematical formulas. The movement of each particle is influenced by its own best known position, and by the ones found by the neighboring particles. Such a strategy is expected to move the swarm towards the best solutions. It has been observed in practice that only a few iterations are necessary to find a suitable worst-case configuration. Once the latter has been computed, it is added to the grid, and a new PDL function is computed. Note that the search for a configuration $\tilde{\delta}$ such that $\overline{\lambda}(\Psi(\tilde{\delta})) > 0$ is necessarily successful. Indeed, the optimization is initialized with the configuration $\tilde{\delta}_0$ obtained during the μ lower bound computation, for which $\overline{\lambda}(\Psi(\tilde{\delta}_0)) \geq 0$ (see Section B.2.2.3).

2.2.5 Description of the algorithm

The results obtained in the previous sections are now combined. The following algorithm is introduced, which provides a solution to Problem B.1.

Algorithm B.4 (robust stability check)

1. *Select the initial degree d and the maximum degree d_{max} of the polynomial Lyapunov function. Choose the maximum number of iterations i_{max}.*
2. *Define an initial grid $\mathcal{G}_1 = (\delta_l)_{1 \leq l \leq M}$ of the unit hypercube $\mathcal{I}$. Set $i = 1$.*
3. *If $i < i_{max}$, solve the LMI feasibility problem (B.22). If $i = i_{max}$ or (B.22) is infeasible, then stop the algorithm if $d = d_{max}$ (**stability not proved**) or increase d and go back to step 2 otherwise.*
4. *Compute X and $\overline{\mu}_{\widehat{\mathbf{\Delta}}}(X)$. If $\overline{\mu}_{\widehat{\mathbf{\Delta}}}(X) < 1$, stop the algorithm (**stability proved**). Otherwise, compute $\underline{\mu}_{\widehat{\mathbf{\Delta}}}(X)$.*
   - → *If $\underline{\mu}_{\widehat{\mathbf{\Delta}}}(X) > 1$, solve problem (B.24). Set $\mathcal{G}_{i+1} = \mathcal{G}_i \cup \{\tilde{\delta}\}$, increase i and go back to step 3.*
   - → *If $\underline{\mu}_{\widehat{\mathbf{\Delta}}}(X) < 1$, apply the recursive algorithm described at the end of Section B.2.2.3. If the highest μ upper bound computed on all subsets is lower than 1, stop the algorithm (**stability proved**). If the μ lower bound computed on one of the subsets is larger than 1 or if the stopping criterion is reached and no conclusion can still be drawn, then stop the algorithm if $d = d_{max}$ (**stability not proved**) or increase d and go back to step 2 otherwise.*

Remark B.6 *The numerically most demanding step in Algorithm B.4 is clearly the third one. An LMI feasibility problem has to be solved, whose complexity increases with the degree of the PDL function and the number of grid points. To reduce the computational cost, it is then recommended to start with a low-order PDL function at step 1 ($d = 1$) and a rough grid at step 2 ($M = 1$).*

2.3 Extension to performance analysis

The method described in Section B.2.2 can be easily adapted to solve Problem B.2. Let $\left(\mathcal{A}, [\mathcal{B}_1\ \mathcal{B}_2], \begin{bmatrix}\mathcal{C}_1\\ \mathcal{C}_2\end{bmatrix}, \begin{bmatrix}\mathcal{D}_{11} & \mathcal{D}_{12}\\ \mathcal{D}_{21} & \mathcal{D}_{22}\end{bmatrix}\right)$ denote a state-space representation of M. The dynamics of the interconnection depicted in Figure B.1(b) are described for all $t \geq 0$ by:

$$\dot{x} = A(\delta(t))x + B(\delta(t))e \tag{B.25}$$
$$y = C(\delta(t))x + D(\delta(t))e \tag{B.26}$$

where:

$$\begin{aligned}
A(\delta(t)) &= \mathcal{A} + \mathcal{B}_1\Delta(t)(I - \mathcal{D}_{11}\Delta(t))^{-1}\mathcal{C}_1\\
B(\delta(t)) &= \mathcal{B}_2 + \mathcal{B}_1\Delta(t)(I - \mathcal{D}_{11}\Delta(t))^{-1}\mathcal{D}_{12}\\
C(\delta(t)) &= \mathcal{C}_2 + \mathcal{D}_{21}\Delta(t)(I - \mathcal{D}_{11}\Delta(t))^{-1}\mathcal{C}_1\\
D(\delta(t)) &= \mathcal{D}_{22} + \mathcal{D}_{21}\Delta(t)(I - \mathcal{D}_{11}\Delta(t))^{-1}\mathcal{D}_{12}
\end{aligned}$$

Lemma B.3 *Under the assumption that $x(0) = 0$, robust stability of system (B.25)-(B.26) is guaranteed and γ is an upper bound on k_{i2} if there exists a parameter-dependent Lyapunov function $P(\delta)$ such that $\forall \delta \in \mathcal{I}$:*

$$P(\delta) = P(\delta)^T > 0 \tag{B.27}$$

and $\forall (\delta, \nu) \in \mathcal{I} \times \mathcal{J}$:

$$\Phi(\delta,\nu) = \begin{bmatrix} A(\delta)^T P(\delta) + P(\delta)A(\delta) + \sum_{k=1}^{N} \nu^{[k]} \frac{\partial P(\delta)}{\partial \delta^{[k]}} + C(\delta)^T C(\delta) & P(\delta)B(\delta) + C(\delta)^T D(\delta) \\ B(\delta)^T P(\delta) + D(\delta)^T C(\delta) & -\gamma^2 I + D(\delta)^T D(\delta) \end{bmatrix} < 0 \tag{B.28}$$

Algorithm B.4 can be applied with minor modifications. The optimization problem at step 3 now consists of minimizing the linear objective γ under the LMI constraints (B.22), where $\Psi(\delta)$ is defined as in (B.20) and $\Phi(\delta, \nu_j)$ is obtained by combining (B.28) and (B.18). Moreover, a parameter $\alpha > 0$ can be introduced to facilitate the validation of the PDL function and to improve the convergence of the algorithm: the value of γ determined at step 3 is simply replaced with $(1 + \alpha)\gamma$ at step 4. The increase in conservatism remains reasonable if α is small enough.

2.4 Conclusion

A practical method is proposed to assess the robustness properties of a system in the presence of linear time-varying perturbations with bounded rates of variation. A parameter-dependent Lyapunov function is first optimized on a finite parametric grid. Its validity is then checked on the whole parametric domain using a fast and reliable μ-analysis based test. In case the validation fails, a worst-case configuration is added to the grid and the whole process is repeated. The proposed method is applied to a satellite in [Biannic *et al.*, 2011] and to a fighter aircraft in [Roos *et al.*, 2012]. Numerical results show that it compares favorably to IQC based analysis: for the same computational time, the results are less conservative. This demonstrates the benefit of using

μ-analysis to speed up the analysis process. Nevertheless, it should be kept in mind that an LMI problem must be solved at each iteration, which might lead to a prohibitive computational cost if the order of the system or the number of uncertainties is too large.

3 Summary of the contributions

The main contributions presented in this chapter are briefly summarized below, and a selection of related publications is given:

- **Section B.1:** frequency-domain approach to compute guaranteed stability margins and worst-case performance levels in the presence of mixed time-invariant and arbitrarily fast time-varying uncertainties ; extension to robust feedforward design [Ferreres and Roos, 2007].
- **Section B.1:** time-domain approach to compute guaranteed stability margins and worst-case performance levels in the presence of mixed time-invariant and arbitrarily fast time-varying uncertainties ; extensions to robust feedforward design and combined robust feedback/feedforward design [Roos and Biannic, 2006a; 2006b].
- **Section B.2:** Lyapunov-based technique to compute guaranteed stability margins and worst-case performance levels in the presence of time-varying uncertainties with bounded rates of variation [Roos *et al.*, 2012; Biannic *et al.*, 2011].

The guiding line between these different methods is to demonstrate that μ-analysis can play a key role in assessing the robustness properties of a system even if some uncertainties are time-varying, and to provide computational tools that can be applied to realistic applications.

Remark B.7 *It would be interesting to compare all existing techniques to solve the aforementioned issues, in terms of both accuracy and computational time. And also to evaluate their conservatism on a large set of real-world benchmarks as in Chapter A, by computing worst-case uncertainties (see e.g. [Köroğlu and Scherer, 2007; Peni and Seiler, 2016]).*

Chapter C

Generation of low-order LFR

A Linear Fractional Representation (LFR) is a model in which all known and fixed dynamics of a given system are put together in a linear time-invariant plant M, while the uncertain and varying parameters are stored in a block-diagonal operator Δ, as shown in Figure C.1. LFR modeling is an essential step before the robustness properties of uncertain closed-loop systems can be evaluated, *e.g.* using the μ-analysis based tools implemented in the SMART Library of the SMAC Toolbox (see Chapter A and [Roos, 2013]). It can also be useful to design robust control laws (especially using H_∞ approaches) or gain-scheduled controllers. But the efficiency of the aforementioned analysis and synthesis techniques strongly depends on the complexity of the considered LFR, which is measured in terms of both the size of the operator Δ and the order of the plant M. An increase in complexity is usually source of conservatism, and can even lead to numerical intractability.

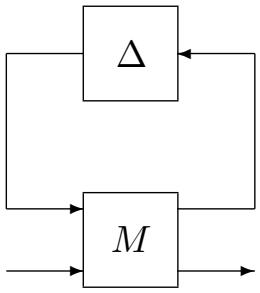


Figure C.1: Linear Fractional Representation

In most industrial applications, physical systems are described using a mix of nonlinear analytical expressions and tabulated data. Therefore, a two-step procedure has to be implemented to obtain a suitable LFR: a linear model with a polynomial or a rational dependence on the system parameters is first generated, and then converted into a linear fractional form. Several techniques such as object-oriented realization exist to perform the latter transformation. Although the minimality of the resulting LFR cannot be guaranteed, symbolic preprocessing techniques as well as numerical reduction usually permit to overcome complexity. Moreover, efficient software are available such as the LFR Toolbox for Matlab [Magni, 2006], which was recently replaced by the GSS Library of the SMAC Toolbox [Biannic and Roos, 2016]. On the other hand, the preliminary issue of converting the tabulated or irrational data into simple yet accurate polynomial or rational expressions has been paid much less attention, although it is of significant practical importance. In the aeronautic field for example, most aircraft models include

tabulated aerodynamic coefficients determined by CFD (Computational Fluid Dynamics), wind tunnel experiments or flight tests, and several controller gains depend on the flight parameters in a tabulated fashion.

The motivations for developing enhanced computational tools dedicated to tabulated data approximation are twofold. The first one is of physical nature. Computing sparse expressions, for which the number of terms in the numerator and the denominator is as low as possible, is a natural way to prevent data overfitting and to ensure a smooth behavior of the model between the points used for approximation. On the other hand, building an LFR from a polynomial or a rational expression $f(x^1, \ldots, x^n)$ results in a block diagonal matrix $\Delta = \mathrm{diag}(x^1 I_{p_1}, \ldots, x^n I_{p_n})$. The number p_j of repetitions of each parameter x^j in Δ is strongly linked to the number of occurrences of x^j in f. Indeed, although this is not an exact rule, the trend is as follows: the fewer the occurrences of x^j in $f(x^1, \ldots, x^n)$, the smaller the size of Δ. In other words, no matter how efficient the LFR generation tools can be, they are of little help if the expressions to be converted are unnecessarily complex. Hence, the need to get tractable LFR for analysis and design purposes is another strong motivation for generating sparse expressions.

The issue of converting tabulated or irrational data into polynomial or rational expressions is the core of this chapter. The case of scalar samples is considered in Section C.1. The problem is first stated in Section C.1.1 and an overview of existing approximation methods is provided in Section C.1.2. Those allowing to get sparse expressions are then detailed in Section C.1.3 and implementation issues to obtain a low-order static LFR are finally discussed in Section C.1.4. The case of a set of MIMO LTI models is considered next, and a whole methodology is described in Section C.2 to obtain a low-order dynamic LFR.

1 From a set of scalar samples to a static LFR

1.1 Problem statement

Let $\{y_k, k \in [1, N]\}$ be a set of samples (measurements, tabulated data...) corresponding to different parametric configurations or operating points $\{x_k, k \in [1, N]\}$ of a given system. More precisely, each $x_k = \left[x_k^1, \ldots, x_k^n\right] \in \mathbb{R}^n$ contains the values of the n explanatory variables $x^1, \ldots, x^n$ for which the sample $y_k \in \mathbb{R}$ was obtained. The main objective is to compute a polynomial or a rational function $f : \mathbb{R}^n \to \mathbb{R}$ of reasonable complexity which approximates these data, *i.e.* such that $f(x_k)$ is close to y_k for all $k \in [1, N]$ in the sense of a certain criterion (root-mean-square error, maximum local error...). Another objective is to ensure that the denominator of f has no root in the considered parametric domain, which is assumed here to be the smallest hyper-rectangle $\mathcal{D} \subset \mathbb{R}^n$ containing every sampled data $(x_k)_{k \in [1,N]}$. This requirement guarantees that well-posed LFR are obtained.

Remark C.1 *The case where an analytical expression $f_A : \mathbb{R}^n \to \mathbb{R}$ is available instead of N samples is not considered here (see e.g. [Petrushev and Popov, 1987]). Moreover, this work only deals with approximation (or regression) and not with interpolation, which would aim at finding a rational function f such that the equalities $f(x_k) = y_k$ are strictly satisfied for a large number N of samples (see [Floater and Hormann, 2007] and references therein).*

1.2 Overview of existing methods

For a given precision, an intuitive idea is to determine a rational function for which the numerator P and the denominator Q are two polynomials of the lowest possible degrees. This fairly simple strategy is followed by most existing methods. A classical linear least-squares

technique is usually applied in case the rational function is restricted to be polynomial [Magni, 2006]. In the general case, a nonlinear least-squares technique can be implemented to minimize the approximation error [The Mathworks, 2014], whereas a quadratic programming problem can be solved to ensure that the resulting rational function intersects a set of intervals containing the data [Celis *et al.*, 2007]. But all those techniques suffer from the same drawback: all admissible monomials of P and Q are usually nonzero, regardless of their real ability to model the data. More generally, the question of which terms should be included in the model is often addressed by trial-and-error, or even ignored in practice. A way to deal with this question is to use orthogonal least-squares. This allows to evaluate the ability of each monomial to efficiently model the data and therefore to select only the most relevant ones, leading to sparse expressions. This approach is applied in [Poussot-Vassal and Roos, 2012; Roos, 2009; Döll *et al.*, 2008; Morelli and DeLoach, 2003] to model aeronautical data with polynomials, but until recently, practical methods to compute rational functions were still missing. Yet, the additional degrees of freedom can lead to simpler expressions and thus to smaller LFR. Consequently, two algorithms have recently been developed to obtain rational expressions with sparse structure, *i.e.* with as few monomials in P and Q as possible. A direct approach computing a rational expression in a single step thanks to a symbolic regression technique is proposed in [Hardier *et al.*, 2013a], while an indirect approach creating a rational model by means of an intermediate surrogate modeling is introduced in [Hardier *et al.*, 2013b]. All these methods are now briefly presented in Sections C.1.2.1 and C.1.2.2.

1.2.1 Polynomial case

The most common approach consists in restricting f to be a **polynomial function**, that is:

$$f(x) = P(x) = \sum_{i=0}^{n_P} a_i r_i(x) \tag{C.1}$$

where $(r_i)_{i\in[0,n_P]}$ is a set of polynomial regressors and $(a_i)_{i\in[0,n_P]}$ are coefficients to be determined.

- A classical solution consists in solving a **linear least-squares (LS)** problem with respect to the coefficients $(a_i)_{i\in[0,n_P]}$, *i.e.* to minimize the following criterion [Magni, 2006]:

$$C = \sum_{k=1}^{N} [y_k - P(x_k)]^2 \tag{C.2}$$

- A well-known improvement to that approach relies on a preliminary orthogonalization process to decouple the regressors. As a result, the ability of each new regressor to reduce the criterion C can be evaluated regardless of those already selected. Hence, only the most relevant ones are considered, which amounts to a certain extent to minimizing the complexity of the polynomial approximation while still guaranteeing a low approximation error. This **orthogonal least-squares (OLS)** based variant was successfully applied by [Morelli and DeLoach, 2003]. It was later improved, allowing to compute a sparse polynomial expression satisfying the following global and local constraints:

$$\begin{cases} \sqrt{C/N} \leq \epsilon_1 \\ |y_k - P(x_k)| \leq \epsilon_2 \quad \forall k \in [1, N] \end{cases} \tag{C.3}$$

  where ϵ_1 and ϵ_2 are some user-defined tolerances [Poussot-Vassal and Roos, 2012; Roos, 2009; Döll *et al.*, 2008]. More details are given in Section C.1.3.1.

1.2.2 Rational case

The more general case where f is a **rational function** is now considered:

$$f(x) = \frac{P(x)}{Q(x)} = \sum_{i=0}^{n_P} a_i r_i^P(x) \Big/ \sum_{i=0}^{n_Q} b_i r_i^Q(x) \tag{C.4}$$

where $(r_i^P)_{i\in[0,n_P]}$ and $(r_i^Q)_{i\in[0,n_Q]}$ are two sets of polynomial regressors, while $(a_i)_{i\in[0,n_P]}$ and $(b_i)_{i\in[0,n_Q]}$ are coefficients to be determined.

- A first method consists of solving a **nonlinear least-squares (NLS)** problem with respect to the coefficients $(a_i)_{i\in[0,n_P]}$ and $(b_i)_{i\in[0,n_Q]}$, that is to minimize the following criterion:

$$C = \sum_{k=1}^{N} \left[y_k - \frac{P(x_k)}{Q(x_k)} \right]^2 \tag{C.5}$$

  It is notably implemented in the Curve Fitting Toolbox of Matlab [The Mathworks, 2014], where several optimization tools can be used to compute a solution (Levenberg-Marquardt algorithms, trust-region methods...). One of its major drawbacks is that several local minima may exist due to the non-convexity. Hence, the results strongly depend on the initialization, which is not a trivial issue.

- A second method was introduced by [Markov *et al.*, 1996] in the context of polynomial approximation and then generalized by [Celis *et al.*, 2007] to the rational case. An uncertainty interval $\left[\underline{y}_k, \overline{y}_k\right]$ is first defined around each y_k. A rational function is then determined which intersects all those intervals, *i.e.* $\underline{y}_k \leq P(x_k)/Q(x_k) \leq \overline{y}_k \ \forall k \in [1, N]$. This can be achieved by solving a **quadratic programming (QP)** problem in the coefficients $(a_i)_{i\in[0,n_P]}$ and $(b_i)_{i\in[0,n_Q]}$ with a strictly convex objective function.

Those two algorithms suffer from the same drawback: all admissible monomials of P and Q are usually nonzero, regardless of their real ability to model the data. In this context, two additional methods have recently been developed to generate sparse rational approximations, which avoid data overfitting and lead to simple yet accurate LFR.

- A third method proposed in [Hardier *et al.*, 2013a] and described in Section C.1.3.2 looks for a rational approximation in a single step thanks to a symbolic regression technique. **Genetic programming (GP)** is implemented to select sparse monomials and coupled with a nonlinear iterative procedure to estimate the coefficients of the rational function.

- A fourth method proposed in [Hardier *et al.*, 2013b] and described in Section C.1.3.3 creates a rational expression by means of an intermediate **surrogate modeling (SM)**. It performs the data approximation by building a sparse modeling based on neural networks, before translating the result into a fractional form. A stepwise selection algorithm is used, combining the benefits of forward orthogonal least-squares to estimate the regression parameters, and of particle swarm optimization to determine the best location of the regressors.

Remark C.2 *Rational approximation is also considered in linear systems theory, for example in the fields of model reduction [Karlsson and Lindquist, 2008] and system identification [Deschrijver* et al.*, 2007], where least-squares methods are usually implemented.*

1.3 Focus on sparse approximation techniques

Our main contribution consists of developing some approximation methods which produce **sparse polynomial or rational expressions**, with the ambition of limiting as much as possible the size of the resulting LFR. In this context, the emphasis is put in this section on the OLS, GP and SM methods presented in Section C.1.2. Note that the last two make use of techniques which are a little outside the main topics of this manuscript, but they are the result of a successful collaboration with a colleague specialized in modeling and identification. So it seems relevant to describe them briefly.

1.3.1 Polynomial approximation using orthogonal least-squares

Let us assume for the moment that the polynomial regressors $(r_i)_{i\in[0,n_P]}$ are fixed. A classic way of doing this is to choose n positive integers $d_1,\dots,d_n$ and to consider all monomials such that the degree of x^i is less than or equal to d_i, in which case $n_P=(d_1+1)\dots(d_n+1)-1$. The quadratic criterion (C.2) then only depends on the coefficients $(a_i)_{i\in[0,n_P]}$ and the optimization problem to be solved can be reformulated as:

$$A_{opt}=\underset{A\in\mathbb{R}^{n_P+1}}{\arg\min}\, C(A) \tag{C.6}$$

where:

$$C(A)=(Y-RA)^T(Y-RA) \tag{C.7}$$

$$R=\begin{bmatrix} r_0(x_1) & \dots & r_{n_P}(x_1)\\ \vdots & \ddots & \vdots\\ r_0(x_N) & \dots & r_{n_P}(x_N)\end{bmatrix}=\begin{bmatrix} R_0 & \dots & R_{n_P}\end{bmatrix} \tag{C.8}$$

$$A^T=\begin{bmatrix} a_0 & \dots & a_{n_P}\end{bmatrix} \tag{C.9}$$

$$Y^T=\begin{bmatrix} y_1 & \dots & y_N\end{bmatrix} \tag{C.10}$$

and $(R_i)_{i\in[0,n_P]}$ are called the modeling functions. This is a classical linear least-squares problem, whose solution is given by:

$$A_{opt}=(R^TR)^{-1}R^TY \tag{C.11}$$

Assume now that the modeling functions which compose the matrix R are made orthogonal, *i.e.* $R_i^TR_j=0$ for all $i\neq j$, as proposed in [Morelli and DeLoach, 2003] and [Roos, 2009]. The components of A_{opt} are then expressed as:

$$a_{i,opt}=\frac{R_i^TY}{R_i^TR_i}\quad,\quad i\in[0\,,n_P] \tag{C.12}$$

and the corresponding value C_{opt} of $C(A)$ is given by:

$$C_{opt}=Y^TY-\sum_{i=0}^{n_P}\frac{(R_i^TY)^2}{R_i^TR_i} \tag{C.13}$$

Using orthogonal modeling functions is relevant for several reasons. First, R^TR becomes diagonal and can be trivially inverted. Then, it can be observed in (C.13) that the reduction in the least-squares criterion $C(A)$ resulting from the inclusion of R_i does not depend on R_j whatever $j\neq i$. This allows to evaluate each orthogonal modeling function (an thus each monomial) in terms of its

ability to reduce $C(A)$, regardless of which other functions are selected. This property is exploited in [Roos, 2009; Poussot-Vassal and Roos, 2012], where an iterative algorithm is introduced. By progressively releasing some tolerances, the most relevant set of monomials is determined, for which the approximation error remains lower than a given user-defined threshold. All the other monomials are eliminated, thus leading to sparse polynomial expressions. This amounts to a certain extent to minimizing the complexity of the approximation formulas, and then the size of the resulting LFR. Some recent applications can be found *e.g.* in [Iannelli *et al.*, 2017; Biannic *et al.*, 2016].

1.3.2 Rational approximation using genetic programming

In the rational case, when the model structure has to be determined as a whole (numerator and denominator degrees, number and type of monomials), the approximation problem cannot generally be solved by means of classical techniques. Over a few variables, a sequential and systematic exploration cannot be carried out either: for example, with 2 explanatory variables and a maximum degree of 10, there are not less than 10^{15} rational candidates available. Moreover, the dual-purpose optimization (model structure and coefficients) is complicated by the fact that a rational model is no more Linear in its Parameters (LP). Fortunately, some promising techniques have recently appeared for global optimization, with the purpose of solving symbolic regression problems close to this one. This is especially the case of Genetic Programming (GP), and after a short description of its main principles, the way it can be adapted to rational approximation will be examined.

GP is part of the evolutionary family, as Genetic Algorithms (GA) are. It uses the principles of natural evolution to evolve a population of individuals randomly created, until a satisfactory solution is found. Opposite to GA, it is not based on a binary coding of information, but it uses a structured representation in the form of syntax trees. These trees are well adapted to solve structural or symbolic optimization problems, since they can have different sizes and shapes. The associated alphabet is also flexible enough to encode mathematical equations, behavior models or computer programs. The first works date back to the early 60s, but GP was really implemented and brought up to date only in the early 90s, thanks also to an increase in computing power [Koza and Poli, 2005]. An iterative process breeds a population and transforms the individuals generation after generation by applying Darwinian mechanisms: reproduction, mutation, crossover, but also gene duplication or deletion. These are applied to the hierarchically structured trees of the individuals, comprising a set of nodes which fall into two categories: the set $\mathbf{F}$ of internal nodes called *functions* or *operators*, and the set $\mathbf{T}$ of tree leaves called *terminals*.

All types of functions are acceptable in $\mathbf{F}$: mathematical $(+, -, \times, /, \sqrt{}, \exp \dots)$, logical, conditional (tests) or user-defined. On the other hand, the terminals correspond to the function arguments but can also include some internal parameters or predefined constants. The content of $\mathbf{T}$ is a central issue for the problem of a joint structural/parameter optimization. With LP models, a good choice is to take the regression coefficients away and to include only the explanatory variables x^i and possibly some predefined constants. Hence, the individuals just represent the functional relationships between the x^i. At each GP iteration, the regressor functions r_j (and their number $m \leq n_P + n_Q$) are derived by analyzing the tree structure of any individual from its root. The numerical values of the regression coefficients (a_j, b_j) are then adapted afterwards, by applying any minimization technique to the squared error. Moreover, coupling GP with an OLS algorithm allows to solve the optimization of the (a_j, b_j) very efficiently.

GP allows to produce polynomials by setting $\mathbf{F} = \{+, \times\}$ and $\mathbf{T} = \{x^0 = 1, x^1, x^2, \dots, x^n\}$, hence restricting the regressors to monomials. The modeling complexity can also be controlled

by penalizing some internal GP parameters like the tree depth, the number of branches/leaves, or by favoring the selection of the simplest operators. Practically, this can be achieved thanks to the fitness function which is used to handle the GP mechanisms of evolution. Similarly to what is done in ridge regression, a penalty component can be added to the fitness function to favor the simplest models and to prevent overfitting.

Let us now move on to the considered rational approximation problem. The following considerations come to mind:

- the rational case extends the polynomial one (structured modeling expressed as the quotient of two polynomials),
- GP is fully justified since there is no other classical option available for jointly optimizing the structure of the numerator and the denominator (*e.g.* a brute-force search does not minimize the number of monomials),
- although a rational model is not LP, the numerator and the denominator remain LP when considered separately, and it would be a pity not to take advantage of that.

Hence, GP appears as a promising alternative for rational modeling, but a prior adaptation of the method is required to use it with maximum efficiency. A dedicated tool named TRACKER (Toolbox for Rational Approximation Computed by Knowing Evolutionary Research) was thus developed, and it is briefly described below (see also [Hardier *et al.*, 2013a]).

Each component of the rational function (numerator or denominator) is represented by a single separate chromosome, which comes in a syntax tree form as usual and *a priori* includes several genes. The sets **T** and **F** are chosen as for the polynomial case, and a specific syntax rule is defined to ensure that all the non-terminal nodes located below a $\times$-type node are also $\times$-type nodes. This trick avoids creating useless branching, which could result in splitting and multiplying some monomials. Thanks to this architecture, a gene appears as a subtree linked to the root node of its chromosome through one or several $\times$-type nodes. A parse analysis of the different genes composing a chromosome also permits to avoid the creation of spurious genes by identifying and grouping them if any. Figure C.2 shows the architecture of a tree corresponding to a simple $f(x)$ example. 5 genes related to the different monomials are highlighted by colors (except the constants a_0 and b_0 which are an integral part of the structure).

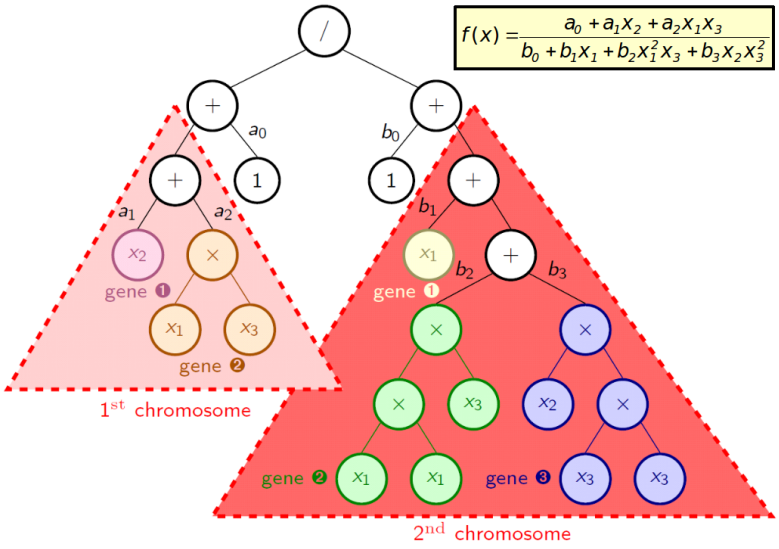


Figure C.2: Parse tree example for rational modeling

To solve the parametric optimization, *i.e.* to estimate the regression coefficients (a_j, b_j) of any created tree, a well-known technique used to identify transfer functions in the frequency domain is implemented [Sanathanan and Koerner, 1963]. It consists of iteratively linearizing the quadratic cost function C defined in equation (C.5) and computing estimates of (a_j, b_j) until convergence. At a given iteration, the coefficients b_i in the denominator are replaced by their most recent estimates $\hat{b}_i$, so they are fixed during optimization. And the denominator is arbitrarily normalized by choosing $b_0 = 1$ and $r_0^Q(x_k) = 1$. Therefore, the kth term of C becomes linear in (a_j, b_j) and reads:

$$\frac{\sum_{i=0}^{n_P} a_i r_i^P(x_k) - \sum_{i=1}^{n_Q} b_i y_k r_i^Q(x_k) - y_k}{1 + \sum_{i=1}^{n_Q} \hat{b}_i r_i^Q(x_k)} \tag{C.14}$$

Let $\hat{D}(x_k) = 1 + \sum_{i=1}^{n_Q} \hat{b}_i r_i^Q(x_k)$ be the estimate of the denominator. The vector of normalized outputs is written as $y^* = \left[y_1/\hat{D}(x_1) \dots y_N/\hat{D}(x_N)\right]^T$, and the kth row R_k of the regression matrix follows $\hat{D}(x_k)R_k = \left[1\ r_1^P(x_k) \dots r_{n_P}^P(x_k)\ -y_k r_1^Q(x_k) \cdots - y_k r_{n_Q}^Q(x_k)\right]$. With these notations, some estimates $\hat{w}$ of the coefficients $w = \left[a_0\ a_1 \dots a_{n_P}\ b_1 \dots b_{n_Q}\right]^T$ are obtained by solving a linear LS problem, whose solution is $\hat{w} = (R^T R)^{-1} R^T y^*$. This method relies on the fact that the approximation $D(x_k) \approx \hat{D}(x_k)$ becomes fully justified when the iterative process has converged. In practice, the latter is initialized with $\hat{D}(x_k) = 1$, and 2 or 3 iterations are usually sufficient to ensure convergence. In case of ill-conditioning, a few iterations of Levenberg-Marquardt optimization are used to recover a satisfactory result. Introduced into the selection process, this technique enables to evaluate the performance of each individual very easily, by coming down to a short series of ordinary LS problems. The overcost remains limited because the major part of the computations required by matrix R can be stored and reused through the loop.

1.3.3 Rational approximation using surrogate modeling

The use of surrogate modeling is becoming widespread in many scientific domains to replace the system or the reference model, when the latter is too restrictive to achieve some tasks like optimization, modeling, parameter identification... Hence, a wide range of techniques have been developed to build surrogate models efficiently, *i.e.* with both accuracy and parsimony. For example, Neural Networks (NN) are recognized nowadays as an efficient alternative to represent complex nonlinear systems, and methods are available to model static nonlinearities such as those encountered when confronted with the considered approximation problem. The idea consists of using such methods to derive a rational model. In this perspective, an existing tool has been adapted. It was initially developed for aircraft modeling and identification purposes and is named KOALA (Kernel Optimization Algorithm for Local Approximation), see [Bucharles *et al.*, 2012; Seren *et al.*, 2011] for more details.

Let us first briefly describe the standard version of KOALA. Forward selection is used to choose the regressors, starting with an empty set and adding them one at a time in order to gradually improve the approximation. A preliminary orthogonalization technique is used to speed up this constructive process, allowing to evaluate each regressor regardless of those previously selected. Particle Swarm Optimization (PSO) is then used to optimize the regressors positioning [Clerc, 2006]. The coupling of this PSO algorithm with the constructive technique

detailed in [Seren *et al.*, 2011] allows to perform structural and parametric optimizations jointly for different types of regressors, typically Radial Basis Function (RBF) networks, but also more general Local Linear Models (LLM) networks. The latter are obtained by replacing the RBF linear weightings w by affine expressions depending on the model inputs. It is thus expected that fewer regressors are required to achieve the same accuracy in most applications. For LLM, the following generic formulation is used for the approximating function f:

$$f(x) = \sum_{j=1}^{m} \left(\sum_{i=0}^{n} w_{ji} x^i \right) r_j(x) \tag{C.15}$$

where $r_j(x)$ represents the kernel value of the jth regressor function, and where $x^0 = 1$ to include the constant terms of the affine modeling into the second sum. By choosing Gaussian radial functions for r_j, the jth term of f becomes:

$$f_j(x) = \left(\sum_{i=0}^{n} w_{ji} x^i \right) \exp\left(- \sum_{i=1}^{n} \frac{(x^i - c_{ji})^2}{\sigma_{ji}^2} \right) \tag{C.16}$$

With this choice, KOALA optimizes both the ellipsoid centers c_{ji} and radii σ_{ji} associated to the radial functions (structure) and the weightings w_{ji} (parameters).

A first idea to use KOALA in the context of rational modeling is to convert equation (C.16) *a posteriori* into a rational form, *i.e.* once c_{ji}, σ_{ji} and w_{ji} have been computed. A Pade approximation of the exponential function is used, so as to replace it with a rational function in reduced form. The latter is expressed as the quotient of two polynomials of degrees p and q, and the corresponding approximation of $f_j(x)$ becomes a rational function of degrees $2p+1$ and $2q$ for each explanatory variable x^i. However, getting high quality approximations requires large values for q. Hence, the degree of the resulting rational function is penalized, with no guarantee about the accuracy of the global regression $f(x)$.

A more relevant approach proposed in [Hardier *et al.*, 2013b] consists of replacing the exponential function straight away with a Pade approximation, before optimizing c_{ji}, σ_{ji} and w_{ji} using this new kind of regressors. The simplest transformation corresponds to the reduced form of degrees $p = 0$ and $q = 1$. Therefore, the resulting approximating function f is the sum of m components given by:

$$f_j(x) = \left(\sum_{i=0}^{n} w_{ji} x^i \right) \Bigg/ \left(1 + \sum_{i=1}^{n} \frac{(x^i - c_{ji})^2}{\sigma_{ji}^2} \right) \tag{C.17}$$

This solution is preferred here, since it leads to simpler and more accurate rational functions. Hence, another class of models based on Pade approximation is included into KOALA, in addition to the aforementioned RBF and LLM kernels.

A post-processing of the resulting rational function f is finally performed before an LFR can be computed, using for example the Matlab Symbolic Toolbox. Several options can be considered for gathering the m components $f_j(x)$ into a single rational function: global expansions of numerator and denominator, factorization of the denominator, sum of elementary rational terms... The latter appears to be the most relevant one, since it allows some simplifications, and a factor of two can usually be gained in the final LFR size.

1.4 Implementation issues

1.4.1 Non singularity of the rational function

The coefficients $(b_i)_{i\in[1,n_Q]}$ being determined, an important issue is now to check whether the denominator Q of f has no roots in the considered parametric domain, *i.e.* $Q(x) \neq 0$ for all $x \in \mathcal{D}$. Such a requirement is indeed a prerequisite to build a well-defined LFR. Unfortunately, checking whether a multivariate polynomial of degree 4 or higher with real coefficients has a real zero is NP-hard in the general case [Blum *et al.*, 1998]. A classical strategy is therefore to focus on sufficient conditions, and some methods based on parameter-dependent slack variables [Sato, 2009] or sum-of-squares [Shor, 1987] have been proposed. They give quite accurate results, but they are likely to become computationally demanding if some multivariate polynomials of high degrees are considered.

An efficient alternative based on μ-analysis is proposed here. In this perspective, $Q(x)$ is first converted into an LFR as depicted in Figure C.1, where M is a fixed matrix and $\Delta = \text{diag}(x^1 I_{p_1}, \ldots, x^n I_{p_n}) \in \mathbb{R}^{p\times p}$. If $M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \in \mathbb{R}^{(p+1)\times(p+1)}$ is partitioned in accordance with this interconnection, the following relation holds:

$$Q(x) = M_{22} + M_{21}\Delta\left(I_p - M_{11}\Delta\right)^{-1} M_{12} \tag{C.18}$$

It is assumed here without loss of generality that $\mathcal{D}$ is the unit hypercube, *i.e.* $x^i \in [-1, 1]$ for all $i \in [1, n]$, which can always be achieved using a suitable affine transformation. The following technical lemma is then introduced, where $M_{22} = Q(0)$ is assumed to be nonzero (otherwise, it could directly be concluded that Q has a real zero in $\mathcal{D}$).

Lemma C.1 *Let $X = M_{11} - M_{12}M_{22}^{-1}M_{21}$. Then for all $x \in \mathcal{D}$:*

$$Q(x) \neq 0 \Leftrightarrow \det(I_p - X\Delta) \neq 0 \tag{C.19}$$

This result shows that the considered nonsingularity check is strongly linked to the notion of structured singular value (see Chapter A). Computing the exact value of $\mu_{\mathbf{\Delta}}(X)$ is NP-hard, but efficient polynomial-time algorithms allow to compute both lower and upper bounds $\underline{\mu}_{\mathbf{\Delta}}(X)$ and $\overline{\mu}_{\mathbf{\Delta}}(X)$ [Seiler *et al.*, 2010; Young *et al.*, 1995]. The following algorithm can then be implemented:

- If $\underline{\mu}_{\mathbf{\Delta}}(X) > 1$, a value $\tilde{x} \in \mathcal{D}$ has been computed, for which $Q(\tilde{x}) = 0$.
- Else, if $\overline{\mu}_{\mathbf{\Delta}}(X) < 1$, it can be concluded that $Q(x) \neq 0$ for all $x \in \mathcal{D}$.
- Otherwise, nothing can be assessed. A strategy is then to divide $\mathcal{D}$ recursively and to apply the whole procedure again on each subdomain until one of the two aforementioned scenarios occurs.

Remark C.3 *Such a procedure is heuristic. Nevertheless, its convergence properties appear quite good in practice. Indeed, it can usually be observed that the smaller a subdomain, the smaller the gap between the bounds (see [Roos and Biannic, 2010] for a detailed study of such a behavior). A conclusion is thus often obtained after only a few iterations, sometimes even after a single one.*

Each time a worst-case value $\tilde{x} \in \mathcal{D}$ is identified for which $Q(\tilde{x}) = 0$, the rational approximation algorithms presented in Section C.1.2.2 are applied again with the additional constraint $Q(\tilde{x}) \geq \epsilon > 0$ (where ϵ is a user-defined tolerance), until a nonsingular rational function is computed (or a maximum number of iterations is reached).

Remark C.4 *The non-singularity check presented here is useful only for the QP-based and the GP-based methods. Indeed, the SM-based approach described in Section C.1.3.3 cannot produce a singular solution, since the denominator of $f_j(x)$ in (C.17) is always strictly positive.*

1.4.2 A Matlab library for polynomial and rational approximation

Despite several theoretical contributions to polynomial and rational approximation, only few practical algorithms have been implemented. Those available are mainly based on linear least-squares in the polynomial case and nonlinear least-squares in the rational one [Magni, 2006; The Mathworks, 2014]. However, none of them is able to produce sparse expressions, where only the most relevant monomials are selected. In this context, the *Approximation of Polynomial and Rational-type for Indeterminate Coefficients via Optimization Tools* (APRICOT) Library of the SMAC Toolbox has been developed to help bridging this gap between theory and practice [Roos *et al.*, 2014]. As shown in Table C.1, it is composed of five main approximation routines, which implement most of the techniques surveyed in Sections C.1.2 and C.1.3. A full version can be freely downloaded from the SMAC website `http://w3.onera.fr/smac/apricot`, which also provides a complete documentation as well as applicative examples.

Matlab routine	Description
`lsapprox`	Polynomial approximation using linear least-squares
`olsapprox`	Polynomial approximation using orthogonal least-squares
`qpapprox`	Rational approximation using quadratic programming
`tracker`	Rational approximation using genetic programming
`koala`	Rational approximation using surrogate modeling

Table C.1: Main approximation routines of the APRICOT Library

The APRICOT library was evaluated on a number of challenging problems [Roos *et al.*, 2014; Iannelli *et al.*, 2017; Biannic *et al.*, 2016; Döll *et al.*, 2012], which allowed us to gain good experience. Based on this, the main features of the different rational approximation methods are summarized below. The QP-based method implemented in the routine `olsapprox` is quite fast and usually gives good results. But there are two main limitations. First, only full expressions with a large number of monomials are obtained, which can result in large-size LFR. Then, numerical problems are frequently encountered for degrees larger than 10. The GP-based algorithm implemented in the routine `tracker` gives very sparse expressions with only few nonzero monomials, and it exhibits good numerical properties. The size of the resulting LFR can thus be significantly reduced. However, it is computationally demanding. Finally, the SM-based approach implemented in the routine `koala` also produces reduced-size LFR, but not for the same reason as `tracker`. Indeed, the rational expressions are composed of a large number of monomials, but they directly come in factorized form. Computational cost is quite low, so very accurate approximations (with high degrees) can be obtained. Moreover, the non-singularity of the resulting rational functions is guaranteed implicitly, so the iterative procedure described in Section C.1.4.1 can be avoided. To conclude, GP and SM prove quite complementary: GP is more accurate for low degree approximations, while SM gives better results for degrees larger than 10.

1.4.3 Towards low-order LFR

Once a suitable rational function with no pole in the considered parametric domain is obtained, it is finally converted into an LFR. Several techniques such as object-oriented realization exist to perform this transformation. Although the minimality of the resulting LFR cannot be guaranteed, symbolic preprocessing techniques as well as numerical reduction usually permit to overcome complexity. Moreover, efficient software are available such as the LFR Toolbox for Matlab (see [Magni, 2006] and references therein). This software offers many functionalities and has been widely used, but it suffers from two main drawbacks. First, it is not maintained anymore,

so compatibility with new versions of Matlab is increasingly difficult to ensure. Moreover, it is not easily accessible to non-expert users, due to a rather complicated way of describing LFR. This motivated the development of the *Generalized State-Space* (GSS) Library of the SMAC Toolbox. It implements the new Matlab class `gss`, which allows to model uncertain and nonlinear systems as Linear Fractional Representations [Biannic and Roos, 2016]. Several tools are proposed to manipulate `gss` objects (addition, multiplication, inversion, concatenation, feedback...), to obtain `gss` objects from symbolic models, to ensure compatibility with Matlab `uss` objects, to manipulate the uncertainties and the nonlinearities (normalization, reordering, random sampling...) and to perform order reduction or approximation. It is designed to be more user-friendly and it allows to handle a large class of continuous- and discrete-time systems with real or complex uncertain or varying parameters, polytopic-type uncertain or varying elements, linear time-invariant uncertainties, sector nonlinearities, saturations and deadzones... It is still under development, and it will ultimately include the state-of-the-art realization and reduction techniques. Finally, it is compatible with most libraries of the SMAC Toolbox: APRICOT (see Section C.1.4.2), SMART (see Section A.6), IQC [Demourant, 2013], SAW (see Section D.4)... A full version can be freely downloaded from the SMAC website `http://w3.onera.fr/smac/gss`, which also provides a complete documentation as well as applicative examples.

1.5 Conclusion

In an industrial context, physical systems are usually represented by a mix of nonlinear analytical expressions and tabulated data. A two-step procedure has to be implemented to obtain a suitable linear fractional representation: a linear model with a polynomial or a rational dependence on the system parameters is first generated, and then converted into a linear fractional form. In this work, three methods are proposed to generate sparse polynomial and rational approximations from a set of scalar or matrix samples, based on orthogonal least squares, genetic programming and surrogate modeling respectively. They usually compare very favorably to classical ones. For a given precision, the symbolic expressions are indeed more compact, and hence the size of the resulting LFR is smaller. Good numerical properties are also observed. All existing algorithms have been implemented in the APRICOT Library of the SMAC Toolbox. The new Matlab class `gss` has also been defined and implemented in the GSS Library. It replaces and improves the LFR Toolbox, which is no longer maintained, and it can be used in conjunction with the APRICOT Library to model uncertain and nonlinear systems as Linear Fractional Representations. Several improvements are possible, among which two are worth being mentioned. The first one is to improve the computational time of the `tracker` routine by using gene expression programming instead of genetic programming. The second one deals with rational approximation of matrix samples. Currently, all coefficients are approximated independently of the others. This results in completely different denominators, which strongly penalizes the size of the resulting LFR. A better trade-off between accuracy and complexity could probably be achieved by imposing (partly) identical denominators for all coefficients.

2 From a set of MIMO LTI models to a dynamic LFR

2.1 Motivations

A prerequisite to apply certain modern control techniques is to build a representative LPV or LFT model of the system, see *e.g.* [Leith and Leithead, 2000]. This can be done in two different ways. On the one hand, if a nonlinear analytical model derived from physical equations is available, either symbolic [Marcos and Balas, 2004] or numerical [Pfifer and Hecker, 2011] linearizations are usually performed around equilibrium points. In the first case, an LPV model

is directly obtained, whereas in the second one, the resulting set of LTI models has to be approximated by an LPV model. On the other hand, the initial nonlinear model is sometimes overly complex or does not even exist, see *e.g.* [Ferreres, 2011]. Global or local identification techniques are thus applied provided a (possibly reduced-scale) prototype of the considered system exists. Global techniques allow to obtain an LPV model in a single step, but they require the control inputs and the scheduling parameters to be excited simultaneously and persistently during a single experiment, which is not always possible. In contrast, local techniques consist of performing several experiments during which the scheduling parameters remain constant, leading once again to a set of LTI models to be approximated. An alternative to local identification is to replace each experiment by a numerical analysis of the system using for example finite element techniques, which is commonly done in the aerospace industry. In this context, this section focuses on the generation of low-order LPV/LFT models from a set of large-scale LTI models. This can be seen as a generalization of what is done in Section C.1 for a set of scalar samples.

This question arises in particular when flexible dynamics are to be taken into account. For example, the issue of representing the aeroelastic behavior of an aircraft in linear fractional form has already been addressed in several papers, but most of them assume that a reasonable-size analytical expression is available [Bennani *et al.*, 2005; Baldelli *et al.*, 2005]. Nevertheless, this assumption does not always hold in practice. We have indeed been confronted a few years ago with purely numerical models of huge complexity, which describe both the rigid and the flexible dynamics of an Airbus aircraft for different mass configurations. They were obtained in an industrial context, first by combining a structural finite element model with a complete representation of the aerodynamic forces and moments acting on the aircraft, and then by removing the less relevant modes and making several adjustments to improve accuracy [Puyou and Losser, 2012]. The resulting large-scale LTI models have about 300 states and no specific structure. Moreover, they cannot be directly approximated because of their size and of the non-consistency of their state vectors. And this example is not an isolated case, because we have also been confronted with large-scale aeroelastic bizjet aircraft models developped by Dassault Aviation, for which the problem is roughly the same [Poussot-Vassal *et al.*, 2014]. First attempts to obtain accurate and tractable LFR from such sets of large-scale LTI models were made in [Ferreres, 2011; Roos, 2009]. Good results are reported in these papers, but the dynamics to be included in the LFR are chosen empirically in both cases. The two aforementioned examples as well as this limitation were our motivation to go one step further and to propose a systematic approach in [Poussot-Vassal and Roos, 2012], which can be applied to any set of purely numerical (and possibly unstructured) models. The resulting two-step strategy is briefly described in this section. All algorithms are omitted for the sake of simplicity, since reading them is a bit tedious.

2.2 Generation of reduced and consistent models

The first step is to **generate a set of N reduced-order models**, which all have the same modal content and are as representative as possible of the N initial large-scale models. In [Ferreres, 2011] and [Roos, 2009], some empirical criteria combined with the physical knowledge of the plant are used to generate reduced A matrices, and a biconvex optimization then produces some suitable B, C and D matrices. These approaches assume that the most relevant modes can be selected by hand, which is quite questionable in the general case. In contrast, a purely numerical method using a projection-based approach is proposed in [Poussot-Vassal and Roos, 2012]. An algorithm built on the work of [Gallivan *et al.*, 2004; Gugercin, 2008; Van-Dooren *et al.*, 2008] is implemented to obtain reduced-order models with consistent modal content and accurate frequency responses.

Let us first assume that a single LTI model $H = (A, B, C, D)$, where $A \in \mathbb{R}^{n \times n}$, has to be reduced. The objective is to compute state-space matrices $\hat{H} = (\hat{A}, \hat{B}, \hat{C}, \hat{D})$, where $\hat{A} \in \mathbb{R}^{r \times r}$ and $r \ll n$, such that H and $\hat{H}$ are close in the sense of a given norm (usually H_2 or H_∞). The projection-based approach consists of finding left and right projectors W and V such that $\hat{A} = W^T AV$, $\hat{B} = W^T B$, $\hat{C} = CV$ and $\hat{D} = D$. Several iterative procedures have been proposed, where the projectors are computed based on so-called interpolation points and directions (see *e.g.* [Vuillemin *et al.*, 2013]). The latter are updated at each iteration, for example by taking the eigenvalues of the current reduced-order model and the associated right eigenvectors respectively. This kind of algorithm works well in practice when a single LTI model has to be reduced. But in the multi-model case, scattered sets of eigenvalues are usually obtained if reduction is performed independently for each model, as can be seen in Figure C.3 (left). This is a critical issue, which can lead to unrealistic modal trajectories after the state-space representations of the reduced-order LTI models are approximated with polynomial or rational function and converted into an LPV/LFT model in Section C.2.4.

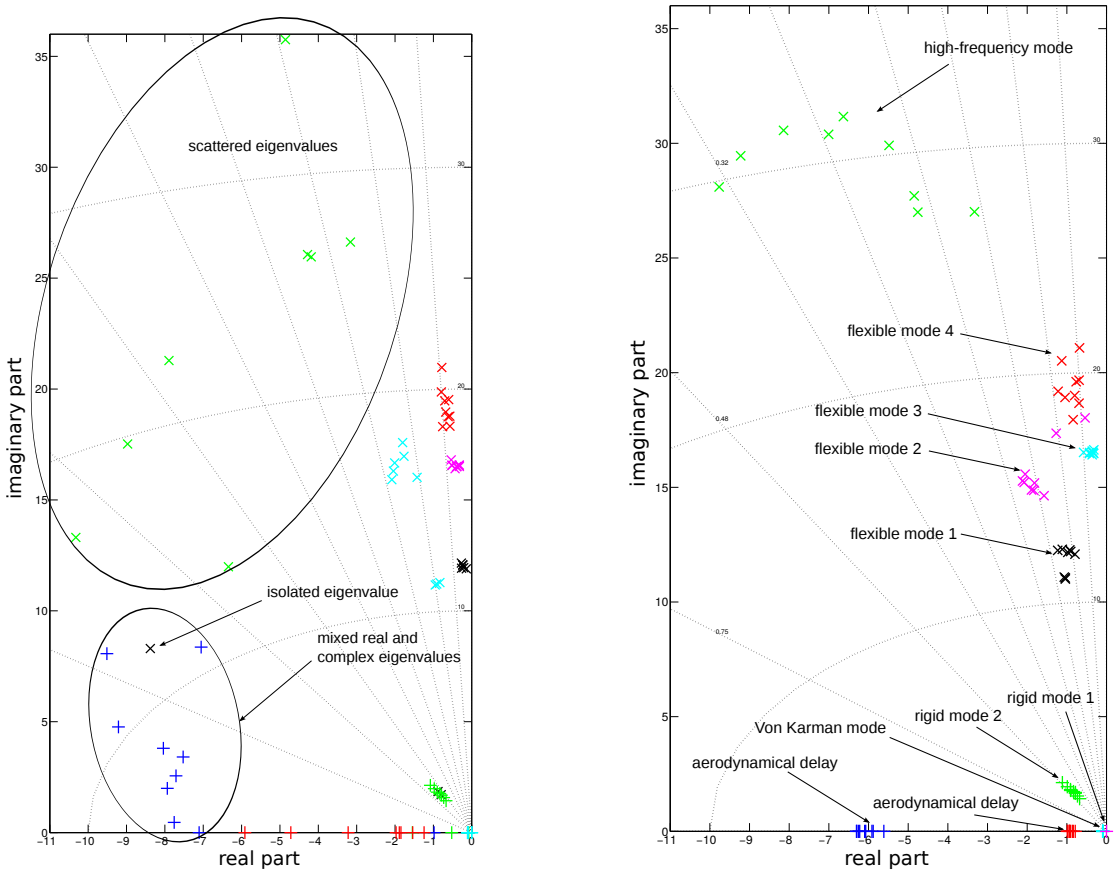


Figure C.3: Eigenvalues with classical (left) and multi-model (right) reduction algorithms

In this context, a multi-model reduction algorithm is proposed in [Poussot-Vassal and Roos, 2012], which forces the eigenvalues of the reduced-order models to remain within the same areas of the complex plane. A three-step procedure is applied. First, each full-order model is approximated independently of the others using the algorithm of [Vuillemin *et al.*, 2013]. Then, the modes of the resulting reduced-order models are sorted and new interpolation points and directions are selected. More precisely, let $(\lambda_k^i)_{k \in [1,N]}$ denote the eigenvalues corresponding to the ith mode for all reduced-order models (represented by the same symbol and color in Fig-

ure C.3 (left)). If the variance of $(\lambda_k^i)_{k\in[1,N]}$ is larger than a user-defined threshold, the mean value of $(\lambda_k^i)_{k\in[1,N]}$ is used as the new interpolation point for all models. Otherwise, the interpolation point for model number k is simply λ_k^i. The new interpolation directions are chosen in a similar way. Finally, a single iteration of the algorithm of [Vuillemin *et al.*, 2013] is performed on each full-order model using these new interpolation points and directions as initialization. A set of N rth-order LTI models is thus generated, whose modal content is now much more consistent, as shown in Figure C.3 (right). The way of selecting the new interpolation points and directions has the advantage to keep the eigenvalues of all the reduced-order models in the same areas of the complex plane if it was already the case after step 2, and to force them to belong to the same areas if it was not initially the case. The counterpart is that the optimality conditions may no longer be guaranteed, and the approximation error may be higher than if each model was reduced independently of the others. Nevertheless, this leads to reduced-order models, whose modal content is more appropriate in the perspective of approximating their state-space representations by polynomial or rational expressions.

Remark C.5 *This reduction step is omitted in most existing works, where small-scale models are often directly considered (see e.g. [de Caigny* et al.*, 2011; Paijmans* et al.*, 2006]). However, it is of significant practical importance, as shown by the two examples mentioned in Section C.2.1.*

2.3 Choice of a suitable state-space form

The second step is to **approximate the reduced-order models**, and a necessary condition to obtain a suitable LPV/LFT model is to use the same state-space form for all of them. Several solutions have been proposed: the balanced state-space realization in [Lovera and Mercère, 2007], the series interconnection of first- and second-order systems in [de Caigny *et al.*, 2011], the scaled companion form in [Ferreres, 2011] and the modal form in [Roos, 2009]. An intuitive way to compute a simple LPV/LFT model is to limit the number of varying elements in the state-space matrices of the reduced-order models [Steinbuch *et al.*, 2003], which speaks in favor of the last two representations. In this work, the emphasis is put on the modal form, although it does not appear to be the easiest choice. Indeed, it requires to exactly pair the modes of the reduced-order models. This is not an obvious task, but an algorithm is proposed in [Poussot-Vassal and Roos, 2012], based on the natural pairing obtained with the companion form. The main reason for this choice is that each element of the state-space matrices of the resulting models can be associated to a single mode. This allows to use an approximation formula with a specific structure and complexity for each mode. A whole procedure that makes the best use of this property is then introduced in Section C.2.4 to select only the most relevant monomials during the approximation step and thus to prevent data overfitting. This significantly reduces the LFR complexity, while maintaining a satisfactory accuracy.

2.4 Generation of a low-order LFR

Once all reduced-order models are written in modal form, an element-wise polynomial or rational approximation of their state-space matrices is performed to obtain an LPV model. This can be achieved by applying any of the techniques described in Section C.1 and implemented in the APRICOT Library of the SMAC Toolbox. The structured tree decomposition of [Cockburn and Morton, 1997] implemented in the GSS Library is finally applied to build an LFR. This fairly simple strategy often works well in practice [Poussot-Vassal and Roos, 2012; Poussot-Vassal *et al.*, 2014], but it sometimes results in overly complicated representations. A three-step procedure is described in [Poussot-Vassal and Roos, 2012] to address this issue and to master the complexity of the resulting LFR:

1. An iterative procedure is first implemented. The state-space coefficients with the lowest influence on the frequency responses are determined (in the sense of a certain norm, usually H_2 or H_∞). They are progressively replaced by their mean value over the whole set of models until the error with respect to the initial full-order models becomes larger than a given threshold ϵ_1, which allows to limit the number of varying elements to be approximated. The B and C matrices of the resulting models are then optimized alternatively until convergence, so as to reduce the H_2 error with respect to the initial full-order models. This is quite fast, since each of these problems is convex [Roos, 2009]. Note that the elements of B and C which are constant over the whole set of models are not modified. The whole algorithm is repeated until no more element can be replaced with its mean value.

2. The coefficients which are not constant are now considered one after the other. An algorithm is first applied to determine the smallest degrees $d_1, \ldots, d_n$ of the n explanatory variables, for which a maximum admissible error $\epsilon_2 > \epsilon_1$ is guaranteed. One of the sparse approximation techniques of Section C.1.3 is then applied, which identifies the most relevant monomials for which the approximation error remains lower than a given threshold $\epsilon_3 > \epsilon_2$. All the other monomials are eliminated, thus leading to sparse polynomial or rational expressions, which amounts to a certain extent to minimizing the complexity of the approximation formulas. The structured tree decomposition is finally applied to get an LFR.

3. The H_2 error between the resulting LFR and the initial full-order models is minimized. More precisely, the matrices B and C of the LFR are optimized alternatively using the method presented in [Torralba *et al.*, 2009], which is a generalization to the LFR framework of the biconvex optimization applied at step 1.

Remark C.6 *The acceptable error is gradually relaxed thanks to the increasing tolerances ϵ_1, ϵ_2 and ϵ_3. This allows to better handle the trade-off between the accuracy and the complexity of the resulting LFR.*

Remark C.7 *If the number N of available models is large enough, only some of them are usually considered for approximation, the other ones being used to evaluate the accuracy of the resulting LFR. Such an approach is relevant, since it provides quantitative information to validate the behavior of the LFR between the parametric configurations used for approximation.*

Remark C.8 *This element-wise strategy is quite efficient in terms of computational time and usually leads to low-order LFR, which are tractable for control laws design and validation. But two issues should be kept in mind. First, nothing can be assessed outside the parametric configurations used for approximation, and a thorough validation is thus necessary. A common strategy to address this issue is to evaluate the stability and the performance properties of the LPV/LFT model at each point of a very fine grid of the operating domain, as suggested in [Steinbuch* et al.*, 2003]. A more rigorous approach is to use the μ-analysis based tools described in Chapter A [Roos* et al.*, 2011], which usually solve the problem very efficiently. Then, no input/output criterion is considered during approximation. A solution to this problem is proposed in [Petersson and Lofberg, 2009]. It consists of solving an SDP problem to minimize the H_2 error between the desired LPV model and the local LTI models. Nevertheless, such an approach can only be applied to low-order systems due to a prohibitive computational cost. Fortunately, recent work reported in [Vizer and Mercère, 2014] shows that non-smooth optimization techniques developed in [Apkarian* et al.*, 2009] could help solving the problem in the H_∞ framework at a reasonable cost.*

2.5 Presentation of the LTI2LFR Library for Matlab

All the tools and algorithms presented in Section C.2 have been implemented in the LTI2LFR Library for Matlab, which allows to convert a set of large-scale MIMO LTI models describing the behavior of the considered system at various operating points into a single reduced-order parameterized model (LPV model or LFR). Only a few high-level tuning parameters have to be defined, which makes it possible to use the toolbox without mastering all the underlying theory, and to efficiently handle the trade-off between accuracy and complexity. A detailed documentation is available in [Roos, 2012; Poussot-Vassal *et al.*, 2013], as well as several applications on challenging Airbus and Dassault benchmarks already mentioned in Section C.2.1. The LTI2LFR Library will be integrated in the SMAC Toolbox in a near future.

3 Summary of the contributions

The main contributions presented in this chapter are briefly summarized below, and a selection of related publications is given:

- **Section C.1.3.1:** orthogonal least squares based method to compute sparse polynomial approximations from a set of scalar or matrix samples [Poussot-Vassal and Roos, 2012; Roos, 2009].
- **Section C.1.3.2:** genetic programming based method to compute sparse rational approximations from a set of scalar or matrix samples [Hardier *et al.*, 2013a].
- **Section C.1.3.3:** surrogate modeling based method to compute sparse rational approximations from a set of scalar or matrix samples [Hardier *et al.*, 2013b].
- **Section C.1.4.1:** μ-analysis based approach to check whether the denominator of a rational function is non-singular, *i.e.* to ensure that the resulting LFR is well-posed [Hardier *et al.*, 2013a].
- **Section C.1.4.2:** implementation of all polynomial and rational approximation algorithms in the APRICOT Library of the SMAC Toolbox [Roos *et al.*, 2014].
- **Section C.1.4.3:** implementation of a new Matlab class in the GSS Library of the SMAC Toolbox, which allows to model uncertain and nonlinear systems as Linear Fractional Representations [Biannic and Roos, 2016].
- **Sections C.2.1 - C.2.4:** whole methodology to generate low-order LPV/LFT models from a set of large-scale MIMO LTI models describing the behavior of a dynamical system at various operating points [Poussot-Vassal and Roos, 2012; Poussot-Vassal *et al.*, 2014; Roos, 2009].
- **Section C.2.5:** implementation of all related tools and algorithms in the LTI2LFR Library [Roos, 2012; Poussot-Vassal *et al.*, 2013].

Chapter D

Analysis & control of saturated systems

In the previous chapters, a linear framework was generally considered. But most physical systems, and especially aerospace ones, are also subject to nonlinearities, among which saturations figure prominently. And this is a major issue. By limiting both the amplitude and the rate of the control signals, saturations induce particularly disturbing nonlinear phenomena (such as limit cycles), that can impair the closed loop performance or even jeopardise its stability. Several strategies are possible to cope with this issue. One of them consists of staying in the linear domain by computing unsaturating control laws, at the price of a lower level of performance. Another one is to allow saturations and to adapt the control signals as soon as one of them is active, so as to return as quickly as possible to the linear domain guaranteeing a well-controlled nominal behavior of the system. The second approach is often preferred, since many applications cannot tolerate an unnecessary loss of performance in the nominal operating domain. In this case, the controller can be designed in one or two steps. The latter solution is more intuitive and flexible. A nominal controller is first designed using any linear technique to ensure that the nominal specifications are met. An anti-windup controller is then computed, which becomes active only when a saturation is reached and modifies the closed-loop behavior such that it is more resilient to saturations. Such a strategy is reassuring for control engineers, who can keep their usual control architectures and simply connect an additional block.

The origins of anti-windup are rather vague. Practioners were already aware of the problems caused by saturations a long time ago, and they developed empirical solutions dedicated in particular to PID controllers. Researchers began to consider saturations a few years after. They managed to explain how saturations could create problems with integrators in PI controllers [Lozier, 1956], but it is only later that they proposed the first well-documented anti-windup techniques, *e.g.* [Fertik and Ross, 1967; Åström and Rundqwist, 1989]. Among the first *modern* contributions, let us mention [Kothare *et al.*, 1994], where a unifying framework inspired by the famous standard forms from robust control theory was developed. The study of saturated systems then became very popular, as evidenced by the large number of papers in this field (see [Zaccarian and Teel, 2011; Tarbouriech and Turner, 2009; Galeani *et al.*, 2009; Bernstein and Michel, 1995] and references therein).

Many LMI-based approaches exist to adjust the anti-windup gains in a systematic way. Most often, these are based on the optimization of either a stability domain [Cao *et al.*, 2002; Tarbouriech *et al.*, 2006c; Gomes da Silva Jr and Tarbouriech, 2005], or a nonlinear $\mathcal{L}_2$-induced performance level [Castelan *et al.*, 2004; Lu *et al.*, 2005; Hu *et al.*, 2005]. Based on the LFT/LPV framework, extended anti-windup schemes were introduced in [Saeki and Wada, 2002; Turner and Postlethwaite, 2004; Wu and Soto, 2004; Lu *et al.*, 2005]. In these contributions, the saturations are viewed as sector nonlinearities and the anti-windup control design issue is recast into a convex

optimization problem under LMI constraints. Following a similar path, alternative techniques using a less conservative representation of the saturation function based on a modified sector condition [Gomes da Silva Jr and Tarbouriech, 2005] were then proposed to compute either static [Biannic *et al.*, 2006b; Gomes da Silva Jr and Tarbouriech, 2005] or dynamic [Tarbouriech *et al.*, 2006a; Hu *et al.*, 2005; Kiyama and Sawada, 2004] anti-windup controllers. These techniques are further exploited in this chapter, and it is shown in Section D.1 that the design problem is convex when the order of the anti-windup controller coincides with that of the nominal closed-loop model.

Unfortunately, this order can be quite large in some cases. Moreover, it is not easy to constrain the dynamics of the anti-windup controller, which usually has some very slow or very fast poles likely to lower performance or to cause implementation issues. Finally, it is generally observed in practice that reduced-order controllers can be as efficient as full-order ones, provided their poles are chosen appropriately. In this context, a convex characterization is proposed in Section D.3 to compute dynamically-constrained anti-windup controllers. More precisely, an upper bound is introduced on the real part of their poles. An algorithm is also given in Section D.2 to compute a reduced-order anti-windup controller based on the poles obtained in the full-order case. The resulting tools are all implemented in the AWAS Toolbox, which has recently been updated and integrated into the SMAC Toolbox, as explained in Section D.4.

1 Full-order anti-windup design

1.1 Problem statement

Consider the nonlinear interconnection of Figure D.1. The saturated plant $G(s)$ to be controlled is written in a standard LFT form, for example using the GSS Library of the SMAC Toolbox (see [Biannic and Roos, 2016] and Chapter C):

$$G(s): \left\{ \begin{array}{rcl} \dot{x}_G & = & A_G\, x_G + B_G \left[\begin{array}{c} \Phi(z) \\ u \end{array} \right] \\ \left[\begin{array}{c} z \\ y \end{array} \right] & = & C_G\, x_G + D_G \left[\begin{array}{c} \Phi(z) \\ u \end{array} \right] \end{array} \right. \tag{D.1}$$

where u and y denote the control inputs and the measured outputs respectively. The nonlinear operator $\Phi : \mathbb{R}^m \to \mathbb{R}^m$ is characterized as follows:

$$\Phi(z) = \left[\begin{array}{ccc} \phi(z_1) & \ldots & \phi(z_m) \end{array} \right]^T \tag{D.2}$$

where $\phi(.)$ is a normalized deadzone nonlinearity defined as:

$$\phi(z_i) = \left\{ \begin{array}{ll} 0 & \text{if} \quad |z_i| \leq 1 \\ z_i - \text{sign}(z_i) & \text{if} \quad |z_i| > 1 \end{array} \right. \tag{D.3}$$

This formulation is not restrictive, since any saturation $\text{sat}_L(.)$ with lower and upper limits $\pm L$ can be converted into a normalized deadzone $\phi(.)$ as follows:

$$\text{sat}_L(z_i) = z_i - L\phi\left(\frac{z_i}{L}\right) \tag{D.4}$$

Remark D.1 *Nested saturations are not considered here for the sake of simplicity. Therefore, it is assumed that* $D_G = \left[\begin{array}{cc} 0 & 0 \\ D_{G_{21}} & D_{G_{22}} \end{array} \right]$, *see e.g. [Tarbouriech* et al., *2006b] for more details.*

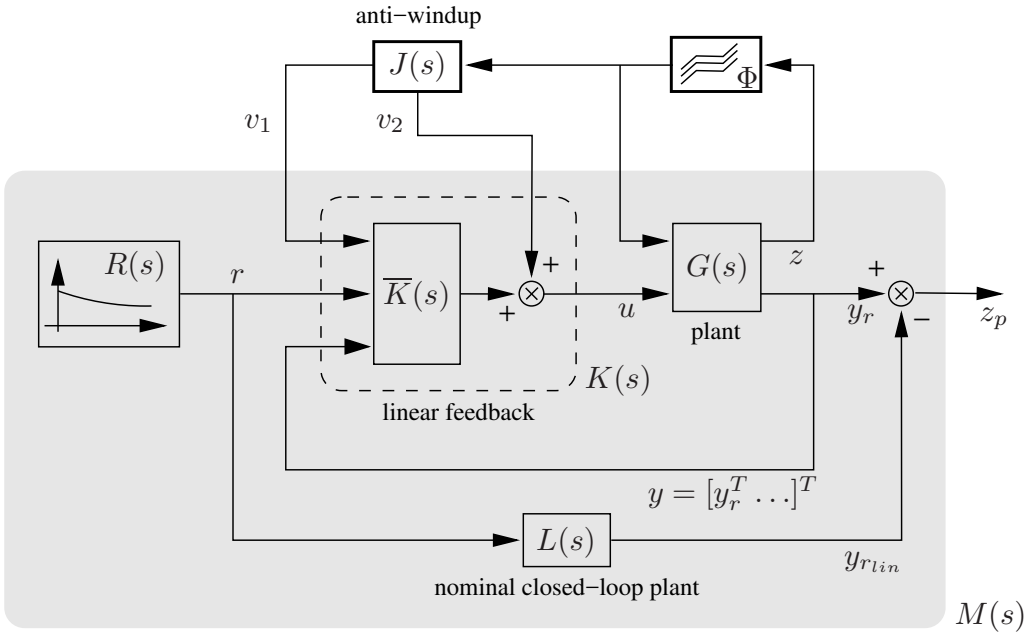


Figure D.1: Standard interconnection with a general anti-windup architecture

Suppose that a nominal linear controller $\overline{K}(s)$ has been first designed, so as to stabilize the plant $G(s)$ and ensure good performance properties in the linear region. To mitigate the adverse effects of saturations, additional signals v_1 and v_2 are injected both at the input and output of this controller [Grimm *et al.*, 2003; Wu and Soto, 2004]. A state-space representation of the resulting controller $K(s)$ is then given by:

$$K(s): \left\{ \begin{array}{rcl} \dot{x}_K & = & A_K\, x_K + B_K \left[\begin{array}{c} r \\ y \end{array} \right] + v_1 \\ \\ u & = & C_K\, x_K + D_K \left[\begin{array}{c} r \\ y \end{array} \right] + v_2 \end{array} \right. \tag{D.5}$$

The signals v_1 and v_2 are obtained as the outputs $v = \left[\begin{array}{c} v_1 \\ v_2 \end{array} \right] \in \mathbb{R}^{n_v}$ of the dynamic anti-windup controller $J(s)$ to be determined:

$$J(s): \left\{ \begin{array}{rcl} \dot{x}_J & = & A_J\, x_J \;+\; B_J\, \Phi(z) \\ v & = & C_J\, x_J \;+\; D_J\, \Phi(z) \end{array} \right. \tag{D.6}$$

where the input signal $\Phi(z)$ can be interpreted as an indicator of the saturations activity.

This nonlinear closed-loop plant can be affected by some exogenous input signals r such as perturbations (wind, turbulences) or commanded inputs. In terms of performance analysis, a classical problem is to ensure that for these inputs, some outputs y_r of the saturated plant remain as close as possible to the outputs $y_{r_{lin}}$ of the corresponding nominal unsaturated behavior $L(s)$, which amounts to minimizing the energy of the error signal z_p. On the one hand, step inputs are usually considered in practice to assess performance, and considering any kind of input signals can lead to conservative results. On the other hand, it is convenient from a theoretical point of view to work with finite-energy signals, for which LMI characterizations based on the induced $\mathcal{L}_2$ norm exist. Unfortunately, step inputs are not $\mathcal{L}_2$-bounded... The performance level introduced

in [Biannic *et al.*, 2006b] allows to address this issue by restricting r to the set of slowly decreasing exponential signals $\mathcal{W}_\epsilon(\rho)$ defined as:

$$\mathcal{W}_\epsilon(\rho) = \left\{ r : \mathbb{R} \to \mathbb{R} \ , \ r(t) = r_0 e^{-\epsilon t} \ \forall t \geq 0 \ , \ |r_0| \leq \rho \right\} \tag{D.7}$$

where ϵ is chosen small enough compared to the system dynamics. The elements of $\mathcal{W}_\epsilon(\rho)$ are both $\mathcal{L}_2$-bounded and representative of step inputs, as can be seen in Figure D.2. A stable autonomous system $R(s)$ with a non-zero initial state r_0 is thus inserted into the interconnection of Figure D.1 to generate $r(t)$. For the sake of simplicity, it has been assumed that r is a scalar signal, but the generalization to vector signals does not raise any technical difficulties.

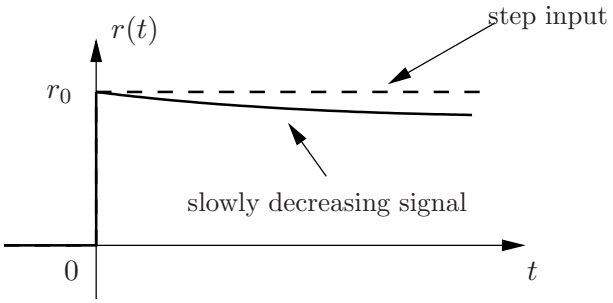


Figure D.2: Approximation of a step input with a slowly decreasing $\mathcal{L}_2$-bounded signal

Let us now define the augmented state vector ξ obtained by merging the states of the reference model (r), the nominal (linear) closed-loop system (x_L), the open-loop plant (x_G) and the nominal controller (x_K):

$$\xi = \begin{bmatrix} r \\ x_L \\ x_G \\ x_K \end{bmatrix} \in \mathbb{R}^{n_M} \tag{D.8}$$

The resulting system $M(s)$ connected with the anti-windup controller is illustrated in Figure D.3 and can be defined as follows:

$$M(s) : \begin{cases} \dot{\xi} &= A\,\xi \ + \ B_\phi\,\Phi(z) \ + \ B_a v \\ z &= C_\phi\,\xi \\ z_p &= C_p\,\xi \ + \ D_{p\phi}\,\Phi(z) \ + \ D_{pa}\,v \ = \ y_r - y_{r_{lin}} \in \mathbb{R}^p \end{cases} \tag{D.9}$$

where y_r corresponds to the first elements of the output vector $y = [y_r^T \ldots]^T$.

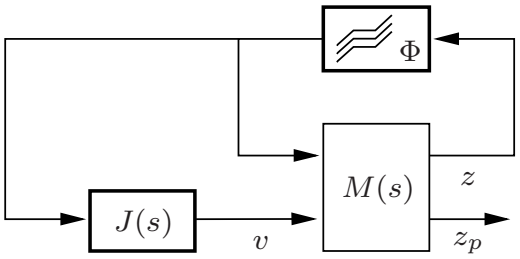


Figure D.3: A synthetic view of Figure D.1

Finally, by adding the state $x_J \in \mathbb{R}^{n_J}$ of the anti-windup controller, the following augmented state vector is defined, where $n = n_M + n_J$:

$$\nu = \begin{bmatrix} \xi \\ x_J \end{bmatrix} \in \mathbb{R}^n \tag{D.10}$$

Therefore, the nonlinear closed-loop interconnection shown in Figure D.4 reads:

$$P(s): \begin{cases} \dot{\nu} &= \begin{bmatrix} A & B_a C_J \\ 0 & A_J \end{bmatrix} \nu + \begin{bmatrix} B_\phi + B_a D_J \\ B_J \end{bmatrix} w \\ z &= \begin{bmatrix} C_\phi & 0 \end{bmatrix} \nu \\ z_p &= \begin{bmatrix} C_p & D_{pa} C_J \end{bmatrix} \nu + \begin{bmatrix} D_{p\phi} + D_{pa} D_J \end{bmatrix} w \\ w &= \Phi(z) \end{cases} \tag{D.11}$$

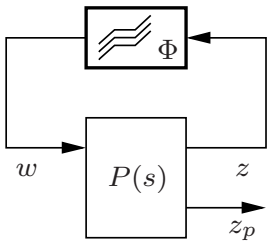


Figure D.4: Nonlinear closed-loop interconnection including the anti-windup controller

Let the state vector ν be partitioned as $\nu = [r\ \zeta^T]^T$ to distinguish more clearly the reference r from the internal states $\zeta = [x_L^T\ x_G^T\ x_K^T\ x_J^T]^T \in \mathbb{R}^{n-1}$. The anti-windup design problem to be solved can then be summarized as follows:

Problem D.1 (anti-windup design) *Compute a dynamic anti-windup controller $J(s)$ (i.e. matrices A_J, B_J, C_J, D_J) and a domain $\mathcal{E}(\rho)$ as large as possible such that, for a given positive scalar ρ and any reference signal $r \in \mathcal{W}_\epsilon(\rho)$, the following properties hold:*

- *the nonlinear closed-loop plant (D.11) is stable for all initial condition ζ_0 inside $\mathcal{E}(\rho)$,*
- *some outputs y_r of the saturated plant remain as close as possible to the linear reference outputs $y_{r_{lin}}$ (associated with the nominal unsaturated behavior), i.e. the energy of the error signal z_p is minimized.*

1.2 Performance analysis of saturated systems

Let us first assume that an anti-windup controller $J(s)$ is available. The following proposition introduced in [Biannic *et al.*, 2006b] allows to compute a performance level for the saturated interconnection of Figure D.4 when step-like input signals $\mathcal{W}_\epsilon(\rho)$ are considered. It makes use of the modified sector condition of [Gomes da Silva Jr and Tarbouriech, 2005] to describe the saturations, which is less conservative than the classical one given in [Khalil, 1996].

Proposition D.1 (performance characterization) *Let γ and $\overline{\rho}$ be some positive scalars, and $u(\overline{\rho}) = [\overline{\rho}\ 0]^T \in \mathbb{R}^n$. If there exist matrices $Q = Q^T \in \mathbb{R}^{n\times n}$, $S = \mathrm{diag}(s_1, \ldots, s_m) > 0$, $Z \in \mathbb{R}^{m\times n}$ such that the following LMI conditions hold:*

$$\begin{bmatrix} Q & \star \\ u(\overline{\rho})^T & 1 \end{bmatrix} > 0 \tag{D.12}$$

$$\begin{bmatrix} \begin{bmatrix} A & B_aC_J \\ 0 & A_J \end{bmatrix} Q + Q \begin{bmatrix} A & B_aC_J \\ 0 & A_J \end{bmatrix}^T & \star & \star \\ S \begin{bmatrix} B_\phi + B_aD_J \\ B_J \end{bmatrix}^T - Z & -2S & \star \\ \begin{bmatrix} C_p & D_{pa}C_J \end{bmatrix} Q & \begin{bmatrix} D_{p\phi} + D_{pa}D_J \end{bmatrix} S & -\gamma I_p \end{bmatrix} < 0 \tag{D.13}$$

$$\begin{bmatrix} Q & \star \\ Z_i + \begin{bmatrix} C_{\phi_i} & 0 \end{bmatrix} Q & 1 \end{bmatrix} > 0\,, \quad i = 1 \ldots m \tag{D.14}$$

where Z_i and $C_{\phi i}$ denote the i^{th} rows of Z and C_ϕ respectively, and where the symmetric terms in the above matrices are replaced by $\star$, then for all $\rho \leq \overline{\rho}$, the interconnection of Figure D.1 is stable for all reference signals $r \in \mathcal{W}_\epsilon(\rho)$ and all initial condition ζ_0 in the domain:

$$\mathcal{E}(\rho) = \left\{ \zeta \in \mathbb{R}^{n-1} : \begin{bmatrix} \rho \\ \zeta \end{bmatrix}^T P \begin{bmatrix} \rho \\ \zeta \end{bmatrix} \leq 1 \right\} \tag{D.15}$$

where $P = Q^{-1}$. Moreover, the output energy satisfies:

$$\int_0^\infty z_p(t)^T z_p(t)\, dt \leq \gamma \tag{D.16}$$

Inequality (D.13) guarantees that the interconnection of Figure D.1 is stable if the internal states ζ are initialized inside the ellipsoid $\mathcal{E}(\rho)$, and that (D.16) is satisfied. Inequality (D.12) enforces stability for all reference signals $r \in \mathcal{W}_\epsilon(\rho)$. Finally, inequality (D.13) is a technical condition, which ensures that the modified section condition of [Gomes da Silva Jr and Tarbouriech, 2005] is satisfied. Proposition D.1 is formulated as a feasibility problem, but there are several ways to turn it into an optimization problem. A classical one consists of minimizing the performance index γ for a given amplitude of the input signals r, *i.e.* for a fixed $\overline{\rho}$. On the contrary, it is also possible to maximize $\overline{\rho}$, or more generally the volume of the ellipsoid $\mathcal{E}(\rho)$, for a given γ, in which case inequality (D.12) has to be adapted.

1.3 Convex characterization of full-order anti-windup design

Let us now focus on the anti-windup design issue stated in Problem D.1. In this case, the matrix Q introduced in Proposition D.1 and the state-space matrices of $J(s)$ have to be computed simultaneously. As a result, inequality (D.13) becomes a BMI and is no longer convex. However, in the full-order case (*i.e.* $n_J = n_M$), convexity can be recovered thanks to the following theorem [Biannic *et al.*, 2007].

Theorem D.1 (full-order anti-windup design) *Let $v(\overline{\rho}) = [\overline{\rho}\ 0]^T \in \mathbb{R}^{n_M}$. Let $\Gamma = diag(N_a, I_m, N_{pa})$, where N_a and N_{pa} denote any basis of the null-spaces of B_a^T and D_{pa}^T respectively.*

There exists an anti-windup controller $J(s)$ such that the conditions of Proposition D.1 are satisfied ***iff*** *there exist matrices $X = X^T, Y = Y^T \in \mathbb{R}^{n_M \times n_M}$, $S = \mathrm{diag}(s_1, \ldots, s_m) > 0$ and $W = \left[\begin{array}{cc} U & V \end{array}\right] \in \mathbb{R}^{m \times (n_M + n_M)}$ such that the following LMI conditions hold:*

$$v(\overline{\rho})^T X v(\overline{\rho}) < 1 \tag{D.17}$$

$$\left[\begin{array}{cc} A^T X + XA & \star \\ C_p & -\gamma I_p \end{array}\right] < 0 \tag{D.18}$$

$$\Gamma^T \left[\begin{array}{ccc} AY + YA^T & \star & \star \\ SB_\phi^T - V & -2S & \star \\ C_p Y & 0 & -\gamma I_p \end{array}\right] \Gamma < 0 \tag{D.19}$$

$$\left[\begin{array}{ccc} X & \star & \star \\ I_{n_M} & Y & \star \\ U_i & V_i + C_{\phi_i} Y & 1 \end{array}\right] > 0\,, \quad i = 1 \ldots m \tag{D.20}$$

The most classical way to apply Theorem D.1 consists of fixing $\overline{\rho}$ and minimizing γ with respect to the matrices X, Y, S, U and V under the **linear constraits** (D.17)-(D.20). This can be done easily using an LMI solver.

Remark D.2 *The matrix Q of Theorem D.1 is obtained from X and Y as described in [Gahinet and Apkarian, 1994]:*

$$Q = \left[\begin{array}{cc} Y & I_{n_M} \\ N & 0 \end{array}\right] \left[\begin{array}{cc} I_{n_M} & X \\ 0 & M \end{array}\right]^{-1} \quad \textit{where} \;\; M^T N = I_{n_M} - XY \tag{D.21}$$

Q being fixed, inequality (D.13) becomes convex with respect to A_J, $\tilde{B}_J = B_J S$, C_J, $\tilde{D}_J = D_J S$, S and Z. These matrices can be computed with an LMI solver, and the anti-windup controller $A_J, B_J = \tilde{B}_J S^{-1}, C_J, D_J = \tilde{D}_J S^{-1}$ is finally obtained, since S is invertible.

2 Fixed-order anti-windup design

The order of the anti-windup controller $J(s)$ computed in Section D.1.3 is equal to n_M, and it may be quite large in some cases. Moreover, it is not easy to constrain the poles of $J(s)$, since the state matrix A_J does not appear in equations (D.17)-(D.20), and unfortunately some of them are usually very slow or very fast. Finally, it is generally observed in practice that reduced-order controllers can be as efficient as full-order ones, provided their poles are chosen appropriately. So the objective now is to develop an effective procedure to design an anti-windup controller of order $n_J < n_M$. Fixing A_J is a convenient way to control precisely the dynamics of $J(s)$. Moreover, as stated in Proposition D.2 below, when C_J is also fixed, the anti-windup problem becomes convex even if $n_J < n_M$, and an algorithm can be implemented [Roos *et al.*, 2010].

Proposition D.2 *The BMI constraint (D.13) is convex if the matrices A_J and C_J are fixed.*

Algorithm D.1 (fixed-dynamics anti-windup design)

1. *Choose appropriate A_J and C_J matrices (see below).*
2. *Fix $\overline{\rho}$ and minimize γ under the LMI constraints (D.12)-(D.14) w.r.t. Q, S, Z, $\tilde{B}_J$, $\tilde{D}_J$.*
3. *Compute $B_J = \tilde{B}_J S^{-1}$ and $D_J = \tilde{D}_J S^{-1}$.*

The main difficulty in the above algorithm consists of choosing the matrices A_J and C_J. This may appear more intuitive by considering the following decomposition:

$$J(s) = M_0 + \sum_{i=1}^{n_1} \frac{M_{i1}}{s+\lambda_i} + \sum_{i=1}^{n_2} \frac{M_{i2}}{s^2 + 2\eta_i\omega_i + \omega_i^2} \tag{D.22}$$

where $D_J = M_0$ and B_J contains the matrices M_{i1} and M_{i2}. For this decomposition, the fixed matrices A_J and C_J can be chosen as:

$$\begin{aligned} A_J &= \text{diag}\,(-\lambda_1, \ldots, -\lambda_{n_1}, A_1, \ldots, A_{n_2}) \quad \text{where} \quad A_i = \begin{bmatrix} 0 & 1 \\ -\omega_i^2 & -2\eta_i\omega_i \end{bmatrix} \\ C_{J_k} &= \big[\underbrace{1 \ldots 1}_{n_1} \; \underbrace{[1\;0] \ldots [1\;0]}_{n_2}\big], \; k = 1, \ldots, n_v \end{aligned} \tag{D.23}$$

where C_{J_k} denotes the kth row of C_J. From this observation, the first step of Algorithm D.1 simply boils down to choosing a list of poles for the anti-windup controller, whose matrices A_J and C_J are then immediately deduced from (D.23). This choice can be made by first computing a full-order controller, as suggested by the following algorithm [Roos *et al.*, 2010]:

Algorithm D.2 (fixed-order anti-windup design)

1. *Compute a full-order anti-windup controller using Theorem D.1.*
2. *Analyze the controller poles and keep only those that are neither too fast nor too slow. Compute A_J and C_J as explained above.*
3. *Run Algorithm D.1 to compute a reduced-order anti-windup controller.*

Remark D.3 *Alternatively, an iterative procedure can be implemented at step 3. Starting from the static case, the list of poles is progressively enriched until the gap between the full and the reduced-order cases becomes small enough.*

This is a heuristic approach and there is no guarantee that the optimal anti-windup controller is obtained, but satisfactory results have been reported in [Biannic *et al.*, 2007].

3 Dynamically-constrained anti-windup design

The fast anti-windup poles can generally be eliminated without consequences in step 2 of Algorithm D.2. They are indeed not desirable from a practical point of view, since they make it difficult to implement the anti-windup controller, as is mentioned in [Turner and Postlethwaite, 2004]. Moreover, their contribution to the enlargment of the stability domain or the improvement of the performance level is often negligible, which is notably confirmed by [Hencey and Alleyne, 2006]. The same cannot be said of slow anti-windup poles. Removing them can indeed significantly reduce the stability domain. But on the other hand, keeping them introduces slow dynamics, which may remain visible on certain plant outputs even when the saturations are no longer active. This issue can be partly addressed by choosing a good tradeoff between $\overline{\rho}$ and γ when applying Theorem D.1, but this is not always obvious. Another option is to constrain the dynamics of the full-order anti-windup controller in step 1 of Algorithm D.2. In this context, Proposition D.1 and Theorem D.1 are modified in this section: an upper bound is introduced on the real part of the controller poles, while maintaining convexity [Roos and Biannic, 2008]. There are no longer any slow poles to remove in step 2, and the aforementioned dilemma is avoided.

Proposition D.3 (modification of Proposition D.1) *If (D.13) is replaced with:*

$$\begin{bmatrix} \begin{bmatrix} A & B_aC_J \\ 0 & A_J \end{bmatrix} Q + Q \begin{bmatrix} A & B_aC_J \\ 0 & A_J \end{bmatrix}^T + 2\lambda \begin{bmatrix} 0 & 0 \\ 0 & I_{n_J} \end{bmatrix} Q \begin{bmatrix} 0 & 0 \\ 0 & I_{n_J} \end{bmatrix} & \star & \star \\ S \begin{bmatrix} B_\phi + B_aD_J \\ B_J \end{bmatrix}^T - Z & -2S & \star \\ \begin{bmatrix} C_p & D_{pa}C_J \end{bmatrix} Q & \begin{bmatrix} D_{p\phi} + D_{pa}D_J \end{bmatrix} S & -\gamma I_p \end{bmatrix} < 0 \quad \text{(D.24)}$$

in Proposition D.1, then in addition to other conclusions, the poles $\lambda_1, \lambda_2, \ldots, \lambda_{n_J}$ of the anti-windup controller $J(s)$ satisfy:

$$\Re(\lambda_j) < -\lambda \quad \text{for all} \quad j \in [1\,,\, n_J] \quad \text{(D.25)}$$

Theorem D.2 (modification of Theorem D.1) *If (D.18) and (D.20) are replaced with:*

$$\begin{bmatrix} AX + XA^T - 2\lambda X & \star & \star \\ 2\lambda Y & -2\lambda Y & \star \\ C_p & 0 & -\gamma I_p \end{bmatrix} < 0 \quad \text{(D.26)}$$

$$\begin{bmatrix} X & \star & \star \\ X & Y & \star \\ U_i & V_i + C_{\phi_i} Y & 1 \end{bmatrix} > 0 \;\;,\;\; i = 1 \ldots m \quad \text{(D.27)}$$

in Theorem D.1, then in addition to other conclusions, the poles $\lambda_1, \lambda_2, \ldots, \lambda_{n_J}$ of the anti-windup controller $J(s)$ satisfy:

$$\Re(\lambda_j) < -\lambda \quad \text{for all} \quad j \in [1\,,\, n_J] \quad \text{(D.28)}$$

The most classical way to apply Theorem D.2 consists of minimizing γ with respect to matrices X, Y, S, U and V under the **linear constraits** (D.17), (D.26), (D.19) and (D.27). This can be done easily using an LMI solver.

Remark D.4 *The matrix Q of Theorem D.2 is obtained from X and Y as follows:*

$$Q = \begin{bmatrix} Y & I_{n_M} \\ N & 0 \end{bmatrix} \begin{bmatrix} I_{n_M} & X^{-1} \\ 0 & M \end{bmatrix}^{-1} \quad \text{where} \quad M^T N = I_{n_M} - X^{-1}Y \quad \text{(D.29)}$$

Q being fixed, the anti-windup controller $J(s)$ can be computed as in Remark D.4.

Remark D.5 *Adding $2\lambda \begin{bmatrix} 0 & 0 \\ 0 & I_{n_J} \end{bmatrix} Q \begin{bmatrix} 0 & 0 \\ 0 & I_{n_J} \end{bmatrix}$ in Proposition D.3 allows to constrain only the poles of the anti-windup controller and not the whole closed-loop plant dynamics as it is the case in other contributions (see notably [Hencey and Alleyne, 2006]).*

Proposition D.3 and Theorem D.2 can now be combined in the same way as in Section D.2 to compute a reduced-order anti-windup controller with pole contraints. The only difference is that there is no slow pole to remove in step 2 of Algorithm D.1. They are indeed avoided in step 1 thanks to the introduction of λ in Theorem D.2. Moreover, it is often observed that only a few poles of $J(s)$ lie within the bandwidth of the closed-loop plant $M(s)$, the other ones being significantly faster. Thus it is possible to design low-order anti-windup controllers, which are almost as efficient as full-order ones. An application to a fighter aircraft model in [Roos and Biannic, 2008] confirms it. A second-order controller is obtained, which guarantees the same stability domain and performance level as the initial full-order controller with 6 states.

4 From the AWAS Toolbox to the SAW Library

Anti-windup controllers are widely used to mitigate the negative effects of actuators saturations on stability and performance. But despite recent theoretical advances in this field, their computation remains a rather difficult and time-consuming task from a control engineer perspective, which is often realized by a trial-and-error simulation-based approach. This can be partly explained by the lack of user-friendly tools to help the designer define and modify the anti-windup structure, perform the optimization of the gains by simply calling an appropriate routine, and test the results with simulations. In this context and similarly to other libraries already mentioned in this manuscript (SMART, GSS, APRICOT...), the ambition of the Anti-Windup Analysis and Synthesis Toolbox for Matlab [Biannic and Roos, 2008] is to contribute to bridging the gap between theory and practice. It implements the Lyapunov-based approach described in this chapter, which is rigorous since guaranteed stability domains or performance levels can be computed, but remains limited to medium-order models (less than 20 states) due to the high number of decision variables involved in the associated LMI problems. Based on this observation, the AWAS Toolbox has recently been updated and integrated into the SMAC Toolbox under the name of *Saturated systems analysis & Anti-Windup design* (SAW) Library [Biannic and Roos, 2017]. Now the H_∞-based approach of [Biannic and Apkarian, 2011] is also available. Recently developed nonsmooth optimization techniques are used to compute fixed-order feedback and anti-windup controllers in a single step by minimizing the weighted transfer seen by the deadzone nonlinearities. This approach is complementary to that described above. It does not allow directly to maximise a stability domain or a performance level, but it can be applied to high-order systems.

5 Summary of the contributions

The main contributions presented in this chapter are briefly summarized below, and a selection of related publications is given:

- **Section D.1 - D.2:** convex characterization of full-order continuous-time anti-windup controllers ; practical algorithm to design reduced-order controllers, whose poles are a subset of those obtained in the full-order case [Roos *et al.*, 2010; Biannic *et al.*, 2007].
- **Section D.3:** introduction of an upper bound on the real part of the anti-windup controller poles to avoid slow dynamics ; convex characterization of full-order continuous-time controllers with pole constraints [Roos and Biannic, 2008].
- **Section D.3:** implementation in the AWAS Toolbox, which is now embedded in the SAW Library of the SMAC Toolbox [Biannic and Roos, 2008; 2017].

An extension of this methodology to parameter-varying saturated systems is proposed in [Roos *et al.*, 2010; Roos, 2007], but it has not been discussed in this chapter due to space constraints. A non-standard anti-windup design strategy is also presented in [Biannic *et al.*, 2006a; Roos *et al.*, 2010] to improve the on-ground control system of a civilian aircraft. A linear fractional representation of the system (see Chapter C) is combined with an original approximation of the nonlinear ground forces by saturation-type nonlinearities. The resulting design method delivers low-order and robust controllers, which are automatically adapted to the runway state and to the aircraft longitudinal velocity (see Chapter E).

Chapter E

Control laws design and implementation

As illustrated in Figure E.1, automatic landing of a commercial aircraft can be divided into three main phases in the vertical plane: the final (or glide-slope) approach during which the aircraft must follow a predefined descent path, the flare segment which is activated when the landing gear height H falls below a threshold value $H_{FLA} \approx 15\,\text{m}$ (50 ft), and the deceleration on the runway where the aircraft velocity must be reduced up to taxi speed. Similarly, in the horizontal plane, the aircraft trajectory must coincide with the runway axis (localizer phase) as long as $H \geq H_{DEC} \approx 10\,\text{m}$ (30 ft). The alignment phase (or decrab mode) is then activated in order to minimize the lateral efforts on the landing gears at touchdown. Finally, once on the runway, the lateral deviation with respect to the centerline must be maintained below a given threshold (usually 16 meters) despite crosswind, varying runway state, comfort constraints...

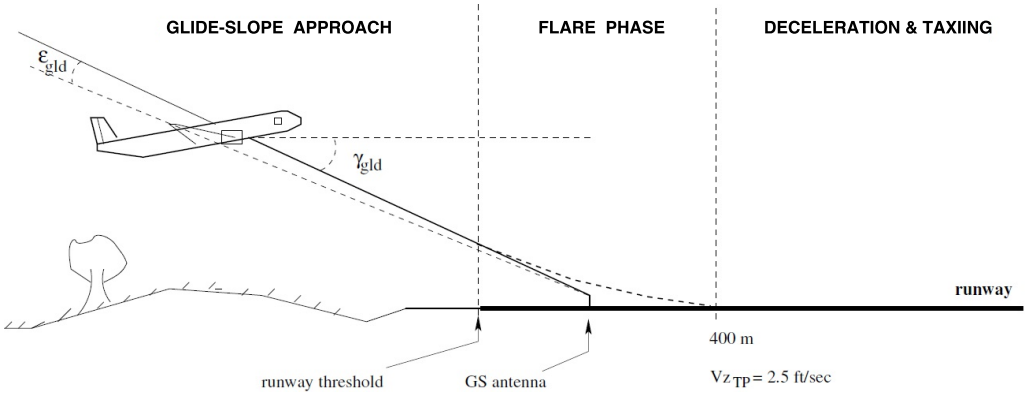


Figure E.1: Approach, landing and ground phases in the vertical plane

The steady growth of air traffic in recent years has led to drastic safety standards with the goal of limiting the number of accidents. Since approach and landing remain the most critical phases (almost 50% of fatal accidents and 75% of non-fatal hull losses between 1997 and 2016 [Airbus, Technical Report 2017]), particular attention has recently focused on improving autoland systems in adverse conditions. With the help of CAT III instrument landing systems (ILS), which are now available in a rapidly growing list of airports, automatic landing control laws allow to better secure these phases, notably in degraded weather conditions such as fog and crosswind. However, despite numerous methodological works over the past two decades (see *e.g.* [Kaminer and Khargonekar, 1990; Biannic and Apkarian, 2001; Looye and Joos, 2006; Sadat-Hoseini *et al.*, 2013]), the design, tuning and validation process of final approach and flare control systems remains a challenging and time-consuming task, which requires rather tricky multi-objective optimization.

Once on ground, the motion of the aircraft is usually controlled manually by the pilot using throttle levers, rudder pedals, handwheels and brake pedals. This is especially demanding in case of contaminated runway and severe crosswind. According to a study carried out by the International Civil Aviation Organization (ICAO), the number of runway excursions has not decreased over the last 20 years [Eurocontrol, 2013]. So a first benefit of developing enhanced on-ground control functionalities would be to improve safety during airport operations. But it would also allow for an increase in airport traffic capacity. Most existing solutions focus on low speed and taxiway maneuvers, where the nose-wheel steering system is sufficient to control the aircraft (see *e.g.* [Duprez *et al.*, 2004; Roos *et al.*, 2010; Lemay *et al.*, 2011; Bihua *et al.*, 2013]). However, the main issue after landing is runway axis hold at high speed despite varying conditions and external perturbations. This gives rise to a difficult control allocation problem, where differential braking must be used in conjunction with classical control devices (nose wheel steering system and rudder).

In the field of flight control design, the most popular methods are still based today on eigenstructure assignment [Magni, 1999] or Linear Quadratic Regulator (LQR) control [Sadat-Hoseini *et al.*, 2013]. Both approaches have been successfully used by Airbus and Boeing, and they have contributed to significant improvements in the flight control design process for nearly 30 years. But they reach their limits when it comes to tackling the above issues. In the meantime, H_∞ control techniques have been progressively developed and evaluated on various flight control problems, see for example [Kaminer and Khargonekar, 1990; Biannic and Apkarian, 2001] where the flare phase receives a particular attention. But despite promising results even in flight tests [Dorobantu *et al.*, 2012], this third approach has not become as popular as the other two in the industry. Things are, however, likely to change in the near future with the emergence of new tools based on nonsmooth optimization techniques [Apkarian and Noll, 2006; Burke *et al.*, 2006]. The latter allow to impose constraints on the structure and the order of the controller. Although convexity is unfortunately lost in that case, the aforementioned algorithms converge to local solutions, which are in a large majority of standard applications not so far from the global (non-structured) optimum. Another interesting feature of these new tools is their capacity to handle multiple models and multiple separate channels [Apkarian *et al.*, 2014].

In this context, the objective of this chapter is to show how such powerful control techniques can be efficiently combined with the analysis and design tools developed in Chapters A to D, in order to solve challenging control problems such as automatic landing, for which classical methods do not allow anymore to satisfy increasingly constrained specifications. More precisely, three methodology are proposed in Sections E.1, E.2 and E.3 to deal respectively with the approach, flare and gound phases. The resulting control architectures comply with several stringent requirements. Moreover, their complexity is compatible with implementation in flight computers. This will be demonstrated in a near future by means of flight tests performed with fixed-wing UAVs (see Section E.4).

1 Robust nonlinear compensation for the approach phase

1.1 Description of the methodology

Robust feedback linearization techniques [Isidori, 1995] have proved their efficiency in many aerospace applications, especially to control highly maneuverable aircraft or UAVs in large operating domains [Snell *et al.*, 1992; Adams and Banda, 1993; Reiner *et al.*, 1995; 1996; Kara Mohamed and Lanzon, 2012; Biannic *et al.*, 2014]. Interestingly, such techniques do not only permit to linearize and decouple a large class of nonlinear systems, but also to adapt the

control laws to the operating point. Therefore, they become a competitive alternative to standard gain-scheduling techniques which often imply many adjustments, or to LPV control design strategies which require high fidelity LPV models [Marcos and Balas, 2004].

However, standard dynamic inversion methods are often criticized for their lack of robustness and the need of an accurate model. This drawback is generally bypassed via robust linear outer loops [Bing-Yu and Blaise, 1998], which still require difficult and possibly time-consuming robustness evaluation *a posteriori* [Papageorgiou and Glover, 2004; 2005]. Severe problems are also likely to occur when the actuators dynamics and limitations prevent an exact cancellation of the nonlinear terms. As emphasized in [Kara Mohamed and Lanzon, 2013], it is then essential to take these dynamics into account in the design process. As observed in [Franco *et al.*, 2006], one of the main reasons why standard dynamic inversion schemes exhibit poor robustness properties is due to the fact that the linearizing inner loops are designed to convert the nonlinear system into a generic Brunovsky's form. Following an intuitive path, it is then proposed in [Franco *et al.*, 2006] to design the inner loops so that for given operating conditions, the nonlinear system will converge to its Jacobian linearization. Hence, the design of the linear robust outer loops is no longer based on a generic model but now explicitly depends on the linearized dynamics of the initial plant.

A similar method is presented in this section, which consists of promoting interactions between the inner and outer loops. More precisely, a feedback linearization is first applied so that in some enlarged neighborhood of given trim conditions, the nonlinear plant behaves like its linearization. Then, a robust outer loop is designed. The originality of this approach lies in the particular structure of the H_∞-based outer controller, which uses a nonlinear signal as a key input to further enlarge the operating domain of the closed-loop system. Various uncertainties are also taken into account thanks to a μ-based robustness analysis step, during which worst-case configurations are identified and then used in an iterative multi-objective and multi-model H_∞ design process.

1.1.1 Robust nonlinear compensation based on dynamic inversion

Consider a continuous-time parameter-dependent nonlinear input-affine system described as:

$$\begin{cases} \dot{\xi}(t) &= f(\xi(t),\theta_p(t)) + G(\xi(t),\theta_p(t))u(t) \\ u(t) &= L_A(u_c(t)) \end{cases} \tag{E.1}$$

where $\xi(t) \in \mathbb{R}^n$ denotes the physical states evolving in the admissible operating domain $\mathcal{X} \subset \mathbb{R}^n$. The realized control inputs $u(t) \in \mathbb{R}^m$ are derived from the commanded inputs $u_c(t) \in \mathbb{R}^m$ via LTI actuator models L_A with unitary static gains. The nonlinearities and parametric variations of the system are captured by functions f and G, which both depend nonlinearly on the state vector and on a set of parameters $\theta_p(t) \in \Theta \subset \mathbb{R}^r$.

Notation 1 *Let $(\bar{\xi},\bar{\theta}_p,\bar{u}) \in \mathbb{R}^n \times \mathbb{R}^r \times \mathbb{R}^m$ be an equilibrium point for system (E.1):*

$$f(\bar{\xi},\bar{\theta}_p) + G(\bar{\xi},\bar{\theta}_p)\bar{u} = \bar{f} + \bar{G}\bar{u} = 0 \tag{E.2}$$

Then $f(\xi(t),\theta_p(t))$ is rewritten as follows:

$$f(\xi(t),\theta_p(t)) = \bar{f} + Ax(t) + \Delta_f(t) \tag{E.3}$$

where:

$$A = \left.\frac{\partial f}{\partial \xi}\right|_{\bar{\xi},\bar{\theta}_p} \quad \text{and} \quad x(t) = \xi(t) - \bar{\xi} \tag{E.4}$$

and $\Delta_f(t)$ denotes the deviation between the nonlinear function f and its linear approximation.

Assumption E.1 *There exists a constant matrix $B \in \mathbb{R}^{n\times m}$ such that $\forall(\xi(t),\theta_p(t)) \in \mathcal{X}\times\Theta$, a nonsingular matrix $\Lambda \in \mathbb{R}^{m\times m}$ and a residual error matrix $\Delta_G \in \mathbb{R}^{n\times m}$ can be found, which satisfy:*

$$G(\xi(t),\theta_p(t)) = B\Lambda + \Delta_G \tag{E.5}$$

$$L_A(\Lambda^{-1}\upsilon(t)) \approx \Lambda^{-1}L_A(\upsilon(t)) \quad \forall\upsilon(t) \in \mathbb{R}^m \tag{E.6}$$

The square matrix Λ typically represents the control inputs efficiency. When considering aerospace systems evolving in standard operating domains, the above non-singularity assumption – connected to the notion of controllability – is not restrictive. Moreover, the variations of this diagonal-dominant matrix are mainly induced by slowly-varying terms such as the dynamic pressure. This observation justifies the commutative property (E.6) between Λ^{-1} and the fast dynamics L_A of the actuators. Yet, a possible relaxation of (E.6) is proposed in Section E.1.1.2.

Given any two signals $v(t) \in \mathbb{R}^m$ and $\zeta(t) \in \mathbb{R}^m$, let us now define the intermediate, input linearizing, control law:

$$u_c(t) = \Lambda(\xi(t),\theta_p(t))^{-1}(v(t) - \zeta(t)) + \bar{u} \tag{E.7}$$

With the above notation in mind, combining equations (E.7) and (E.1) leads to:

$$\dot{x} = Ax + BL_A(v) + w_f + w_u \tag{E.8}$$

where parametric- and time-dependence have been omitted for clarity and:

$$w_f = \Delta_f - BL_A(\zeta) \tag{E.9}$$

$$w_u = (G - \bar{G})\bar{u} + \Delta_G\Lambda^{-1}L_A(v - \zeta) \tag{E.10}$$

Equation (E.8) shows that the parameter-dependent nonlinear system (E.1) has been reduced to a linear model with a new control input v and two measured perturbations w_f and w_u. As usual in dynamic inversion schemes, w_f can be partly canceled by an optimal choice of the auxiliary input signal $\zeta(t)$:

$$\hat{\zeta}(t) = \underset{\zeta(t)\in\mathbb{R}^m}{\arg\min} \|\Delta_f(t) - BL_A(\zeta(t))\| \tag{E.11}$$

Remark E.1 *In the special case of square systems with ideal actuators, i.e. $L_A(u(t)) = u(t)$, $w_f = 0$ is easily obtained with $\hat{\zeta}(t) = B^{-1}\Delta_f(t)$.*

Let $w = w_f + w_u$ be the vector of remaining input perturbations which cannot be canceled, and assume that it is available for feedback, via estimation, at least on a limited bandwidth. The following structure for the linear outer loop can thus be considered:

$$v = K(s)\begin{pmatrix}\hat{w}\\ w_c\\ y\end{pmatrix} \tag{E.12}$$

where $\hat{w}$, w_c and y denote respectively the estimation of w, the reference inputs and the measured outputs. Without significant loss of generality in most applications, both $y(t) = Cx(t) \in \mathbb{R}^p$ and $z(t) = Lx(t) \in \mathbb{R}^q$ are assumed to depend linearly on the state vector x, where z are the variables to be tracked. The output feedback controller $K(s)$ in (E.12) is to be designed so as to satisfy the following requirements:

1. good tracking properties by minimizing the error between z and the reference outputs $z_r = R(s)w_c$, where the LTI model $R(s)$ describes the nominal closed-loop dynamics,

2. reasonable control activity, which is indirectly obtained by limiting v to avoid control inputs saturations,

3. good rejection of the perturbations $w = w_u + w_f$ which could not be entirely removed by the linearizing inner loop, so as to enlarge the operating domain.

Let $\Sigma_A(s)$ and $\Sigma(s) = \left[\begin{array}{c} L \\ C \end{array}\right] (sI - A)^{-1} \left[\begin{array}{cc} I & B \end{array}\right]$ denote the transfer functions associated to the linear operator L_A and to the linearized model (E.8) respectively. The above outer loop design issue can be recast as a multi-objective H_∞ minimization problem. More precisely, considering the linear interconnection of Figure E.2, it is proposed to compute $K(s)$ by solving:

$$\hat{K}(s) = \arg\min_{K(s)} \|\mathcal{T}_{w_c \to z_p}(s)\|_\infty \tag{E.13}$$

under the constraints:

$$\left\{ \begin{array}{l} \|\mathcal{T}_{w_c \to z_u}(s)\|_\infty \leq \gamma_u \\ \|\mathcal{T}_{w \to z_p}(s)\|_\infty \leq \gamma_r \end{array} \right. \tag{E.14}$$

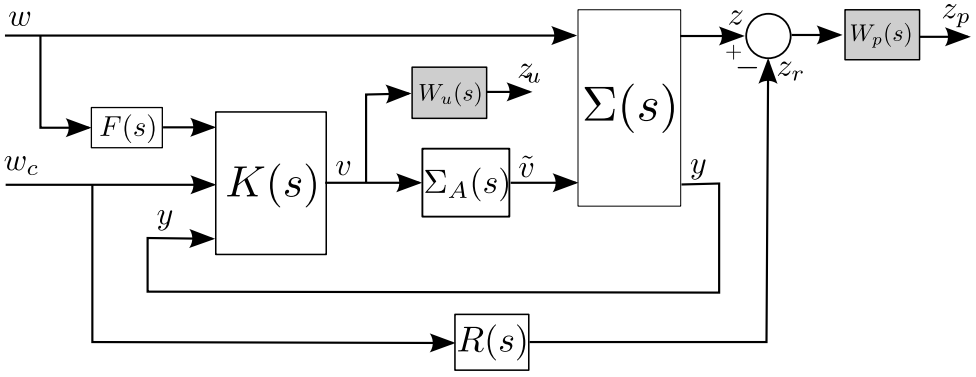


Figure E.2: H_∞ design-oriented scheme

The H_∞-norm minimization of equation (E.13) corresponds to the nominal tracking requirement. A weighting function $W_p(s)$ – typically a low-pass filter – is used to specify the frequency domain where this tracking should be the most efficient. Next, the second and third requirements are respectively taken into account by the two additional constraints in (E.14). The first one involves a high-pass weighting function $W_u(s)$ to minimize the control activity in the high-frequency domain. Finally, a low-pass transfer function $F(s)$ is introduced and can be viewed as an approximation of the estimation process, *i.e.* $\hat{w} \approx F(s)w$.

Problem (E.13)-(E.14) is a difficult nonconvex and nonsmooth optimization problem. However, it can be efficiently solved thanks to recent advances in nonsmooth optimization techniques [Apkarian and Noll, 2006; Burke *et al.*, 2006]. Moreover the structure of the controller as well as its order can be fixed *a priori*, which makes the implementation easier. And last but not least, multiple models can be considered simultaneously during the design process. This flexibility will be used below to improve the robustness properties of the proposed design scheme.

1.1.2 Preliminary LTI robustness analysis

Now that an initial controller has been designed, let us focus more closely on the validity of the approximations made in Section E.1.1.1 and on their potential impact on the closed-

loop behavior. More precisely, it is shown below how μ-analysis can be used to detect possible problems induced by the three main sources of uncertainties.

Plant uncertainties

When combining equations (E.1) and (E.7) to get (E.8), it is assumed that both f and G are well-known. But in practice, uncertainties δ are always present so that equation (E.8) reads:

$$\dot{x} = A(\delta)x + B(\delta)L_A(v) + w \tag{E.15}$$

where $\delta = \begin{bmatrix}\delta_1 & \dots & \delta_l\end{bmatrix}^T \in \mathbb{R}^l$. Assuming that both A and B depend rationally on δ, $\Sigma(s)$ is replaced with $\Sigma(s,\delta)$ and written as follows:

$$\Sigma(s,\delta) = \mathcal{F}_u(M(s), \Delta_M) \tag{E.16}$$

where $\mathcal{F}_u$ denotes the upper Linear Fractional Transformation (LFT, see Chapter C), $M(s)$ is an LTI system and:

$$\Delta_M \in \boldsymbol{\Delta_M} = \{\text{diag}(\delta_1 I_{n_1}, \dots, \delta_l I_{n_l}), \delta_i \in \mathbb{R}\} \tag{E.17}$$

Actuators uncertainties

In Assumption E.1, the commutative property (E.6) may not be valid in all cases. Following [Biannic *et al.*, 2012], it is then proposed to relax it as follows.

Assumption E.2 *There exist a nonlinear bounded operator* $\Gamma(.)$ *and a positive bound* $k_\Gamma \in \mathbb{R}_+$ *such that* $\forall v(t) \in \mathbb{R}^m$*:*

$$L_A(\Lambda^{-1}v(t)) = \Lambda^{-1}L_A(v(t)) + \Gamma(v(t)) \tag{E.18}$$

$$\|\Gamma(v(t))\| \leq k_\Gamma \|v(t)\| \tag{E.19}$$

It results from Assumption E.2 that the linear model $\Sigma_A(s)$ in Figure E.2 should now be replaced by $\Sigma_A(s) + \Gamma(.)$, as illustrated in Figure E.3.

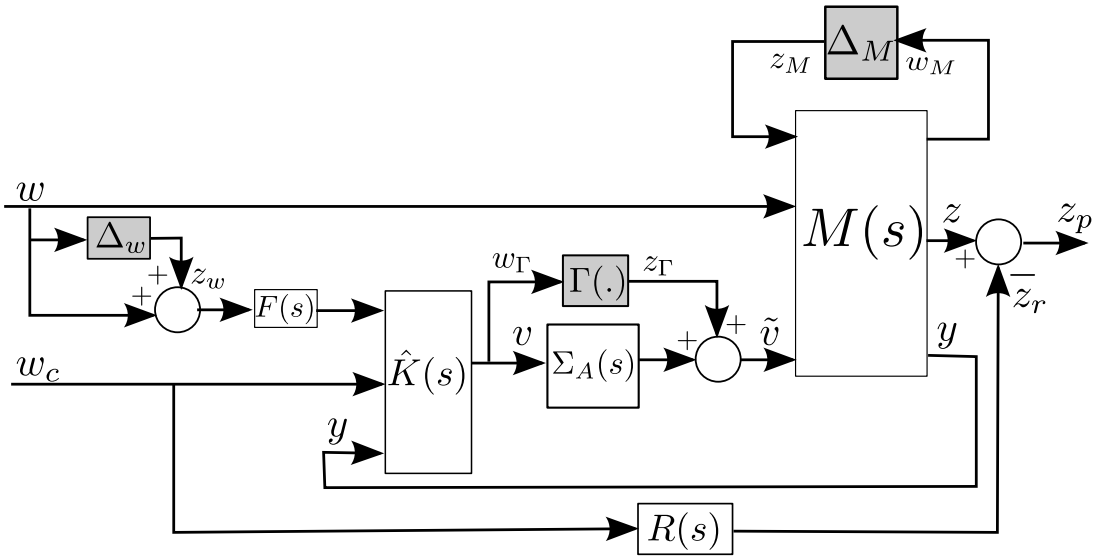


Figure E.3: Closed-loop scheme for robustness analysis

Estimation uncertainties

Finally, it should be emphasized that only an estimate $\hat{w}$ of the nonlinear input perturbations w is available to the outer loop controller $\hat{K}(s)$. The estimation process has been taken into account in the design phase through the approximation $\hat{w} \approx F(s)w$, but it might be too optimistic in practice. Therefore, a diagonal perturbation block Δ_w corresponding to multiplicative uncertainties is added to the robustness analysis diagram of Figure E.3, so that $\hat{w}$ is replaced with $(I + \Delta_w)\hat{w}$.

The three operators described above are then normalized, merged into a single block-diagonal operator $\Delta = \mathrm{diag}\,(\Delta_M, \Gamma(.), \Delta_w)$ and pulled out to generate an augmented linear model $P(s)$, as depicted in Figure E.4. This can be achieved using the APRICOT and the GSS Libraries of the SMAC Toolbox (see [Roos *et al.*, 2014; Biannic and Roos, 2016] and Chapter C).

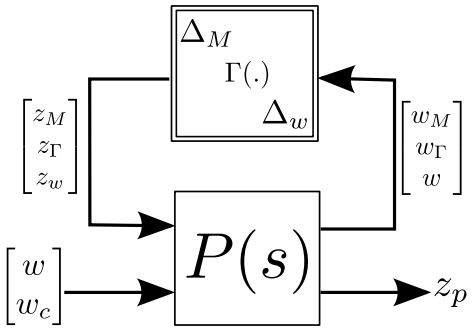


Figure E.4: LFT-based representation for robustness analysis

Assuming for the moment that Δ is a time-invariant operator, μ-analysis can be used to assess the robustness properties of the uncertain closed-loop interconnection depicted in Figure E.4. The SMART Library of the SMAC Toolbox (see [Roos, 2013] and Chapter A) can be used for this purpose. Since Δ is normalized, the system is robustly stable if the resulting μ upper bound is lower than 1. Otherwise, if the μ lower bound is larger than 1, there exists a destabilizing worst-case uncertainty matrix Δ^* such that $\overline{\sigma}(\Delta^*) < 1$.

1.1.3 Multi-model design

There exist two main ways to improve the initial controller $\hat{K}(s)$ using the above robustness analysis results. Since it was designed in the H_∞ framework, it seems natural to use a μ-synthesis approach. Existing software [The Mathworks, 2017b] is based on [Young, 1996] and references therein. It works well in the case of complex-valued uncertainties, but numerical difficulties are often reported with real-valued uncertain parameters. Moreover, results may be quite conservative in that case. A potentially more efficient technique is proposed in [Apkarian, 2011], but no practical implementation is currently available. For these reasons, an alternative is preferred here, which consists of performing a multi-model design. First considered in [Ackermann, 1985], this intuitive approach leads to nonconvex optimization problems. However, as already pointed out in Section E.1.1.1, it has regained interest recently thanks to the flexibility of nonsmooth H_∞ optimization algorithms. The latter indeed offer new perspectives, since multiple models can be considered simultaneously. From this observation, a simple iterative algorithm is proposed below. Starting from a single-model design, the idea is to alternate robustness analysis and multi-model design. During the analysis step, worst-case configurations Δ^* are isolated to enrich the bank of models to be considered in the next design step.

Algorithm E.1 (robust multi-model design)

1. *Choose a parametric configuration $\bar{\theta}_p$ and apply the linearizing inner loop control law (E.7) to the nonlinear system (E.1), so as to get a linear nominal model $\Sigma_0(s)$ described by (E.8).*
2. *Set $i = 0$. Solve problem (E.13)-(E.14) to compute an initial controller $\hat{K}_0(s)$.*
3. *Perform a μ-analysis based robustness test as described in Section E.1.1.2. If the μ upper bound is less than 1, **stop** since the controller stabilizes the system on the whole parametric domain. If the μ lower bound in larger than one, extract the corresponding destabilizing matrix Δ^* and compute the associated model $\Sigma(s, \delta^*)$.*
4. *Set $i \leftarrow i + 1$. Add $\Sigma(s, \delta^*)$ to the bank of models. Compute $\hat{K}_i(s)$ via multi-model H_∞ design and go back to step 3.*

1.1.4 Towards a global robustness analysis

In the simplified robustness analysis approach of Section E.1.1.2, the nonlinear input signal w is considered as an external perturbation. However, in view of (E.8) and (E.9), it is clear that w may depend on x and θ_p in a quite complicated way. Robustness analysis thus becomes tricky in the most general case. Fortunately, with a good knowledge of the considered system, such as the one discussed in Section 1.2, reasonably simpler approximations can be obtained in practice, for example:

$$w = H(\theta_p)x + W(x) \tag{E.20}$$

with the following assumptions:

- $H(.)$ depends rationally on θ_p and can be written in LFT form,
- the nonlinear operator $W(.)$ satisfies Lipschitz conditions.

Hence, the LFT model $\mathcal{F}_u(P(s), \Delta)$ of Figure E.4 is modified to include additional blocks in Δ, so that w no longer appears as an external perturbation:

$$\Delta = \text{diag}\,(\Delta_M, \Gamma(.), \Delta_w, W(.), \Theta_p) \tag{E.21}$$

The operator (E.21) now contains uncertainties Δ_M and Δ_w, time-varying parameters Θ_p, as well as memoryless nonlinearities $\Gamma(.)$ and $W(.)$. Robustness analysis must then be performed with more general tools based for example on the IQC framework [Megretski and Rantzer, 1997].

1.2 Application to the approach phase

The robust nonlinear compensation technique of Section E.1.1 is now applied to a commercial aircraft during the approach phase. A full description of the considered benchmark is out of the scope of this manuscript, but detailed equations are available in the appendix of [Biannic and Roos, 2018] and the whole Matlab package can be downloaded from the SMAC website `http://w3.onera.fr/smac/aircraftModel`. This section only focuses on the longitudinal behavior of the aircraft. The lateral behavior is described in [Biannic *et al.*, 2017].

1.2.1 Nonlinear longitudinal aircraft model

Let us consider the following fourth-order longitudinal aircraft model:

$$\left\{\begin{array}{lcl} m\dot{V} & = & -\bar{q}SC_D(\alpha) - mg\sin\gamma + F\cos\alpha \\ mV\dot{\gamma} & = & \bar{q}SC_L(\alpha, q, V, \delta_e) - mg\cos\gamma + F\sin\alpha \\ J\dot{q} & = & \bar{q}SLC_m(\alpha, q, V, \delta_e) + z_e F \\ \dot{\theta} & = & q \end{array}\right. \tag{E.22}$$

with V the airspeed, γ the flight path angle, q the pitch rate, θ the pitch angle, $\overline{q} = \frac{1}{2}\rho V^2$ the dynamic pressure, J the longitudinal inertia, m the mass, S the reference surface, L the reference length, ρ the air density, α the angle of attack, F the engine thrust on the longitudinal axis and z_e the vertical shift between the position of the center of gravity and the thrust application point. The drag, lift and pitching moment coefficients C_D, C_L and C_m can be expressed as follows:

$$\begin{aligned} C_D(\alpha) &= C_{D0} + C_{D\alpha}\alpha + C_{D\alpha 2}\alpha^2 \\ C_L(\alpha, q, V, \delta_e) &= \underbrace{C_{L0} + C_{L\alpha}\alpha + C_{Lq}\frac{q}{V}}_{\tilde{C}_L} + C_{L\delta_e}\delta_e \\ C_m(\alpha, q, V, \delta_e) &= \underbrace{C_{m0} + C_{m\alpha}\alpha + C_{mq}L\frac{q}{V}}_{\tilde{C}_m} + C_{m\delta_e}\delta_e \end{aligned} \tag{E.23}$$

where δ_e is the elevator deflection angle and the C_{xy} are fixed stability derivatives affected by $\pm 30\,\%$ of multiplicative uncertainties. Here $\xi = [V\ \gamma\ q\ \theta]^T$, $u = [F\ \delta_e]^T$ and $\theta_p = [m\ x_{cg}\ V_0]^T$, where x_{cg} and V_0 are the center of gravity position and the initial airspeed respectively. The operating domain for the considered approach phase is $m \in [123\ 180]$ tons, $x_{cg} \in [15\ 41]\,\%$ and $V_0 \in [60\ 80]$ m/s, defining the admissible set Θ. First-order linear actuators $\Sigma_A(s)$ are introduced.

Remark E.2 *Using first-order models for the actuators is common practice in the aeronautical industry, and it is sufficiently representative here. Rate limitations are neglected, but it would be possible to add an anti-windup controller (see Chapter D) to take them into account.*

Equation (E.22) is rewritten as in (E.1), (E.5) and (E.6), where:

$$f = \begin{bmatrix} \frac{-\overline{q}S}{m}C_D - g\sin\gamma \\ \frac{\overline{q}S}{mV}\tilde{C}_L - \frac{g}{V}\cos\gamma \\ \frac{\overline{q}SL}{J}\tilde{C}_m \\ q \end{bmatrix} \quad , \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad , \quad \Lambda = \begin{bmatrix} \frac{\cos\alpha}{m} & 0 \\ \frac{z_e}{J} & \frac{\overline{q}SL}{J}C_{m\delta_e} \end{bmatrix} \quad , \quad \Delta_G = \begin{bmatrix} 0 & 0 \\ \frac{\sin\alpha}{mV} & \frac{\overline{q}S}{mV}C_{L\delta_e} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Remark E.3 *Given the operating domain of the system, Λ is nonsingular. Moreover, Λ is slowly varying, since its dynamics mainly comes from the dynamic pressure $\overline{q}$. Therefore, Assumption E.1 is satisfied.*

Remark E.4 *The choice of B was made based on the maximum control efficiency. Indeed, as can be seen in (E.22), the thrust F mainly affects the airspeed V, while the elevator deflection δ_e has a large impact on the pitch rate q.*

1.2.2 Nonlinear compensation technique

The method described in Section E.1.1.1 can now be applied. The inner loop control law (E.7) is first implemented, which leads to the linearized system (E.8). $\hat{\zeta}$ is chosen to cancel the nonlinear dynamics on V_a and q, so that only the nonlinear terms on γ are kept in w_f. The nominal configuration $\bar{\theta}_p$ corresponds to a mass of 150 tons, a center of gravity position at 21% and an initial airspeed of 70 m/s. This is a central configuration in terms of the location of the linearized system poles. Choosing an extreme configuration may also be an option, but nominal performance is severely degraded in this case and it is difficult to control the opposite extreme configurations. Finally, the frequency weighting functions $W_p(s)$ and $W_u(s)$ as well as the filter $F(s)$ are defined as explaind in Section E.1.1.1 (see [Biannic *et al.*, 2017] for numerical values).

The optimization problem (E.13)-(E.14) is solved. A third-order controller $\hat{K}_0(s)$ proves to be a good compromise, since it offers a satisfactory performance level and it is sufficiently simple to be implemented in a flight computer. Simulations are performed with this controller on the full nonlinear benchmark, and a few results are displayed in Figure E.5. The upper plots show the aircraft response to a 3 deg step demand on the flight path angle γ for various parametric configurations: $m \in \{120, 150, 180\}$ tons, $x_{cg} \in \{15, 20, 40\}$ % and $V_0 \in \{60, 70, 80\}$ m/s. Similarly, the lower plots show the response to a 3 m/s step demand on the airspeed V. The red plots correspond to the reference signals z_r to be tracked, *i.e.* to the outputs of the reference model $R(s)$.

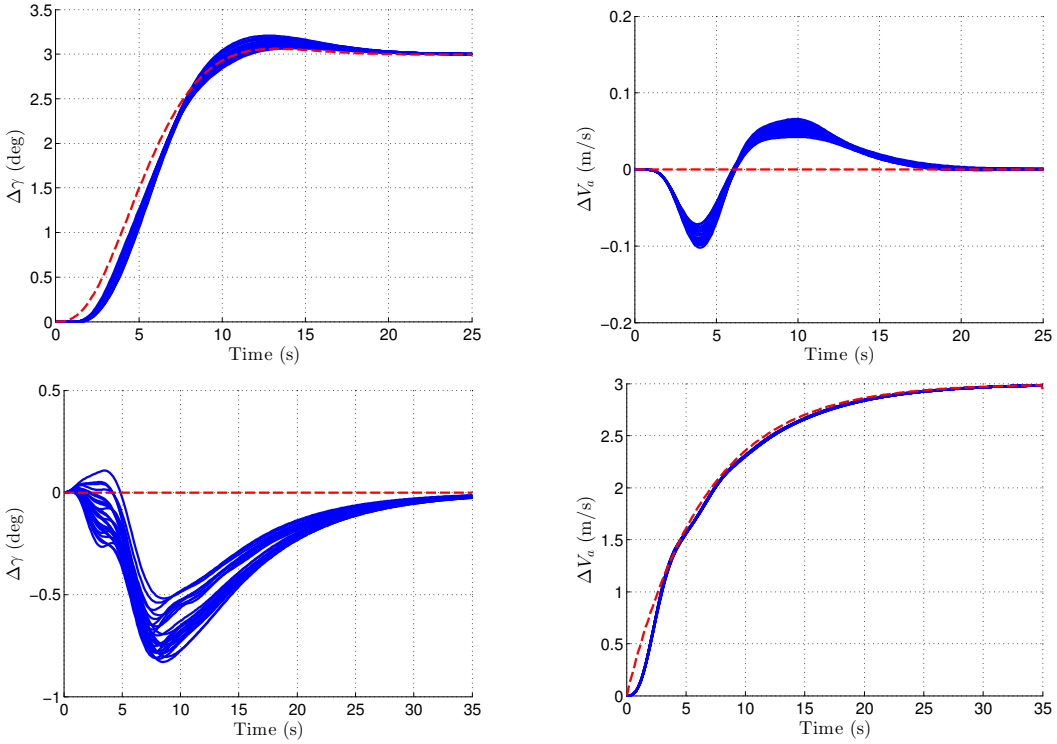


Figure E.5: Nonlinear simulations with the initial controller $\hat{K}_0(s)$

These simulations confirm the good stability and performance properties of the robust nonlinear compensation scheme, which maintains near-nominal performance in a large flight domain. Moreover, the controller follows the reference models quite well and a good decoupling between γ and V is also ensured.

1.2.3 Robustness analysis and multi-model design

The controller $\hat{K}_0(s)$ has been designed to cope with a large operating domain in nominal conditions. Therefore, the uncertain operators represented by the gray boxes in Figure E.3 have not yet been taken into account. The objective of this section is twofold. First, the effect of these uncertainties is evaluated with μ-analysis tools. Next, the resulting worst-case configurations are used in the multi-model design strategy of Algorithm E.1 to compute a more robust controller.

Multiplicative uncertainties δ_{C_D}, δ_{C_L}, δ_{C_m} are introduced on the aerodynamic coefficients C_D, C_L, C_m, as well as additive uncertainties δ_V, δ_m, $\delta_{x_{cg}}$ on the airspeed, the mass and the center of gravity location. Equations (E.22) are linearized and a set of LTI models is obtained.

An LFR is then generated as in equations (E.16)-(E.17) using the APRICOT and the LTI2LFR Libraries presented in Chapter C. Thanks to sparse approximation techniques, the size of Δ_M is reasonable, so that μ-analysis tools can be easily applied:

$$\Delta_M = \text{diag}\left(\delta_{C_D} I_2, \delta_{C_L} I_3, \delta_{C_m} I_2, \delta_V I_6, \delta_m, \delta_{x_{cg}}\right) \in \mathbb{R}^{15\times 15} \tag{E.24}$$

As a result, the size of the operator $\Delta = \text{diag}\left(\Delta_M, \Gamma(.), \Delta_w\right) \in \mathbf{\Delta}$ in Figure E.4 is 20×20. A normalization step is performed, so that $\pm 30\%$ of uncertainties are introduced when Δ is restricted to the unit ball, *i.e.* $\overline{\sigma}(\Delta) < 1$. Upper and lower bounds on the structured singular value μ are finally computed with the SMART Library of the SMAC Toolbox. With the nominal controller $\hat{K}_0(s)$, a lower-bound $\underline{\mu}_{\mathbf{\Delta}} > 1$ is obtained as well as a worst-case configuration Δ^* such that $\overline{\sigma}(\Delta^*) = \underline{\mu}_{\mathbf{\Delta}}^{-1} < 1$. Δ^* is used to initialize the multi-model design procedure of Algorithm E.1. A new controller $\hat{K}(s)$ is obtained after 5 iterations. Robustness is significantly improved, since $\overline{\mu}_{\mathbf{\Delta}} < 1$ thanks to the branch-and-bound technique implemented in the SMART Library.

The same nonlinear simulations as in Section E.1.2.2, *i.e.* without uncertainties, are now performed with $\hat{K}(s)$ instead of $\hat{K}_0(s)$. Results are displayed in Figure E.6. As expected, performance has slightly decreased, since $\hat{K}_0(s)$ was optimized for the nominal case: responses are a little bit slower and decoupling is not as good.

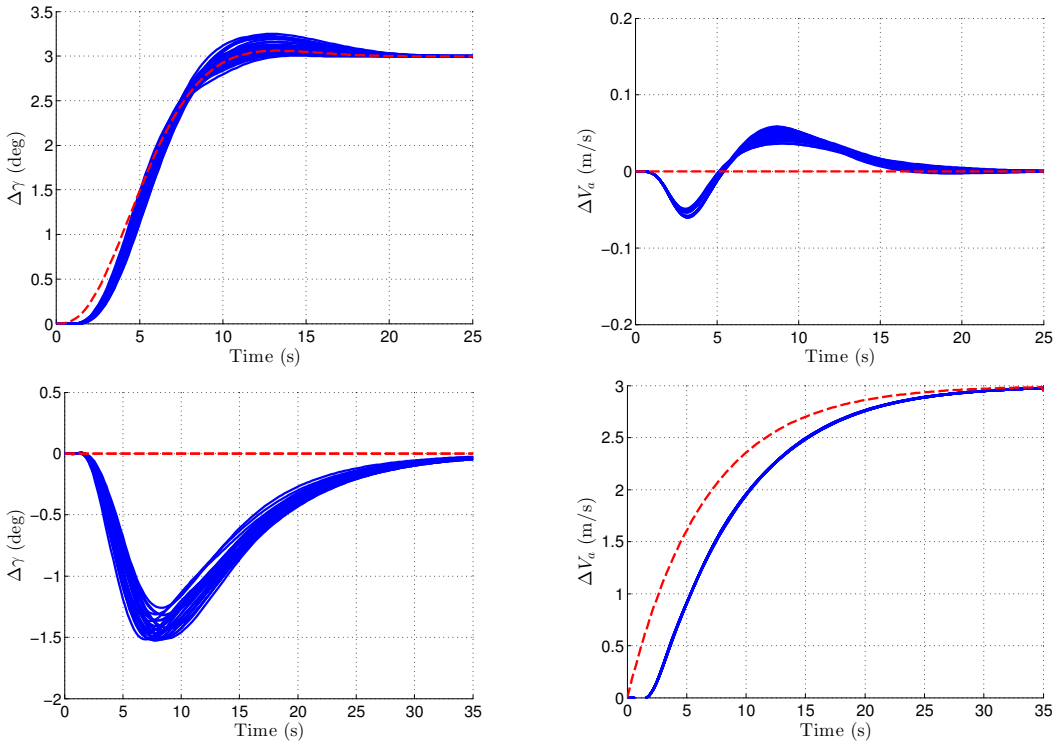


Figure E.6: Nonlinear simulations with the more robust controller $\hat{K}(s)$

However, major improvements are obtained with $\hat{K}(s)$ in the presence of uncertainties. This can be observed in Figure E.7, where $\pm 25\%$ uncertainties are allowed on the aerodynamic coefficients C_D, C_L and C_m. Results are not presented for $\pm 30\%$. Parametric configurations can indeed be found, for which the nominal controller $\hat{K}_0(s)$ is unstable (unlike $\hat{K}(s)$ which is always stable). This demonstrates the efficiency of the proposed methodology.

Remark E.5 *The lateral controller design is not presented here for the sake of brevity, but the whole process is described in [Biannic* et al.*, 2017].*

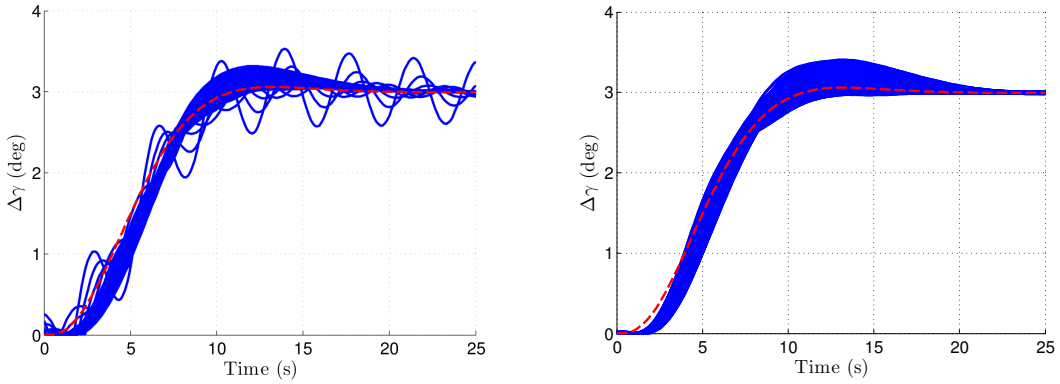
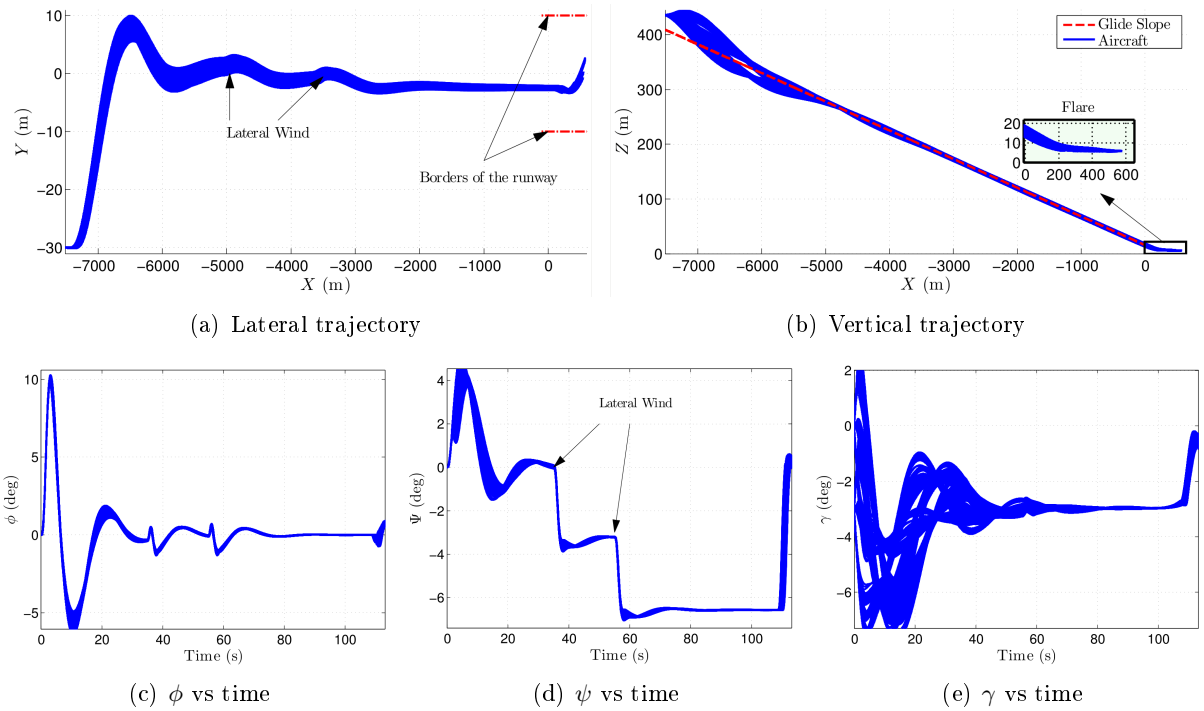


Figure E.7: Comparison between $\hat{K}_0(s)$ (left) and $\hat{K}(s)$ (right)

1.2.4 Final approach simulations

PID-based guidance laws are designed in addition to the control laws described above, so as to bring to zero the lateral and the vertical deviations with respect to the glide and the localizer beams respectively. Basic flare and debrab laws are also implemented, whose robustness will be improved in Section E.2. Several landing simulations are then run. Results are displayed in Figure E.8 for different initial values of mass, center of gravity position, uncertainties and flight path angle. Two lateral wind gusts are generated at 35 sec and 55 sec.



(a) Lateral trajectory

(b) Vertical trajectory

(c) ϕ vs time

(d) ψ vs time

(e) γ vs time

Figure E.8: Landing simulations in the presence of lateral wind for different initial conditions

In all cases, the reference glide path $\gamma_{gld} = -3\,\text{deg}$ is reached long enough before touchdown (Figure E.8(b)). The flare controller then allows to land around 400 m after runway threshold, while reducing γ (Figure E.8(e)) and consequently the vertical velocity. On the lateral axis,

the decrab controller brings the azimuth angle Ψ back to 0 once arriving above the runway (Figure E.8(d)), while keeping the wings in a horizontal position (bank angle ϕ very close to 0 in Figure E.8(c)). Finally, no lateral excursion can be observed at touchdown (Figure E.8(a)).

1.3 Conclusion

Inspired by dynamic inversion techniques, an original methodology is proposed to design nonlinear controllers over possibly large operating domains. A partial feedback linearization of the plant is combined with a fixed-structure multi-model H_∞ design technique. A preliminary μ-based validation step is then performed, during which worst-case configurations are identified and used to progressively enrich the set of design models. A global nonlinear validation strategy is finally sketched. The resulting algorithm is evaluated on a realistic aircraft landing problem. Future work will focus on its application to a small fixed-wing UAV, including flight tests, as explained in Section E.4.

2 Robust multi-model H_∞ design for the flare phase

2.1 Overall structure of the flare control system

The nominal slope during approach is generally fixed to $\gamma_{gld} = -3\,\text{deg}$, and for the considered aircaft the nominal approach airspeed is around $V = 70\,\text{m/s}$ (see Section E.1.2). Therefore, the vertical speed V_z remains above $3.5\,\text{m/s}$. The objective of the flare phase is to provide enhanced vertical speed control capacities, so that the aircraft hits the runway about 400 m after threshold with a much lower vertical speed of $0.76\,\text{m/s}$ ($2.5\,\text{ft/s}$). During this short maneuver, rarely exceeding 7 s, the engines are set to the idle position and the thrust δ_{th} is constant. Therefore, the only control input along the longitudinal axis is the elevator deflection δ_e. Basic flare control laws were already used in Section E.1.2.4 to perform full landing simulations, but they are not robust enough. A more efficient structure is shown in Figure E.9, which consists of two nested loops. A reference vertical speed profile is generated by the outer loop, which sends a commanded vertical speed V_{zc} to the inner loop. This profile depends on both the initialization altitude $H_{FLA} \approx 15\,\text{m}$ (50 ft) and the airspeed. It must be tracked despite external perturbations and model uncertainties thanks to an appropriate tuning of the inner loop vertical speed controller, whose design is described in this section.

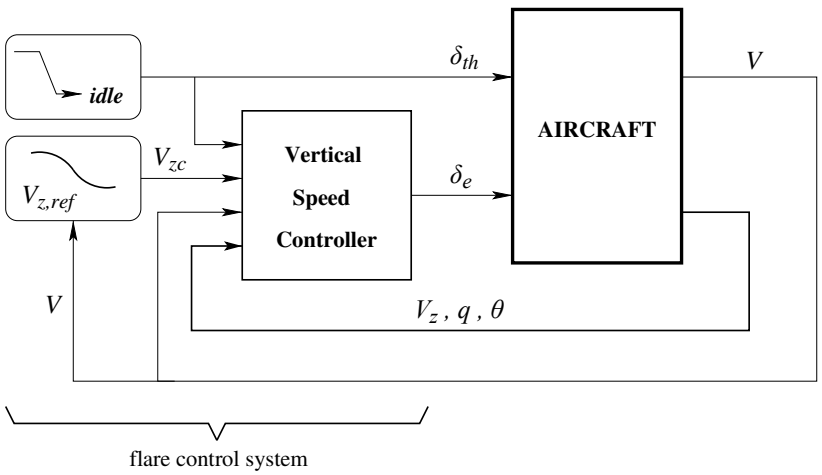


Figure E.9: Flare control system

2.2 Vertical speed control via multi-channel structured H_∞ optimization

The central objectives and constraints of the flare phase are fundamentally different from those encountered during the final approach. From a control viewpoint, the design problem appears to be simpler since only one variable (the vertical speed) is being tracked and only one control input (the elevator deflection) is being managed. However, the fluctuations of V (which is no longer controlled) and the engines deceleration generate perturbations. In addition to the ground and wind effects, this finally makes the control problem quite difficult. Fortunately, the latter falls within the scope of H_∞ techniques.

The key issue is to define an appropriate design scheme, which begins with the choice of relevant linear approximations of the aircraft longitudinal dynamics near the ground. Such models are obtained using trimming and linearization tools for various points of the two-dimensional flare trajectory. A set of third-order models $\Sigma(s, H)$ parameterized by the landing gear height H is obtained, whose state-space representation is of the following form:

$$\begin{bmatrix} \dot{V}_z \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = A(H) \begin{bmatrix} V_z \\ q \\ \theta \end{bmatrix} + B_1(H)\,\delta_e + B_2(H) \underbrace{\begin{bmatrix} V \\ \delta_{th} \\ w \end{bmatrix}}_{w_{pert}} \tag{E.25}$$

In these models, V and δ_{th} are no longer viewed as a state and a control input respectively, but as external perturbations. However, unlike wind inputs w, they are measured and can be used for feedback. This is illustrated in the design diagram of Figure E.10. The controller to be optimized is a static gain $K \in \mathbb{R}^7$. The proposed vertical speed control law then reads:

$$\delta_{e_c} = K \left[V_{z_r} \;\; V_{z_c} \int (V_{z_c} - V_z) \;\; V_z \;\; q \;\; \theta \;\; V \right]^T \tag{E.26}$$

where δ_{e_c} is the commanded elevator deflection and V_{z_r} is the output of a linear reference model $R(s)$ to be tracked. The main interest of the proposed structure is to remain close to the one used during the approach phase, which not only simplifies the switching strategy but also contributes to an improved safety level.

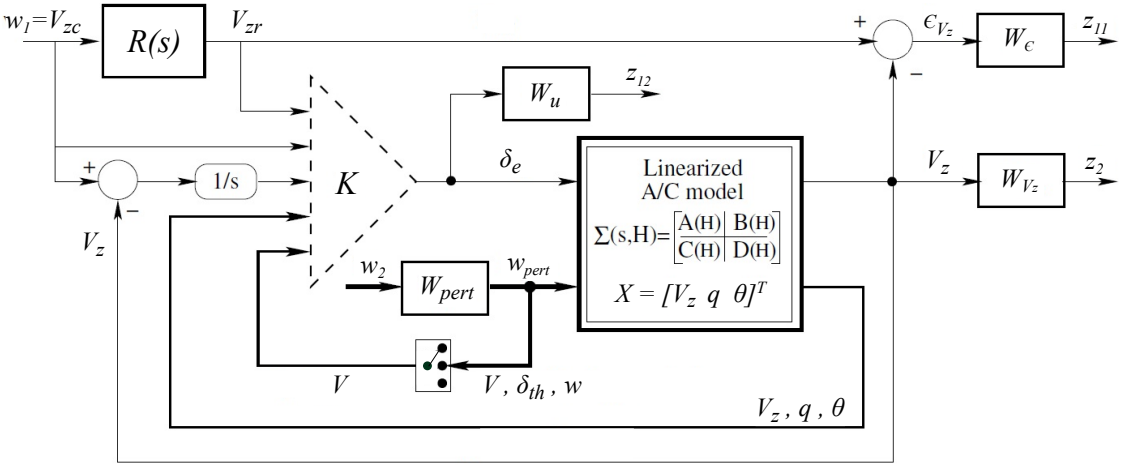


Figure E.10: H_∞ design scheme

Let us now focus on the optimization criteria. Beyond the standard closed-loop stability requirements, the objective is to determine the best controller $\hat{K}$ such that:

- the norm of the error $\epsilon_{V_z} = V_{z_r} - V_z$ is minimized,
- the control activity is such that magnitude and rate saturations are avoided,
- the perturbations w_{pert} have a negligible effect on the vertical speed variations.

In the H_∞ framework [Zhou *et al.*, 1996] and using the notation of Figure E.10, the first two objectives are realized through the H_∞ norm minimization of the map from w_1 to $[z_{11}\ z_{12}]^T$. The latter can be written as:

$$\mathcal{T}_{w_1 \to [z_{11}\ z_{12}]^T}(s) = \mathcal{F}_l(P_1(s,H), K) \tag{E.27}$$

where $\mathcal{F}_l$ stands for the lower linear fractional transformation and $P_1(s,H)$ is a parameterized (height dependent) weighted LTI model. Following standard rules, the weighting functions $W_\epsilon(s)$ and $W_u(s)$ can be chosen as a high-gain low-pass filter and a static gain respectively. The third objective is achieved by the H_∞ norm minimization of the map from w_2 (where $w_{pert} = W_{pert}(s)w_2$) to $z_2 = W_{V_z}(s)V_z$, which can be written as:

$$\mathcal{T}_{w_2 \to z_2}(s) = \mathcal{F}_l(P_2(s,H), K) \tag{E.28}$$

The three-dimensional filter $W_{pert}(s)$ is used to shape the input perturbations. As (E.28) is considered independently of (E.27), it is not restrictive to choose a static function $W_{pert}(s) = \mathrm{diag}(W_V, W_{\delta th}, W_w)$, where the gains $W_V, W_{\delta th}$ and W_w are used to adjust the relative weights of each input perturbation. The three corresponding transfer functions are then weighted by a single scalar dynamic filter $W_{V_z}(s)$. In order to remove any static error on V_z (mostly induced by the speed variations), which could be critical near ground, a high-gain low-pass filter must be used. To summarize the above discussion, the controller optimization can be stated as the following multi-model and multi-channel structured H_∞ control design problem:

$$\hat{K} = \arg\min_{K} \max_{\substack{H \le H_{FLA} \\ i=1,2}} \left\{ \left\| \mathcal{F}_l(P_i(s,H), K) \right\|_\infty \right\} \tag{E.29}$$

2.3 Robustness against parametric uncertainties and modeling errors

By simultaneously considering several design models for different values of the landing gear height H, robustness against ground effects modeling errors is enforced in the proposed design process. This is, however, not sufficient in practice, where parametric robustness against mass and center of gravity location is also required. To this end, the above multi-objective H_∞ design framework is generalized to take these additional uncertainties into account. Therefore, the design problem (E.29) becomes:

$$\hat{K} = \arg\min_{K} \max_{\substack{H \le H_{FLA} \\ i=1,2 \\ j=1\ldots N}} \left\{ \left\| \mathcal{F}_l(P_{i,j}(s,H), K) \right\|_\infty \right\} \tag{E.30}$$

In this formulation, N models are selected in the (mass $\times$ centering) domain, which typically correspond to the vertices. Problem (E.30) is then solved using the same H_∞ design tools as in Section E.1.1.1. A total of $2 \times 4 \times 2 = 16$ design models are used, corresponding to:

- 2 values of the landing gear height ($H = 5\,\text{m}$ and $H = 15\,\text{m}$),
- 4 models (indexed by j) corresponding to the minimum and maximum values of the mass and the center of gravity location,
- 2 separate optimization channels (indexed by i).

Numerical values of the reference model and weighting functions can be found in [Biannic and Roos, 2018]. A static controller composed of 7 gains is computed, and the resulting minimized H_∞ norm is around 1, which means that the requirements are satisfied.

Such a design can be followed by an analysis step in order to detect any potential worst-case configuration that could appear inside the operating domain. An iterative design process similar to the one described in Section E.1.1.3 can therefore be considered by incorporating the worst-case scenarios into the initial selection of design models.

2.4 Nonlinear implementation and simulation results

The structured flare control law resulting from the above optimization process is implemented in the nonlinear aircraft benchmark already used in Section E.1.2, as shown in Figure E.11. It is highlighted in green and activated as soon as the landing gear height H is lower than a threshold value $H_{FLA} \approx 15\,\text{m}$.

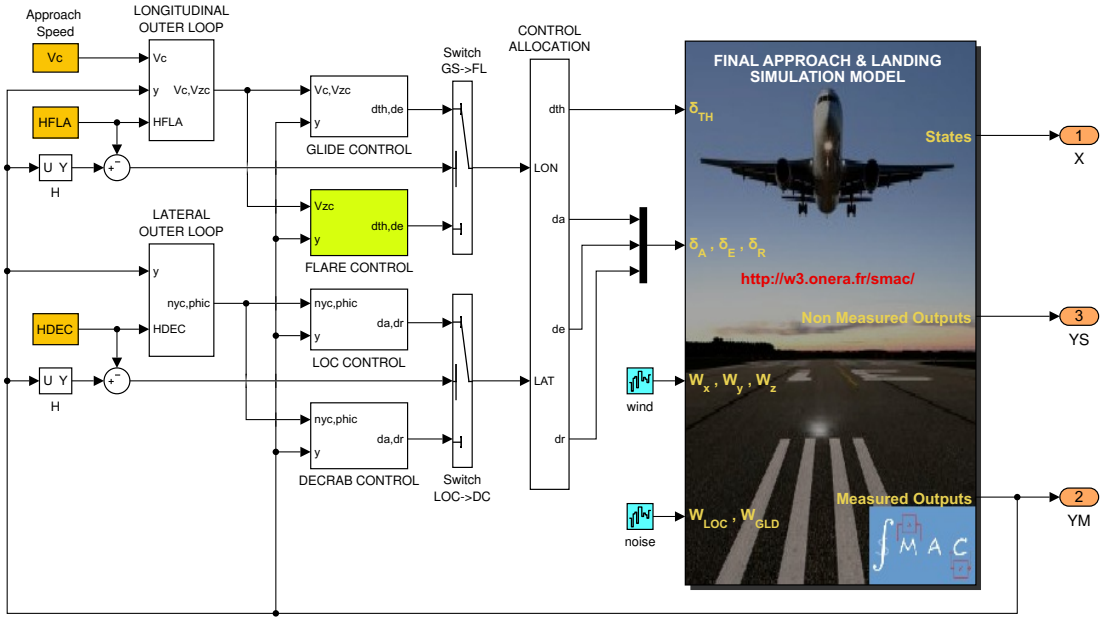


Figure E.11: Nonlinear closed-loop simulation model

A first simulation is displayed in Figure E.12 to evaluate the influence of external perturbations. A deterministic wind step occurs at 30 s during the approach phase. Strong turbulence as well as ILS noise are also introduced. The upper left plot shows the aircraft trajectory in the vertical plane during the whole maneuver, and it appears that the proposed guidance and control system behaves well. The impact point is slightly after the 400 m threshold and meets the requirements. A zoom on the flare trajectory reveals that the vertical speed V_z remains close to the reference one (upper right plot). Next, the robustness of the proposed controller is assessed for extreme values of the mass (120 tons and 180 tons) and the center of gravity location (15% and 40%).

Results are presented in Figure E.13. A reasonable dispersion of the impact point is observed (between 400 and 600 m), and the vertical speed is well tracked whatever the flight condition. More detailed simulation results can be found in [Biannic and Roos, 2018].

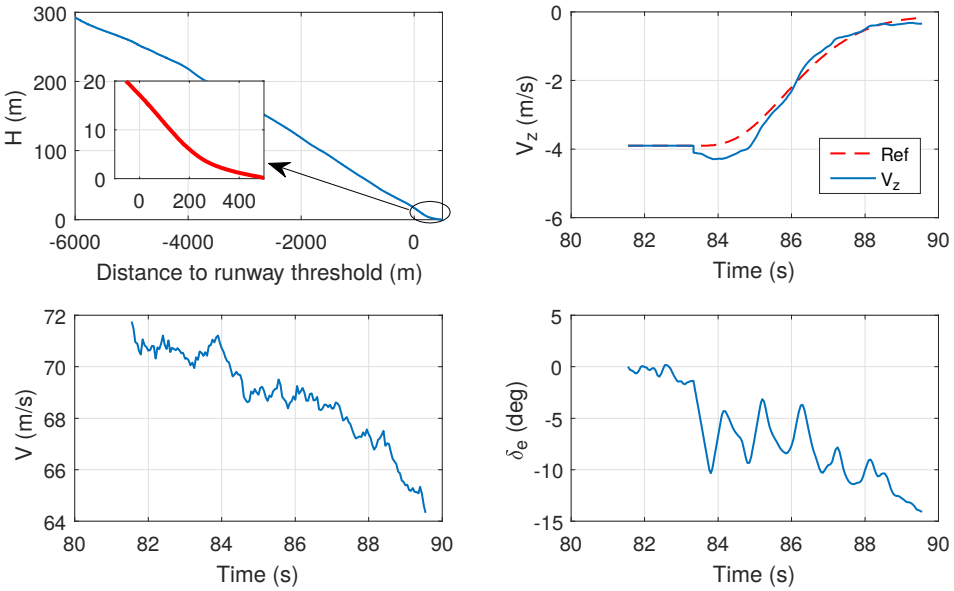


Figure E.12: Final approach and flare trajectory (upper left plot) and zoom on the flare phase (other plots) with deterministic wind step, strong turbulence and ILS noise

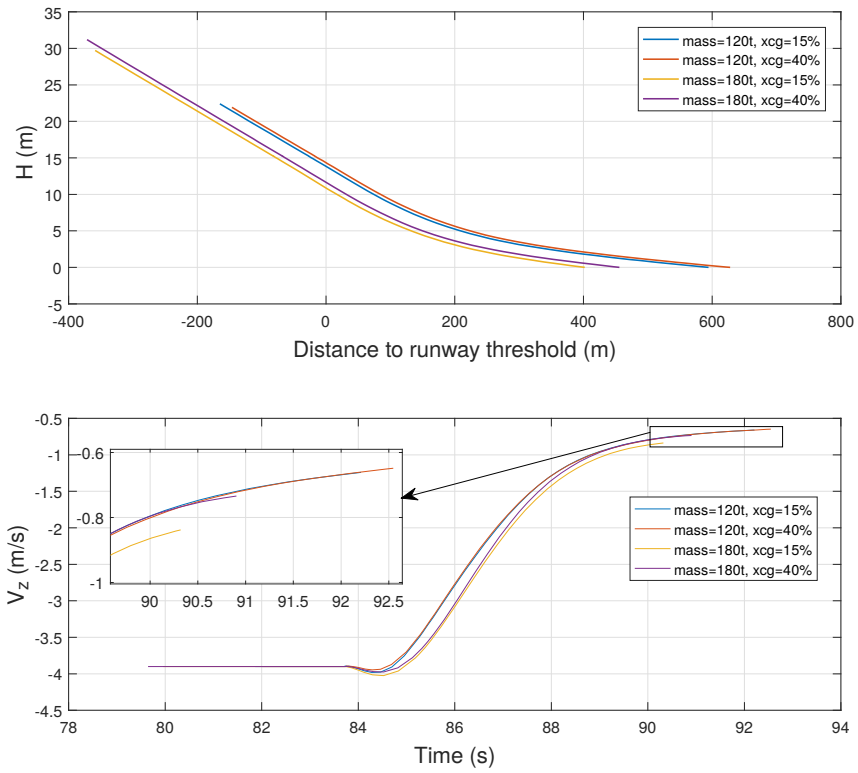


Figure E.13: Robustness to mass and center of gravity variations

2.5 Conclusion

A multi-model H_∞-based design approach is proposed to improve and shorten the tuning process of flare control systems. A fast and accurate tracking of the vertical speed is achieved despite measured (longitudinal speed variations) and external (wind, gust, ILS noise) perturbations. Parametric uncertainties and modeling errors are also taken into account thanks to the flexibility of the multi-model design tools, which are used here to improve robustness against inaccurate modeling of the ground effects and significant variations of the mass and the center of gravity location. The resulting flare control laws are combined with the glide and the outer loop controllers designed in Section E.1.2 to provide a complete control architecture that can be used during final approach and until touchdown.

3 Adaptive anti-windup design for the ground phase

Safety and economy are the primary concerns during the ground phase for commercial aircraft, the ultimate goals being the automation and the optimization of taxi operations. But for now, most aircraft are still controlled manually by the pilot before landing and after touchdown. One of the main reasons for this is that the on-ground motion is quite complex due to the coupling of aerodynamic effects and tire-road interactions. The latter are indeed highly nonlinear and depend on several parameters (runway characteristics and condition, tires characteristics, anti-skid braking system, longitudinal speed of the aircraft...). Advanced control techniques thus prove necessary to reach satisfactory performance levels. In this context, an adaptive anti-windup approach is presented in Section E.3.2 to control the aircraft at low speed with the nose-wheel steering system. It is based on an original approximation of the nonlinear ground forces by saturation-type nonlinearities, which is described in Section E.3.1. But one of the most challenging on-ground control problems occurs at intermediate speed between 40 and 80 kts. The main objective is to follow the runway centreline in the worst possible conditions, *i.e.* on a contaminated runway and in the presence of crosswind. Achieving good robustness and performance requires to combine all available actuators, and notably to use differential braking in conjunction with classical control devices, such as nose-wheel steering system and rudder. If differential braking should be used sparingly in dry weather to avoid excessive tire and brake wear, it could significantly reduce the lateral deviation on a wet or snowy runway. In this context, a dynamic control allocation scheme combined with a robust outer loop is briefly outlined in Section E.3.3 to address this issue.

3.1 Design oriented modeling of an on-ground aircraft

High-fidelity models with 6 or more degrees of freedom, such as the ones presented in [Rankin *et al.*, 2009; Jeanneau, 2007], are more convenient for analysis and validation than for control. Therefore, simplified design-oriented models have been developed. However, they generally focus only on the lateral behavior of the aircraft [Lemay *et al.*, 2011; Bihua *et al.*, 2013; Looye, 2007]. A few ones consider also the longitudinal behavior, but most of the time they do not discuss the coupling effects or do not provide a complete modeling of the tire-road interactions [Biannic *et al.*, 2006a; Turbuk and Paglione, 2010; Georgieva and Serbezov, 2017]. Moreover, the operational domain is often limited to low speed, and if higher speed is considered, only the rudder is usually taken into account in addition to the nose-wheel steering system [Bihua *et al.*, 2013; Biannic *et al.*, 2006a; Looye, 2007].

In order to bridge the gap between high-fidelity and oversimplified representations, a simple

but accurate design-oriented model with 3 degrees of freedom is proposed in [Sadien *et al.*, 2018]. It includes both the longitudinal and the lateral aerodynamic effects and tire-ground interactions, as well the dependence on runway conditions. The on-ground dynamics can be controlled by engines thrust, rudder deflection, nose wheel steering and (differential) braking, whose physical limitations are taken into account. It can be used up to 100 kts, which makes it suitable to address the above control issue. All equations are not described here, but a particular emphasis is placed on the slip force model, which is the key point that will be used in Sections E.3.2 and E.3.3.

A first simplified model was proposed in [Biannic *et al.*, 2006a]. The slip force F_{sy} is represented by a saturated linear function in the case of pure lateral slip:

$$F_{sy} = N_t \, \mathrm{sat}_{[\bar{\mu} F_{sy_{max}}]} \left(G_y \beta_s \right) \tag{E.31}$$

where the saturation level depends on the runway state $\bar{\mu}$ and the maximum lateral force $F_{sy_{max}}$ for a dry runway. N_t denotes the number of tires in the considered landing gear, β_s is the sideslip angle, and the cornering gain $G_y = \left. \frac{\partial F_{sy}}{\partial \beta_s} \right|_{\beta_s = 0}$ is computed for a mean value of the normal reaction F_z. No explicit dependence on F_z is considered, but an uncertainty is introduced on G_y, whose bounds are determined by extensive simulations of the high-fidelity model of [Jeanneau, 2007]. This is illustrated in Figure E.14, where the nominal model (E.31) is shown in green, while the bounds on the uncertain cornering gain are displayed in brown.

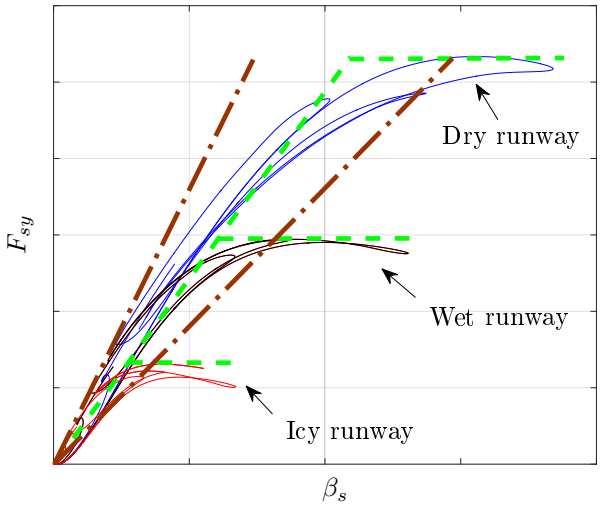


Figure E.14: Lateral slip model proposed in [Biannic *et al.*, 2006a]

Based on [Biannic *et al.*, 2006a], an improved model is proposed in [Sadien *et al.*, 2018], which accounts for combined slip, runway state and normal load variation. The combined longitudinal and lateral slip force $F_s = \sqrt{F_{sx}^2 + F_{sy}^2}$ generated by an isotropic tire, under non-zero sideslip and longitudinal slip, cannot exceed μF_z, where $\mu = \bar{\mu} \mu_{max}$ denotes the tire-road friction coefficient and μ_{max} its maximum value corresponding to a dry runway [Brach and Brach, 2011]. The maximum force boundary is illustrated by the pink circle in Figure E.15. The key idea to avoid this coupling between F_{sx} and F_{sy} is to assume that the sideslip angle β_s remains small during the roll-out phase, typically less than 5 deg in practice. Furthermore, the anti-skid system prevents the longitudinal slip ratio of becoming too large. In this context, it is reasonable to assume that the resultant of the slip forces lies inside the pink area in Figure E.15 instead of the pink circle, where the longitudinal and lateral maximum forces are represented in green and blue respectively.

The following decoupled model is then assumed:

$$F_{sx} = N_t \, \text{sat}_{\left[\bar{\mu}\lambda_{sx}\frac{F_z}{N_t}\right]} \left(K_x T_{brk}\right) \tag{E.32}$$

$$F_{sy} = N_t \, \text{sat}_{\left[\bar{\mu}\lambda_{sy}\frac{F_z}{N_t}\right]} \left(K_y \frac{F_z}{N_t}\beta_s\right) \tag{E.33}$$

T_{brk} is the braking torque. K_x and K_y correspond to the inverse of the effective rolling radius and to the reduced lateral cornering gain respectively, while λ_{sx} and λ_{sy} denote the longitudinal and the lateral friction fractions, defined for a dry runway and satisfying ${\lambda_{sx}}^2 + {\lambda_{sy}}^2 = {\mu_{max}}^2$. K_y does not depend on speed in the considered operational domain, but it is function of the runway state according to [Mitchell, 2006]:

$$K_y = \frac{K_{y_{max}}}{\frac{2}{3} + \frac{1}{3\bar{\mu}}} \tag{E.34}$$

where $K_{y_{max}}$ is the maximum value of K_y, obtained for a dry runway.

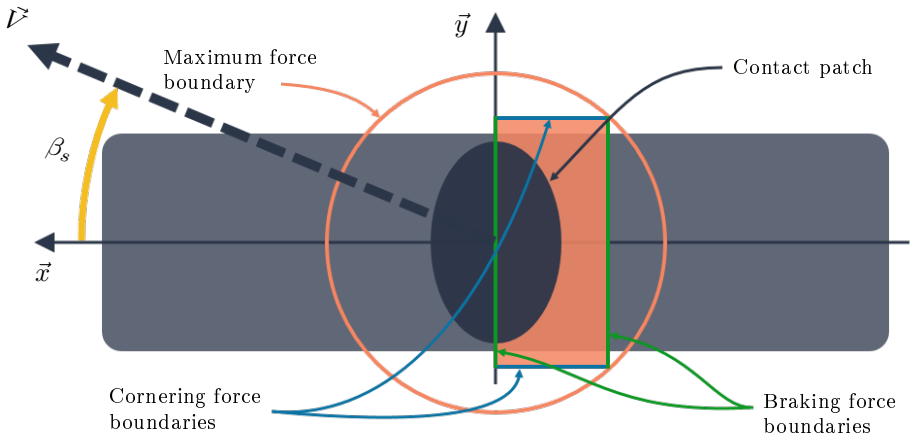


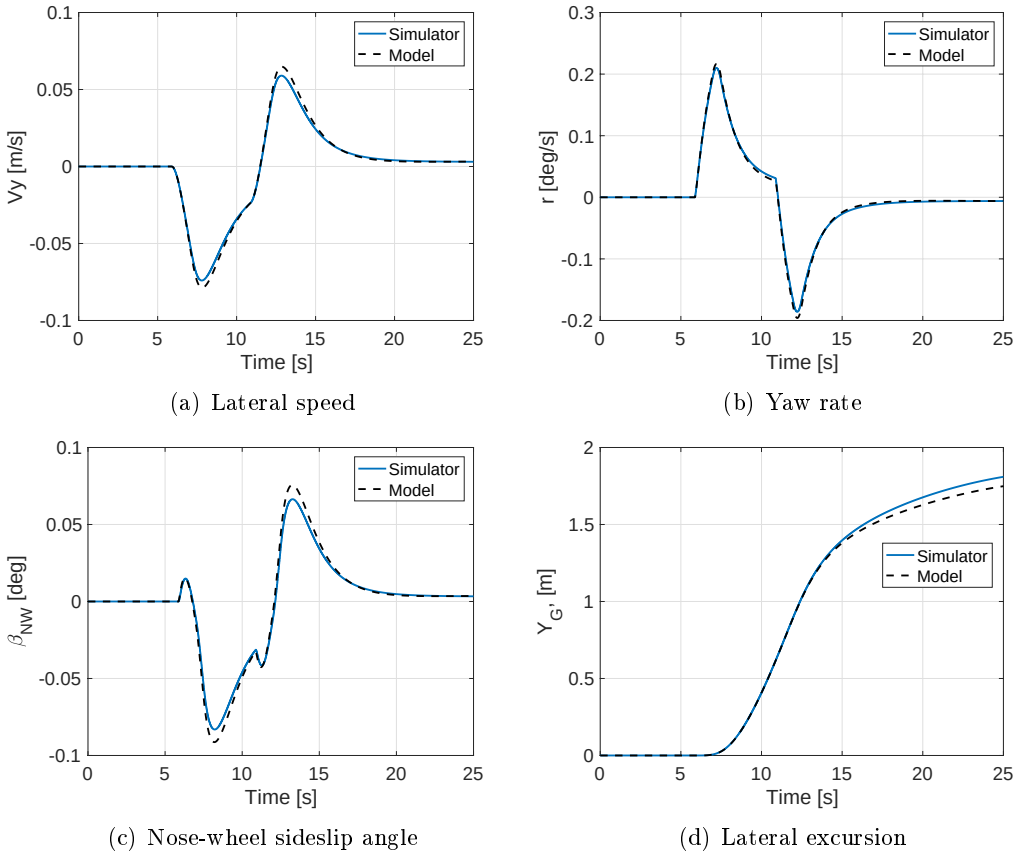
Figure E.15: Main landing gear wheel seen from above

The whole on-ground aircraft model is described in [Sadien *et al.*, 2018] and compared to a high-fidelity Airbus simulator [Goupil *et al.*, 2014]. One of the considered maneuvers is represented in Figure E.16. It corresponds to a 3 deg deflection of the brake pedal to the right and then to the left on a wet runway at medium speed. The lateral behavior of the proposed model is close to that of the simulator, and similar results are obtained for all maneuvers. Hence, it is well-suited for the development of ground control laws.

3.2 Adaptive anti-windup for low speed maneuvers

The control objectives depend on the aircraft velocity. Let us first focus on low speed (below 40 kts), where the main concern is maneuverability. The challenge is to design a control architecture, which ensures a good tracking of the yaw rate r and the heading Ψ:

- with as fast a response as possible,
- without overshoot (especially in heading),
- whatever the runway state (dry, wet or icy),
- for any aircraft longitudinal speed.



(a) Lateral speed

(b) Yaw rate

(c) Nose-wheel sideslip angle

(d) Lateral excursion

Figure E.16: Model validation - differential braking order

The only control input is the nose-wheel deflection θ_{NW}. Typical maneuvers are runway exit at the right speed while minimizing the occupation time, or autonomous motion along the taxiways towards the selected gate. A methodology based on the on-ground aircraft model presented in Section E.3.1 is described in [Roos *et al.*, 2010]. The main steps are briefly outlined below.

The huge variation of the lateral slip forces with respect to the runway state (see Figure E.14) suggests that standard robust control methods would yield conservative results and reduced performance. On the opposite, the proposed approach is to perform an on-line estimation of these forces. This information can then be used to maintain the sideslip angle β_s just below the optimal value for which F_{sy} is the highest, so that the aircraft turns quickly without for all that slipping on the runway. In this perspective, it is necessary to capture the nonlinear behavior of the slip forces. Very accurate models are for example used in [Falcone *et al.*, 2007], but the resulting model predictive control approach is computationally demanding and cannot be reasonably implemented in flight computers. Therefore, a simpler model seems preferable, such as the one proposed in Section E.3.1. And the associated saturations strongly suggest to design an anti-windup controller using the techniques described in Chapter D, which becomes active only when the sideslip angle reaches a certain limit. With this strategy, there is no need to modify the nominal control laws, which is a strong industrial requirement.

The proposed ground control architecture is represented in Figure E.17. A nominal controller depending on the aircraft longitudinal velocity V_x is first designed, so as to ensure good stability and performance in the linear operating domain. It is composed of a first-order inner loop controller $K_r(s)$ for yaw rate tracking and a first-order outer loop controller $K_\Psi(s)$ for heading

tracking. The commanded yaw rate r_c is followed without steady-state error, but an important overshoot occurs when the saturations involved in the slip forces model are active, as shown by the dotted line in Figure E.19(a). Therefore, a parameter-varying static anti-windup controller J_r function of V_x is designed to counter the adverse effects of these nonlinearities. The overshoot becomes negligible, which is confirmed by the solid line in Figure E.19(a).

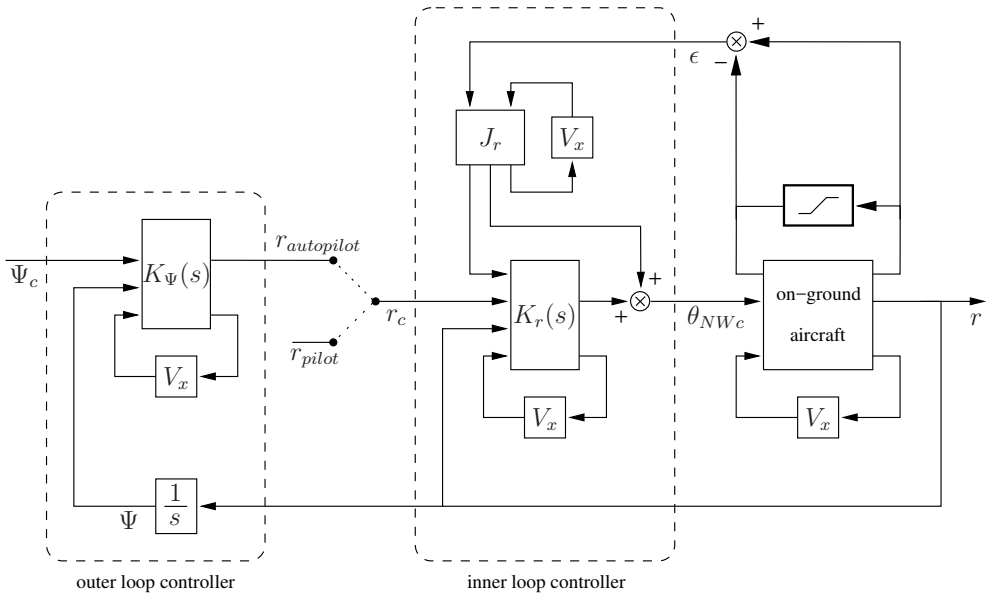


Figure E.17: Ground control architecture

The input ϵ of the anti-windup controller J_r is computed as the difference between the unsaturated and the saturated lateral slip forces. However, it is not available on a real aircraft and has to be estimated on-line. This is a difficult issue, since the saturation levels depend on the runway state (see equation (E.31)), which is unknown and can change rapidly. An estimation procedure is implemented, based on the dynamic inversion of the lateral aircraft model. Estimates $\hat{F}_{sy}$ of the lateral slip forces are obtained, and the signals:

$$\left\{ \begin{array}{rcl} \epsilon_{NW} & = & \text{sign}\left(\beta_{s_{NW}}\right) |\hat{F}_{sy_{NW}} - G_{y_{NW}} \beta_{s_{NW}}| \\ \epsilon_{MG} & = & \text{sign}\left(\beta_{s_{MG}}\right) |\hat{F}_{sy_{MG}} - G_{y_{MG}} \beta_{s_{MG}}| \end{array} \right. \tag{E.35}$$

can be used by the anti-windup device, where the subscripts NW and MG stand for the nose-wheel and the main landing gears respectively (see [Roos *et al.*, 2010] for details).

The resulting adaptive controller is composed of the parameter-varying nominal and anti-windup compensators as well as the estimator. It is implemented on the nonlinear high-fidelity model of [Jeanneau, 2007] for validation, as shown in Figure E.18, where $x_{NL} = [r_{NL}\ V_{yNL}]^T$. Selected simulation results are presented in Figure E.19(a) for the inner loop and E.19(b) for the outer loop. It can be seen that the control objectives are satisfied. In particular, the heading is well tracked whatever the runway state, which demonstrates the ability of the on-line estimator to identity the slip forces.

Finally, an extensive validation is performed in [Roos and Biannic, 2007]. Stability and performance are first evaluated using the μ-analysis based tools presented in Chapter A. The analysis techniques of Chapter B are then applied to take into account time-invariant uncertainties (aerodynamic coefficients) as well as time-varying uncertainties (cornering gains) and parameters (aircraft longitudinal velocity).

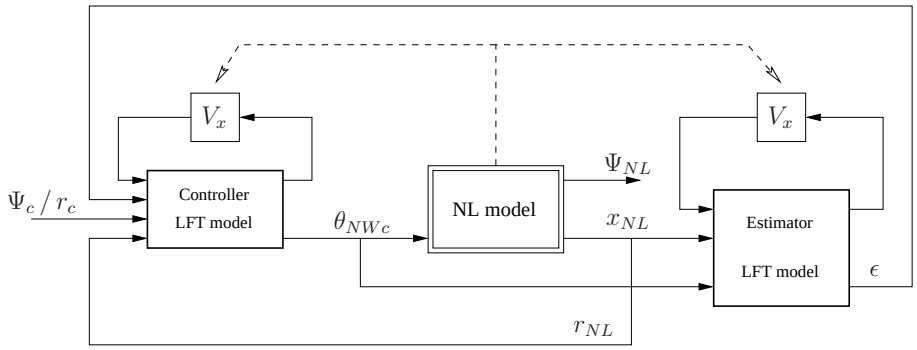
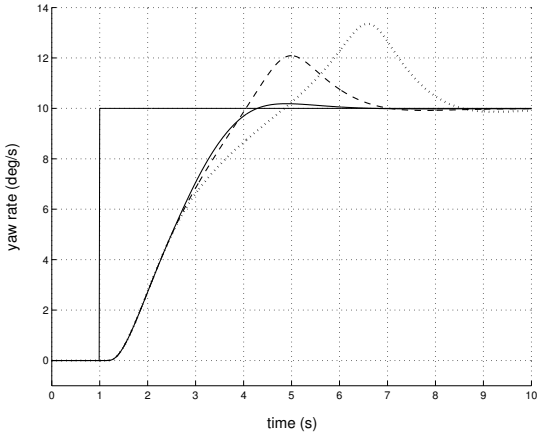
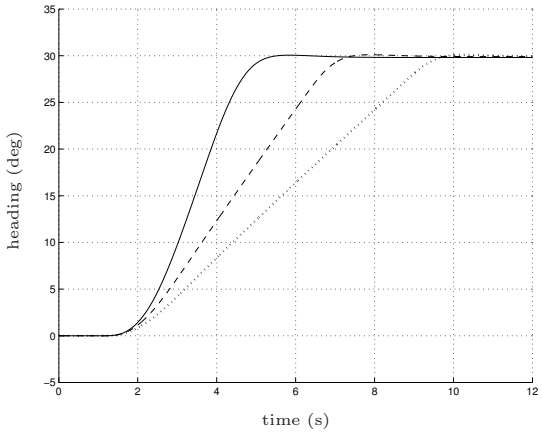


Figure E.18: Controller implementation



(a) Yaw rate responses with no (dotted line), partial (dashed line) and full (solid line) anti-windup



(b) Heading responses for dry (solid line), wet (dashed line) and icy (dotted line) runways

Figure E.19: Simulation results on the high-fidelity nonlinear model

3.3 Robust dynamic allocation for runway axis hold

The methodology presented in Section E.3.2 is limited to low speed maneuvers, where a single actuator is sufficient to control the lateral behavior of the aircraft. At higher speed, the main objective is to follow the runway centreline whatever the runway condition and despite strong crosswind. All actuators must be combined to meet safety, performance and comfort requirements, which raises a control allocation problem (see *e.g.* [Johansen and Fossen, 2013; Zaccarian, 2009; Härkegård, 2004] for a review of dynamic allocation techniques). However, differential braking is systematically ignored in the on-ground control literature, and even the rudder is rarely mentioned. One of the very few works facing the above control allocation problem is [Looye, 2007], which presents a simple solution depending on the aircraft velocity V_x: only the nose wheel or the rudder is used if $V_x < V_{low}$ or $V_x > V_{high}$, and in between a linear blend is applied. This kind of approach is typically implemented by aircraft manufacturers due to its simplicity, but it is clearly suboptimal. Work is underway to improve these results. A two-step strategy is proposed in [Cassaro *et al.*, 2018], based on the on-ground aircraft model developed in [Sadien *et al.*, 2018] and briefly reviewed in Section E.3.1. A robust outer loop controller designed using structured H_∞ tools allows to cope with severe crosswind and to limit the aircraft lateral deviation, even on a contaminated runway. It produces a single output, which corresponds to a differential braking command. Among the three available control actions,

differential braking is indeed the only one to be effective across the entire speed range. A dynamic control allocation scheme is then implemented based on the method of [Zaccarian, 2009], so as to account for abrupt changes in the actuators effectiveness due to the fast variations of velocity and loads during deceleration. Promising simulations are presented in [Cassaro *et al.*, 2018], although further validation is required.

4 Control laws implementation

A wide variety of military and civilian tasks can be assigned to fixed-wing UAV: search and rescue, exploration and detection, surveillance, imaging... Many of them take place in hazardous and remote environments, and their success often relies on efficient and robust control systems. However, the design of such systems is challenging because many specifications have to be taken into account with a limited computing power.

The objective in this context is twofold. First, apply advanced control techniques such as those presented in the previous sections to improve performance and increase the operating domain of fixed-wing UAV. Then, prove that the resulting control architectures are simple enough to be implemented on low-performance avionics. Work is still in progress, but the ambition is to carry out flight tests on some of the UAV operated by ONERA, which are shown in Figure E.20. Both of them are quite large (approximately 4 m wingspan) and can carry a substantial payload (up to 20 kg for the K50). This makes it possible to integrate a whole set of sensors and all necessary devices to monitor the flight.

(a) Avion Jaune

(b) K50

Figure E.20: Fixed-wing UAV operated by ONERA

Fairly accurate models of these UAV have already been obtained, as reported in [Lesprier *et al.*, 2015a]. A few key steps of the process are illustrated in Figure E.21:

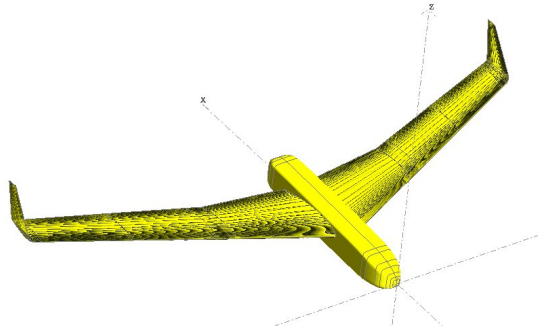
(a)-(b) 3D modeling of the aircraft to compute the stability derivatives using software such as XFLR5, AVL or Tornado, but also the inertia matrix using CATIA.

(c) Identification of the dynamics as well as the rate and magnitude saturations of the actuators thanks to ground tests.

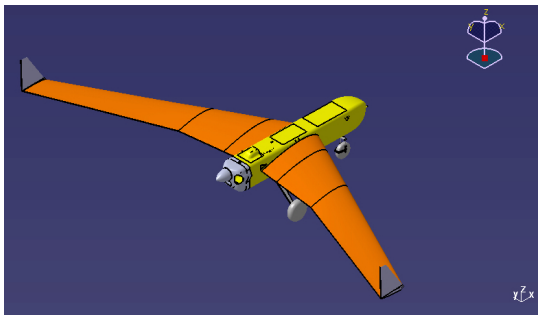
(d) Application of the APRICOT Library of the SMAC Toolbox to flight test data, so as to express the engine speed as a polynomial function of the thrust command and the airspeed.

(e)-(f) Development of a thrust model from propeller geometrical data and static thrust measurement on ground using dedicated software such as QPROP.

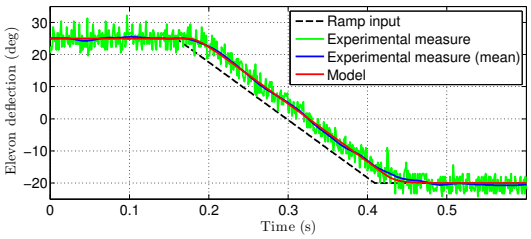
(g)-(h) First in-flight validation of the open-loop aircraft model by performing simple longitudinal and lateral maneuvers.



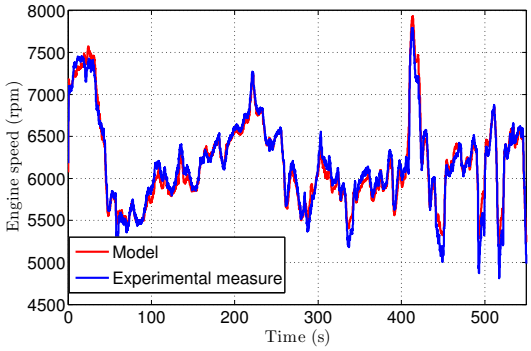
(a) 3D modeling for stability derivatives estimation



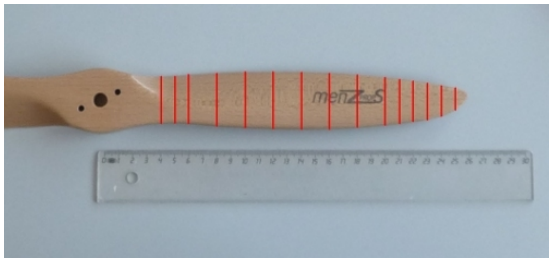
(b) 3D modeling for inertia matrix computation



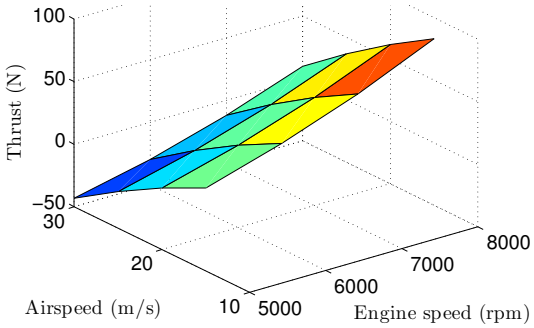
(c) Actuators models identification



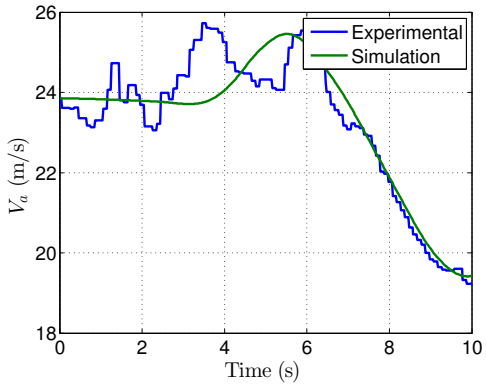
(d) Engine model identification



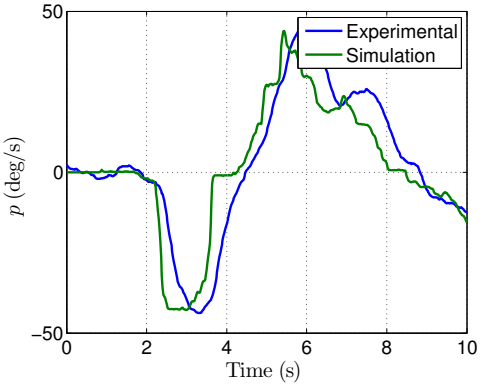
(e) Propeller cutting to get precise geometrical data



(f) Thrust modeling



(g) In-flight validation, longitudinal behavior



(h) In-flight validation, lateral behavior

Figure E.21: Different modeling steps of a fixed-wing UAV

Triming and linearization routines have then been developed, and a whole control architecture has finally been designed using the robust nonlinear compensation technique of Section E.1. The next steps are to improve the open-loop model thanks to enhanced identification techniques [Bucharles *et al.*, 2012], to fine-tune the control laws, and to implement them in the avionics, before final closed-loop validation can be performed in flight.

Remark E.6 *The microprocessor used on the Avion Jaune is an ARM Cortex-M4*[1]*, which is composed of two cores with a frequency of 80 MHz. There is a built-in Float Processing Unit, so that floats operations do not need additional libraries. The tasks are currently run at a time period of 20 ms, within which at least 10 ms are available for control (the other tasks being e.g. sensors acquisition and data fusion). This is more than enough to perform the operations required by the robust nonlinear compensation technique of Section E.1:* 2×2 *nonsingular matrix inversion for the linearizing inner loop, third-order controller computation for the robust outer loop. Finally, the inevitable discretization process prior to the control laws implementation should not be a problem. Simulations were successfully carried out with a time step of up to 50 ms, which is much larger that the 20 ms mentioned above.*

5 Summary of the contributions

The main contributions presented in this chapter are briefly summarized below, and a selection of related publications is given:

- **Section E.1**: robust nonlinear compensation technique based on a partial feedback linearization of the plant combined with a fixed-structure multi-model H_∞ design technique ; application to the *approach phase* of a commercial aircraft [Biannic *et al.*, 2017; Lesprier *et al.*, 2014].
- **Section E.2**: robust multi-model and multi-channel H_∞ design for the *flare phase* [Biannic and Roos, 2018; 2015].
- **Section E.3**: control oriented modeling of an aircraft during the *ground phase*, including an original saturation-based representation of the slip forces ; adaptive anti-windup design technique for low speed maneuvers ; robust dynamic allocation method for runway axis hold at high speed despite contaminated runway and crosswind [Sadien *et al.*, 2018; Biannic *et al.*, 2006a; Roos *et al.*, 2010; Roos and Biannic, 2007; Cassaro *et al.*, 2018].
- **Section E.4**: practical approach to fixed-wing UAV modeling and preliminary control laws implementation in a real avionics system developed by ONERA [Lesprier *et al.*, 2015a].

[1] `https://www.arm.com/products/processors/cortex-m4-processor.php`

Research prospects

The final chapter of this manuscript briefly summarizes my research perspectives, which are organized along three lines:

- Development of enhanced robustness analysis tools,
- Computation of simple yet accurate linear fractional representations,
- Design and implementation of advanced control architectures.

Each of these lines is described in more detail below.

Development of enhanced robustness analysis tools

As emphasized in this manuscript, a lot of work has already been done during the last three decades in the field of μ-analysis, and this technique is now well recognized as mature and efficient by the control community. In most cases, stability margins and performance levels can be computed with both a very good accuracy and a reasonably low computational time, as shown in Chapter A. But we can always go further! Here are a few ideas:

- In the same way as for the μ lower bound, perform an extensive comparison of all practical μ upper bound algorithms, and propose some combinations to take the most out of them,
- Improve the branch-and-bound algorithms when they do not converge in a reasonable number of iterations, for example by better dividing the uncertainty domain (use other criteria than the highest μ-sensitivity, create subdomains of different sizes...),
- Focus on numerical issues (for example, when reducing ϵ in Algorithm A.1 to improve precision, the size of the interval I_v sometimes becomes smaller and smaller around certain frequencies, and many iterations are needed to validate the whole frequency range, which significantly slows down the algorithm),
- Compare several optimization tools (notably LMI solvers)...

The objective is to continually improve the SMART Library until all real-world applications can be solved.

But despite the good results achieved on many benchmarks, μ-analysis based tools are only marginally used in industry. This is mostly because they are developed within a deterministic framework. In this context, the robustness margins can be too conservative, since the associated worst-case configurations sometimes correspond to non-physical operating points or to situations that are unlikely to occur in practice. As an example, if some flight control laws fail to stabilize an aircraft for a parametric configuration whose probability of occurrence is below 10^{-9} per flight hour, the proposed control system might be wrongly rejected. This is in stark contrast with industrial practices, where Monte Carlo methods are commonly used by control engineers to

assess stability and performance of many kinds of aerospace systems in a probabilistic framework. These methods are easy to implement and do not pose any specific constraint on the representation of the system to be analyzed. But on the other hand, their high computational complexity makes them time-consuming and costly. In this context, there is a real need to bridge the gap between Monte Carlo simulations and deterministic optimisation-based methods, by developing new validation tools which combine the advantages of both approaches. The main objective is to provide fast and computationally efficient tools able to capture the probability of occurrence of worst-case events, thus speeding up the control laws validation process. To do so, the idea is to confine the system uncertainties in a set as in the deterministic case, but also to consider them as random variables with given multivariate probability distributions. Probabilistic μ-analysis could be investigated first, since it is a fairly direct extension of classical μ-analysis, and the algorithm introduced in Section IV of [Roos and Biannic, 2010] would be a good starting point. A more prospective study could then be conducted to evaluate the applicability of other probabilistic and randomized techniques (see *e.g.* [Calafiore *et al.*, 2011]) to high-order systems with many uncertainties. Beyond control laws validation, these techniques could also be integrated in the design process and used to optimize the controller gains, a context to which classical Monte Carlo methods cannot be directly applied. The ambition is to shorten the development time by avoiding a tedious iterative process between design and validation, and to better handle the trade-off between robustness margins and performance levels. We have proposed a PhD position on this topic in collaboration with Politecnico di Milano.

We have spent a considerable amount of time working on μ-analysis, which is essentially based on the H_∞ norm. This can be a little restrictive. Consider for example the clearance of flight control laws process mentionned in the introduction. Several criteria must be evaluated, among which some of them are formulated in terms of the H_2 norm. This is typically the case of comfort and loads requirements [Papageorgiou *et al.*, 2012]. An upper bound on the worst-case H_2 performance level was proposed in [Paganini and Feron, 2000] and could be useful in this context. But its practical computation is a challenging issue with an infinite number of frequency-domain constraints, each of them being an LMI involving scaling matrices with a large number of decision variables in case of highly repeated uncertainties. Fortunately, it should be possible to adapt some of the methods presented in Chapter A to cope with the H_2 framework. For example, Algorithm A.1 was already adapted in [Garulli *et al.*, 2013] to avoid solving the problem on a finite frequency grid, and thus to get a guaranteed upper bound on the worst-case performance level. But it is still restricted to small and medium size problems due to the limited capabilities of LMI solvers. An idea to get a more tractable algorithm could be to use a gradient descent technique as in the H_∞ case to obtain a solution that would probably be slightly suboptimal, but with a much shorter computational time. Moreover, in the spirit of [Ferreres and Biannic, 2001; Roos, 2010], it should be investigated whether an accurate lower bound on the worst-case H_2 performance level could be obtained.

Several methods have been proposed in Chapter B to deal with both LTI and time-varying uncertainties. But when bounded rates of variation have to be considered (which is the most common in practice), existing techniques are usually based on LMI characterizations involving the optimization of square matrices with $n(n+1)/2$ decision variables, where n is the order of the considered system. This is typically the case when IQC-based analysis is coupled with the KYP lemma, or when parameter-dependent Lyapunov functions are used as in Section B.2. In case of high-order systems, the computational time can be prohibitive and a frequency-domain approach is a valuable alternative. In this context, generalizing the approach of [Ferreres and Roos, 2007] presented in Section B.2 could be relevant.

Finally, a more complicated problem is to consider both structured uncertainties and saturations when assessing the robustness properties of a feedback system. IQC-based analysis is a natural way to address this issue, see *e.g.* [Ge *et al.*, 2012; Ferreres and Biannic, 2007]. But saturations are usually treated as classical sector-bounded nonlinearities, which can be conservative. And the computational time can become prohibitive in case of highly repreated parametric uncertainties. A better solution could be to characterize the saturations using the modified sector condition introduced in [Gomes da Silva Jr and Tarbouriech, 2005] instead of the classical one to reduce conservatism, and to handle the uncertainties in the μ framework to master computational time. But combining both approaches is not obvious...

Computation of simple yet accurate LFR

Before an LFR can be generated from tabulated data, it is important to ensure that the latter are consistent and can be approximated. But experience shows that there is currently no method that works all the time, especially when a set of MIMO LTI systems must be turned into an LFR. A first solution is to find polynomial or rational approximations of the state-space matrices, but it is difficult to guarantee that the state vector is consistent across all models. So the eigenvalues and the frequency responses of the resulting LFR may not evolve smoothly on the whole parametric domain, leading to some unrealistic behavior. Strategies to constrain them in prescribed intervals could be investigated. Another solution is to generate an LFR, which only preserves the input-output behavior, without considering the eigenvalues. This can be achieved by minimizing an H_∞ or an H_2 criterion, see *e.g.* [Petersson and Lofberg, 2009] for the H_2 case. But computational time often becomes an issue when the size of the problem increases. Keeping in mind that our objective is to provide robustness analysis tools, which are much faster than classical simulation-based techniques, more research has to be done to fasten the LFR generation step. Indeed, it becomes useless to have analysis tools which run in a few seconds if it takes hours or days to get the LFR! Recently, non-smooth optimization techniques have been developed and could help minimizing the H_∞ norm at a reasonable cost ([Apkarian *et al.*, 2009]).

As already highlighted in Chapter B, a particular attention must also be paid to the size of the LFR when applying robustness analysis tools, so as to avoid conservatism, or even numerical intractability. A first strategy is to create a high-fidelity LFR without trying to master complexity, so as to closely match the behavior of the considered system. But checking whether it is minimal of not (*i.e.* whether the size of Δ is as small as possible) and computing the associated minimal representation is a very difficult open problem. An alternative to reduce complexity is to apply reduction techniques such as the ones proposed in [D'Andrea and Khatri, 1997; Ferreres, 2004]. Recently, parametric model reduction has gained a lot of interest and several techniques dedicated to parameterized dynamical systems have been proposed [Benner *et al.*, 2015]. It would be interesting to investigate the relevance of these algorithms when applied to the LFR framework to see whether better results could be obtained.

A second strategy is to act upstream and to directly generate a simple yet accurate LFR. This question arises primarily when trying to obtain an LFR from tabulated data. Most of the time, the user simply adjusts by trial and error the maximum degrees of the polynomial and rational approximations, which can lead to solutions that are far from being optimal. Attempts have been made to automate this procedure (see *e.g.* [Roos *et al.*, 2014; Pfifer and Hecker, 2011]) but much remains to be done. What are the best size and structure of Δ (*i.e.* the number de repetitions of each parameter and uncertainty)? Is it relevant to add delays, nonlinearities...? What is the best way to take into account the approximation errors? These are all questions that

remain largely unanswered for the moment. And what is the best way to choose the samples used for approximation, *i.e.* which parametric configurations of the system have to be considered? A solution based on the ν-gap metric is proposed in [Vizer and Mercère, 2014], but it seems tedious to implement when several parameters are considered.

To sum up, plenty of interesting contributions exist in the field of LFR generation, but there is still much to be done to combine the most relevant ones, so as to obtain both reliable and generic methodologies.

Design and implementation of advanced control architectures

Many real-world applications can be cited, where strong cost constraints or weight limitations dictate the use of potentially low performance hardware (such as actuators, sensors and onboard computers). For example, the access to space is becoming significantly easier and cheaper with the advent of nanosats. To promote this new economic model, low-cost launchers must be developed. In the military domain, the need for guided projectiles has continuously increased during the last years with several such systems in development or in operational use. One of the most topical issues is to improve the precision and to enhance the range of traditional artillery shells in the modern battlefield, while significantly reducing costs. And UAV are now widely used to perform a large number of demanding missions. So there is a real need to control various kinds of systems on extended operational domains in the presence of hardware limitations, uncertainties, varying parameters as well as actuator saturations.

Many efficient analysis tools are now available, but in most cases they are applied only once the control laws have been computed. However, it could be beneficial to better integrate them into the design step, so as to anticipate the tedious validation inherent to any industrial process. The methodology presented in Section E.1 is a first step in that direction. It uses μ-analysis to compute worst-case configurations, which are used to enrich the bank of design models and to iteratively improve the robustness properties of the closed-loop system. When applied to the approach phase of an aircraft, it could be extended by using evolutionary algorithms to optimize some high-level parameters such as the altitudes at which the flare and the decrab are initiated, in order to maximize the probability of safe landing. When applied to the aforementioned projectiles, it could be coupled with an anti-windup device to cope with actuator saturations or with aerodynamic nonlinearities at high angles, so as to increase the operational domain. We have proposed a PhD project in collaboration with the French German Institute of Saint Louis (ISL) to go one step further in the development of novel control architectures able to cope with demanding specifications such as those mentioned above, and to develop generic computational tools that could be passed to control engineers.

But it must be paid attention not to make the laws too complicated. And for this, there is nothing like facing implementation constraints. Thanks to our growing involvement in ONERA's UAV lab, the proposed control architectures will be implemented on a 4m wide aircraft and validated first by hardware-in-the-loop simulations and then by flight tests. The feedback will allow us to take a fresh look at our design methodologies and to further improve them. We are currently setting up an ambitious research project on this topic.

References

[Ackermann, 1985] J. Ackermann. Multi-model approaches to robust control system design. In *Uncertainty and Control*, volume 70 of *Lecture Notes in Control and Information Sciences*, pages 108–130. Springer Berlin Heidelberg, 1985.

[Adams and Banda, 1993] R. Adams and S. Banda. Robust flight control design using dynamic inversion and structured singular value synthesis. *IEEE Transactions on Control Systems Technology*, 1(2):80–92, 1993.

[Airbus, Technical Report 2017] Airbus. A statistical analysis of commercial aviation accidents 1958-2016, Technical Report, 2017.

[Akcay and Ninness, 1999] H. Akcay and B. Ninness. Orthonormal basis functions for continuous-time systems and L_p convergence. *Mathematics of Control, Signals and Systems*, 12(3):295–305, 1999.

[Amato *et al.*, 1997] F. Amato, M. Corless, M. Mattei, and R. Setola. A multivariable stability margin in the presence of time-varying bounded rate gains. *International Journal of Robust and Nonlinear Control*, 7(2):127–143, 1997.

[Apkarian and Noll, 2006] P. Apkarian and D. Noll. Nonsmooth H_∞ synthesis. *IEEE Transactions on Automatic Control*, 51(1):71–86, 2006.

[Apkarian and Noll, 2017] P. Apkarian and D. Noll. Worst-case stability and performance with mixed parametric and dynamic uncertainties. *International Journal of Robust and Nonlinear Control*, 27:1284–1301, 2017.

[Apkarian *et al.*, 2009] P. Apkarian, D. Noll, and O. Prot. A proximity control algorithm to minimize nonsmooth and nonconvex semi-infinite maximum eigenvalue functions. *Journal of Convex Analysis*, 16:641–666, 2009.

[Apkarian *et al.*, 2014] P. Apkarian, P. Gahinet, and P. Buhr. Multi-model, multi-objective tuning of fixed-structure controllers. In *Proceedings of the 13th European Control Conference*, pages 856–861, Strasbourg, France, 2014.

[Apkarian *et al.*, 2016] P. Apkarian, D. Noll, and L. Ravanbod. Nonsmooth bundle trust-region algorithm with applications to robust stability. *Set-Valued and Variational Analysis*, 24(1):115–148, 2016.

[Apkarian, 2011] P. Apkarian. Nonsmooth μ synthesis. *International Journal of Robust and Nonlinear Control*, 21(8):1493–1508, 2011.

[Åström and Rundqwist, 1989] K. Åström and L. Rundqwist. Integrator windup and how to avoid it. In *Proceedings of the American Control Conference*, pages 1693–1698, Pittsburgh, Pennsylvania, 1989.

[Baldelli *et al.*, 2005] D. Baldelli, R. Lind, and M. Brenner. Robust aeroelastic match-point solutions using describing function method. *Journal of Aircraft*, 42(6):1597–1605, 2005.

[Balemi *et al.*, 1991] S. Balemi, S. Boyd, and V. Balakrishnan. Computation of the maximum H_∞-norm of parameter-dependent linear systems by a branch and bound algorithm. In *Proceedings of the International Symposium on Mathematical Theory of Networks and Systems*, pages 305–310, Kobe, Japan, 1991.

[Barmish and Kang, 1993] B. Barmish and H. Kang. A survey of extreme point results for robustness of control systems. *Automatica*, 29(1):13–35, 1993.

[Bartlett *et al.*, 1988] A. Bartlett, C. Hollot, and H. Lin. Root locations on an entire polytope of polynomials: it suffices to check the edges. *Mathematics of Control, Signals and Systems*, 1(1):61–71, 1988.

[Bates and Mannchen, 2004] D. Bates and T. Mannchen. Improved computation of mixed μ bounds for flight control law robustness analysis. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 218(8):609–619, 2004.

[Bennani *et al.*, 2005] S. Bennani, B. Beuker, J. van Staveren, and J. Meijer. Flutter analysis for the F-16A/B in heavy store configuration. *Journal of Aircraft*, 42(6):1566–1575, 2005.

[Benner *et al.*, 2015] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531, 2015.

[Bernstein and Michel, 1995] D. Bernstein and A. Michel. A chronological bibliography on saturating actuators. *International Journal of Robust and Nonlinear Control*, 5(5):375–380, 1995.

[Beugnon *et al.*, 2003] C. Beugnon, B. Girouart, B. Frapard, and K. Lagadec. Mixed μ-analysis applied to the attitude control of the MetOp spacecraft. In *Proceedings of the European Control Conference*, pages 2643–2649, Cambridge, United Kingdom, 2003.

[Biannic and Apkarian, 2001] J-M. Biannic and P. Apkarian. A new approach to fixed-order H_∞ synthesis : Application to autoland design. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Montreal, Canada, 2001.

[Biannic and Apkarian, 2011] J-M. Biannic and P. Apkarian. Anti-windup design via nonsmooth multi-objective H_∞ optimization. In *Proceedings of the American Control Conference*, pages 4457–4462, San Francisco, California, 2011.

[Biannic and Ferreres, 2005] J-M. Biannic and G. Ferreres. Efficient computation of a guaranteed robustness margin. In *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, 2005.

[Biannic and Roos, 2008] J-M. Biannic and C. Roos. Introduction to AWAST: the Anti-Windup Analysis and Synthesis Toolbox. In *Proceedings of the 9th IEEE International Symposium on Computer-Aided Control System Design*, pages 589–594, San Antonio, Texas, 2008.

[Biannic and Roos, 2012] J-M. Biannic and C. Roos. Chapter 7: Enhanced μ-analysis techniques for clearance. In *Optimization based clearance of flight control laws*, pages 131–147. Lecture Notes in Control and Information Sciences, vol. 416, Springer, 2012.

[Biannic and Roos, 2015] J-M. Biannic and C. Roos. Flare control law design via multi-channel H_∞ synthesis: illustration on a freely available nonlinear aircraft benchmark. In *Proceedings of the American Control Conference*, pages 1303–1308, Chicago, Illinois, 2015.

[Biannic and Roos, 2016] J-M. Biannic and C. Roos. Generalized State Space: a new Matlab class to model uncertain and nonlinear systems as Linear Fractional Representations. Available with the SMAC toolbox at http://w3.onera.fr/smac/gss, 2016.

[Biannic and Roos, 2017] J-M. Biannic and C. Roos. Introduction to anti-windup design with the SAW library. Available with the SMAC toolbox at http://w3.onera.fr/smac/saw, 2017.

[Biannic and Roos, 2018] J-M. Biannic and C. Roos. Robust autoland design by multi-model H_∞ synthesis with a focus on the flare phase. *Aerospace*, 5(1), 2018. doi:10.3390/aerospace5010018.

[Biannic *et al.*, 2006a] J-M. Biannic, A. Marcos, M. Jeanneau, and C. Roos. Nonlinear simplified LFT modelling of an aircraft on ground. In *Proceedings of the IEEE Conference on Control Applications*, pages 2213–2218, Munich, Germany, 2006.

[Biannic *et al.*, 2006b] J-M. Biannic, S. Tarbouriech, and D. Farret. A practical approach to performance analysis of saturated systems with application to fighter aircraft flight controllers. In *Proceedings of the 5th IFAC Symposium on Robust Control Design*, Toulouse, France, 2006.

[Biannic *et al.*, 2007] J-M. Biannic, C. Roos, and S. Tarbouriech. A practical method for fixed-order anti-windup design. In *Proceedings of the 7th IFAC Symposium on Nonlinear Control Systems*, pages 527–532, Pretoria, South Africa, 2007.

[Biannic *et al.*, 2011] J-M. Biannic, C. Roos, and C. Pittet. Linear parameter varying analysis of switched controllers for attitude control systems. *AIAA Journal of Guidance, Control and Dynamics*, 34(5):1561–1566, 2011.

[Biannic *et al.*, 2012] J-M. Biannic, L. Burlion, S. Tarbouriech, and G. Garcia. On dynamic inversion with rate limitations. In *Proceedings of the American Control Conference*, pages 191–196, Montreal, Canada, 2012.

[Biannic *et al.*, 2014] J-M. Biannic, L. Burlion, and H. de Plinval. Robust control design over large flight envelopes: a promising approach for aerial robotics. *Aerospace Lab Journal*, 8, 2014. Available at http://www.aerospacelab-journal.org/al8.

[Biannic *et al.*, 2016] J-M. Biannic, G. Hardier, C. Roos, C. Seren, and L. Verdier. Surrogate models for aircraft flight control: some off-line and embedded applications. *Aerospace Lab Journal*, 12, 2016. Available at http://www.aerospacelab-journal.org/al12.

[Biannic *et al.*, 2017] J-M. Biannic, C. Roos, and J. Lesprier. Nonlinear structured H_∞ controllers for parameter-dependent uncertain systems with application to aircraft landing. *Aerospace Lab Journal*, 13, 2017. Available at http://www.aerospacelab-journal.org/al13.

[Bihua *et al.*, 2013] C. Bihua, J. Zongxia, and S. Ge. Aircraft-on-ground path following control by dynamical adaptive backstepping. *Chinese Journal of Aeronautics*, 26(3):668–675, 2013.

[Bing-Yu and Blaise, 1998] Z. Bing-Yu and M. Blaise. Robustness analysis of dynamic inversion control laws applied to nonlinear aircraft pitch-axis models. *Nonlinear Analysis Theory, Methods and Applications*, 32:501–532, 1998.

[Blum *et al.*, 1998] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and real computation.* Springer Verlag, 1998.

[Braatz and Morari, 1991] R. Braatz and M. Morari. μ-sensitivities as an aid for robust identification. In *Proceedings of the American Control Conference*, pages 231–236, Boston, Massachusetts, 1991.

[Braatz *et al.*, 1994] R. Braatz, P. Young, J. Doyle, and M. Morari. Computational complexity of μ calculation. *IEEE Transactions on Automatic Control*, 39(5):1000–1002, 1994.

[Brach and Brach, 2011] R. Brach and M. Brach. The tire-force ellipse (friction ellipse) and tire characteristics. Technical Report 2011-01-0094, SAE International, 2011.

[Brito and Kim, 2010] A. Brito and J. Kim. Application of a two-step calculation of the real structured singular value to the science mode of the LISA mission. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Toronto, Canada, 2010.

[Bucharles *et al.*, 2012] A. Bucharles, C. Cumer, G. Hardier, B. Jacquier, A. Janot, T. Le Moing, C. Seren, C. Toussaint, and P. Vacher. An overview of relevant issues for aircraft model identification. *Aerospace Lab Journal*, 4, 2012. Available at http://www.aerospacelab-journal.org/al4.

[Burke *et al.*, 2006] J. Burke, D. Henrion, A. Lewis, and M. Overton. Stabilization via nonsmooth, nonconvex optimization. *IEEE Transactions on Automatic Control*, 51(11):1760–1769, 2006.

[Calafiore *et al.*, 2011] G. Calafiore, F. Dabbene, and R. Tempo. Research on probabilistic methods for control system design. *Automatica*, 47:1279–1293, 2011.

[Cao *et al.*, 2002] Y. Cao, Z. Lin, and D. Ward. An antiwindup approach to enlarging domain of attraction for linear systems subject to actuator saturation. *IEEE Transactions on Automatic Control*, 47(1):140–145, 2002.

[Cassaro *et al.*, 2018] M. Cassaro, C. Roos, and J-M. Biannic. Robust dynamic allocation for aircraft roll-out phase directional control. In *Proceedings of the 17th European Control Conference*, Limassol, Cyprus, 2018.

[Castelan *et al.*, 2004] E. Castelan, I. Queinnec, S. Tarbouriech, and J. Gomes da Silva Jr. LMI approach for L_2-control of linear systems with saturating actuators. In *Proceedings of the IFAC Symposium on Nonlinear Control Systems*, pages 267–272, Stuttgart, Germany, 2004.

[Celis *et al.*, 2007] O.S. Celis, A. Cuyt, and B. Verdonk. Rational approximation of vertical segments. *Numerical Algorithms*, 45(1-4):375–388, 2007.

[Charbonnel, 2010] C. Charbonnel. H_∞ controller design and μ-analysis: powerful tools for flexible satellite attitude control. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Toronto, Canada, 2010.

[Chen and Sugie, 1997] G. Chen and T. Sugie. An upper bound of μ based on the parameter dependent multipliers. In *Proceedings of the American Control Conference*, pages 2604–2608, Albuquerque, New Mexico, 1997.

[Cheng and De Moor, 1994] Y. Cheng and B. De Moor. Continuous power iteration for computing lower bound of μ. In *Proceedings of the American Control Conference*, pages 1416–1417, Baltimore, Maryland, 1994.

[Chesi *et al.*, 2007] G. Chesi, A. Garulli, A. Tesi, and A. Vicino. Robust stability of time-varying polytopic systems via parameter-dependent homogeneous Lyapunov functions. *Automatica*, 43(2):309–316, 2007.

[Chesi, 2010] G. Chesi. LMI techniques for optimization over polynomials in control: a survey. *IEEE Transactions on Automatic Control*, 55(11):2500–2510, 2010.

[Clerc, 2006] M. Clerc. *Particle swarm optimization.* ISTE, London, 2006.

[Cockburn and Morton, 1997] J. Cockburn and B. Morton. Linear fractional representations of uncertain systems. *Automatica*, 33(7):1263–1271, 1997.

[Dailey, 1990] R. Dailey. A new algorithm for the real structured singular value. In *Proceedings of the American Control Conference*, pages 3036–3040, San Diego, California, 1990.

[de Caigny *et al.*, 2011] J. de Caigny, J. Camino, and J. Swevers. Interpolation-based modeling of MIMO LPV systems. *IEEE Transactions on Control Systems Technology*, 19(1):46–63, 2011.

[De Gaston and Safonov, 1988] R. De Gaston and M. Safonov. Exact calculation of the multiloop stability margin. *IEEE Transactions on Automatic Control*, 33(2):156–171, 1988.

[Dehaene *et al.*, 1997] J. Dehaene, C. Yi, and B. De Moor. Calculation of the structured singular value with gradient-based optimization algorithms on a Lie group of structured unitary matrices. *IEEE Transactions on Automatic Control*, 42(11):1596–1600, 1997.

[Demourant, 2013] F. Demourant. New algorithmic approach based on integral quadratic constraints for stability analysis of high order models. In *Proceedings of the 12th European Control Conference*, pages 359–364, Zurich, Switzerland, 2013.

[Deschrijver *et al.*, 2007] D. Deschrijver, B. Haegeman, and T. Dhaene. Orthonormal vector fitting: a robust macromodeling tool for rational approximation of frequency domain responses. *IEEE Transactions on Advanced Packaging*, 30(2):216–225, 2007.

[Devasia, 2002] S. Devasia. Should model-based inverse inputs be used as feedforward under plant uncertainty? *IEEE Transactions on Automatic Control*, 47(11):1865–1871, 2002.

[Döll *et al.*, 2008] C. Döll, C. Bérard, A. Knauf, and J-M. Biannic. LFT modelling of the 2-DOF longitudinal nonlinear aircraft behaviour. In *Proceedings of the 9th IEEE Symposium on Computer-Aided Control System Design*, pages 864–869, San Antonio, Texas, 2008.

[Döll *et al.*, 2012] C. Döll, F. Lescher, and C. Roos. Chapter 5: Generation of LFRs for a nonlinear controller and closed-loop aircraft models. In *Optimization based clearance of flight control laws*, pages 79–109. Lecture Notes in Control and Information Sciences, vol. 416, Springer, 2012.

[Dorobantu *et al.*, 2012] A. Dorobantu, A. Murch, and G. Balas. Robust control design for the NASA AirSTAR flight test vehicle. In *Proceedings of the 50th AIAA Aerospace Sciences Meeting*, Nashville, Tennessee, 2012.

[Douglas and Athans, 1995] J. Douglas and M. Athans. The calculation of μ-sensitivities. In *Proceedings of the American Control Conference*, pages 437–447, Seattle, Washington, 1995.

[Doyle, 1982] J. Doyle. Analysis of feedback systems with structured uncertainties. *IEE Proceedings, Part D*, 129(6):242–250, 1982.

[Duprez *et al.*, 2004] J. Duprez, F. Mora-Camino, and F. Villaumé. Aircraft-on-ground lateral control for low speed manoeuvers. In *Proceedings of the 16th IFAC Symposium on Automatic Control in Aerospace*, pages 475–478, St. Petersburg, Russia, 2004.

[D'Andrea and Khatri, 1997] R. D'Andrea and S. Khatri. Kalman decomposition of linear fractional transformation representations and minimality. In *Proceedings of the American Control Conference*, pages 3557–3561, Albuquerque, New Mexico, 1997.

[Elgersma *et al.*, 1996] M. Elgersma, J. Freudenberg, and B. Morton. Polynomial methods for the structured singular value with real parameters. *International Journal of Robust and Nonlinear Control*, 6(2):147–170, 1996.

[Eurocontrol, 2013] Eurocontrol. European action plan for the prevention of runway excursions edition 1.0, Technical Report, 2013.

[Fabrizi *et al.*, 2014] A. Fabrizi, C. Roos, and J-M. Biannic. A detailed comparative analysis of μ lower bound algorithms. In *Proceedings of the 13th European Control Conference*, pages 220–226, Strasbourg, France, 2014.

[Falcone *et al.*, 2007] P. Falcone, F. Borrelli, J. Asgari, H. Tseng, and D. Hrovat. Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on Control Systems Technology*, 15(3):566–580, 2007.

[Fan and Tits, 1992] M. Fan and A. Tits. A measure of worst-case H_∞ performance and of largest acceptable uncertainty. *Systems & Control Letters*, 18(6):409–421, 1992.

[Fan *et al.*, 1991] M. Fan, A. Tits, and J. Doyle. Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics. *IEEE Transactions on Automatic Control*, 36(1):25–38, 1991.

[Ferreres and Biannic, 2001] G. Ferreres and J-M. Biannic. Reliable computation of the robustness margin for a flexible aircraft. *Control Engineering Practice*, 9(12):1267–1278, 2001.

[Ferreres and Biannic, 2007] G. Ferreres and J-M. Biannic. Convex design of a robust antiwindup controller for an LFT model. *IEEE Transactions on Automatic Control*, 52(11):2173–2177, 2007.

[Ferreres and Fromion, 1996] G. Ferreres and V. Fromion. Robustness analysis using the ν tool. In *Proceedings of the 35th IEEE Conference on Decision and Control*, pages 4566–4571, Kobe, Japan, 1996.

[Ferreres and Fromion, 1997] G. Ferreres and V. Fromion. Computation of the robustness margin with the skewed μ tool. *Systems & Control Letters*, 32(4):193–202, 1997.

[Ferreres and M'Saad, 1994] G. Ferreres and M. M'Saad. Direct computation of the maximal s.s.v. over the frequency range using the ν tool. In *Proceedings of the 33rd IEEE Conference on Decision and Control*, pages 2147–2148, Lake Buena Vista, Florida, 1994.

[Ferreres and Roos, 2007] G. Ferreres and C. Roos. Robust feedforward design in the presence of LTI/LTV uncertainties. *International Journal of Robust and Nonlinear Control*, 17(14):1278–1293, 2007.

[Ferreres *et al.*, 1996] G. Ferreres, G. Scorletti, and V. Fromion. Advanced computation of the robustness margin. In *Proceedings of the 35th IEEE Conference on Decision and Control*, pages 4580–4584, Kobe, Japan, 1996.

[Ferreres *et al.*, 2003] G. Ferreres, J-F. Magni, and J-M. Biannic. Robustness analysis of flexible structures: practical algorithms. *International Journal of Robust and Nonlinear Control*, 13:715–734, 2003.

[Ferreres *et al.*, 2004] G. Ferreres, J-M. Biannic, and J-F. Magni. A skew mu toolbox (SMT) for robustness analysis. In *Proceedings of the 13th IEEE International Symposium on Computer Aided Control Systems Design*, pages 309–314, Taipei, Taiwan, 2004.

[Ferreres, 1999] G. Ferreres. *A practical approach to robustness analysis with aeronautical applications*. Kluwer Academic, 1999.

[Ferreres, 2004] G. Ferreres. Reduction of dynamic LFT systems with LTI model uncertainties. *International Journal of Robust and Nonlinear Control*, 14:307–323, 2004.

[Ferreres, 2011] G. Ferreres. Computation of a flexible aircraft LPV/LFT model using interpolation. *IEEE Transactions on Control Systems Technology*, 19(1):132–139, 2011.

[Fertik and Ross, 1967] H. Fertik and C. Ross. Direct digital control algorithm with anti-windup feature. *ISA Transactions*, 6:317–328, 1967.

[Floater and Hormann, 2007] M. Floater and K. Hormann. Barycentric rational interpolation with no poles and high rates of approximation. *Numerische Mathematik*, 107(2):315–331, 2007.

[Franco *et al.*, 2006] A. Franco, H. Bourlès, E. De Pieri, and H. Guillard. Robust nonlinear control associating robust feedback linearization and $\mathcal{H}_\infty$ control. *IEEE Transactions on Automatic Control*, 51:1200–1207, 2006.

[Fu and Barabanov, 1997] M. Fu and N. Barabanov. Improved upper bounds for the mixed structured singular value. *IEEE Transactions on Automatic Control*, 42(10):1447–1452, 1997.

[Gahinet and Apkarian, 1994] P. Gahinet and P. Apkarian. A Linear Matrix Inequality approach to H_∞ control. *International Journal of Robust and Nonlinear Control*, 4:421–448, 1994.

[Gahinet *et al.*, 1996] P. Gahinet, P. Apkarian, and M. Chilali. Affine parameter-dependent Lyapunov functions and real parametric uncertainty. *IEEE Transactions on Automatic Control*, 41(3):436–442, 1996.

[Galeani *et al.*, 2009] S. Galeani, S. Tarbouriech, M. Turner, and L. Zaccarian. A tutorial on modern anti-windup design. *European Journal of Control*, 3-4(15):418–440, 2009.

[Gallivan *et al.*, 2004] K. Gallivan, A. Vanderope, and P. Van-Dooren. Model reduction of MIMO systems via tangential interpolation. *SIAM Journal of Matrix Analysis and Application*, 26(2):328–349, 2004.

[Ganet-Schoeller *et al.*, 2009] M. Ganet-Schoeller, J. Bourdon, and G. Gelly. Non-linear and robust stability analysis for ATV rendezvous control. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Chicago, Illinois, 2009.

[Garulli *et al.*, 2012] A. Garulli, A. Masi, S. Paoletti, and E. Turkoglu. Chapter 9: Lyapunov-based robustness analysis techniques for clearance. In *Optimization based clearance of flight control laws*, pages 161–178. Lecture Notes in Control and Information Sciences, vol. 416, Springer, 2012.

[Garulli *et al.*, 2013] A. Garulli, A. Hansson, S. Khoshfetrat Pakazad, A. Masi, and R. Wallin. Robust finite-frequency H_2 analysis of uncertain systems with application to flight comfort analysis. *Control Engineering Practice*, 21(6):887–897, 2013.

[Ge *et al.*, 2012] D. Ge, G. Sun, and H. Karimi. Robust anti-windup control considering multiple design objectives. *Mathematical Problems in Engineering*, 2012.

[Georgieva and Serbezov, 2017] H. Georgieva and V. Serbezov. Mathematical model of aircraft ground dynamics. In *Proceedings of the International Conference on Military Technologies*, pages 514–519, Brno, Czech Republic, 2017.

[Giusto and Paganini, 1999] A. Giusto and F. Paganini. Robust synthesis of feedforward compensators. *IEEE Transactions on Automatic Control*, 44(8):1578–1582, 1999.

[Glavaski and Tierno, 1998] S. Glavaski and J. Tierno. Advances in worst-case H_∞ performance computation. In *Proceedings of the IEEE Conference on Control Applications*, pages 668–673, Trieste, Italy, 1998.

[Gomes da Silva Jr and Tarbouriech, 2005] J. Gomes da Silva Jr and S. Tarbouriech. Anti-windup design with guaranteed regions of stability: an LMI-based approach. *IEEE Transactions on Automatic Control*, 50(1):106–111, 2005.

[Goupil *et al.*, 2014] P. Goupil, J. Boada-Bauxell, A. Marcos, E. Cortet, M. Kerr, and H. Costa. Airbus efforts towards advanced real-time fault diagnosis and fault tolerant control. In *Proceedings of the 19th IFAC World Congress*, pages 3471–3476, Cape Town, South Africa, 2014.

[Grimm *et al.*, 2003] G. Grimm, J. Hatfield, I. Postlethwaite, A. Teel, M. Turner, and L. Zaccarian. Anti-windup for stable linear systems with input saturation: an LMI-based synthesis. *IEEE Transactions on Automatic Control*, 48(9):1509–1525, 2003.

[Gugercin, 2008] S. Gugercin. An iterative SVD-Krylov based method for model reduction of large-sale dynamical systems. *Linear Algebra and its Applications*, 428(8-9):1964–1986, 2008.

[Haddad and Kapila, 1998] W. Haddad and V. Kapila. Robust stabilization for continuous-time systems with slowly time-varying uncertain real parameters. *IEEE Transactions on Automatic Control*, 43(7):987–992, 1998.

[Halton *et al.*, 2008] M. Halton, M. Hayes, and P. Iordanov. State-space μ analysis for an experimental drive-by-wire vehicle. *International Journal of Robust and Nonlinear Control*, 18(9):975–992, 2008.

[Hardier *et al.*, 2013a] G. Hardier, C. Roos, and C. Seren. Creating sparse rational approximations for Linear Fractional Representations using genetic programming. In *Proceedings of the 3rd IFAC International Conference on Intelligent Control and Automation Science*, pages 232–237, Chengdu, China, 2013.

[Hardier *et al.*, 2013b] G. Hardier, C. Roos, and C. Seren. Creating sparse rational approximations for Linear Fractional Representations using surrogate modeling. In *Proceedings of the 3rd IFAC International Conference on Intelligent Control and Automation Science*, pages 238–243, Chengdu, China, 2013.

[Härkegård, 2004] O. Härkegård. Dynamic control allocation using constrained quadratic programming. *Journal of Guidance, Control, and Dynamics*, 27(6):1028–1034, 2004.

[Hayes *et al.*, 2001] M. Hayes, D. Bates, and I. Postlethwaite. New tools for computing tight bounds on the real structured singular value. *AIAA Journal of Guidance, Control and Dynamics*, 24(6):1204–1213, 2001.

[Helmersson, 1995] A. Helmersson. A finite frequency method for μ-analysis. In *Proceedings of the 3rd European Control Conference*, pages 171–176, Rome, Italy, 1995.

[Helmersson, 1999] A. Helmersson. An IQC-based stability criterion for systems with slowly varying parameters. In *Proceedings of the 14th IFAC World Congress*, pages 525–530, Beijing, China, 1999.

[Hencey and Alleyne, 2006] B. Hencey and A. Alleyne. A static anti-windup compensator design technique for robust regional pole placement. In *Proceedings of the ASME International Mechanical Engineering Congress and Exposition*, Chicago, Illinois, 2006.

[Holland *et al.*, 2005] R. Holland, P. Young, and C. Zhu. Development of a skew μ lower bound. *International Journal of Robust and Nonlinear Control*, 15(11):495–506, 2005.

[Hu *et al.*, 2005] T. Hu, A. Teel, and L. Zaccarian. Regional anti-windup compensation for linear systems with input saturation. In *Proceedings of the American Control Conference*, pages 3397–3402, Portland, Oregon, 2005.

[Iannelli *et al.*, 2017] A. Iannelli, P. Simplicio, D. Navarro-Tapia, and A. Marcos. LFT modeling and μ analysis of the aircraft landing benchmark. In *Proceedings of the 20th IFAC World Congress*, pages 4026–4031, Toulouse, France, 2017.

[Iordanov and Halton, 2015] P. Iordanov and M. Halton. Computation of the real structured singular value via pole migration. *International Journal of Robust and Nonlinear Control,*, 25:3163–3178, 2015.

[Iordanov *et al.*, 2003] P. Iordanov, M. Hayes, and M. Halton. On μ-analysis and synthesis for systems subject to real uncertainty. In *Proceedings of the 7th European Control Conference*, pages 420–425, Cambridge, United Kingdom, 2003.

[Isidori, 1995] A. Isidori. *Nonlinear Control Systems*, volume 1 of *Communications and Control Engineering*. Springer, 1995.

[Iwasaki and Hara, 2005] T. Iwasaki and S. Hara. Generalized KYP lemma: unified frequency domain inequalities with design applications. *IEEE Transactions on Automatic Control*, 50(1):41–59, 2005.

[Jeanneau, 2007] M. Jeanneau. Chapter 1: The Airbus on-ground transport aircraft benchmark. In *Nonlinear Analysis and Synthesis Techniques for Aircraft Control*, pages 3–24. Lecture Notes in Control and Information Sciences, vol. 365, Springer, 2007.

[Johansen and Fossen, 2013] T. Johansen and T. Fossen. Control allocation – A survey. *Automatica*, 49(5):1087–1103, 2013.

[Jonsson and Rantzer, 1996] U. Jonsson and A. Rantzer. Systems with uncertain parameters - time-variations with bounded derivatives. *International Journal of Robust and Nonlinear Control*, 6(9-10):969–982, 1996.

[Kaminer and Khargonekar, 1990] I. Kaminer and P. Khargonekar. Design of the flare control law for longitudinal autopilot using H_∞ synthesis. In *Proceedings of the 29th IEEE Conference on Decision and Control*, pages 2981–2986, Honolulu, Hawaii, 1990.

[Kara Mohamed and Lanzon, 2012] M. Kara Mohamed and A. Lanzon. Design and control of novel tri-rotor UAV. In *Proceedings of the 2012 UKACC International Conference on Control*, pages 304–309, Cardiff, United Kingdom, 2012.

[Kara Mohamed and Lanzon, 2013] M. Kara Mohamed and A. Lanzon. Effect of unmodelled actuator dynamics on feedback linearised systems and a two stage feedback linearisation method. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, pages 841–846, Florence, Italy, 2013.

[Karlsson and Lindquist, 2008] J. Karlsson and A. Lindquist. Stability-preserving rational approximation subject to interpolation constraints. *IEEE Transactions on Automatic Control*, 53(7):1724–1730, 2008.

[Kasimbeyli *et al.*, 2009] R. Kasimbeyli, O. Ustun, and A. Rubinov. The modified subgradient algorithm based on feasible values. *Optimization*, 58(5):535–560, 2009.

[Khalil, 1996] H. Khalil. *Nonlinear systems*. Prentice Hall, 1996.

[Kim *et al.*, 2009] J. Kim, D. Bates, and I. Postlethwaite. A geometrical formulation of the μ-lower bound problem. *IET Control Theory & Applications*, 3(4):465–472, 2009.

[Kiyama and Sawada, 2004] T. Kiyama and K. Sawada. LMI-based $\mathcal{L}_2$ analysis / synthesis with saturating control via generalized sector approach. In *Proceedings of the IFAC Symposium on Large Scale Systems*, pages 504–510, Osaka, Japan, 2004.

[Köroğlu and Scherer, 2007] H. Köroğlu and C. Scherer. Robust performance analysis for structured linear time-varying perturbations with bounded rates-of-variation. *IEEE Transactions on Automatic Control*, 52(2):197–211, 2007.

[Köse and Scherer, 2009] I. Köse and C. Scherer. Robust $\mathcal{L}_2$-gain feedforward control of uncertain systems using dynamic IQCs. *International Journal of Robust and Nonlinear Control*, 19:1224–1247, 2009.

[Kothare *et al.*, 1994] M. Kothare, P. Campo, M. Morari, and C. Nett. A unified framework for the study of anti-windup designs. *Automatica*, 30(12):1869–1883, 1994.

[Koza and Poli, 2005] J. Koza and R. Poli. Genetic programming. In E. Burke and G. Kendall, editors, *Search Methodologies. Introductory Tutorials in Optimization and Decision Support Techniques*, pages 127–164. Springer, 2005.

[Lawrence *et al.*, 2000] C. Lawrence, L. Tits, and P. Van Dooren. A fast algorithm for the computation of an upper bound on the μ-norm. *Automatica*, 36(3):449–456, 2000.

[Lee and Edgar, 2003] J. Lee and T. Edgar. Upper bounds of structured singular values for mixed uncertainties. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 798–802, Maui, Hawaii, 2003.

[Leith and Leithead, 2000] D. Leith and W. Leithead. Survey of gain-scheduling analysis and design. *International Journal of Control*, 73:1001–1025, 2000.

[Lemay *et al.*, 2011] D. Lemay, Y. Chamaillard, M. Basset, and J-P. Garcia. Gain-scheduled yaw control for aircraft ground taxiing. In *Proceedings of the 18th IFAC World Congress*, pages 12970–12975, Milano, Italy, 2011.

[Lemos *et al.*, 2014] R. Lemos, A. Simões, and P. Apkarian. A non-smooth lower bound on ν. *International Journal of Robust and Nonlinear Control*, 24(3):477–494, 2014.

[Lescher and Roos, 2011] F. Lescher and C. Roos. Robust stability of time-delay systems with structured uncertainties: a μ-analysis based algorithm. In *Proceedings of the 50th IEEE Conference on Decision and Control*, pages 4955–4960, Orlando, Florida, 2011.

[Lesprier *et al.*, 2014] J. Lesprier, J-M. Biannic, and C. Roos. Nonlinear structured H_∞ controllers for parameter-dependent uncertain systems with application to aircraft landing. In *Proceedings of the IEEE Multiconference on Systems and Control*, pages 433–438, Antibes, France, 2014.

[Lesprier *et al.*, 2015a] J. Lesprier, J-M. Biannic, and C. Roos. Modeling and robust nonlinear control of a fixed-wing UAV. In *Proceedings of the IEEE Multiconference on Systems and Control*, pages 1334–1339, Sydney, Australia, 2015.

[Lesprier *et al.*, 2015b] J. Lesprier, C. Roos, and J-M. Biannic. Efficient computation of the multiplier based μ upper bound on large frequency intervals. In *Proceedings of the 54th IEEE Conference on Decision and Control*, pages 4621–4626, Osaka, Japan, 2015.

[Lesprier *et al.*, 2015c] J. Lesprier, C. Roos, and J-M. Biannic. Improved μ upper bound computation using the μ-sensitivities. In *Proceedings of the 8th IFAC Symposium on Robust Control Design*, pages 214–219, Bratislava, Slovak Republic, 2015.

[Lind *et al.*, 1995] R. Lind, G. Balas, and A. Packard. Robustness analysis with linear time-invariant and time-varying real uncertainty. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, pages 132–140, Baltimore, Maryland, 1995.

[Looye and Joos, 2006] G. Looye and H.D. Joos. Design of autoland controller functions with multiobjective optimization. *Journal of Guidance, Control and Dynamics*, 29(2):475–484, 2006.

[Looye, 2007] G. Looye. Chapter 8: Rapid prototyping using inversion-based control and object-oriented modelling. In *Nonlinear Analysis and Synthesis Techniques for Aircraft Control*, pages 147–173. Lecture Notes in Control and Information Sciences, vol. 365, Springer, 2007.

[Lovera and Mercère, 2007] M. Lovera and G. Mercère. Identification for gain-scheduling: a balanced subspace approach. In *Proceedings of the American Control Conference*, pages 858–863, New York City, New York, 2007.

[Lozier, 1956] J. Lozier. A steady-state approach to the theory of saturable servo systems. *IRE Transactions on Automatic Control*, 1(1):19–39, 1956.

[Lu *et al.*, 2005] B. Lu, F. Wu, and S. Kim. Linear parameter varying anti-windup compensation for enhanced flight control performance. *AIAA Journal of Guidance, Control and Dynamics*, 28(3):494–504, 2005.

[Ly *et al.*, 1998] J. Ly, R. Chiang, K. Goh, and M. Safonov. LMI multiplier K_m/μ analysis of the Cassini spacecraft. *International Journal of Robust and Nonlinear Control*, 8(2):155–168, 1998.

[Magni *et al.*, 1999] J-F. Magni, C. Döll, C. Chiappa, B. Frappard, and B. Girouart. Mixed μ-analysis for flexible systems. Part 1: theory. In *Proceedings of the 14th IFAC World Congress*, pages 325–360, Beijing, China, 1999.

[Magni, 1999] J-F. Magni. Multimodel eigenstructure assignment in flight-control design. *Aerospace Science and Technology*, 3(3):141–151, 1999.

[Magni, 2006] J-F. Magni. Linear Fractional Representation Toolbox for use with Matlab. Available with the SMAC Toolbox at http://w3.onera.fr/smac/lfrt, 2006.

[Marcos and Balas, 2004] A. Marcos and G. Balas. Development of linear parameter varying models for aircraft. *AIAA Journal of Guidance, Control and Dynamics*, 27(2):218–228, 2004.

[Markov *et al.*, 1996] S. Markov, E. Popova, U. Schneider, and J. Schulze. On linear interpolation under interval data. *Mathematics and Computers in Simulation*, 42:35–45, 1996.

[Matsuda and Mori, 2009] T. Matsuda and T. Mori. Stability feeler: a tool for parametric robust stability analysis and its applications. *IET Control Theory & Applications*, 3(12):1625–1633, 2009.

[Matsuda *et al.*, 2009] T. Matsuda, M. Kawanishi, and T. Narikiyo. Computation of real structured singular value by stability feeler. In *Proceedings of the Asian Control Conference*, pages 672–677, Hong Kong, China, 2009.

[Megretski and Rantzer, 1997] A. Megretski and A. Rantzer. System analysis via Integral Quadratic Constraints. *IEEE Transactions on Automatic Control*, 42(6):819–830, 1997.

[Meinsma *et al.*, 2000] G. Meinsma, T. Iwasaki, and M. Fu. When is (D,G)-scaling both necessary and sufficient. *IEEE Transactions on Automatic Control*, 45(9):1755–1759, 2000.

[Menon *et al.*, 2009] P. Menon, I. Postlethwaite, S. Bennani, A. Marcos, and D. Bates. Robustness analysis of a reusable launch vehicle flight control law. *Control Engineering Practice*, 17:751–765, 2009.

[Mitchell, 2006] D. Mitchell. Frictional and retarding forces on aircraft tyres part IV: Estimation of effects of yaw. Technical Report 86016a, ESDU, 2006.

[Montagner and Peres, 2004] V. Montagner and P. Peres. Robust stability and performance of linear time-varying systems in polytopic domains. *International Journal of Control*, 77(15):1343–1352, 2004.

[Morelli and DeLoach, 2003] E. Morelli and R. DeLoach. Wind tunnel database development using modern experiment design and multivariate orthogonal functions. In *Proceedings of the 41st AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 2003.

[Newlin and Glavaski, 1995] M. Newlin and S. Glavaski. Advances in the computation of the μ lower bound. In *Proceedings of the American Control Conference*, pages 442–446, Seattle, Washington, 1995.

[Newlin and Young, 1997] M. Newlin and P. Young. Mixed μ problems and branch and bound techniques. *International Journal of Robust and Nonlinear Control*, 7:145–164, 1997.

[Packard and Doyle, 1993] A. Packard and J. Doyle. The complex structured singular value. *Automatica*, 29(1):71–109, 1993.

[Packard and Pandey, 1993] A. Packard and P. Pandey. Continuity properties of the real/complex structured singular value. *IEEE Transactions on Automatic Control*, 38(3):415–428, 1993.

[Packard *et al.*, 2000] A. Packard, G. Balas, R. Liu, and J. Shin. Results on worst-case performance assessment. In *Proceedings of the American Control Conference*, pages 2425–2427, Chicago, Illinois, 2000.

[Paganini and Feron, 2000] F. Paganini and E. Feron. Chapter 7: Linear matrix inequality methods for robust H_2 analysis: a survey with comparisons. In *Advances in linear matrix inequality methods in control*, pages 129–151. Society for Industrial and Applied Mathematics, 2000.

[Paganini, 1996] F. Paganini. Robust stability under mixed time varying, time invariant and parametric uncertainty. *Automatica*, 32(10):1381–1392, 1996.

[Paijmans *et al.*, 2006] B. Paijmans, W. Symens, H. van Brussel, and J. Swevers. A gain-scheduling-control technique for mechatronic systems with position-dependent dynamics. In *Proceedings of the American Control Conference*, pages 2933–2938, Minneapolis, Minnesota, 2006.

[Papageorgiou and Glover, 2004] C. Papageorgiou and K. Glover. Robustness analysis of nonlinear dynamic inversion control laws with application to flight control. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, pages 3485–3490, Nassau, Bahamas, 2004.

[Papageorgiou and Glover, 2005] C. Papageorgiou and K. Glover. Robustness analysis of nonlinear flight controllers. *Journal of Guidance, Control and Dynamics*, 28(4):639–648, 2005.

[Papageorgiou *et al.*, 2012] C. Papageorgiou, R. Falkeborn, and A. Hansson. Chapter 10: IQC-based analysis techniques for clearance. In *Optimization based clearance of flight control laws*, pages 179–201. Lecture Notes in Control and Information Sciences, vol. 416, Springer, 2012.

[Peni and Seiler, 2016] T. Peni and P. Seiler. Computation of lower bounds for the induced $\mathcal{L}_2$ norm of LPV systems. *International Journal of Robust and Nonlinear Control*, 26:646–661, 2016.

[Petersson and Lofberg, 2009] D. Petersson and J. Lofberg. Optimization based LPV-approximation of multi-model systems. In *Proceedings of the European Control Conference*, pages 3172–3177, Budapest, Hungary, 2009.

[Petrushev and Popov, 1987] P. Petrushev and V. Popov. *Rational approximation of real functions*, volume 28 of *Encyclopedia of Mathematics and its Applications*. University Press, Cambridge, 1987.

[Pfifer and Hecker, 2011] H. Pfifer and S. Hecker. Generation of optimal linear parametric models for LFT-based robust stability analysis and control design. *IEEE Transactions on Control Systems Technology*, 19(1):118–131, 2011.

[Pfifer and Seiler, 2016] H. Pfifer and P. Seiler. Less conservative robustness analysis of linear parameter varying systems using integral quadratic constraints. *International Journal of Robust and Nonlinear Control*, 26(16):3580–3594, 2016.

[Poussot-Vassal and Roos, 2012] C. Poussot-Vassal and C. Roos. Generation of a reduced-order LPV/LFT model from a set of large-scale MIMO LTI flexible aircraft models. *Control Engineering Practice*, 20(9):919–930, 2012.

[Poussot-Vassal *et al.*, 2013] C. Poussot-Vassal, T. Loquen, and C. Roos. JTI Clean Sky SFWA-ITD / WP1.2.3 - Advanced load control techniques - Vibration control design for BizJet aircraft: a multi model approach. Technical Report 2/20894, ONERA/DCSD, 2013.

[Poussot-Vassal *et al.*, 2014] C. Poussot-Vassal, C. Roos, P. Vuillemin, O. Cantinaud, and J-P. Lacoste. Chapter 11: Control-oriented aeroelastic BizJet low-order LFT modeling. In *Control-oriented modelling and identification: Theory and practice*, pages 241–268. Control Engineering Series, vol. 80, IET, 2014.

[Pulecchi *et al.*, 2012] T. Pulecchi, A. Marcos, C. Roux, V. Mantini, and S. Bennani. Thrust vector control robustness of axial-symmetric launch vehicles with fuel slosh. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Minneapolis, Minnesota, 2012.

[Puyou and Losser, 2012] G. Puyou and Y. Losser. Chapter 2: Clearance benchmark for a civil aircraft. In *Optimization based clearance of flight control laws*, pages 11–36. Lecture Notes in Control and Information Sciences, vol. 416, Springer, 2012.

[Puyou *et al.*, 2012] G. Puyou, R. Fernandes de Oliveira, and A. Berard. Chapter 17: Evaluation of clearance techniques in an industrial context. In *Optimization based clearance of flight control laws*, pages 319–357. Lecture Notes in Control and Information Sciences, vol. 416, Springer, 2012.

[Rankin *et al.*, 2009] J. Rankin, E. Coetzee, B. Krauskopf, and M. Lowenberg. Bifurcation and stability analysis of aircraft turning on the ground. *Journal of Guidance, Control and Dynamics*, 32(2):499–510, 2009.

[Reiner *et al.*, 1995] J. Reiner, G. Balas, and W. Garrard. Robust dynamic inversion for control of highly maneuverable aircraft. *Journal of Guidance, Control and Dynamics*, 18(1):18–24, 1995.

[Reiner *et al.*, 1996] J. Reiner, G. Balas, and W. Garrard. Flight control design using robust dynamic inversion and time-scale separation. *Automatica*, 32:1493–1504, 1996.

[Roos and Biannic, 2006a] C. Roos and J-M. Biannic. On robustness analysis versus mixed LTI/LTV uncertainties. In *Proceedings of the 5th IFAC Symposium on Robust Control Design*, Toulouse, France, 2006.

[Roos and Biannic, 2006b] C. Roos and J-M. Biannic. A positivity approach to robust controllers analysis and synthesis versus mixed LTI/LTV uncertainties. In *Proceedings of the American Control Conference*, pages 3661–3666, Minneapolis, Minnesota, 2006.

[Roos and Biannic, 2007] C. Roos and J-M. Biannic. Chapter 9: Robustness analysis versus mixed LTI/LTV uncertainties for on-ground aircraft. In *Nonlinear Analysis and Synthesis Techniques for Aircraft Control*, pages 175–194. Lecture Notes in Control and Information Sciences, vol. 365, Springer, 2007.

[Roos and Biannic, 2008] C. Roos and J-M. Biannic. A convex characterization of dynamically-constrained anti-windup controllers. *Automatica*, 44(9):2449–2452, 2008.

[Roos and Biannic, 2010] C. Roos and J-M. Biannic. Efficient computation of a guaranteed stability domain for a high-order parameter dependent plant. In *Proceedings of the American Control Conference*, pages 3895–3900, Baltimore, Maryland, 2010.

[Roos and Biannic, 2015] C. Roos and J-M. Biannic. A detailed comparative analysis of all practical algorithms to compute lower bounds on the structured singular value. *Control Engineering Practice*, 44:219–230, 2015.

[Roos and Biannic, 2018] C. Roos and J-M. Biannic. Computing the exact value of μ is no longer an issue. *Submitted to the 9th IFAC Symposium on Robust Control Design*, 2018.

[Roos *et al.*, 2010] C. Roos, J-M. Biannic, S. Tarbouriech, C. Prieur, and M. Jeanneau. On-ground aircraft control design using a parameter-varying anti-windup approach. *Aerospace Science and Technology*, 14(7):459–471, 2010.

[Roos *et al.*, 2011] C. Roos, F. Lescher, J-M. Biannic, C. Döll, and G. Ferreres. A set of μ-analysis based tools to evaluate the robustness properties of high-dimensional uncertain systems. In *Proceedings of the IEEE Multiconference on Systems and Control*, pages 644–649, Denver, Colorado, 2011.

[Roos *et al.*, 2012] C. Roos, L. Lafourcade, and J-M. Biannic. Stability and performance analysis in the presence of LTV perturbations with bounded rates of variation: a Lyapunov based approach. In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 6659–6664, Maui, Hawaii, 2012.

[Roos *et al.*, 2014] C. Roos, G. Hardier, and J-M. Biannic. Polynomial and rational approximation with the APRICOT Library of the SMAC Toolbox. In *Proceedings of the IEEE Multiconference on Systems and Control*, pages 1473–1478, Antibes, France, 2014.

[Roos, 2007] C. Roos. Contribution à la commande des systèmes saturés en présence d'incertitudes et de variations paramétriques - Application au pilotage de l'avion au sol. PhD thesis, Institut Supérieur de l'Aéronautique et de l'Espace, Toulouse, France, 2007.

[Roos, 2009] C. Roos. Generation of flexible aircraft LFT models for robustness analysis. In *Proceedings of the 6th IFAC Symposium on Robust Control Design*, Haifa, Israel, 2009.

[Roos, 2010] C. Roos. A practical approach to worst-case H_∞ performance computation. In *Proceedings of the IEEE Multiconference on Systems and Control*, pages 380–385, Yokohama, Japan, 2010.

[Roos, 2012] C. Roos. COCKPIT/OCKF/Co3.1 - Outils pour l'analyse de la robustesse des performances. Génération d'un modèle LPV/LFT par interpolation d'un ensemble de modèles LTI. Technical Report 1/17443, ONERA/DCSD, 2012.

[Roos, 2013] C. Roos. Systems Modeling, Analysis and Control (SMAC) Toolbox: an insight into the robustness analysis library. In *Proceedings of the IEEE Multiconference on Systems and Control*, pages 176–181, Hyderabad, India, 2013.

[Sadat-Hoseini *et al.*, 2013] H. Sadat-Hoseini, A. Fazelzadeh, A. Rasti, and P. Marzocca. Final approach and flare control of a flexible aircraft in crosswind landings. *Journal of Guidance, Control and Dynamics*, 36(4):946–957, 2013.

[Sadien *et al.*, 2018] E. Sadien, C. Roos, A. Birouche, C. Grimault, L-E. Romana, J. Boada-Bauxell, and M. Basset. Control design oriented modeling of an on-ground aircraft. In *Proceedings of the 17th European Control Conference*, Limassol, Cyprus, 2018.

[Saeki and Wada, 2002] M. Saeki and N. Wada. Synthesis of a static anti-windup compensator via linear matrix inequalities. *International Journal of Robust and Nonlinear Control*, 12(10):927–953, 2002.

[Sanathanan and Koerner, 1963] C. Sanathanan and J. Koerner. Transfer function synthesis as a ratio of two complex polynomials. *IEEE Transactions on Automatic Control*, 8(1):56–58, 1963.

[Sato, 2009] M. Sato. Parameter-dependent slack variable approach for positivity check of polynomials over hyper-rectangle. In *Proceedings of the American Control Conference*, pages 5357–5362, St Louis, Missouri, 2009.

[Scorletti and Fromion, 2006] G. Scorletti and V. Fromion. Further results on the design of robust H_∞ feedforward controllers and filters. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 3560–3565, San Diego, California, 2006.

[Scorletti *et al.*, 2007] G. Scorletti, X. Bombois, M. Barenthin, and V. Fromion. Improved efficient analysis for systems with uncertain parameters. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 5038–5043, New Orleans, Louisiana, 2007.

[Seiler *et al.*, 2010] P. Seiler, A. Packard, and G. Balas. A gain-based lower bound algorithm for real and mixed μ problems. *Automatica*, 46(3):493–500, 2010.

[Seren *et al.*, 2011] C. Seren, G. Hardier, and P. Ezerzere. On-line estimation of longitudinal flight parameters. In *Proceedings of the SAE AeroTech Congress and Exhibition*, Toulouse, France, 2011.

[Shamma, 1994] J. Shamma. Robust stability with time-varying structured uncertainty. *IEEE Transactions on Automatic Control*, 39(4):714–724, 1994.

[Shor, 1987] N. Shor. Class of global minimum bounds of polynomial functions. *Cybernetics*, 23(6):731–734, 1987.

[Sideris and Sanchez Pena, 1990] A. Sideris and R. Sanchez Pena. Robustness margin calculation with dynamic and real parametric uncertainty. *IEEE Transactions on Automatic Control*, 35(8):970–974, 1990.

[Sideris, 1992] A. Sideris. Elimination of frequency search from robustness tests. *IEEE Transactions on Automatic Control*, 37(10):1635–1640, 1992.

[Snell *et al.*, 1992] A. Snell, D. Enns, and W. Garrard. Nonlinear inversion flight control for a supermaneuverable aircraft. *Journal of Guidance, Control, and Dynamics*, 15(4):976–984, 1992.

[Sparks *et al.*, 1996] A. Sparks, P. Blue, and S. Banda. An LMI formulation of robustness analysis for systems with time-varying and LTI uncertainty. In *Proceedings of the AIAA Guidance, Navigation and Control Conference*, San Diego, California, 1996.

[Steinbuch *et al.*, 2003] M. Steinbuch, R. van de Molengraft, and A. van der Voort. Experimental modelling and LPV control of a motion system. In *Proceedings of the American Control Conference*, pages 1374–1379, Denver, Colorado, 2003.

[Tarbouriech and Turner, 2009] S. Tarbouriech and M. Turner. Anti-windup design: an overview of some recent advances and open problems. *IET Control Theory and Applications*, 3(1):1–19, 2009.

[Tarbouriech *et al.*, 2006a] S. Tarbouriech, J. Gomes da Silva Jr, and F. Bender. Dynamic anti-windup synthesis for discrete-time linear systems subject to input saturation and $\mathcal{L}_2$ disturbances. In *Proceedings of the 5th IFAC Symposium on Robust Control Design*, Toulouse, France, 2006.

[Tarbouriech *et al.*, 2006b] S. Tarbouriech, C. Prieur, and J. Gomes da Silva Jr. Stability analysis and stabilization of systems presenting nested saturations. *IEEE Transactions on Automatic Control*, 51(8):1364–1371, 2006.

[Tarbouriech *et al.*, 2006c] S. Tarbouriech, I. Queinnec, and G. Garcia. Stability region enlargement through anti-windup strategy for linear systems with dynamics restricted actuator. *International Journal of Systems Science*, 37(2):79–90, 2006.

[Tempo *et al.*, 2013] R. Tempo, G. Calafiore, and F. Dabbene. *Randomized algorithms for analysis and control of uncertain systems.* Communications and Control Engineering, Springer, 2013.

[The Mathworks, 2014] The Mathworks. Curve Fitting Toolbox for Matlab. http://www. mathworks.com/products/curvefitting, 2014.

[The Mathworks, 2017a] The Mathworks. Optimization Toolbox for Matlab. https://www. mathworks.com/products/optimization/, 2017.

[The Mathworks, 2017b] The Mathworks. Robust Control Toolbox for Matlab. https://www. mathworks.com/products/robust/, 2017.

[Tierno and Young, 1992] J. Tierno and P. Young. An improved μ lower bound via adaptive power iteration. In *Proceedings of the 31st IEEE Conference on Decision and Control*, pages 3181–3186, Tucson, Arizona, 1992.

[Tits and Balakrishnan, 1998] A. Tits and V. Balakrishnan. Small-μ theorems with frequency-dependent uncertainty bounds. *Mathematics of Control, Signals, and Systems*, 11(3):220–243, 1998.

[Torralba *et al.*, 2009] J. Torralba, F. Demourant, G. Puyou, and G. Ferreres. A method for flexible aircraft LFT modelling. In *Proceedings of the 10th European Control Conference*, pages 3933–3938, Budapest, Hungary, 2009.

[Turbuk and Paglione, 2010] M. Turbuk and P. Paglione. Aircraft ground dynamics modeling. In *Proceedings of the VI National Congress of Mechanical Engineering*, Campina Grande, Brazil, 2010.

[Turner and Postlethwaite, 2004] M. Turner and I. Postlethwaite. A new perspective on static and low order anti-windup synthesis. *International Journal of Control*, 77:27–44, 2004.

[van der Waerden, 1991] B. van der Waerden. *Algebra.* Springer Verlag, New York, 1991.

[Van-Dooren *et al.*, 2008] P. Van-Dooren, K. Gallivan, and P. Absil. $\mathcal{H}_2$-optimal model reduction of MIMO systems. *Applied Mathematics Letters*, 21(12):53–62, 2008.

[Vizer and Mercère, 2014] D. Vizer and G. Mercère. H_∞-based LPV model identification from local experiments with a gap metric-based operating point selection. In *Proceedings of the European Control Conference*, pages 388–393, Strasbourg,France, 2014.

[Vuillemin *et al.*, 2013] P. Vuillemin, C. Poussot-Vassal, and D. Alazard. $\mathcal{H}_2$ optimal and frequency limited approximation methods for large-scale LTI dynamical systems. In *Proceedings of the 6th IFAC Symposium on Systems Structure and Control*, pages 719–724, Grenoble, France, 2013.

[Wu and Soto, 2004] F. Wu and M. Soto. Extended anti-windup control schemes for LTI and LFT systems with actuator saturations. *International Journal of Robust and Nonlinear Control*, 14(15):1255–1281, 2004.

[Yazıcı *et al.*, 2011] A. Yazıcı, A. Karamancıoğlu, and R. Kasimbeyli. A nonlinear programming technique to compute a tight lower bound for the real structured singular value. *Optimization and Engineering*, 12(3):445–458, 2011.

[Young and Doyle, 1997] P. Young and J. Doyle. A lower bound for the mixed μ problem. *IEEE Transactions on Automatic Control*, 42(1):123–128, 1997.

[Young *et al.*, 1995] P. Young, M. Newlin, and J. Doyle. Computing bounds for the mixed μ problem. *International Journal of Robust and Nonlinear Control*, 5(6):573–590, 1995.

[Young, 1996] P. Young. Controller design with real parametric uncertainty. *International Journal of Control*, 65:469–509, 1996.

[Young, 2001] P. Young. Structured singular value approach for systems with parametric uncertainty. *International Journal of Robust and Nonlinear Control*, 11:653–680, 2001.

[Zaccarian and Teel, 2011] L. Zaccarian and A. Teel. *Modern anti-windup synthesis. Control augmentation for actuator saturation.* Princeton University Press, 2011.

[Zaccarian, 2009] L. Zaccarian. Dynamic allocation for input redundant control systems. *Automatica*, 45(6):1431–1438, 2009.

[Zadeh and Desoer, 1963] L. Zadeh and C. Desoer. *Linear system theory.* New York: McGraw Hill, 1963.

[Zhou *et al.*, 1996] K. Zhou, J. Doyle, and K. Glover. *Robust and optimal control.* Prentice-Hall, New Jersey, 1996.

Part II

Detailed curriculum vitae

Advanced control laws design and validation - A set of methods and tools to bridge the gap between theory and practice

Many real-world applications suffer from strong cost constraints or weight limitations, which dictate the use of potentially low performance hardware (actuators, sensors, onboard computers). For example, the access to space is becoming significantly easier and cheaper with the advent of nanosats, and low-cost launchers must be developed to promote this new economic model. In the military domain, one of the most topical issues is to improve the precision and to enhance the range of traditional artillery shells in the modern battlefield, while significantly reducing costs. And UAV are now widely used to perform a large number of demanding missions. So there is a real need to control various kinds of systems on extended operational domains in the presence of hardware limitations, uncertainties, varying parameters as well as actuator saturations. In this context, our contribution lies at the frontier between research and engineering. Based on known theories and results in the fields of LFR modeling, μ/IQC/Lyapunov-based analysis, as well as analysis of saturated systems, we first try to develop validation methods that can be applied to real-world issues with a reasonable computational cost. Heuristics sometimes replace rigorous mathematical proofs, but we believe this is the price to pay for bridging the gap between theory and practice. Many efforts are also invested in the development of generic computational tools designed for control engineers. Then, we show how these methods and tools can be directly integrated into the control laws design process, so as to reduce a little bit the number of iterations between design and validation in an industrial context. The next step will be to face the implementation constraints. Thanks to our growing involvement in ONERA's UAV lab, the proposed control architectures will be implemented and validated first by hardware-in-the-loop simulations and then by flight tests. The feedback will allow us to take a fresh look at our design and validation methodologies, and to further improve them.

Keywords : ROBUSTNESS ANALYSIS ; ROBUST CONTROL ; ANTI-WINDUP DESIGN ; LFT MODELING ; AEROSPACE APPLICATIONS

Synthèse et validation de lois de commande avancées - Un ensemble de méthodes et d'outils pour faire le lien entre théorie et pratique.

De nombreuses applications dans le monde réel souffrent de fortes contraintes de coût ou de poids, qui dictent l'utilisation de du matériel potentiellement peu performant (actionneurs, capteurs, ordinateurs de bord). Par exemple, l'accès à l'espace devient nettement plus facile et moins cher avec l'avènement des nanosats, et les lanceurs à bas prix doivent être de ce nouveau modèle économique. Dans le domaine militaire, l'une des questions les plus d'actualité est d'améliorer la précision et la portée des obus d'artillerie traditionnels sur le champ de bataille moderne, avec une réduction significative des coûts. Et les drones sont maintenant largement utilisés pour effectuer un grand nombre de missions exigeantes. Il y a donc un réel besoin de contrôler différents types de systèmes sur des domaines opérationnels étendus en présence de limitations matérielles, d'incertitudes, de paramètres variants ainsi que de saturations d'actionneurs. Dans ce contexte, notre travail se situe à la frontière entre la recherche et l'ingénierie. D'après des théories et des résultats connus dans les domaines de la modélisation LFR, de l'analyse μ/IQC/Lyapunov, ainsi que de l'analyse des systèmes saturés, nous essayons d'abord d'élaborer des méthodes de validation qui peuvent être appliquées à des problèmes du monde réel avec un temps de calcul raisonnable. L'heuristique remplace parfois des preuves mathématiques rigoureuses, mais nous pensons que c'est le prix à payer pour faire le lien entre la théorie et la pratique. De nombreux efforts sont également investis dans le développement d'outils de calcul génériques conçus pour les ingénieurs automaticiens. Ensuite, nous montrons comment ces méthodes et outils peuvent être directement intégrés dans le processus de conception des lois de commande, de manière à réduire le plus possible le nombre d'itérations entre la conception et la validation dans un contexte industriel. La prochaine étape consistera à faire face aux contraintes de mise en œuvre. Grâce à notre implication croissante dans le labo drones de l'ONERA, les architectures de contrôle proposées seront mises en œuvre et validées d'abord par des simulations Hardware-in-the-Loop, puis par des essais en vol. Le retour d'expérience nous permettra de jeter un regard neuf sur nos méthodologies de conception et de validation, et ddes améliorer encore davantage.

Mots-clés : ANALYSE DE ROBUSTESSE ; COMMANDE ROBUSTE ; SYNTHESE ANTI-WINDUP ; MODELISATION LFT ; APPLICATIONS AEROSPATIALES

Office National d'Études et de Recherches Aérospatiales
BP 72 - 29 avenue de la Division Leclerc
92322 CHATILLON CEDEX
Tél. : +33 1 46 73 40 40 - Fax : +33 1 46 73 41 41
http://www.onera.fr