On randomized algorithms and their applications in robust optimization

Algoritmos aleatorios y aplicaciones en optimización robusta

Amalia Luque Sendra

Thesis submitted for
the Degree of Doctor of Philosophy

Dpto. de Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Seville, May 2014

On randomized algorithms and their applications in robust optimization

Algoritmos aleatorios y aplicaciones en optimización robusta

Author:
Amalia Luque Sendra

Supervisors:
Teodoro Álamo Cantarero
Daniel Rodríguez Ramírez
Eduardo Fernández Camacho

Seville, May 2014

Genius is one percent inspiration and ninety-nine percent perspiration.

Thomas A. Edison.

To my family.

To Teo, Dani and Eduardo

Abstract

In this work we study the sample complexity of probabilistic methods for control of uncertain systems. In particular, we show the role of the binomial distribution for some problems involving analysis and design of robust controllers with finite families. We also address the particular case in which the design problem can be formulated as an uncertain convex optimization problem. The results provide simple explicit sample bounds to guarantee that the obtained solutions meet some pre-specified probabilistic specifications.

We also present a randomized strategy for design under uncertainty. The main contribution is to provide a general class of sequential algorithms which satisfy the required specifications using probabilistic validation. At each iteration of the sequential algorithm, a candidate solution is probabilistically validated by means of a set of randomly generated uncertainty samples.

The idea of validation sets has been used in some randomized algorithms when a given candidate solution is classified as probabilistic solution when it satisfies all the constraints on the validation set. In this thesis, we show the limitations of this strategy and present a more general setting where the candidate solution may violate the specifications for a reduced number of elements of the validation set. This generalized scheme exhibits some advantages, in particular in terms of obtaining a probabilistic solution.

We propose a randomized algorithm for feasibility of uncertain LMIs. The algorithm is based on the solution of a sequence of semidefinite optimization problems involving a reduced number of constraints. A bound of the maximum number of iterations required by the algorithm is given. Analogies and differences with the gradient and localization methods are discussed. Finally, the performance and behaviour of the algorithm are illustrated by means of a numerical example.

We present here a general strategy for the design of a fault-detection block with probabilistic validation (PCV- Processing, Classification, Validation). A general scheme of PCV is proposed, that allows to design a fault detection block with probabilistic validation in the maximum percentage of non detected faults (set as design condition) and in the percentage of false alarms (obtained a posteriori). In each iteration of the sequential algorithm, a candidate solution is probabilistically validated by a set of samples randomly generated. A general framework is presented in which the candidate solution can violate the constraints for a limited number of elements of the validation set. This generalized scheme shows significant advantages, in particular in terms of the obtention of the probabilistic solution.

We illustrate some of these methodologies by illustrative examples, such as the application to feedback controllers or frequential identification.

Acknowledgements

During the last years in which I have been working on my PhD Thesis many people were kind enough to share these experiences with me, both the positive and the negative, which I will be forever grateful for.

Foremost, I would like to thank my parents, Joaquín and Amalia. They raised me with love and care, and taught me values such as how to be responsible and work hard. I hope that they will enjoy seeing me successfully complete my PhD. I would like them to know that my successes are a testament to them and are directly a fruit of their labor. I would also like to thank my husband Óscar, who has stood by me with love and support since many years, and without whose trust and patience this work would not have been possible . I extend my gratitude to all of my family, especially to my daughter Amalia and son Óscar, who are my deepest motivation. A grateful mention for my brother and sister, and for my brothers in law, who are more than good partners; and a friendly word for my mother in law. On behalf of my parents and myself my warmest thanks go to my grandparents Paco, Maruja, Joaquín, who have unfortunately passed away, and Amalia.

It is also right that I thank my advisors Teodoro Álamo, Daniel Rodríguez and Eduardo F. Camacho. They have supported me from the beginning, despite my clumsiness, with infinite patience and always shown their best intentions towards me. They guided me the whole way, always contributing with insightful discussions and suggestions.

I would like to thank the Grupo de Control Predictivo, with a special mention to Carlos Bordons, Daniel Limón, David Muñoz, Manuel Ruíz, Antonio Gallego and Pepe Maestre, whose cover page was an inspiration for mine. I also want to express my gratitude to all of the professors at the Departamento de Ingeniería de Sistemas y Automática. I would like to thank all the PhD students in our department, especially my friends Pablo, Rachad, Mirko, Jorn, Isabel, Alicia and Pepe. An special thank to Pablo, whose Thesis structure has been of great help while building mine.

Moreover, I would like to thank to Ministry of Education and Science for the doctorate grant (FPU) and the DPI projects that supported me during the development of my PhD (DPI2007-66718-C04-01, DPI2008-05818, DPI2010-215589-C05-01, DPI-2011-26243). Be-

sides, I thank the European Commission for founding part of my work through FP7-223866 and Junta de Castilla y Leon under Grant VA037A08

I was fortunate enough to undertake some of the work for this thesis in Torino, where I spent three months at Politecnico di Torino. I was supervised by professors Roberto Tempo and Fabrizio Dabbene and I would like to express my gratitude to them. I am grateful for all the wonderful people I met in Italy. I am grateful to all the people that gave me that opportunity.

Last but not least, I would like to thank all the people who, with their comments and enriching discussions, have helped to make this work possible and to improve it. My gratitude to Vincent Puig, Joaquín Blesa, Sebastián Ponce, Enrique Molleja, Teresa Álvarez, Fernando Tadeo and Alfredo Pérez.

Amalia Luque Sendra
Sevilla, May 2014.

Contents

Chapter 1

Introduction

1.1 State of the art

1.1.1 Complexity of robust design problems in control

The first critical issue of the classical robustness paradigm is computational complexity. In particular, various control problems have been shown to fall into the category of the so-called ”intractable” problems, which are practically unsolvable if the number of variables becomes sufficiently large. These problems are generally denoted as NP-hard.

In a typical instance of a computational problem, we are given input data $x_1, x_2, \ldots, x_d$ and we are asked to compute some function of them. For example, in the determinant problem, an instance consists of the entries of a given matrix and the desired outcome is its determinant. We focus on digital computation and constrain the input data to be given with a finite number of bits. For example, we will typically assume that the inputs take integer or rational values. As far as negative complexity results are concerned, this is hardly restrictive. If a function is difficult or impossible to evaluate on integer or rational inputs, then it certainly remains so with arbitrary real inputs. Different instances of the same problem can have different sizes. The size of an instance is defined as the number of bits used to encode the input data according to a certain pre-specified format. In particular, any (nonzero) integer i can be viewed as having size (bit length) approximately equal to $\log |i|$, since this is roughly the number of bits in a binary representation of i.

Loosely speaking, an algorithm is described by a program, i.e., a finite sequence of instructions of the type encountered in common programming languages. For a more precise description, we need to specify a model of computation. A rather simple such model is provided

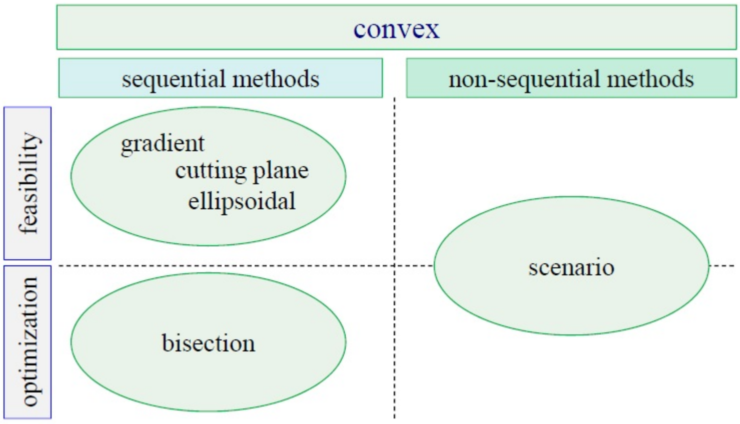


Figure 1.1: Big Picture

by a random access machine (RAM) ((Aho, Hopcrof and Ullman, 1974), (Papadimitriou, 1994)).

A RAM consists of a read-only tape that contains the input data $x_1, x_2, \ldots, x_d$, an output tape on which the outcome of the computation is written, an unlimited sequence of registers used to store intermediate quantities generated in the course of the computation, and a program. Each register and each memory location on the input and output tapes may contain an arbitrary, possibly negative, integer. The program is a sequence of instructions, some of which may be associated with labels that are used in ”jump to” instructions. The instruction set can be assumed to contain the halting instruction, the four arithmetic operations, instructions for reading from a location on the input tape into a register (respectively, writing the contents of a register to a location on the output tape), indirect addressing (e.g., read the contents of the register whose address is stored in register i, and store them in register 1), and jump instructions that move the program to a next instruction, possibly depending on the outcome of the comparison of some register to zero. The exact nature of the allowed instructions is not important, because alternative choices of the instruction set lead to computing machines with equivalent computing capabilities, as long as all critical abilities, such as jumping, are present. In a typical execution of the algorithm (or ”computation”), the input data are loaded on the input tape, and the algorithm starts carrying out the program instructions. For any given input, the computation may or may not halt (that is, reach a halting instruction). The algorithm is said to solve a particular problem if it always halts (for every instance of the problem, that is, for every choice of the input data) and produces the correct answer on the output tape. We also say that a problem is unsolvable if there exists no algorithm (under our model of computation) that will always halt with the correct answer. One may wonder whether the class of solvable problems depends on the choice of a model of computation. According to the so-called Church-Turing thesis (Copeland and Jack, 2008), all reasonable models of digital computation lead to the same class of solvable problems, and are therefore equally powerful. This thesis is supported by the fact that all reasonable models that have been proposed and studied lead indeed to the same class of solvable problems. Still, it only remains a thesis - rather than a theorem - because we do not have a precise definition of ”all reasonable models” (Blondel and Tsitsiklis, 2000).

1.1.1.1 Decidable and undecidable problems

An undecidable problem is a decision problem for which it is impossible to construct a single algorithm that always leads to a correct yes-or-no answer.

A decision problem is any arbitrary yes-or-no question on an infinite set of inputs. Because of this, it is traditional to define the decision problem equivalently as the set of inputs for which the problem returns yes. These inputs can be natural numbers, but also other values of some other kind, such as strings of a formal language. Using some encoding, such as a G*ö*del numbering, the strings can be encoded as natural numbers. Thus, a decision problem informally phrased in terms of a formal language is also equivalent to a set of natural numbers. To keep the formal definition simple, it is phrased in terms of subsets of the natural numbers.

Formally, a decision problem is a subset of the natural numbers. The corresponding informal problem is that of deciding whether a given number is in the set. A decision problem A is called decidable or effectively solvable if A is a recursive set. A problem is called partially decidable, semi-decidable, solvable, or provable if A is a recursively enumerable set. This means that there exists an algorithm that halts eventually when the answer is yes but may run for ever if the answer is no. Partially decidable problems and any other problems that are not decidable are called undecidable.

1.1.1.2 Time complexity

Assuming that an algorithm halts, its running time is defined as the sum of the ”costs” of each instruction. In the so-called random access machine (RAM) model, each arithmetic operation involves a single instruction which is assumed to have a unit cost. More realistically, in the bit model, the cost of an arithmetic is given by the sum of the bit length of the integers involved in the operation, i.e. the running time of the algorithms depends on the size of the problem instance.

Since the running time may be different for different instances of the same size, the running time T(n) is defined as the worst-case running time over all instances of size n.

Formally, we say that an algorithm runs in polynomial time is there exists an integer k such that in the worst-case running time is:

$$T(n) = O(n^k).$$

P is defined as the class of decision problems having polynomial-time algorithms. In practice, the class P consists on all the problems that are efficiently solvable. Notice that the notion of time complexity is associated with a specific algorithm and not with the problem itself. For the same problems, algorithms with different complexity may be derived.

An alternative definition of polynomial-time algorithms is related to the notion of ”average” running time. An example in this direction is the simplex method for solving linear programming problems. An example of a problem which is solvable in polynomial time is stability of a continuous-time system (Tempo, Calafiore and Dabbene, 2013).

1.1.1.3 NP- Completeness and NP-hardness

Non deterministic polynomial optimization problems (NPOP) are intended to provide a basis for a natural generalization of the theory of NY sets. Intuitively an NPOP is a set whose elements are encoded according to some reasonable scheme. Each element in the set has a set of nonnegative integers associated with ill assumed to represent a certain combinatorial property of the element, a property we are interested in. The elements of the set or rather their encodings are assumed to be polynomially recognizable and the sets of numbers associated with those elements are assumed to be: computable by a nondeterministic Turing machine in polynomial time (Paz, 1981).

NP-hard (Non-deterministic Polynomial-time hard) is a class of problems that are, informally, ”at least as hard as the hardest problems in NP”. A problem H is NP-hard if and only if there is an NP-complete problem L that is polynomial time Turing-reducible to H. In other words, L can be solved in polynomial time by an oracle machine with an oracle for H. Informally, we can think of an algorithm that can call such an oracle machine as a subroutine for solving H, and solves L in polynomial time, if the subroutine call takes only one step to compute. NP-hard problems may be of any type: decision problems, search problems, or optimization problems.

As consequences of definition, we have (note that these are claims, not definitions):

- Problem H is at least as hard as L, because H can be used to solve L.

  Since L is NP-complete, and hence the hardest in class NP, also problem H is at least as hard as NP, but H does not have to be in NP and hence does not have to be a decision problem (even if it is a decision problem, it need not be in NP)

Since NP-complete problems transform to each other by polynomial-time many-one reduction (also called polynomial transformation), all NP-complete problems can be solved in polynomial time by a reduction to H, thus all problems in NP reduce to H; note, however, that this involves combining two different transformations: from NP-complete decision problems to NP-complete problem L by polynomial transformation, and from L to H by polynomial Turing reduction.

- If there is a polynomial algorithm for any NP-hard problem, then there are polynomial algorithms for all problems in NP, and hence P= NP
- If $P \neq NP$, then NP-hard problems cannot be solved in polynomial time, while P = NP does not resolve whether the NP-hard problems can be solved in polynomial time.
- If an optimization problem H has an NP-complete decision version L, then H is NP-hard.

A common mistake is to think that the NP in NP-hard stands for non-polynomial. Although it is widely suspected that there are no polynomial-time algorithms for NP-hard problems, this has never been proven. Moreover, the class NP also contains all problems which can be solved in polynomial time.

1.1.2 Deterministic approaches

In the early 1970s, a set-theoretic description of a plant family, often called unknown-but-bounded model (Schweppe, 1973), emerged as a novel paradigm for estimation and Kalman filtering. A few years later, researchers realized some drawbacks of optimal control, such as the lack of guaranteed margins of linear quadratic Gaussian (LQG) (Doyle, 1978). Subsequently, in the early 1980s, a successful alternative to the existing classical approach for control has been developed. In the new setting, the design objective is to determine feedback controllers that are guaranteed against all possible uncertainty realizations, i.e. worst-case (or robust) controllers, see (Safonov, 2012) for an historical account on the history of robust control. In other words, a controller is designed with the aim of guaranteeing a specified performance for all plants that are compatible with the uncertainty description. (Le, Stoica, Álamo, Camacho and Dumur, October 2013)

A major stepping stone in the robustness direction was the formulation in 1981 by Zames of the H_∞ problem (Zames, 1981). In the subsequent fifteen years, the research in robust control evolved in various directions, each based on diverse problem formulations and mathematical tools. Even tough several subareas played a major role within robustness, we fell

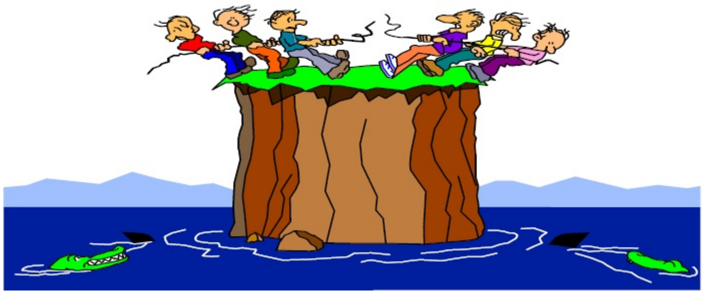


Figure 1.2: Probabilistic Robustness

that H_∞ deserves the credit for its centrality and also for its connections with classical optimal control. However, other successful methods to handle uncertainty have been developed. In particular, we recall the methods based on the structured singular value, also known as μ theory (Packard and Doyle, 1993), the approach dealing with systems affected by parametric uncertainty, or Kharitonov theory ((Barmish, 1994), (Bhattacharyya, Chapellat and Keel, 1995)), the optimization-based methods based on linear matrix inequalities (Boyd, Ghaoui, Feron and Balakrishnan, 1994), the l_1 optimal control theory (Daleh and Díaz Bobillo, 1995) and the so-called quantitative feedback theory (QFT) ((Horowitz, 1991),(Houpis and Rasmussen, 1999)).

In the late 1980s, robust control became a well-known discipline (Athans, 1971), (Morari and Zafiriou, 1989) so that the technical results and the algorithms developed were successfully used in various industrial applications, including aerospace, chemical, electrical and mechanical engineering. Moreover, the impact of robust control theory has begun to spread to other fields than engineering, such as economics (Hansen and Sargent, 2008).

A few years later, in the early 1990s, researchers in robust control realized more fully some of its theoretical limitations, which can be roughly summarized as the issues of conservatism and computational complexity. In fact, when compared with classical stochastic methods, the worst-case paradigm may lead to problems whose exact solution cannot be determined in polynomial time, see (Blondel and Tsitsiklis, 2000). Therefore, relaxation techniques are typically introduced so that the resulting problem can be solved with numerically efficient algorithms (Bravo, Álamo, Redondo and Camacho, 2008). Clearly, this entails a compromise between tightness of the solution and numerical complexity (Tempo et al., 2013), (Álamo, Fiacchini, Cepeda, Limón and Camacho, 2007), (Camacho, Álamo, Limón, Bravo, Ramírez, noz and Ruiz Arahal, 2005), (Bravo, Álamo and Camacho, 2004).

Robust linear matrix inequalities play an important role with rational dependence on uncertainties in robust control (Bravo, Limón, Álamo and Camacho, 2005). Various classical relaxations based on the S-procedure can be subsumed to a unified framework. Based on Lagrange duality for semi-definite programs, under some conditions, such relaxations can be verified to be exact. The systematic construction of families of relaxations can be shown to be asymptotically exact, based on recent results on the sum-of-squares representation of polynomial matrices (Scherer, 2006).

1.1.3 Randomized approaches to analysis and design of control systems

In recent years, research on probabilistic analysis and design methods for systems and control has significantly progressed. Specific areas where we have seen convincing developments include uncertain and hybrid systems (Tempo, Calafiore and Dabbene, 2005),(Vidyasagar, 1997). A key technical ingredient of this approach is the use of the theory of rare events and large deviation inequalities which suitably bound the tail of the probability distribution. These inequalities are crucial in the area of Statistical Learning Theory (Vapnik, 1998; Vidyasagar, 1997). The use of this theory for feedback design of uncertain systems has been initiated in (Vidyasagar, 1997). Recently, significant improvements regarding the sample complexity have been provided in (Álamo, Tempo and Camacho, 2009). For the special case of convex optimization problems, the scenario approach has been introduced in (Calafiore and Campi, 2006) for probabilistic controller design.

The utility of randomized algorithms stems from the fact they can circumvent the complexity of nonconvex design problems. A sequence or other collection of random variables is independent and identically distributed (i.i.d.) if each random variable has the same probability distribution as the others and all are mutually independent. In this setting, one can draw N i.i.d. samples $\{w^{(1)},\ldots,w^{(N)}\}$ from $\mathscr{W}$ according to probability $\mathrm{Pr}_{\mathscr{W}}$ and solve the sampled optimization problem.

Since obtaining a global solution to the previous problem is a difficult task in the general case, we analyze in this work the probabilistic properties of any suboptimal feasible solution. If one allows at most m violations of the N constraints.The idea of allowing some violations of the constraints is not new and can be found, for example, in the context of identification (Bai, Cho, Tempo and Ye, 2002). The randomized strategies corresponding to problems (2.2) and (2.3) have been recently studied in (Álamo et al., 2009); see also (Tempo et al., 2005; Vidyasagar, 1997).

We remark that the probability of failure is slightly different from the probability of one-sided constrained failure introduced in (Álamo et al., 2009).

1.1.4 Randomized validation schemes

The design in the presence of uncertainty is of paramount relevance in different fields. Unfortunately, the related semi-infinite optimization problems often exhibit an NP-hard nature that seriously compromises their solution in a reasonable computational time (Blondel and Tsitsiklis, 2000). There exist two ways to circumvent this NP-hard issue. One option consists in resorting to deterministic relaxations of the original problem which are normally solved in polynomial time but which might lead to overly conservative solutions (Scherer, 2006). An alternative paradigm is to assume that the plant uncertainty is probabilistically described so that a randomized algorithm may be derived to obtain, normally in polynomial time, a solution with some given properties normally stated in terms of the probability of error (Tempo et al., 2005), (Vidyasagar, 1997).

The field of randomized algorithms have evolved significantly in the last years. A recent survey on this topic can be found in (Calafiore, Dabbene and Tempo, 2011). Two complementary approaches, non-sequential and sequential, have been proposed. A classical approach for non-sequential methods is based upon statistical learning theory (Vapnik, 1998). In particular, the use of this theory for feedback design of uncertain systems has been initiated in (Vidyasagar, 1997); subsequent work along this direction include (Koltchinskii, Abdallah, Ariola, Dorato and Panchenko, 2000), (Vidyasagar, 2001), (Vidyasagar and Blondel, 2001), (Álamo et al., 2009). In (Álamo, Tempo and Luque, 2010*a*) and (Luedtke and Ahmed, 2008) the particular case in which the design parameter set has finite cardinality is analyzed.

The advantage of these methods is that the problem under attention may be non-convex. For convex optimization problems, a successful non-sequential paradigm, denoted as the scenario approach, has been introduced in (Calafiore and Campi, 2005) and (Calafiore and Campi, 2006). See also (Campi and Garatti, 2008), (Campi and Garatti, 2011), (Calafiore, 2010) and (Álamo et al., 2010*a*) for related results.

In non-sequential methods the original robust control problem is reformulated in terms of a single optimization problem with sampled constraints which are randomly generated. A relevant feature of these approaches is that they do not require any validation step. The number of samples required to guarantee that the obtained solutions meet some probabilistic specifications should take into account the specific nature of the problem under consideration. The main result of this line of research is to derive explicit lower bounds to this required sample size. Recently, improvements regarding this sample complexity have been provided in (Álamo et al., 2009). However, the obtained explicit sample bounds can be overly conservative because they rely on a worst-case analysis and grow (at least linearly) with the number of decision variables.

For sequential methods, the resulting iterative algorithms are based on stochastic gradient (Calafiore and Polyak, 2001), (Polyak and Tempo, 2001); ellipsoid iterations (Kanev, Schutter and Verhaegen, 2003), (Oishi, 2007); or analytic center cutting plane methods (Calafiore

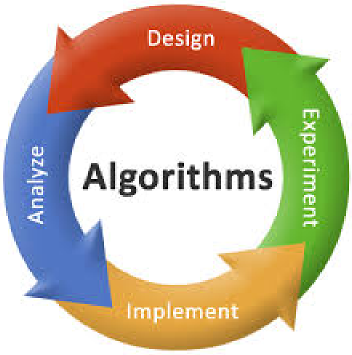


Figure 1.3: Algorithms

and Dabbene, 2007); see also (Álamo, Tempo, Ramírez and Camacho, 2007) for other classes of sequential algorithms. Convergence properties in finite-time are in fact one of the focal points of these papers. Various control problems have been solved using these sequential randomized algorithms, including robust LQ regulators, switched systems, and uncertain linear matrix inequalities (LMIs). Sequential methods are mostly used for uncertain convex problems because the computational effort at each iteration is affordable. However, they can be applied in principle to any kind of robust design problem. For example, a sequential algorithm that can be applied to a rather general class of problems is presented in (Álamo et al., 2009).

The main point in common of all these sequential algorithms is the use of the validation strategy presented in (Oishi, 2003) (see (Oishi, 2007) for a journal version). The candidate solutions provided at each iteration of these algorithms are validated using a validation set which is drawn according to the probability measure defined in the uncertain set. If the candidate solution satisfies the design specifications for every element of this validation set then it is classified as probabilistic solution and the algorithm terminates. The main point in this validation scheme is that the cardinality of the validation set increases with each iteration of the algorithm. The strategy guarantees that if a probabilistic solution is obtained, then it meets some probabilistic specifications. A similar approach, introduced in (Dabbene, Shcherbakov and Polyak, 2010), has been presented in (Calafiore et al., 2011) in the context of sequential algorithms. The contribution is a reduction on the cardinality required for the validation sets.

The main contribution of this work is to propose a relaxed validation scheme in which we allow the candidate solution to violate the design specifications for one or more of the members of the validation set. The idea of allowing some violations of the constraints is not new and can be found, for example, in the context of identification (Bai et al., 2002), chance-constrained optimization (Campi and Garatti, 2011) and statistical learning theory (Álamo et al., 2009). This scheme makes sense in the presence of soft constraints or when it is not possible to find a solution satisfying the specifications for all the admissible realizations of uncertainty.

As it will be shown later, this relaxed scheme allows us to reduce, in some cases dramatically, the number of iterations required by the sequential algorithm. Another advantage of the proposed approach is that it does not rely on the existence of a robust deterministic solution. Furthermore, the presented strategy is quite general and is not based on a convexity assumption.

1.1.5 Sequentially optimal R.A. for robust LMI feasibility problems

The use of randomized algorithms (see (Tempo et al., 2005)) has attracted the attention of the control community in the last few years. One of the reasons for this widespread interest is that randomization can be used to circumvent the NP-hard nature of a large number of robust control problems (Nemirovskii, 1993),(Poljak and Rohn, 1993). Randomization allows one to obtain a solution that satisfies the constraints of a given robustness problem for most of the possible realizations of the uncertainty. This concept of approximate feasibility has been introduced in the context of robust control in (Barmish and Scherbakov, 2002). Under relatively mild assumptions, the randomized methods are able to compute (in polynomial time) an approximate solution to a robust problem. The measure of the set of original constraints that are violated by the approximate solution can be made smaller than any pre-specified quantity.

The randomized gradient approach presented in (Polyak and Tempo, 2001),(Calafiore and Polyak, 2001),(Fujisaki, Dabbene and Tempo, 2003) and (Liberzon and Tempo, 2004) finds a solution to a robust problem involving linear matrix inequalities in a finite number of iterations with probability one, if a strong feasibility condition holds. These gradients algorithms are based on an iterative scheme where the current solution is updated towards a descent direction obtained by a random gradient of a suitable feasibility violation function.

Another important class or randomized methods are based on probabilistic versions of standard localization methods. These localization methods have better theoretical convergence properties than the gradient ones. Among them one finds the probabilistic ellipsoid method (Kanev et al., 2003),(Oishi, 2003) and the probabilistic analytic center cutting plane method (Calafiore and Dabbene, 2006).

In the context of robust optimization, the scenario approach also plays a relevant role. It is shown in (Calafiore and Campi, 2005) and (Calafiore and Campi, 2006) that by appropriate sampling of the constraints one obtains a standard convex optimization problem (the scenario) whose solution is approximately feasible for the original (usually infinite) set of constraints, i.e., the measure of the set of original constraints that are violated by the scenario solution decreases to zero as the number of samples is increased.

In some sense, the scenario approach, the gradient and the ellipsoid methods have a very different nature. The scenario approach obtains an approximate solution to a robust optimization problem solving a simple optimization problem with a large number of constraints. On the other hand, the gradient method and the ellipsoid algorithm obtain an approximate solution to a robust feasibility problem in a sequential way, by means of a considerable number of iterations in which the candidate solution is updated by means of a simple rule so that no optimization is really required.

In this work we present a randomized algorithm that addresses the problem of obtaining a feasible robust solution to a possibly uncountable number of linear matrix inequalities. The presented algorithm does not belong to any of the aforementioned strategies. The algorithm requires a finite number of iterations to converge. It addresses the robust feasibility problem by means of the solution of a sequence of relatively simple optimization problems. The proposed algorithm, as the gradient method and the ellipsoid algorithm, has a sequential nature. However, instead of using a simple updating rule, each candidate solution is updated by means of an optimization problem involving a reduced number of constraints obtained from the original constraints of the problem. One of the advantages of the proposed algorithm is that it is capable of determining the non feasibility of a given robust feasibility problem. Our numerical experience shows that the algorithm performs satisfactorily: in an affordable number of iterations it obtains an (approximately) feasible solution in case of feasibility, or it detects that the problem is not feasible.

1.1.6 Applications of randomized algorithms

Probabilistic design methods and randomized algorithms have been developed for several applications related to systems and control. We present a review (Tempo et al., 2013) of some of the main areas where these methods have been successfully used. For extensive information on applications on computer science, computational geometry and optimization, see (Mitzenmacher and Upfal, 2005), (Motwani and Raghavan, 1995), (Mulmuley, 1994).

1.1.6.1 Fault detection

A fault is defined as any change in the behavior of some of the system components (not allowed deviation of the one of characteristic parameters or properties) so that it and can not fulfill the function for which it was designed (Blanke, 1999). Besides faults, there exist other factors that alter the normal behavior of the system, as disturbances and noise. Disturbances are non known entries that can occur in the system at any time but they have been taken into account when designing the conventional control loop. Any disturbance which had not been taken into account in this design will be considered a fault. Noise is also a non known input manifested in the system but, unlike disturbance, has zero mean, and it is possible to have a priori knowledge of which is its amplitude. A fault detection system must react to faults and be immune (robust), so far as possible, to the other factors in the system that create uncertainty. Furthermore, many of the fault detection methods are based on a model (mathematical or quantitative) of the monitored system which can never accurately describe the real behavior of the system and therefore it will present a modeling error that must be considered.

The goal of a fault detection block is, once a fault has occurred in an instant T_F, to detect it in a time range less or equal than $T_{D_{max}}$, set in advance. Depending on the magnitude and incidence of the faults desired to be detected and the possible presence of other uncertainty factors in the system, it may not be possible to design a detection block detecting all faults without false alarms in situations in which there are no faults. So there is always a compromise between the proportion of undetected faults (*MF* ”Missed Faults”) and the proportion of times that the detection block is activated without the presence of faults due to the uncertainties present in the system (*FA* ”False alarms”). This compromise, that should be taken into account in the design process of the detector block, it is logical to prioritize minimizing not detected faults respect to the minimization of false alarms.

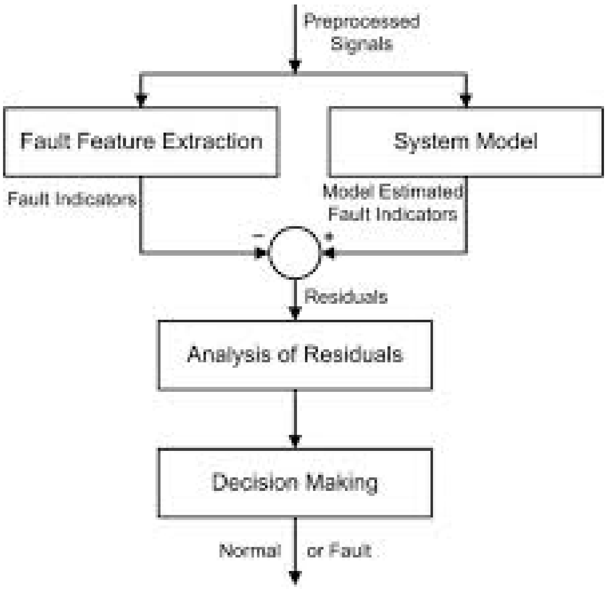


Figure 1.4: Fault Detection

The random nature of faults and uncertainties inherent in the system makes the design problem of the detection block a robustness problem.

Typically, for a robustness problem, the design parameters, and different auxiliary variables, are described in terms of a vector of decision variables θ, denoted as ”design parameter” and is restricted to the set Θ. Moreover, the uncertainty w is bounded on the set $\mathscr{W}$. That is, each element $w \in \mathscr{W}$ represents one of the admissible realizations of uncertainty, with probability $\Pr_{\mathscr{W}}$. In our context of fault detection, θ corresponds to the decision variables that determine the fault detection block. This block allows us to determine if there is a fault or not in a given scenario, so there will be two uncertainty sets $\mathscr{W}_F$ and $\mathscr{W}_N$ consisting in all possible scenarios of the system to be monitored operation, with faults and without faults respectively. Furthermore, w_F and w_N represent a realization of a scenario with and without fault. $\mathscr{W}_F$ and $\mathscr{W}_N$ have spaces probability Pr_F and Pr_N respectively.

We also consider two measurable binary functions:

$$g(\theta, w) := \begin{cases} 0 & \text{if } \theta \text{ detects a fault} \\ 1 & \text{in other case.} \end{cases}$$

$$h(\theta, w) := \begin{cases} 0 & \text{if } \theta \text{ doesn't detect a fault} \\ 1 & \text{in other case.} \end{cases}$$

Applying these two functions on the spaces $\mathscr{W}_F$ and $\mathscr{W}_N$ we obtain the following expected values:

$$\begin{aligned} E_g(\theta) &:= \text{Pr}_F\{w_F \in \mathscr{W}_F : g(\theta, w_F) = 1\} \\ E_h(\theta) &:= \text{Pr}_N\{w_N \in \mathscr{W}_N : h(\theta, w_N) = 1\}. \end{aligned}$$

Where $E_g(\theta)$ and $E_h(\theta)$ are the proportion of undetected faults (MF) and false alarms (FA) respectively.

The utility of randomized algorithms arises when being able to treat the following design problem

$$\min_{\theta \in \Theta} E_h(\theta) \text{ subject to } E_g(\theta) \leq \eta_F \tag{1.1}$$

where η_F is the maximum proportion of undetected faults imposed as a constraint of the detector block.

In this context, we can extract N_N and N_F i.i.d. samples (independent and identically distributed) $\{w_N^{(1)}, \ldots, w_N^{(N_N)}\}$ of $\mathscr{W}_N$ and $\{w_F^{(1)}, \ldots, w_F^{(N_F)}\}$ of $\mathscr{W}_F$ according to the probability Pr_N and Pr_F respectively, and with a ratio between scenarios with fault and without fault $F_N = \frac{N_F}{N_N}$ determined by the probability of failure of the system to be monitored. This way can solve the following sampled optimization problem

$$\min_{\theta\in\Theta} \sum_{\ell_N=1}^{N_N} h(\theta, w_N^{(\ell_N)}) \tag{1.2}$$
$$\text{subject to} \sum_{\ell_F=1}^{N_F} g(\theta, w_F^{(\ell_F)}) \le \eta_F N_F$$

In this work we propose a design method of a fault detector block based on the use of historical or real simulations with and without faults, avoiding the difficulty of analysis, that is not always possible, due to the complexity of problem.

The result thus obtained, through a probabilistic validation test, guarantees that the proposed solution behaves the desired way with a certain probability, fixed a priori. It also guarantees the satisfaction of probabilistic constraints. This technique is well suited for addressing complex problems.

1.1.6.2 Feedback controllers testing

In computer networks congestion appears when there are too many sources sending data too fast for the network to handle. Techniques to reduce congestion are of great interest (Mesquine, Tadeo and Álvarez, 2011). This thesis concentrates on congestion control methodologies where feedback control techniques provide efficient solutions ((Jacobson, 1988), (S. Ryu and Qiao, 2004), (Hollot, Misra, Towsley and Gong, 2002), (Sun, Chan, Ko, Chen and Zukerman, 2007), (Floyd and Jacobson, 1993)).

A central problem in designing controllers for these systems is the difficulty of ensuring adequate performance in all possible conditions, as these systems operate under a very wide range of conditions, are inherently nonlinear and suffer from significant time-varying delays. Thus, designers frequently have to show the effectiveness of their proposal by extensive simulations, which is a time-consuming methodology, and does not offer a definite guarantee of performance: simulation results in most of the references show only specific cases and scenarios.

Prompted by this problem, we concentrate on the following issue: given a required degree of confidence, how many simulations are needed to check the adequate performance of the controllers? Thus, we develop a randomized approach based on some ideas in ((Su-Woon, Chang-Jin, Sin and Ho-Chan, 2012),(Álamo et al., 2009), (Álamo, Tempo and Luque, 2010*b*), (Fujisaki and Kozawa, 2006)), to test whether a controller robustly satisfies a set of specifications with a given probabilistic error margin.

The presented results are stated in an implicit way, that is, the number of experiments required is obtained from a simple numerical procedure. The theoretical framework proposed in (Álamo et al., 2009) and (Álamo et al., 2010*b*) is conveniently tailored for this particular application. Hence this work constitutes a proof of concept of the methodology proposed in the aforementioned references.

The main idea is to test the controller under a finite set of different scenarios. When the controller satisfies the specifications for a sufficient number of these scenarios, then certain properties can be concluded with a given degree of confidence, and no more simulations are needed. One of the main characteristics of the technique is that it is independent of the family of controllers (PI, PID, predictive, robust, etc). The methodology is applied in this chapter to the active queue management (AQM) scheme, which complements the end-to-end Transmission Control Protocol (TCP), at the routers' transport layer. The AQM objectives (S. Ryu and Qiao, 2004), (Hollot et al., 2002), (Sun et al., 2007), are efficient queue utilization, queuing delay and robustness.

Numerous AQM algorithms have been proposed (see (S. Ryu and Qiao, 2004) for a good survey on the subject), with Random Early Detection (RED) (Floyd and Jacobson, 1993) being the most widely used algorithm, as it can detect and respond to long-term traffic patterns. This chapter uses the AQM mathematical models published in (Hollot et al., 2002), and extensively used in the literature ((Jacobson, 1988), (S. Ryu and Qiao, 2004), (Hollot et al., 2002), (Floyd and Jacobson, 1993), (Vidyasagar, 2001)) and the references therein) for controller design and testing. The main metrics proposed to determine controller performance are: router queue size (real value and standard deviation), link utilization and the probability of packet losses (Álvarez and Martínez, 2013). As a demonstration, the proposed technique is applied to a problem of two routers connected in a Dumbbell topology, which represents a single bottleneck scenario.

The length of their queues is controlled with a PID ((Aström and Hägglund, 2006), (nez, Camarillo, Moreno-Valenzuela and Campa, 2011)) whose probabilistic properties are guaranteed following the results presented in the chapter. The simulations are done using the software ns-2, which is a discrete event simulator targeted at networking research, providing substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless networks. It must be pointed out that although the proposed methodology was prompted by a congestion control problem, and is demonstrated on this problem, it can be directly applied to other control testing problems, as plants to be controlled are frequently nonlinear, uncertain and subject to parameter variations ((nez et al., 2011)).

1.1.7 Thesis overview and contributions

The following is an outline of this thesis and its contributions:

- **Chapter 3**: Explicit Bounds for Required Number of Random Samples. In this chapter the sample complexity of probabilistic methods for control of uncertain systems is studied. The particular case in which the design problem can be formulated as an uncertain convex optimization problem is also addressed. Simple explicit sample bounds to guarantee that the obtained solutions meet some pre-specified probabilistic specifications are provided. Published partially in (Álamo et al., 2010*b*), (Álamo et al., 2010*a*).

- **Chapter 4**: Randomized Validation Schemes. A randomized strategy for design under uncertainty is presented. A general class of sequential algorithms which satisfy the required specifications using probabilistic validation is provided. At each iteration of the sequential algorithm, a candidate solution is probabilistically validated by means of a set of randomly generated uncertainty samples. Published partially in (Álamo, Luque, Ramírez and Tempo, 2012).

- **Chapter 5**: A sequentially optimal R.A. for robust LMI feasibility problems. A randomized algorithm for feasibility of uncertain LMIs is proposed. The algorithm is based on the solution of a sequence of semidefinite optimization problems involving a reduced number of constraints. A bound of the maximum number of iterations required by the algorithm is given. Analogies and differences with the gradient and localization methods are discussed. The performance and behaviour of the algorithm are illustrated by means of a numerical example. Published partially in (Álamo, Tempo, Ramírez, Luque and Camacho, 2013).

- **Chapter 6**: Fault detection with probabilistic validation. A general strategy for the design of a fault-detection block with probabilistic validation (PCV- Processing, Classification, Validation) is presented. A general scheme of PCV is proposed, that allows to design a fault detection block with probabilistic validation in the maximum percentage of non detected faults (set as design condition) and in the percentage of false alarms (obtained a posteriori). In each iteration of the sequential algorithm, a candidate solution is probabilistically validated by a set of samples randomly generated. A general framework is presented in which the candidate solution can violate the constraints for a limited number of elements of the validation set. This generalized scheme shows significant advantages, in particular in terms of the obtention of the probabilistic solution. Published partially in (Blesa, Luque, Álamo and Dabbene, 2013).

- **Chapter 7**: Application to frequential identification. Identification of a fuel cell by aplication of randomized algorithms. A Matlab trials tool is developed. With this tool cells are evaluated in laboratory and an optimal exciting signal for modeling the fuel cell is obtained. This way in the identification will become easier and quicker. The objective is to find the exciting signal to reach the best approximation to fuel cells model. Published partially in (Ponce, 2013).

- **Chapter 8**: A probabilistic approach for testing feedback controllers. A randomized approach to test whether a controller robustly satisfies a set of specifications with a given probabilistic error margin is presented. The results are stated in an implicit way, that is, the number of experiments required is obtained from a simple numerical procedure. This chapter constitutes a proof of concept of the methodology proposed in the previous chapters. The main idea is to test the controller under a finite set of different scenarios. When the controller satisfies the specifications for a sufficient number of these scenarios, then certain properties can be concluded with a given degree of confidence, and no more simulations are needed. Published partially in (Maestre, Álvarez, Álamo, Salim and Luque, 2012).

1.1.8 Publications

The following articles have been issued or submitted for publication during the elaboration of this thesis:

BOOK CHAPTERS:

1. ”On the sample complexity of probabilistic analysis and design methods” T. Álamo, R. Tempo, A. Luque. Perspectives in mathematical system theory, control and signal processing. Lecture notes in control and information series 398. Springer. USA. 2010.

JOURNALS

1. A probabilistic approach for testing feedback controllers, with application to congestion control. Jose M. Maestre, Teresa Álvarez, Teodoro Álamo, Anuar Salim and Amalia Luque. Technical Notes. International Journal of Control, Automation, and Systems.Volume 10, Number 4, August 2012.
2. Un algoritmo secuencial, aleatorio y óptimo para problemas de factibilidad robusta. T. Álamo, R. Tempo, D.R. Ramírez, A. Luque, E.F. Camacho. RIAI. 2013.
3. The Sample Complexity of Randomized Methods for Analysis and Design of Uncertain Systems. T. Alamo, R. Tempo b, A. Luque, D.R. Ramírez. Accepted for publication in Automática.

   Future work:

   - Fault detection using randomized algorithms. To be submitted to Automática.
   - Controllers validation. To be submitted to a Journal.
   - Paper about frequential identification

CONFERENCES

1. ”Dynamic model of the relationships between technology and employment.” A. Luque, A. Conseglieri, T. Álamo. Proceedings of the European Control Conference 2009. Budapest-Hungría. 23-26 August 2009. ISBN 978-963-311-369-1

2. ”Algoritmos aleatorios”. A. Luque, T. Álamo, R. Tempo. XXX Jornadas de Automática. Valladolid. 2-4 September 2009. ISBN 13-978-84-692-2387-1

3. ”Modelado de sistemas híbridos de energías renovables y su aplicación a una planta termosolar de agua caliente sanitaria (A.C.S.)” . A. Luque, A. Quintero, T. Álamo, D. Limón, M. R. Arahal, A. Conseglieri. XXX Jornadas de Automática. Valladolid. 2-4 September 2009. ISBN 13-978-84-692-2387-1.

4. ”Randomized Algorithms and their application to renewable energy systems”. A. Luque, T. Álamo. IFAC - Conference on control methodologies and technology for energy efficiency. March 2010.

5. ”Hybrid modeling of renewable energy systems and its application to a hot water solar plant”. A.Luque, A. Quintero, T. Álamo, D. Limón, M. R. Arahal, A. Conseglieri, E. F. Camacho. IFAC Conference on control methodologies and technology for energy efficiency. March 2010.

6. ”On the Sample Complexity of Randomized Approaches to the Analysis and Design under Uncertainty”. T. Álamo, R. Tempo, A. Luque. Proceedings of the 2010 American Control Conference (ACC10), ISBN: 978-1-4244-7425-7. June 2010.

7. ”Algoritmo para el diseño robusto de sistemas complejos”. A. Luque, T. Álamo. XXXI Jornadas de Automática. Jaén. 2-4 September 2010. ISBN: 978-84-693-0715-1

8. ”Modeling of a hybrid renewable/fossil hot water production system”. A. Luque T. Álamo M. R. Arahal, D. Limón. 2010 IEEE International Conference on emerging technologies and factory automation (ETFA’ 2010), ISBN: 978-1-4244-6849-2. September 2010.

9. T. Álamo, A. Luque, D. R. Ramírez and R. Tempo, ”Randomized Control Design through Probabilistic Validation,” Proc. of the American Control Conference, Montreal, Canada, June 2012.

10. Detección de fallos con validación probabilústica. Blesa, Joaquín ; Luque-Sendra, Amalia; Álamo-Cantarero, Teodoro; Dabbene, F. XXXIV Jornadas de Automática. 2013. Terrassa. Barcelona.

Chapter 2

Explicit bounds for sample complexity of random samples

2.1 A randomized approach to analysis and design of control systems

In this chapter, we study the sample complexity of probabilistic methods for control of uncertain systems. In particular, we show the role of the binomial distribution for some problems involving analysis and design of robust controllers with finite families. We also address the particular case in which the design problem can be formulated as an uncertain convex optimization problem. The results of the chapter provide simple explicit sample bounds to guarantee that the obtained solutions meet some pre-specified probabilistic specifications.

In recent years, research on probabilistic analysis and design methods for systems and control has significantly progressed. Specific areas where we have seen convincing developments include uncertain and hybrid systems (Tempo et al., 2005),(Vidyasagar, 1997). A key technical ingredient of this approach is the use of the theory of rare events and large deviation inequalities which suitably bound the tail of the probability distribution. These inequalities are crucial in the area of Statistical Learning Theory (Vapnik, 1998), (Vidyasagar, 1997). The use of this theory for feedback design of uncertain systems has been initiated in (Vidyasagar, 1997). Recently, significant improvements regarding the sample complexity have been provided in (Álamo et al., 2009). For the special case of convex optimization problems, the scenario approach has been introduced in (Calafiore and Campi, 2006) for probabilistic controller design. This chapter is a more complete version of (Álamo et al., 2010*b*) and contains the proofs of all technical results.

In this section we first introduce some preliminary notation and definitions as well as two randomized strategies. In Section 2.2 we provide bounds for the binomial distribution which are used in Section 2.3 to analyze the probabilistic properties of different schemes involving randomization. The chapter draws to a close in Section 2.5.

We assume that a probability measure $\mathrm{Pr}_{\mathcal{W}}$ over the sample space $\mathcal{W}$ is given. Given $\mathcal{W}$, a collection of N (i.i.d.) samples $\mathrm{w} = \{w^{(1)},\dots,w^{(N)}\}$ drawn from $\mathcal{W}$ is said to belong to the Cartesian product $\mathcal{W}^N = \mathcal{W} \times \cdots \times \mathcal{W}$ (N times). Moreover, if the collection w of N i.i.d. samples $\{w^{(1)},\dots,w^{(N)}\}$ is generated from $\mathcal{W}$ according to the probability measure $\mathrm{Pr}_{\mathcal{W}}$, then the *multisample* w is drawn according to the probability measure $\mathrm{Pr}_{\mathcal{W}^N}$. The scalars $\eta \in (0,1)$ and $\delta \in (0,1)$ denote probabilistic parameters. Furthermore, $\ln(\cdot)$ is the natural logarithm and e is the Euler number. For $x \in \mathbb{R}$, $x > 0$, $\lfloor x \rfloor$ denotes the largest integer smaller than or equal to x.

Typically, for a robustness problem, the design parameters, along with different auxiliary variables, are parameterized by means of a decision variable vector θ, which is denoted as "design parameter", and is restricted to a set Θ. On the other hand, the uncertainty w is bounded in the set $\mathcal{W}$. That is, each element $w \in \mathcal{W}$ represents one of the admissible uncertainty realizations. We also consider a binary measurable function $g : \Theta \times \mathcal{W} \to \{0,1\}$ and a real measurable function $f : \Theta \times \mathcal{W} \to \mathbb{R}$ which serve to formulate the specific design problem under attention. In a control context, the binary function $g : \Theta \times \mathcal{W} \to \{0,1\}$, is defined as

$$g(\theta,w) := \begin{cases} 0 & \text{if } \theta \text{ meets control specifications for } w \\ 1 & \text{otherwise.} \end{cases}$$

Given $\theta \in \Theta$, there might be a subset of the elements of $\mathcal{W}$ for which the constraint $g(\theta,w) = 0$ is not satisfied. This concept is rigorously formalized by means of the notion of "probability of violation", which is now introduced.

Definition 2.1 *[probability of violation] Consider a probability measure* $\mathrm{Pr}_{\mathcal{W}}$ *over* $\mathcal{W}$ *and let* $\theta \in \Theta$ *be given. The probability of violation of* θ *for the function* $g : \Theta \times \mathcal{W} \to \{0,1\}$ *is defined as*

$$E(\theta) := \mathrm{Pr}_{\mathcal{W}}\{\, w \in \mathcal{W} \;:\; g(\theta,w) = 1 \,\}.$$

Given $\theta \in \Theta$, it is generally difficult to obtain the exact value of the probability of violation $E(\theta)$ since this requires the solution of a multiple integral. However, we can approximate its value using the concept of empirical mean. For given $\theta \in \Theta$, and multisample $\mathrm{w} = \{w^{(1)},\dots,w^{(N)}\}$, drawn according to the probability measure $\mathrm{Pr}_{\mathcal{W}^N}$, the empirical mean of $g(\theta,w)$ with respect to w is defined as

$$\hat{E}(\theta,\mathrm{w}) := \frac{1}{N}\sum_{i=1}^{N} g(\theta,w^{(i)}).$$

Clearly, the empirical mean $\hat{E}(\theta,\mathrm{w})$ is a random variable. Since $g(\cdot,\cdot)$ is a binary function, $\hat{E}(\theta,\mathrm{w})$ is always within the closed interval $[0,1]$.

The utility of randomized algorithms stems from the fact they can circumvent the complexity of nonconvex design problems of the type

$$\min_{\theta\in\Theta} J(\theta) \text{ subject to } g(\theta,w)=0, \text{ for all } w\in\mathscr{W} \tag{2.1}$$

where $J:\Theta\to(-\infty,\infty)$ is a measurable function which normally represents the controller performance. In this setting, one can draw N i.i.d. samples $\{w^{(1)},\dots,w^{(N)}\}$ from $\mathscr{W}$ according to probability $\mathrm{Pr}_{\mathscr{W}}$ and solve the sampled optimization problem

$$\min_{\theta\in\Theta} J(\theta) \text{ subject to } g(\theta,w^{(\ell)})=0,\ \ell=1,\dots,N. \tag{2.2}$$

Since obtaining a global solution to the previous problem is a difficult task in the general case, we analyze in this chapter the probabilistic properties of any suboptimal feasible solution. If one allows at most m violations of the N constraints, the following sampled problem can be used to obtain a probabilistic relaxation to the original problem (2.1)

$$\min_{\theta\in\Theta} J(\theta) \text{ subject to } \sum_{\ell=1}^{N} g(\theta,w^{(\ell)})\le m. \tag{2.3}$$

The randomized strategies corresponding to problems (2.2) and (2.3) have been recently studied in (Álamo et al., 2009), see also (Tempo et al., 2005; Vidyasagar, 1997). In order to analyze the probabilistic properties of any feasible solution to problem (2.3), we introduce the following definition.

Definition 2.2 *[probability of failure] Given N, $\eta\in(0,1)$, the integer m where $0\le m\le N$ and $g:\Theta\times\mathscr{W}\to\{0,1\}$, the probability of failure, denoted by $p(N,\eta,m)$ is defined as*

$$\begin{aligned} p(N,\eta,m) := \mathrm{Pr}_{\mathscr{W}^N}\{\mathrm{w}\in\mathscr{W}^N \ :\ & \textit{There exists } \theta\in\Theta \\ & \textit{such that } \hat{E}(\theta,\mathrm{w})\le\frac{m}{N} \textit{ and } E(\theta)>\eta\}. \end{aligned}$$

We remark that the probability of failure is slightly different from the probability of one-sided constrained failure introduced in (Álamo et al., 2009). Therefore, if the probability of failure $p(N,\eta,m)$ is no greater than δ then every feasible solution $\theta\in\Theta$ to problem (2.3) satisfies $E(\theta)\le\eta$ with probability no smaller than $1-\delta$. From a practical point of view, the objective is to obtain explicit expressions yielding a minimum number of samples N such that the inequality $p(N,\eta,m)\le\delta$ holds.

We notice that the probability of failure can be easily bounded by the binomial distribution if Θ consists of a unique element. That is, if $\Theta=\{\hat{\theta}\}$ is a singleton, then

$$p(N,\eta,m) =$$

$$\mathrm{Pr}_{\mathscr{W}^N}\{\mathrm{w} \in \mathscr{W}^N \ : \ \hat{E}(\hat{\theta},\mathrm{w}) \leq \frac{m}{N} \,\text{and}\, E(\hat{\theta}) > \eta\} =$$

$$\mathrm{Pr}_{\mathscr{W}^N}\{\mathrm{w} \in \mathscr{W}^N \ : \ \sum_{\ell=1}^{N} g(\hat{\theta},w^{(\ell)}) \leq m \,\text{and}\, E(\hat{\theta}) > \eta\} \leq$$

$$\mathrm{Pr}_{\mathscr{W}^N}\{\mathrm{w} \in \mathscr{W}^N \ : \ \sum_{\ell=1}^{N} g(\hat{\theta},w^{(\ell)}) \leq m \,\text{and}\, E(\hat{\theta}) = \eta\} =$$

$$\sum_{i=0}^{m} \left(\begin{array}{c} N \\ i \end{array} \right) \eta^i (1-\eta)^{N-i}.$$

On the other hand, if Θ consists of an infinite number of elements, a deeper analysis involving Statistical Learning Theory is needed (Tempo et al., 2005),(Vidyasagar, 1997). In Subsection 2.3.3 of this chapter, we address this problem under the assumption that Θ consists of a finite number of elements.

In Subsection 2.3.4 we study the probabilistic properties of the optimal solution of problem (2.2) under the assumption that $g(\theta,w) = 0$ is equivalent to $f(\theta,w) \leq 0$, where $f : \Theta \times \mathscr{W} \to \mathbb{R}$ is a convex function with respect to θ in Θ. In this case the result is not expressed in terms of probability of failure because it applies only to the optimal solution of problem (2.2), and not to every feasible solution.

2.2 Explicit sample size bounds for the binomial distribution

Given a positive integer N and a nonnegative integer m, $m \leq N$, and $\eta \in (0,1)$, the binomial distribution is given by

$$B(N,\eta,m) := \sum_{i=0}^{m} \left(\begin{array}{c} N \\ i \end{array} \right) \eta^i (1-\eta)^{N-i}.$$

The problem we address in this section is the explicit computation of the *sample complexity*, i.e. a function $\tilde{N}(\eta,m,\delta)$ such that the inequality $B(N,\eta,m) \leq \delta$ holds for any $N \geq \tilde{N}(\eta,m,\delta)$, where $\delta \in (0,1)$. As it will be illustrated in the following section, the inequality $B(N,\eta,m) \leq \delta$ plays a fundamental role in probabilistic analysis and design methods. Although some explicit expressions are available, e.g. the multiplicative and additive forms of Chernoff bound (Chernoff, 1952), the results obtained in this chapter are tuned on the specific inequalities stemming from the control problems described in Section 2.3.

The following technical lemma provides an upper bound for the binomial distribution $B(N,\eta,m)$.

Lemma 2.3 *Suppose that $\eta \in (0,1)$ and that the nonnegative integer m and the positive integer N satisfy $m \leq N$. Then,*

$$\begin{aligned} B(N,\eta,m) &= \sum_{i=0}^{m} \binom{N}{i} \eta^i (1-\eta)^{N-i} \\ &\leq a^m \left(\frac{\eta}{a} + 1 - \eta\right)^N, \ \forall a \geq 1. \end{aligned}$$

Proof:

$$\begin{aligned} B(N,\eta,m) &= a^m \sum_{i=0}^{m} \binom{N}{i} a^{-m} \eta^i (1-\eta)^{N-i} \\ &\leq a^m \sum_{i=0}^{m} \binom{N}{i} a^{-i} \eta^i (1-\eta)^{N-i} \\ &\leq a^m \sum_{i=0}^{N} \binom{N}{i} \left(\frac{\eta}{a}\right)^i (1-\eta)^{N-i} \\ &= a^m \left(\frac{\eta}{a} + 1 - \eta\right)^N. \end{aligned}$$

■

We notice that each particular choice of $a \geq 1$ provides an upper bound for $B(N,\eta,m)$. When using Lemma 2.3 to obtain a given sample complexity result, the chosen value for a plays a significant role.

Lemma 2.4 *Given $\delta \in (0,1)$ and the nonnegative integer m, suppose that the integer N and the scalars $\eta \in (0,1)$ and $a > 1$ satisfy the inequality*

$$N \geq \frac{1}{\eta}\left(\frac{a}{a-1}\right)\left(\ln \frac{1}{\delta} + m \ln a\right). \tag{2.4}$$

Then, $m \leq N$ and

$$B(N,\eta,m) = \sum_{i=0}^{m} \binom{N}{i} \eta^i (1-\eta)^{N-i} \leq \delta.$$

Proof:

We first prove that if inequality (2.4) is satisfied then $N \geq m$. Since $\eta \in (0,1)$ and $\delta \in (0,1)$, inequality (2.4) implies

$$N \geq \left(\frac{a}{a-1}\ln a\right) m.$$

We notice now that

$$\frac{\mathrm{d}}{\mathrm{d}a}\left(\frac{a}{a-1}\ln a\right) = \left(\frac{-1}{(a-1)^2}\right)\ln a + \frac{1}{a-1}.$$

Since $\ln a < a-1$ for every $a > 1$, we conclude that

$$\frac{\mathrm{d}}{\mathrm{d}a}\left(\frac{a}{a-1}\ln a\right) > \left(\frac{-1}{(a-1)^2}\right)(a-1) + \frac{1}{a-1} = 0.$$

Using this fact, we conclude that $\frac{a}{(a-1)}\ln a$ is a strictly increasing function for $a > 1$. This means that

$$N \geq \left(\frac{a}{a-1}\ln a\right) m \geq \lim_{\hat{a}\to 1}\left(\frac{\hat{a}}{\hat{a}-1}\ln \hat{a}\right) m = m.$$

We now prove that inequality (2.4) guarantees that $a^m(\frac{\eta}{a}+1-\eta)^N \leq \delta$. The inequality (2.4) can be rewritten as

$$N\eta\left(\frac{a-1}{a}\right) \geq \ln\frac{1}{\delta} + m\ln a. \tag{2.5}$$

Since $x \leq -\ln(1-x)$ for every $x \in (0,1)$, and $\eta(\frac{a-1}{a}) \in (0,1)$, from inequality (2.5), we obtain a sequence of inequalities

$$-N\ln\left(1-\eta\left(\frac{a-1}{a}\right)\right) \geq \ln\frac{1}{\delta} + m\ln a$$

$$\ln\delta \geq m\ln a + N\ln\left(1-\eta\left(\frac{a-1}{a}\right)\right)$$

$$\delta \geq a^m\left(\frac{\eta}{a}+1-\eta\right)^N.$$

We have therefore proved that inequality (2.4) implies $m \leq N$ and $a^m(\frac{\eta}{a}+1-\eta)^N \leq \delta$. The claim of the property follows now directly from Lemma 2.3. ■

Obviously, the best sample size bound is obtained taking the infimum with respect to $a > 1$. However, a suboptimal value easily follows setting a equal to the Euler constant, which yields the sample size bound

$$N \geq \frac{1}{\eta}\left(\frac{\mathrm{e}}{\mathrm{e}-1}\right)\left(\ln\frac{1}{\delta}+m\right).$$

Since $\frac{\mathrm{e}}{\mathrm{e}-1} < 1.59$, we obtain $N \geq \frac{1.59}{\eta}\left(\ln\frac{1}{\delta}+m\right)$. If $m > 0$ then the choice $a = 1 + \frac{\ln\frac{1}{\delta}}{m} + \sqrt{2\frac{\ln\frac{1}{\delta}}{m}}$ provides a less conservative bound (which is very close to the optimal one based on extensive numerical experiments) at the price of a more involved expression.

Corollary 2.5 *Given $\delta \in (0,1)$ and the nonnegative integer m, suppose that the integer N and the scalar $\eta \in (0,1)$ satisfy the inequality*

$$N \geq \frac{1}{\eta}\left(m+\ln\frac{1}{\delta}+\sqrt{2m\ln\frac{1}{\delta}}\right). \tag{2.6}$$

Then,

$$B(N,\eta,m)=\sum_{i=0}^{m}\binom{N}{i}\eta^i(1-\eta)^{N-i}\leq\delta. \tag{2.7}$$

The proof of this corollary is given in the appendix. This corollary improves upon the explicit expression obtained when using the multiplicative form of the Chernoff bound (Tempo et al., 2005), which turns out to be

$$N \geq \frac{1}{\eta}\left(m+\ln\frac{1}{\delta}+\sqrt{\left(\ln\frac{1}{\delta}\right)^2+2m\ln\frac{1}{\delta}}\right).$$

2.3 Sample complexity for probabilistic analysis and design

We now illustrate some control problems in the context of randomized algorithms where one encounters inequalities of the form

$$B(N,\eta,m)=\sum_{i=0}^{m}\binom{N}{i}\eta^i(1-\eta)^{N-i}\leq\delta.$$

In particular we show how the results of the previous section can be used to obtain explicit sample size bounds guaranteeing that the probabilistic solutions resulting from different randomized approaches meet some pre-specified probabilistic properties.

2.3.1 Worst case performance analysis

We recall here a result presented in (Tempo, Bai and Dabbene, 1997) for the probabilistic worst case performance analysis.

Theorem 2.6 *Suppose that given function $f:\Theta\times\mathscr{W}\to\mathbb{R}$, and $\hat{\theta}\in\Theta$, the multisample $\mathrm{w}=\{w^{(1)},\ldots,w^{(N)}\}$ is drawn from $\mathscr{W}^N$ according to probability $\mathrm{Pr}_{\mathscr{W}^N}$. Suppose also that*

$$\gamma=\max_{\ell=1,\ldots,N} f(\hat{\theta},w^{(\ell)}).$$

If

$$N \geq \frac{\ln \frac{1}{\delta}}{\ln \frac{1}{1-\eta}},$$

then $\mathrm{Pr}_{\mathscr{W}}\{w \in \mathscr{W} \,:\, f(\hat{\theta},w) > \gamma\} \leq \eta$ *with probability no smaller than* $1-\delta$.

The proof of this statement, that can be found in (Tempo et al., 1997), relies on the fact that $\mathrm{Pr}_{\mathscr{W}}\{w \in \mathscr{W} \,:\, f(\hat{\theta},w) > \gamma\} \leq \eta$ with probability no smaller than $1-(1-\eta)^N$. Therefore, it suffices to take N such that $B(N,\eta,0) = (1-\eta)^N \leq \delta$.

2.3.2 Analysis of the probability of violation

In the following theorem we provide a sample complexity result that characterizes how the empirical mean converges in probability to the true probability of violation.

Theorem 2.7 *Given* $\hat{\theta} \in \Theta$, ρ, η *with* $0 \leq \rho < \eta < 1$ *and* $\delta \in (0,1)$, *if*

$$N \geq \frac{\ln \frac{1}{\delta}}{(\sqrt{\eta}-\sqrt{\rho})^2}$$

then $\mathrm{Pr}_{\mathscr{W}^N}\{\mathrm{w} \in \mathscr{W}^N \,:\, \hat{E}(\hat{\theta},\mathrm{w}) \leq \rho$ *and* $E(\hat{\theta}) > \eta\} \leq \delta$.

Proof: We notice that

$$\begin{aligned} &\mathrm{Pr}_{\mathscr{W}^N}\{\mathrm{w} \in \mathscr{W}^N \,:\, \hat{E}(\hat{\theta},\mathrm{w}) \leq \rho \text{ and } E(\hat{\theta}) > \eta\} = \\ &\mathrm{Pr}_{\mathscr{W}^N}\{\mathrm{w} \in \mathscr{W}^N \,:\, \hat{E}(\hat{\theta},\mathrm{w}) \leq \frac{\lfloor \rho N \rfloor}{N} \text{ and } E(\hat{\theta}) > \eta\} \\ &\leq B(N,\eta,\lfloor \rho N \rfloor). \end{aligned}$$

Therefore it suffices to show that the proposed sample size bound guarantees $B(N,\eta,\lfloor \rho N \rfloor) \leq \delta$. Using Corollary 3.4 and taking into account that $\rho N \geq \lfloor \rho N \rfloor$ we obtain that this is in fact the case if

$$\begin{aligned} N &\geq \frac{1}{\eta}\left(\ln\frac{1}{\delta} + \rho N + \sqrt{2\rho N \ln\frac{1}{\delta}}\right) \\ &= \frac{1}{\eta}\left(\sqrt{\ln\frac{1}{\delta}} + \sqrt{\rho N}\right)^2 - \frac{2-\sqrt{2}}{\eta}\left(\sqrt{\rho N \ln\frac{1}{\delta}}\right). \end{aligned}$$

This inequality is satisfied if

$$N \geq \frac{1}{\eta}\left(\sqrt{\ln\frac{1}{\delta}}+\sqrt{\rho N}\right)^2.$$

Equivalently, $\left(\sqrt{\eta}-\sqrt{\rho}\right)\sqrt{N} \geq \sqrt{\ln\frac{1}{\delta}}$ which yields $N \geq \frac{\ln\frac{1}{\delta}}{(\sqrt{\eta}-\sqrt{\rho})^2}$. ■

For small values of $\gamma = \frac{\rho}{\eta}$, the obtained sample size using Theorem 2.7 is

$$\frac{\ln\frac{1}{\delta}}{\eta(1-\sqrt{\gamma})^2} \approx \frac{\ln\frac{1}{\delta}}{\eta}.$$

This bound is significantly better (for small values of η and γ) than that corresponding to the additive form of the Chernoff bound (Chernoff, 1952), which for this case has a sample complexity given by

$$\frac{\ln\frac{1}{\delta}}{2(\eta-\rho)^2} = \frac{\ln\frac{1}{\delta}}{2\eta^2(1-\gamma)^2} \approx \frac{\ln\frac{1}{\delta}}{2\eta^2}.$$

On the other hand, the multiplicative form of the Chernoff bound (Tempo et al., 2005) provides the sample size bound

$$\frac{2\eta\ln\frac{1}{\delta}}{(\eta-\rho)^2} = \frac{2\ln\frac{1}{\delta}}{\eta(1-\gamma)^2}$$

which is worse than that provided by Theorem 2.7 for small values of $\gamma = \frac{\rho}{\eta}$. Finally, we remark that the bound presented in Theorem 2.7 can be also obtained by means of a result stated in (Okamoto, 1959), which is the so-called Okamoto bound.

2.3.3 Finite families for design

We consider here the nonconvex sampled problem (2.3) for the case when Θ consists of a set of finite cardinality n_C. As a motivation consider the case when, after an appropriate normalization procedure, the design parameter set is rewritten as $\hat{\Theta} = \{\ \theta \in \mathbb{R}^{n_\theta}\ :\ \|\theta\|_\infty \leq 1\ \}$. Suppose also that a gridding approach is adopted. For each component θ_j, $j = 1,\dots,n_\theta$ of the design parameters $\theta \in \mathbb{R}^{n_\theta}$, only n_{C_j} equally spaced values are considered. That is, θ_j is constrained into the set $\Upsilon_j = \{\ -1+\frac{2(t-1)}{(n_{C_j}-1)}\ :\ t = 1,\dots,n_{C_j}\ \}$. With this gridding, the following finite cardinality set $\Theta = \{\ [\theta_1,\dots,\theta_{n_\theta}]^\top\ :\ \theta_j \in \Upsilon_j,\ j = 1,\dots,n_\theta\ \}$ is obtained. We notice that the cardinality of the set is $n_C = \prod_{j=1}^{n_\theta} n_{C_j}$. Another situation in which the finite cardinality assumption holds is when a finite number of random samples in the space of design parameter are drawn according to a given probability, see e.g. (Fujisaki and Kozawa, 2006; Koltchinskii et al., 2000; Vidyasagar, 2001).

The following property states the relation between the binomial distribution and the probability of failure under this finite cardinality assumption.

Lemma 2.8 *Suppose that the cardinality of* Θ *is no larger than* n_C. *Then,*

$$p(N,\eta,m) \leq n_C \sum_{i=0}^{m} \left(\begin{array}{c} N \\ i \end{array} \right) \eta^i (1-\eta)^{N-i} = n_C B(N,\eta,m).$$

Proof:

Denote $\tilde{n}_C \leq n_C$ the cardinality of Θ. Therefore, Θ can be rewritten as $\Theta = \{\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(\tilde{n}_C)}\}$. Then,

$$\begin{gathered} p(N,\eta,m) = \Pr_{\mathscr{W}^N}\{\mathrm{w} \in \mathscr{W}^N \ : \text{There exists } \theta \in \Theta \\ \text{such that } \hat{E}(\theta,\mathrm{w}) \leq \frac{m}{N} \text{ and } E(\theta) > \eta\} \leq \\ \sum_{j=1}^{\tilde{n}_C} \Pr_{\mathscr{W}^N}\{\mathrm{w} \in \mathscr{W}^N : \hat{E}(\theta^{(j)},\mathrm{w}) \leq \frac{m}{N} \text{and} E(\theta^{(j)}) > \eta\} \leq \\ \tilde{n}_C \sum_{i=0}^{m} \left(\begin{array}{c} N \\ i \end{array} \right) \eta^i (1-\eta)^{N-i} \leq n_C \sum_{i=0}^{m} \left(\begin{array}{c} N \\ i \end{array} \right) \eta^i (1-\eta)^{N-i}. \end{gathered}$$

■

Consider now the optimization problem (2.3). It follows from Lemma 2.8 that in order to guarantee that every feasible solution $\hat{\theta} \in \Theta$ satisfies $E(\hat{\theta}) \leq \eta$ with probability no smaller than $1-\delta$, it suffices to take N such that $n_C B(N,\eta,m) \leq \delta$, where n_C is an upper bound on the cardinality of Θ. As it will be shown next, the required sample complexity in this case grows with the natural logarithm of n_C. This means that we can consider finite families with high cardinality and still obtain reasonable sample complexity bounds.

Theorem 2.9 *Suppose that the cardinality of* Θ *is no larger than* n_C. *Given the nonnegative integer m,* $\eta \in (0,1)$ *and* $\delta \in (0,1)$, *if*

$$N \geq \inf_{a>1} \frac{1}{\eta} \left(\frac{a}{a-1} \right) \left(\ln \frac{n_C}{\delta} + m \ln a \right)$$

then $p(N,\eta,m) \leq \delta$. *Moreover, if*

$$N \geq \frac{1}{\eta} \left(m + \ln \frac{n_C}{\delta} + \sqrt{2m \ln \frac{n_C}{\delta}} \right)$$

then $p(N,\eta,m) \leq \delta$.

Proof:

From Lemma 2.8 we have that $p(N,\eta,m) \leq \delta$ provided that $B(N,\eta,m) \leq \frac{\delta}{n_C}$. The two claims of the property now follow directly from Lemma 2.4 and Corollary 3.4 respectively.

■

Consider the sampled optimization problem (2.3)

$$\min_{\theta\in\Theta} J(\theta) \text{ subject to } \sum_{\ell=1}^{N} g(\theta, w^{(\ell)}) \leq m.$$

From the definition of $p(N,\eta,m)$ and Theorem 2.9 we conclude that if one draws N i.i.d. samples $\{w^{(1)},\ldots,w^{(N)}\}$ from $\mathscr{W}$ according to probability $\Pr_{\mathscr{W}}$, then with probability no smaller than $1-\delta$, all the feasible solutions to problem (2.3) have a probability of violation no larger than η, provided that the cardinality of Θ is upper bounded by n_C and the sample complexity is given by

$$N \geq \frac{1}{\eta}\left(m + \ln\frac{n_C}{\delta} + \sqrt{2m\ln\frac{n_C}{\delta}}\right).$$

We remark that taking a equal to the Euler constant, the following sample size bound

$$N \geq \frac{1}{\eta}\left(\frac{\mathrm{e}}{\mathrm{e}-1}\right)\left(\ln\frac{n_C}{\delta} + m\right)$$

is immediately obtained from Theorem 2.9. If $m > 0$ then a suboptimal value for a is given by

$$a = 1 + \frac{\ln\frac{n_C}{\delta}}{m} + \sqrt{2\frac{\ln\frac{n_C}{\delta}}{m}}.$$

2.3.4 Optimal robust optimization for design

In this subsection, we study the so-called scenario approach for robust control introduced in (Calafiore and Campi, 2006), see also (Campi and Garatti, 2008) for recent results in this area. Suppose that in order to address the general semi-infinite optimization problem (2.1), one resorts to randomization. That is, N i.i.d. samples $\{w^{(1)},\ldots,w^{(N)}\}$ from $\mathscr{W}$ according to probability $\Pr_{\mathscr{W}}$ are drawn and one solves the following problem

$$\min_{\theta\in\Theta} J(\theta) \text{ subject to } g(\theta, w^{(\ell)}) = 0,\ \ell = 1,\ldots,N. \tag{2.8}$$

We consider here the particular case in which $J(\theta) = c^\top \theta$, the constraint $g(\theta, w) = 0$ is convex in Θ for all $w \in W$, the solution of (2.8) is unique[1]. These assumptions are now stated precisely.

Assumption 2.10 [convexity] *Let $\Theta \subset \mathbb{R}^{n_\theta}$ be a convex and closed set. We assume that*

$$J(\theta) := c^\top \theta \quad \text{and} \quad g(\theta, w) := \begin{cases} 0 & \text{if } f(\theta, w) \leq 0, \\ 1 & \text{otherwise} \end{cases}$$

where $f : \Theta \times \mathscr{W} \to [-\infty, \infty]$ is convex in Θ for any fixed value of $w \in \mathscr{W}$.

Assumption 2.11 [feasibility and uniqueness] *The optimization problem (2.8), for all possible multisample extractions $\{w^{(1)}, \ldots, w^{(N)}\}$, is always feasible and attains a unique optimal solution. Moreover, its feasibility domain has a nonempty interior.*

We state here a result proved in (Campi and Garatti, 2008) that relates the binomial distribution to the probabilistic properties of the optimal solution obtained from (2.8).

Lemma 2.12 *Let Assumptions 1 and 2 hold. Suppose that N, $\eta \in (0,1)$ and $\delta \in (0,1)$ satisfy the following inequality*

$$\sum_{i=0}^{n_\theta - 1} \binom{N}{i} \eta^i (1-\eta)^{N-i} \leq \delta. \tag{2.9}$$

Then, with probability no smaller than $1-\delta$, the optimal solution $\hat{\theta}_N$ to the optimization problem (2.8) satisfies the inequality $E(\hat{\theta}_N) \leq \eta$.

We now state an explicit sample size bound to guarantee that the probability of violation is smaller than η with probability at least $1-\delta$.

Theorem 2.13 *Let Assumptions 1 and 2 hold. Given $\eta \in (0,1)$ and $\delta \in (0,1)$, if*

$$N \geq \inf_{a>1} \left(\frac{a}{\eta(a-1)} \right) \left(\ln \frac{1}{\delta} + (n_\theta - 1) \ln a \right) \tag{2.10}$$

or

$$N \geq \frac{1}{\eta} \left(\ln\left(\frac{1}{\delta}\right) + n_\theta - 1 + \sqrt{2(n_\theta - 1)\ln\frac{1}{\delta}} \right) \tag{2.11}$$

then, with probability no smaller than $1-\delta$, the optimal solution $\hat{\theta}_N$ to the optimization problem (2.8) satisfies the inequality $E(\hat{\theta}_N) \leq \eta$.

[1] We remark that this uniqueness assumption can be relaxed in most cases, as shown in Appendix A of (Calafiore and Campi, 2006).

Proof: From Lemma 2.12 it follows that it suffices to take N such that $B(N,\eta,n_\theta-1)\leq\delta$. Both inequalities (2.10) and (2.11) guarantee that $B(N,\eta,n_\theta-1)\leq\delta$ (see Lemma 2.4 and Corollary 3.4 respectively). This completes the proof. ■

We remark that a sample size bound which depends linearly on $\frac{1}{\eta}$ is obtained taking a equal to the Euler constant

$$N \geq \frac{1}{\eta}\left(\frac{\mathrm{e}}{\mathrm{e}-1}\right)\left(\ln\frac{1}{\delta}+n_\theta-1\right).$$

This bound always improves upon other recent bounds given in the literature, see e.g. (Álamo et al., 2009). If $n_\theta>1$ then a suboptimal value for a is given by

$$a=1+\frac{\ln\frac{1}{\delta}}{n_\theta-1}+\sqrt{2\frac{\ln\frac{1}{\delta}}{n_\theta-1}}.$$

2.4 Numerical example

The objective of this numerical example is to obtain probabilistic upper and lower bounds of a given time function $y:\mathscr{W}\to R$ of the form

$$y(w)=[A(1+\frac{1}{2}t^2)\sin(7t+0.5)+B]\mathrm{e}^{-\frac{3}{2}t},$$

where $w\in\mathscr{W}$.

The uncertainty set $\mathscr{W}$ is

$$\mathscr{W}=\left\{w=[t\quad A\quad B]^T, t\in[0,1],\, A\in[1,3],\, B\in[1,3]\right\}.$$

For a given order d, we define the regressor $\varphi_d:\mathscr{W}\to R^{d+1}$ as

$$\varphi_d(w)=\varphi_d([t\quad A\quad B]^T)=\left[\begin{array}{ccccc}1 & t & t^2 & \cdots & t^d\end{array}\right]^T.$$

The objective of this example is to find a parameter vector $\theta=[\gamma_d,\lambda_d]^T$, $\gamma_d\in R^{d+1}$ and $\lambda_d\in R^{d+1}$ such that, with probability no smaller than $1-\delta$,

$$\mathrm{Pr}_{\mathscr{W}}\{\, w\in\mathscr{W} \;:\; |y(w)-\gamma_d^T\varphi_d(w)|\geq\lambda_d^T|\varphi_d(w)|\,\}\leq\eta.$$

The vector $|\varphi_d(w)|$ is obtained from the absolute values of $\varphi_d(w)$. The binary function $g : \Theta \times \mathscr{W} \to \{0,1\}$, is defined as

$$g(\theta,w) := \begin{cases} 0 & \text{if } \theta \text{ meets design specifications for } w \\ 1 & \text{otherwise,} \end{cases}$$

where "design specifications" means satisfying the following constraint:

$$|y(w) - \gamma_d^T \varphi_d(w)| \leq \lambda_d^T |\varphi_d(w)|$$

for randomly generated samples $w \in \mathscr{W}$.

A similar problem is addressed in (Campi and Garatti, 2008) using the scenario approach. For the numerical computations, we take $\delta = 10^{-6}$ and $\eta = 0.05$. We address the problem in this chapter from the finite families and scenario approach.

2.4.1 Finite families approach

We apply the results of Section 2.3.3 to determine both the degree d and the parameter vectors (γ_d, λ_d) that meet the design specification and optimize a given performance index.

In this example, as it will be seen later, a finite family of cardinality $n_C = 150$ is considered. In order to compare the finite family approach with the scenario one, we consider no allowed failures (i.e $m = 0$). For this choice of parameters ($m = 0$, $n_C = 150$, $\delta = 10^{-6}$ and $\eta = 0.05$), the number of samples N required to obtain a solution with the specified probabilistic probabilities is 377 (see Property 2.9). A set D of $M = N$ samples is drawn (i.i.d.) from $\mathscr{W}$. We use these samples to select the optimal parameters $(\tilde{\gamma}_d, \tilde{\lambda}_d)$ corresponding to each of the different regressors $\varphi_d(\cdot)$. Each pair $(\tilde{\gamma}_d, \tilde{\lambda}_d)$ is obtained minimizing the empirical mean of the absolute value of the approximation error. That is, each pair $(\tilde{\gamma}_d, \tilde{\lambda}_d)$ is the solution to the optimization problem

$$\begin{array}{ll} \min\limits_{\gamma_d, \lambda_d} & \dfrac{1}{M} \sum\limits_{w \in D} \lambda_d^T |\varphi_d(w)| \\ s.t. & |y(w) - \gamma_d^T \varphi_d(w)| \leq \lambda_d^T |\varphi_d(w)|, \ \forall w \in D. \end{array}$$

We notice that the obtained parameters do not necessarily satisfy the probabilistic design specifications. In order to resolve this problem, we consider a new set of candidate solutions of the form

$$\begin{aligned} \Theta = \{\ \theta_{d,j} = (\tilde{\gamma}_d, \mathrm{e}^{\left(\frac{j}{10}-1\right)} \tilde{\lambda}_d) \ : \\ d = 1, \ldots, d_{\max}, j = 1, \ldots, j_{\max}\ \}. \end{aligned}$$

This family has cardinality $n_c = d_{\max} j_{\max}$. We take a large factor $\mathrm{e}^{\left(\frac{j}{10}-1\right)}$, to increase the probability of meeting the design specifications. Therefore, choosing a large enough value for $j_{\max}$ leads to a non-empty intersection of Θ with the set of parameters that meet the design specifications. In this example, we take $j_{\max} = 15$ and $d_{\max} = 10$. This yields to $n_c = 150$.

Using the finite family approach, we choose from Θ the design parameter that optimizes a given performance index. We draw from $\mathscr{W}$ a set V of N (i.i.d.) samples and select the pair that minimizes the empirical mean of the absolute value of the approximation error in the validation set V. That is, we consider the performance index

$$\frac{1}{N} \sum_{w \in V} \mathrm{e}^{\left(\frac{j}{10}-1\right)} \tilde{\lambda}_d^T |\varphi_d(w)|$$

subject to the constraints

$$|y(w) - \tilde{\gamma}_d^T \varphi_d(w)| \leq \mathrm{e}^{\left(\frac{j}{10}-1\right)} \tilde{\lambda}_d^T |\varphi_d(w)|, \ \forall w \in V.$$

We remark that the feasibility of this problem can be guaranteed in two ways. The first one is to choose $j_{\max}$ large enough. The second one is to allow m failures. As previously discussed, in this example we take $j_{\max} = 15$ and $m = 0$.

As the cardinality N of V has been chosen properly, the probability of violation and the probability of failure of the best solution from Θ are bounded by η and δ respectively.

The obtained solution corresponds to $d = 5$ and $j = 13$. The corresponding value for the performance index is 0.8121. Figure 3.1 shows the approximation for the set V and the obtained probabilistic upper and lower bounds for the random function.

Finally, for illustrative purposes, we check with a validation set of sample size $N_v = 10N$, obtaining a number of 60 failures. The experimental violation probability turned out to be $\eta_{\exp} = 0.0146$, while the specification was $\eta = 0.05$.

2.4.2 Convex scenario approach

In this case we take advantage of the result of Subsection 2.4.1 and take $d = 5$ as the order of the approximation polynomial. Following the scenario approach we draw a set $\mathscr{W}_k$ of N samples (i.i.d) from $\mathscr{W}$ and solve the convex optimization problem

$$\begin{array}{ll} \min\limits_{\gamma_d, \lambda_d} & \lambda_d^T E\{|\varphi_d(t)|\} \\ s.t. & |y(w) - \gamma_d^T \varphi_d(w)| \leq \lambda_d^T |\varphi_d(w)|, \ \forall w \in \mathscr{W}_k. \end{array}$$

In order to guarantee the design specifications we use Property 2.8 to determine the value of N. Since the number of decision variables is $2(d+1)$, $\eta = 0.05$ and $\delta = 10^{-6}$, the resulting value for N is 845. We notice that the convex scenario approach does not apply directly to the minimization of the empirical mean. This is why one has to resort to the exact computation of the mean of the approximation error $\lambda_d^T E\{|\varphi_d(t)|\}$, see (Campi and Garatti, 2008).

Figure (3.1) shows the initial data set generated using the procedure described above, plus the envelope that contains all the solution polynomials.

Again, for illustrative purposes, we check with a validation set of size $N_v = 10N$, obtaining zero failures. The experimental value $\eta_{\text{exp}} = 0$ is obtained, while the specification was $\eta = 0.05$.

Using this strategy, 845 design data are required, bigger than the number of required samples to use the finite families approach. We obtained a performance index of 0.8748, slightly larger than that obtained by the finite families strategy. The advantage of the finite families approach is that, using a smaller number of samples, a similar performance is obtained. This allows us to determine the best order of the polynomial with the further advantage that the exact computation of the mean of the error is not required.

2.5 Conclusion

In this chapter we have derived sample complexity results for various analysis and design problems related to uncertain systems. In particular we provided new results which guarantee that a binomial distribution expression is smaller than a pre-specified value. These results are subsequently exploited for the analysis of worst case performance and constraint violation. With regard to design problems we considered the case of finite cardinality of controller families and the special case when the design problem can be recast as a robust convex optimization problem.

2.6 Appendix

Proof of Corollary 3.4.

We first consider the case $m = 0$. Then, we obtain $B(N,\eta,0) = (1-\eta)^N = \mathrm{e}^{N\ln(1-\eta)} \leq \mathrm{e}^{-\eta N}$. Therefore, it follows from $\eta N \geq \ln\frac{1}{\delta}$ that $\mathrm{e}^{-\eta N} \leq \mathrm{e}^{\ln\delta} = \delta$. This proves the result for $m = 0$.

Consider now the case $m > 0$. We first prove that

$$h(r) := \sqrt{2(r-1)} - \ln\left(r + \sqrt{2(r-1)}\right) \geq 0, \;\; \forall r \geq 1. \tag{2.12}$$

Since $h(1) = 0$, the inequality $h(r) \geq 0$ holds if the derivative of $h(r)$ is strictly positive for every r larger than one.

$$\frac{\mathrm{d}}{\mathrm{d}r}h(r) =$$

$$\frac{1}{\sqrt{2(r-1)}} - \frac{1}{r+\sqrt{2(r-1)}}\left(1 + \frac{1}{\sqrt{2(r-1)}}\right) =$$

$$\left(\frac{1}{\sqrt{2(r-1)}}\right)\left(1 - \frac{1+\sqrt{2(r-1)}}{r+\sqrt{2(r-1)}}\right) =$$

$$\left(\frac{1}{\sqrt{2(r-1)}}\right)\left(\frac{r-1}{r+\sqrt{2(r-1)}}\right) \geq 0, \;\; \forall r > 1.$$

This proves the inequality $h(r) \geq 0$, for all $r \geq 1$. Denote now $\hat{a} = r + \sqrt{2(r-1)}$, with $r = 1 + \frac{1}{m}\ln\frac{1}{\delta}$. Clearly $\hat{a} > 1$, therefore, from the direct application of Lemma 2.4 we conclude that it suffices to choose N such that

$$N\eta \geq \frac{\hat{a}}{\hat{a}-1}\left(\ln\frac{1}{\delta} + m\ln\hat{a}\right) =$$

$$\frac{r+\sqrt{2(r-1)}}{r-1+\sqrt{2(r-1)}}\left(r - 1 + \ln(r + \sqrt{2(r-1)}\right)m.$$

Since $h(r) \geq 0$ we infer that

$$\frac{r-1+\ln(r+\sqrt{2(r-1)})}{r-1+\sqrt{2(r-1)}} \leq 1.$$

From this, we finally conclude that inequality $B(N,\eta,m) \leq \delta$ holds if

$$N\eta \geq (r + \sqrt{2(r-1)})m = m + \ln\frac{1}{\delta} + \sqrt{2m\ln\frac{1}{\delta}}.$$

■

Chapter 3

Randomized validation schemes

3.1 Introduction

In this chapter, we present a randomized strategy for design under uncertainty. The main contribution is to provide a general class of sequential algorithms which satisfy the required specifications using probabilistic validation. At each iteration of the sequential algorithm, a candidate solution is probabilistically validated by means of a set of randomly generated uncertainty samples.

The idea of validation sets has been used in some randomized algorithms when a given candidate solution is classified as probabilistic solution when it satisfies all the constraints on the validation set. In this chapter, we show the limitations of this strategy and present a more general setting where the candidate solution may violate the specifications for a reduced number of elements of the validation set. This generalized scheme exhibits some advantages, in particular in terms of obtaining a probabilistic solution.

The design in the presence of uncertainty is of paramount relevance in different fields. Unfortunately, the related semi-infinite optimization problems often exhibit an NP-hard nature that seriously compromises their solution in a reasonable computational time (Blondel and Tsitsiklis, 2000). There exist two ways to circumvent this NP-hard issue. One option consists in resorting to deterministic relaxations of the original problem which are normally solved in polynomial time but which might lead to overly conservative solutions (Scherer, 2006). An alternative paradigm is to assume that the plant uncertainty is probabilistically described so that a randomized algorithm may be derived to obtain, normally in polynomial time, a solution with some given properties normally stated in terms of the probability of error (Tempo et al., 2005), (Vidyasagar, 1997).

The field of randomized algorithms have evolved significantly in the last years. A recent survey on this topic can be found in (Calafiore et al., 2011). Two complementary approaches, non-sequential and sequential, have been proposed. A classical approach for non-sequential methods is based upon statistical learning theory (Vapnik, 1998). In particular, the use of this theory for feedback design of uncertain systems has been initiated in (Vidyasagar, 1997); subsequent work along this direction include (Koltchinskii et al., 2000), (Vidyasagar, 2001), (Vidyasagar and Blondel, 2001), (Álamo et al., 2009). In (Álamo et al., 2010*a*) and (Luedtke and Ahmed, 2008) the particular case in which the design parameter set has finite cardinality is analyzed.

The advantage of these methods is that the problem under attention may be non-convex. For convex optimization problems, a successful non-sequential paradigm, denoted as the scenario approach, has been introduced in (Calafiore and Campi, 2005) and (Calafiore and Campi, 2006). See also (Campi and Garatti, 2008), (Campi and Garatti, 2011), (Calafiore, 2010) and (Álamo et al., 2010*a*) for related results.

In non-sequential methods the original robust control problem is reformulated in terms of a single optimization problem with sampled constraints which are randomly generated. A relevant feature of these approaches is that they do not require any validation step. The number of samples required to guarantee that the obtained solutions meet some probabilistic specifications should take into account the specific nature of the problem under consideration. The main result of this line of research is to derive explicit lower bounds to this required sample size. Recently, improvements regarding this sample complexity have been provided in (Álamo et al., 2009). However, the obtained explicit sample bounds can be overly conservative because they rely on a worst-case analysis and grow (at least linearly) with the number of decision variables.

For sequential methods, the resulting iterative algorithms are based on stochastic gradient (Calafiore and Polyak, 2001), (Polyak and Tempo, 2001), ellipsoid iterations (Kanev et al., 2003), (Oishi, 2007), or analytic center cutting plane methods (Calafiore and Dabbene, 2007), see also (Álamo, Tempo, Ramírez and Camacho, 2007) for other classes of sequential algorithms. Convergence properties in finite-time are in fact one of the focal points of these papers. Various control problems have been solved using these sequential randomized algorithms, including robust LQ regulators, switched systems, and uncertain linear matrix inequalities (LMIs). Sequential methods are mostly used for uncertain convex problems because the computational effort at each iteration is affordable. However, they can be applied in principle to any kind of robust design problem. For example, a sequential algorithm that can be applied to a rather general class of problems is presented in (Álamo et al., 2009).

The main point in common of all these sequential algorithms is the use of the validation strategy presented in (Oishi, 2003) (see (Oishi, 2007) for a journal version). The candidate solutions provided at each iteration of these algorithms are validated using a validation set which is drawn according to the probability measure defined in the uncertain set. If the candidate solution satisfies the design specifications for every element of this validation set then it is classified as probabilistic solution and the algorithm terminates. The main point in this validation scheme is that the cardinality of the validation set increases with each iteration of the algorithm. The strategy guarantees that if a probabilistic solution is obtained, then it meets some probabilistic specifications. A similar approach, introduced in (Dabbene et al., 2010), has been presented in (Calafiore et al., 2011) in the context of sequential algorithms. The contribution is a reduction on the cardinality required for the validation sets.

The main contribution of this chapter is to propose a relaxed validation scheme in which we allow the candidate solution to violate the design specifications for one or more of the members of the validation set. The idea of allowing some violations of the constraints can be found, for example, in the context of identification (Bai et al., 2002), chance-constrained optimization (Campi and Garatti, 2011) and statistical learning theory (Álamo et al., 2009). This scheme makes sense in the presence of soft constraints or when it is not possible to find a solution satisfying the specifications for all the admissible realizations of uncertainty.

As it will be shown later in this chapter, this relaxed scheme allows us to reduce, in some cases dramatically, the number of iterations required by the sequential algorithm. Another advantage of the proposed approach is that it does not rely on the existence of a robust deterministic solution. Furthermore, the presented strategy is quite general and is not based on a convexity assumption.

The rest of the chapter is organized as follows. Section 3.2 presents the problem statement. In Section 5.5.1 we introduce the proposed family of probabilistically validated algorithms. The issue of sample size of the validating sets is analyzed in Section 5.5.2. A comparison with the validation scheme presented in (Oishi, 2007) is provided in Section 3.5. Section 3.6 discusses how to use the results of the chapter in the context of non-sequential randomized algorithms. The chapter ends with a section of conclusions.

3.2 Problem statement

We assume that a probability measure $\text{Pr}_{\mathscr{W}}$ over the sample space $\mathscr{W}$ is given. Given $\mathscr{W}$, a collection of N independent identically distributed (i.i.d.) samples $\mathrm{w} = \{w^{(1)},\dots,w^{(N)}\}$ drawn from $\mathscr{W}$ is said to belong to the Cartesian product $\mathscr{W}^N = \mathscr{W} \times \cdots \times \mathscr{W}$ (N times). Moreover, if the collection w of N i.i.d. samples $\{w^{(1)},\dots,w^{(N)}\}$ is generated from $\mathscr{W}$ according to the probability measure $\text{Pr}_{\mathscr{W}}$, then the *multisample* w is drawn according to the probability measure $\text{Pr}_{\mathscr{W}^N}$. The scalars $\varepsilon \in (0,1)$ and $\delta \in (0,1)$ denote probabilistic parameters. Furthermore, $\ln(\cdot)$ is the natural logarithm, e is the Euler number and $\log_2$ is the logarithm to the base 2. For $x \in \mathbb{R}$, $x \geq 0$, $\lfloor x \rfloor$ denotes the largest integer smaller than or equal to x; $\lceil x \rceil$ denotes the smallest integer greater or equal than x. For $\alpha \in \mathbb{R}$, $\alpha > 1$, $\xi(\alpha)$

denotes the Riemann zeta function (i.e. $\xi(\alpha) = \sum\limits_{k=1}^{\infty} \frac{1}{k^{\alpha}}$).

Typically, for a robustness problem, the design parameters, along with different auxiliary variables, are parameterized by means of a decision variable vector θ, which is denoted as design parameter, and is restricted to a design parameter set Θ. On the other hand, the uncertainty w is bounded in the set $\mathcal{W}$. That is, each element $w \in \mathcal{W}$ represents one of the admissible uncertainty realizations. We also consider a binary measurable function $g : \Theta \times \mathcal{W} \to \{0,1\}$. In a control context, the binary function $g : \Theta \times \mathcal{W} \to \{0,1\}$, is defined as

$$g(\theta,w) := \begin{cases} 0 & \text{if } \theta \text{ meets control specifications for } w \\ 1 & \text{otherwise.} \end{cases}$$

Given $\theta \in \Theta$, there might be a subset of the elements of $\mathcal{W}$ for which the constraint $g(\theta,w) = 0$ is not satisfied. This concept is rigorously formalized by means of the notion of "probability of violation", which is now introduced.

Definition 3.1 *[probability of violation] Consider a probability measure* $\text{Pr}_{\mathcal{W}}$ *over* $\mathcal{W}$ *and let* $\theta \in \Theta$ *be given. The probability of violation of* θ *for the function* $g : \Theta \times \mathcal{W} \to \{0,1\}$ *is defined as*

$$E(\theta) := \text{Pr}_{\mathcal{W}}\{\, w \in \mathcal{W} \;:\; g(\theta,w) = 1 \,\}.$$

We consider the robust optimization problem

$$\min_{\theta\in\Theta} J(\theta) \ \text{ subject to } \ E(\theta) \leq \varepsilon \tag{3.1}$$

where $J : \Theta \to (-\infty,\infty)$ is a measurable function. Given accuracy $\varepsilon \in (0,1)$ and confidence $\delta \in (0,1)$, the main focus is to design an algorithm such that any probabilistic solution $\hat{\theta}$ obtained running the algorithm satisfies $E(\hat{\theta}) \leq \varepsilon$ with probability no smaller than $1-\delta$. We address this issue by means of a rather general family of randomized algorithms with probabilistic validation that are introduced in the following section.

3.3 Sequential algorithms with probabilistic validation

In this section we present a general family of randomized algorithms, which we denote as "**S**equential **P**robabilistic **V**alidation algorithms", SPV algorithms for short. The main feature of this class of algorithms is that they are based on a probabilistic validation step. This family encompasses most of the sequential randomized algorithms that have been mentioned in the introduction of this chapter. As a matter of fact, the non-sequential strategies that can be found in the context of statistical learning theory (Vidyasagar, 1997) and convex scenario (Calafiore and Campi, 2006) can be also provided with an outer iterative structure that makes them fit in the proposed scheme. We explore this possibility in Section 3.6.

Each iteration of an SPV algorithm is composed of the computation of a candidate solution for the problem and a validation step. The results provided in this chapter are basically independent of the particular strategy chosen to obtain candidate solutions. Therefore, in the following discussion we restrict ourselves to a generic candidate solution computation step.

The accuracy $\varepsilon \in (0,1)$ and confidence $\delta \in (0,1)$ required for the probabilistic solution play a relevant role when determining the sample size of each validation step. The main purpose of this chapter is to provide a validation scheme such that it guarantees that for given accuracy ε and confidence δ, all the probabilistic solutions obtained running the SPV algorithm have a probability of violation no larger than ε with probability no smaller than $1-\delta$.

We enumerate each iteration of the algorithm by means of integer k. We denote m_k the number of violations that are allowed at the validation step of iteration k. We assume that m_k is given by a function of k, that is, $m_k = m(k)$ where the function $m : \mathbb{N} \to \mathbb{N}$ is given. We also denote M_k the sample size of the validation step of iteration k. We assume that M_k is given by a function of k, ε and δ. That is, $M_k = M(k,\varepsilon,\delta)$ where $M : \mathbb{N} \times \mathbb{R} \times \mathbb{R} \to \mathbb{N}$ has to be appropriately designed in order to guarantee the probabilistic properties of the algorithm (the main contribution of (Oishi, 2007) was to provide this function for the particular case $m_k = 0$ for every $k \geq 1$). For future references we denote the functions $m(\cdot)$ and $M(\cdot,\cdot,\cdot)$ as *level function* and *cardinality function* respectively.

We are now in a position to introduce the structure of an SPV algorithm

(i) Set accuracy $\varepsilon \in (0,1)$ and confidence $\delta \in (0,1)$ equal to the desired levels. Set k equal to 1.

(ii) Obtain a candidate solution $\hat{\theta}_k$ to the robust optimization problem (A.4.1).

(iii) Set $m_k = m(k)$ and $M_k = M(k,\varepsilon,\delta)$.

(iv) Obtain validation set $\mathscr{V}_k = \{v^{(1)},\dots,v^{(M_k)}\}$ drawing M_k i.i.d validation samples from $\mathscr{W}$ according to probability $\Pr_{\mathscr{W}}$.

(v) If $\sum\limits_{\ell=1}^{M_k} g(\hat{\theta}_k, v^{(\ell)}) \leq m_k$, then $\hat{\theta}_k$ is a probabilistic solution.

(vi) Exit if the exit condition is satisfied.

(vii) k=k+1. Goto (ii).

Although the exit condition can be quite general, a reasonable one is to exit after a given number of candidate solutions have been classified as probabilistic solutions or when a given computational time has elapsed since the starting of the algorithm. After exiting one could choose the probabilistic solution which maximizes a given performance index. In the following section we propose a strategy to choose the cardinality of the validation set at iteration k in such a way that with probability no smaller than $1-\delta$ all candidate solutions classified as probabilistic solutions by the algorithm meet the accuracy ε.

3.4 Adjusting the validation sample size

The cardinality adjusting strategy provided in this section constitutes a generalization of that presented in (Oishi, 2007) and (Dabbene et al., 2010). To infer the different results of this section we rely on some contributions on sample size complexity presented in (Álamo et al., 2010*a*).

The following definition introduces the notion of *failure function*.

Definition 3.2 (failure function) *The function $\mu : \mathbb{N} \to \mathbb{R}$ is said to be a failure function if it satisfies*

(i) $\mu(k) \in (0,1)$ *for every positive integer k.*

(ii) $\sum\limits_{k=1}^{\infty} \mu(k) \leq 1.$

We notice that the function

$$\mu(k) = \frac{1}{\xi(\alpha)k^{\alpha}},$$

where $\xi(\cdot)$ is the Riemann zeta function, is a failure function for every $\alpha > 1$. This is due to the fact that $\sum\limits_{i=1}^{\infty} \frac{1}{k^{\alpha}}$ converges for every scalar α greater than 1 to $\xi(\alpha)$. This family has been used in the context of validation schemes in (Calafiore et al., 2011) and in (Oishi, 2007) for the particular value $\alpha = 2$.

Property 3.3 *Consider an SPV algorithm with given accuracy parameter $\varepsilon \in (0,1)$, confidence $\delta \in (0,1)$, level function $m(\cdot)$ and cardinality function $M(\cdot,\cdot,\cdot)$. If there exists a failure function $\mu(\cdot)$ such that*

$$\sum_{i=0}^{m(k)} \begin{pmatrix} M(k,\varepsilon,\rho) \\ i \end{pmatrix} \varepsilon^{i}(1-\varepsilon)^{M(k,\varepsilon,\rho)-i} \leq \delta\mu(k), \ \forall k \geq 1$$

then with a probability greater than $1-\delta$ *all the probabilistic solutions obtained running the SPV algorithm have a probability of violation no greater than* ε*.*

The proof of the previous property follows the same lines as the proof of Theorem 9 in (Oishi, 2007).

Denote by δ_k the probability of classifying at iteration k the candidate solution $\hat{\theta}_k$ as a probabilistic solution under the assumption that the probability of violation

$$\begin{aligned}
\delta_k &= \mathrm{Pr}_{\mathscr{W}^{M_k}}\Big\{\{v^{(1)},\ldots,v^{(M_k)}\}\in\mathscr{W}^{M_k}\ : \\
&\qquad \sum_{j=1}^{M_k} g(\hat{\theta}_k,v^{(j)})\le m_k \ \text{and}\ E(\hat{\theta}_k)>\varepsilon\Big\} \\
&< \mathrm{Pr}_{\mathscr{W}^{M_k}}\Big\{\{v^{(1)},\ldots,v^{(M_k)}\}\in\mathscr{W}^{M_k}\ : \\
&\qquad \sum_{j=1}^{M_k} g(\hat{\theta}_k,v^{(j)})\le m_k \ \text{and}\ E(\hat{\theta}_k)=\varepsilon\Big\} \\
&= \sum_{i=0}^{m_k}\left(\begin{array}{c} M_k \\ i \end{array}\right)\varepsilon^i(1-\varepsilon)^{M_k-i} \\
&= \sum_{i=0}^{m(k)}\left(\begin{array}{c} M(k,\varepsilon,\rho) \\ i \end{array}\right)\varepsilon^i(1-\varepsilon)^{M(k,\varepsilon,\rho)-i} \\
&\le \delta\mu(k).
\end{aligned}$$

Therefore the probability of misclassification of a candidate solution at iteration k is smaller than $\delta\mu(k)$. From here we conclude that the probability of erroneously classifying one or more candidate solutions as probabilitic solutions running the algorithm is bounded by

$$\sum_{k=1}^{\infty}\delta_k<\sum_{k=1}^{\infty}\delta\mu(k)=\delta\sum_{k=1}^{\infty}\mu(k)\le\delta.$$

■

In order to design a cardinality function $M(\cdot,\cdot,\cdot)$ satisfying the conditions of Property 3.3 we will use the following result (see Corollary 1 in (Álamo et al., 2010*a*)).

Corollary 3.4 *Given* $\delta\in(0,1)$ *and the nonnegative integer m, suppose that the integer N and the scalar* $\eta\in(0,1)$ *satisfy the inequality*

$$N\ge\frac{1}{\eta}\left(m+\ln\frac{1}{\delta}+\sqrt{2m\ln\frac{1}{\delta}}\right).$$

Then,

$$\sum_{i=0}^{m} \left(\begin{array}{c} N \\ i \end{array} \right) \eta^i (1-\eta)^{N-i} \leq \delta.$$

We now present the main contribution of the chapter, which is a general expression to compute the cardinality of the validation set at each iteration of the algorithm.

Theorem 3.5 *Consider an SPV algorithm with given accuracy parameter $\varepsilon \in (0,1)$, confidence $\delta \in (0,1)$ and level function $m(\cdot)$. Suppose also that $\mu(\cdot)$ is a failure function. Then the cardinality function*

$$M(k,\varepsilon,\delta) =$$

$$\left\lceil \frac{1}{\varepsilon} \left(m(k) + \ln \frac{1}{\delta\mu(k)} + \sqrt{2m(k) \ln \frac{1}{\delta\mu(k)}} \right) \right\rceil$$

guarantees that with probability greater than $1-\delta$ all the probabilistic solutions obtained running the SPV algorithm have a probability of violation no greater than ε.

Proof:

Corollary 3.4 guarantees that the proposed choice for the cardinality function satisfies

$$\sum_{i=0}^{m(k)} \left(\begin{array}{c} M(k,\varepsilon,\rho) \\ i \end{array} \right) \varepsilon^i (1-\varepsilon)^{M(k,\varepsilon,\rho)-i} \leq \delta\mu(k), \; \forall k \geq 1.$$

The result then follows from a direct application of Property 3.3. ■

We notice here that the proposed cardinality function depends on the previous selection of the level function $m(\cdot)$ and the failure function $\mu(\cdot)$. Reasonable choices for these functions are $m(k) = \lfloor ak \rfloor$, where a is a non negative scalar and $\mu(k) = \frac{1}{\xi(\alpha)k^\alpha}$ where α is greater than one. We recall that this choice guarantees that $\mu(k)$ is a failure function. As it will be shown in the following section, the proposed level and failure functions allows us to recover, for the particular choice $a = 0$ the validation strategies proposed in (Dabbene et al., 2010) and (Oishi, 2007). We rewrite the resulting algorithm in the following corollary

Corollary 3.6 *Consider the following SPV algorithm*

(i) Set accuracy $\varepsilon \in (0,1)$, confidence $\delta \in (0,1)$ and scalars $a \geq 0$, $\alpha > 1$ equal to the desired levels. Set k equal to 1.

(ii) Obtain a candidate solution $\hat{\theta}_k$ to the robust optimization problem (A.4.1).

(iii) Set $m_k = \lfloor ak \rfloor$ and

$$M_k = \left\lceil \frac{1}{\varepsilon}\left(m_k + \ln\frac{\xi(\alpha)k^\alpha}{\delta} + \sqrt{2m_k \ln\frac{\xi(\alpha)k^\alpha}{\delta}} \right) \right\rceil.$$

(iv) Obtain validation set $\mathscr{V}_k = \{v^{(1)}, \dots, v^{(M_k)}\}$ drawing M_k i.i.d validation samples from $\mathscr{W}$ according to probability $\mathrm{Pr}_{\mathscr{W}}$.

(v) If $\sum_{\ell=1}^{M_k} g(\hat{\theta}_k, v^{(\ell)}) \leq m_k$, then $\hat{\theta}_k$ is a probabilistic solution.

(vi) Exit if the exit condition is satisfied.

(vii) k=k+1. Goto (ii).

Then, with probability greater than $1-\delta$ all the probabilistic solutions obtained running the SPV algorithm have a probability of violation no greater than ε.

Proof: The result is inferred directly from Theorem 3.5 using as level funtion $m(k) = \lfloor ak \rfloor$ and failure function $\mu(k) = \frac{1}{\xi(\alpha)k^\alpha}$. ■

Since the probabilistic properties of the algorithm presented in Corollary 3.6 are independent of the particular value of $\alpha > 1$, a reasonable choice for α is the one that minimizes the cardinality of the validation samples. In (Dabbene et al., 2010) it is shown that $\alpha = 1.1$ minimizes the term $\ln \xi(\alpha)k^\alpha$. Thus setting $\alpha = 1.1$ leads to a minimization of the cardinality of the resulting validation sets regardless of the particular level function $m(\cdot)$.

3.5 Comparison with other validation schemes

In this section, we provide comparisons with the validation schemes presented in (Oishi, 2007). This strategy has been successfully used in different randomized algorithms dealing with uncertain convex problems (Álamo, Tempo, Ramírez and Camacho, 2007), (Calafiore and Dabbene, 2007), (Oishi, 2007). We notice that setting $a = 0$ and $\alpha = 2$ in Corollary 3.6 we obtain $m(k) = 0$ for every iteration k and

$$M(k) = \left\lceil \frac{1}{\varepsilon} \ln\left(\frac{\xi(2)k^2}{\delta}\right) \right\rceil = \left\lceil \frac{1}{\varepsilon} \ln\left(\frac{\pi^2 k^2}{6\delta}\right) \right\rceil.$$

This is the same cardinality function presented in (Oishi, 2007) if one takes into account that for small values of ε, $-\ln(1-\varepsilon)$ can be approximated by ε. In the same way, $a = 0$ and $\alpha = 1.1$ lead to the cardinality function presented in (Dabbene et al., 2010).

Not allowing any failure in each validation test makes perfect sense for convex problems if the robust feasibility set

$$\Theta_r = \{ \ \theta \in \Theta \ : \ g(\theta, w) = 0 \text{ for all } w \in \mathscr{W} \ \}$$

is not empty. Under this assumption, the algorithm takes advantage of the validation samples that have not satisfied the specifications to obtain a new candidate solution. A common feature of the papers using this strict validation scheme is the proof that if Θ_r is not empty, a probabilistic solution (not necessarily belonging to the robust feasibility set Θ_r) is obtained in a finite number of iterations of the algorithm, see e.g., (Álamo, Tempo, Ramírez and Camacho, 2007), (Calafiore and Dabbene, 2007), (Oishi, 2007).

A very different situation is encountered when Θ_r is empty. We show by means of the following property that one should not use a strict validation scheme ($a = 0$) to address the case of empty robust feasible set because the algorithm might fail to obtain a probabilistic solution even if the set $\{ \ \theta \in \Theta \ : \ E(\theta) \leq \varepsilon \ \}$ is not empty.

Property 3.7 *Consider the SPV algorithm presented in Corollary 3.6 with* $a = 0$ *and* $\alpha > 1$. *Suppose that* $E(\theta) \geq \mu > 0$ *for all* $\theta \in \Theta$. *Then the SPV algorithm does not find any probabilistic solution in the first N iterations of the algorithm with probability greater than*

$$1 - \left(\frac{\delta}{\xi(\alpha)} \right)^{\frac{\mu}{\varepsilon}} \Phi(\frac{\alpha\mu}{\varepsilon}, \lceil \log_2 N \rceil),$$

where given the scalar $s > 0$ *and the integer* $t \geq 0$ *the function* $\Phi(s,t)$ *is*

$$\Phi(s,t) := \begin{cases} \dfrac{1 - 2^{(1-s)(t+1)}}{1 - 2^{1-s}} & \textit{if } s \neq 1 \\ \\ t+1 & \textit{otherwise.} \end{cases}$$

Proof: We notice that $a = 0$ implies that at iteration k the algorithm classifies a candidate solution $\hat{\theta}_k$ as a probabilistic solution only if it satisfies the constraint $g(\hat{\theta}_k, v^{(k)}) = 0$, $k = 1, \ldots, M_k$ where $\{v^{(1)}, \ldots, v^{(M_k)}\}$ is the randomly obtained validating set $\mathscr{V}_k$. Since $E(\theta) \geq \mu$ for all $\theta \in \Theta$ and $a = 0$, the probability of classifying a candidate solution as a probabilistic solution at iteration k is not greater than

$$\begin{array}{rcl} (1-\mu)^{M_k} & = & \mathrm{e}^{M_k \ln(1-\mu)} \\ & < & \mathrm{e}^{-\mu M_k} \\ & \leq & \mathrm{e}^{-\frac{\mu}{\varepsilon} \ln\left(\frac{\xi(\alpha)k^\alpha}{\delta}\right)} \\ & = & \left(\dfrac{\delta}{\xi(\alpha)k^\alpha}\right)^{\frac{\mu}{\varepsilon}}. \end{array}$$

Therefore, the probability of providing a probabilistic solution at any of the first N iterations of the algorithm is smaller than

$$\sum_{k=1}^{N} \left(\frac{\delta}{\xi(\alpha)k^\alpha}\right)^{\frac{\mu}{\varepsilon}} = \left(\frac{\delta}{\xi(\alpha)}\right)^{\frac{\mu}{\varepsilon}} \sum_{k=1}^{N} \left(\frac{1}{k^\alpha}\right)^{\frac{\mu}{\varepsilon}}.$$

Denoting $s = \frac{\alpha\mu}{\varepsilon}$ and taking into account Property 3.11 in the appendix we have

$$\sum_{k=1}^{N} \left(\frac{1}{k^\alpha}\right)^{\frac{\mu}{\varepsilon}} = \sum_{k=1}^{N} \frac{1}{k^s} \leq \Phi(s, \lceil \log_2 N \rceil).$$

We conclude that the probability of not finding any probabilistic solution in the first N iterations of the algorithm is smaller than

$$1 - \left(\frac{\delta}{\xi(\alpha)}\right)^{\frac{\mu}{\varepsilon}} \Phi(\frac{\alpha\mu}{\varepsilon}, \lceil \log_2 N \rceil).$$

■

Example 3.8 *Suppose that* $\Theta = [0,1]$, $\mathscr{W} = [-0.08, 1]$, $\varepsilon = 0.1$, $\delta = 10^{-4}$ *and that*

$$g(\theta, w) = \begin{cases} 0 & \textit{if } \theta \leq w \\ 1 & \textit{otherwise.} \end{cases}$$

Suppose also that $\Pr_{\mathscr{W}}$ *is the uniform distribution. It is clear that* $\theta = 0$ *minimizes the probability of violation and satisfies* $E(0) = \frac{0.08}{1.08} > 0.074$*. From here we obtain*

$$E(\theta) \geq 0.074 = \mu \textit{ for all } \theta \in \Theta.$$

Consider now the choice $\alpha = 1.1$ *and a maximum number of iterations N equal to* 10^6*. We infer from Property 3.7 that regardless of the strategy used to obtain candidate solutions the choice* $a = 0$ *and* $\alpha = 1.1$ *in Corollary 3.6 does not find any probabilistic solution with probability greater than* 0.98*. The choice* $\alpha = 2$ *leads to a probability greater than* 0.99*. This illustrates that a strict validation scheme is not well suited for this robust design problem.* ■

The next result states that the validation scheme presented in this chapter obtains under minor technical assumptions a probabilistic solution with probability one.

Property 3.9 *Consider an SPV algorithm with given accuracy parameter $\varepsilon \in (0,1)$, confidence $\delta \in (0,1)$ and level function $m(\cdot)$. Suppose that*

(i) $\mu(\cdot)$ *is a failure function.*

(ii) The cardinality function $M(k,\varepsilon,\delta)$ *is given by*

$$\left\lceil \frac{1}{\varepsilon}\left(m(k) + \ln\frac{1}{\delta\mu(k)} + \sqrt{2m(k)\ln\frac{1}{\delta\mu(k)}} \right) \right\rceil .$$

(iii) There exits an integer k^**, scalars* $\mu \in (0,1)$ *and* $p \in (0,1)$ *such that at every iteration* $k > k^*$ *a candidate solutions* $\hat{\theta}_k$ *satisfying* $E(\hat{\theta}_k) \leq \mu < \varepsilon$ *is obtained with probability greater than p.*

(iv) $\lim\limits_{k\to\infty} \frac{1}{m(k)} \ln \frac{1}{\delta\mu(k)} = 0$

then the SPV algorithm obtains with probability 1 a probabilistic solution in a finite number of iterations.

Proof:

From the assumption

$$\lim_{k\to\infty} \frac{1}{m(k)} \ln \frac{1}{\delta\mu(k)} = 0$$

we infer that

$$\begin{aligned} \lim_{k\to\infty} \frac{M(k)}{m(k)} &= \lim_{k\to\infty} \frac{1}{\varepsilon}\left(1 + \frac{1}{m(k)} \ln \frac{1}{\delta\mu(k)} \right. \\ &\qquad \left. + \sqrt{2\frac{1}{m(k)} \ln \frac{1}{\delta\mu(k)}} \right) \\ &= \frac{1}{\varepsilon}. \end{aligned}$$

This implies, along with the assumption $\mu < \varepsilon$, that there is $\tilde{k}$ such that

$$\mu < \frac{m(k)}{M(k)}, \text{ for every } k > \tilde{k}.$$

We conclude that the algorithm provides candidate solutions $\hat{\theta}_k$ satisfying

$$E(\hat{\theta}_k) \leq \mu < \frac{m(k)}{M(k)} \tag{3.2}$$

for every $k \geq \max\{k^*, \tilde{k}\}$ with probability no smaller than p. The validation test is satisfied if

$$\sum_{\ell=1}^{M(k)} g(\hat{\theta}_k, v^{(\ell)}) \leq m(k),$$

or equivalently, if

$$\frac{1}{M(k)} \sum_{\ell=1}^{M(k)} g(\hat{\theta}_k, v^{(\ell)}) \leq \frac{m(k)}{M(k)}.$$

We notice that $\frac{1}{M(k)} \sum_{\ell=1}^{M(k)} g(\hat{\theta}_k, v^{(\ell)})$ is the empirical mean associated to $g(\hat{\theta}_k, v)$ (Tempo et al., 2005). Moreover, recall that the probability of obtaining an empirical mean greater than the actual probability of violation is smaller than $\frac{1}{2}$. We therefore infer from equation (3.2) that the probability of classifying a candidate solution as a probabilistic one is no smaller than $\frac{p}{2}$ for every iteration $k > \max\{k^*, \tilde{k}\}$. Since $\frac{p}{2} > 0$ we conclude that the algorithm obtains a probabilistic solution with probability 1. ■

3.6 Application to non-sequential randomized algorithms

Motivated by the results presented in this chapter we present the following SPV algorithm which takes advantages of the theoretical results obtained in the literature of non-sequential randomized algorithms, see e.g., (Álamo et al., 2009), (Campi and Garatti, 2011), (Calafiore, 2010).

(i) Set accuracy $\varepsilon \in (0,1)$, confidence $\delta \in (0,1)$, and scalars $a > 0$ and $\alpha > 1$, equal to the desired levels.

(ii) Set k equal to 1 and $\mathscr{W}_1$ equal to the empty set.

(iii) Set $m_k = \lfloor ak \rfloor$ and

$$M_k = \left\lceil \frac{1}{\varepsilon} \left(m_k + \ln \frac{\xi(\alpha)k^\alpha}{\delta} + \sqrt{2m_k \ln \frac{\xi(\alpha)k^\alpha}{\delta}} \right) \right\rceil.$$

(iv) Obtain, if possible, a candidate suboptimal feasible solution $\hat{\theta}_k$ to the optimization problem

$$\min_{\theta \in \Theta} J(\theta) \text{ subject to } \sum_{w \in \mathscr{W}_k} g(\theta, w) \leq (\text{card } \mathscr{W}_k) \frac{m_k}{M_k}.$$

(v) Obtain validation set $\mathscr{V}_k = \{v^{(1)}, \ldots, v^{(M_k)}\}$ drawing M_k i.i.d validation samples from $\mathscr{W}$ according to probability $\Pr_{\mathscr{W}}$.

(vi) If a feasible solution $\hat{\theta}_k$ was found at step (iv) then classify it as a probabilistic solution if

$$\sum_{\ell=1}^{M_k} g(\hat{\theta}_k, v^{(\ell)}) \leq m_k.$$

(vii) Exit if the exit condition is satisfied.

(viii) $\mathscr{W}_{k+1} = \mathscr{W}_k \bigcup \mathscr{V}_k$. $k = k+1$. Goto (iii).

Under rather general assumptions like finite VC-dimension of $g(\cdot,\cdot)$ (Álamo et al., 2009) or convexity of the optimization problem with respect to the design parameter θ (see (Campi and Garatti, 2011) and (Calafiore, 2010)), we have that the feasible solutions obtained at step (iii) of the algorithm have a probability of violation smaller than $\frac{m_k}{M_k} < \varepsilon$ with a probability that tends to 1 with the cardinality of $\mathscr{W}_k$. From this we conclude that if the optimization problem at step (iii) is feasible with probability greater than 0, the proposed algorithm satisfies the assumptions of Property 3.9 and therefore provides a probabilistic solution that meets accuracy ε and confidence δ. The main advantage of the proposed algorithm with respect to the non-sequential algorithms available in the literature is that no explicit bound on the number of samples is required. This might lead to a substantial reduction of the number of required samples specially in the case of non-convex uncertain problems.

Illustrative example:

Suppose that $\Theta = [0,1]$, $\mathscr{W} = [-0.1, 0.95]$, $\varepsilon = 0.1$, $\delta = 10^{-4}$ and that

$$g(\theta, w) = \begin{cases} 0 & \text{if } \theta \leq w \\ 1 & \text{otherwise.} \end{cases}$$

Suppose also that $\Pr_{\mathscr{W}}$ is the uniform distribution. It is clear that $\theta = 0$ minimizes the probability of violation and satisfies the constraint $E(\theta) \leq \varepsilon = 0.1$ since $E(0) = \frac{0.1}{1.05} = 0.0952$. Consider now the choice $\alpha = 1.1$. Since $E(\theta) \geq 0.0952 \geq \frac{\varepsilon}{1.1} = 0.0909$ for every $\theta \in \Theta$ we infer from Property 3.7 that regardless of the strategy used to obtain candidate solutions and the number of iterations, the choice $a = 0$ and $\alpha = 1.1$ in Corollary 3.6 never founds a probabilistic solution with probability no smaller than $1 - \frac{5}{2}\delta^{\frac{1}{\sqrt{1.1}}} > 0.999$. A similar result is obtained for the choice $a = 0$ and $\alpha = 2$. This illustrates that a strict validation scheme is not well suited for this robust design problem. ■

3.7 Numerical example

As in previous chapter, the objective of this numerical example is to obtain probabilistic upper and lower bounds of a given time function $y : \mathscr{W} \to R$ of the form

$$y(w) = [A(1+\frac{1}{2}t^2)\sin(7t+0.5)+B]\mathrm{e}^{-\frac{3}{2}t},$$

where $w \in \mathscr{W}$.

The uncertainty set $\mathscr{W}$ is

$$\mathscr{W} = \left\{ w = [t \quad A \quad B]^T, t \in [0,1],\, A \in [1,3],\, B \in [1,3] \right\}.$$

For a given order d, we define the regressor $\varphi_d : \mathscr{W} \to R^{d+1}$ as

$$\varphi_d(w) = \varphi_d([t \quad A \quad B]^T) = \begin{bmatrix} 1 & t & t^2 & \cdots & t^d \end{bmatrix}^T.$$

The objective of this example is to find a parameter vector $\theta = [\gamma_d, \lambda_d]^T$, $\gamma_d \in R^{d+1}$ and $\lambda_d \in R^{d+1}$ such that, with probability no smaller than $1-\delta$,

$$\mathrm{Pr}_{\mathscr{W}}\{\, w \in \mathscr{W} \;:\; |y(w) - \gamma_d^T \varphi_d(w)| \geq \lambda_d^T |\varphi_d(w)| \,\} \leq \eta.$$

The vector $|\varphi_d(w)|$ is obtained from the absolute values of $\varphi_d(w)$. The binary function $g : \Theta \times \mathscr{W} \to \{0,1\}$, is defined as

$$g(\theta,w) := \begin{cases} 0 & \text{if } \theta \text{ meets design specifications for } w \\ 1 & \text{otherwise,} \end{cases}$$

where "design specifications" means satisfying the following constraint:

$$|y(w) - \gamma_d^T \varphi_d(w)| \leq \lambda_d^T |\varphi_d(w)|$$

for randomly generated samples $w \in \mathscr{W}$.

For the numerical computations, we take $\delta = 10^{-6}$ and $\eta = 0.05$. We address the problem from the SPV approach.

3.7.1 SPV algorithm

We again take advantage of the result of Subsection 2.4.1 and take $d = 5$ as the order of the approximation polynomial. Following the SPV algorithm approach, we begin setting

$\eta = 0.05$, confidence $\delta = 10^{-6}$, scalars $a = 0.75$, $\alpha = 2$ and iteration index $k = 1$. The initial $\mathscr{W}_k$ is a set of 400 samples drawn from $\mathscr{W}$ according to probability $\Pr_{\mathscr{W}}$.

(i) A candidate solution $\hat{\theta}_k$ to the problem

$$\begin{aligned} \min_{\gamma_d, \lambda_d} \quad & \lambda_d^T \sum |\varphi_d(t)| \\ s.t. \quad & |y(w) - \gamma_d^T \varphi_d(w)| \leq \frac{1}{1.2} \lambda_d^T |\varphi_d(w)|, \; \forall w \in \mathscr{W}_k \end{aligned}$$

is obtained.

(ii) Set $m_k = \lfloor a(k) \rfloor$ and

$$M_k = \left\lceil \frac{1}{\eta} \left(m_k + \ln \frac{\xi(\alpha) k^\alpha}{\delta} + \sqrt{2 m_k \ln \frac{\xi(\alpha) k^\alpha}{\delta}} \right) \right\rceil .$$

(iii) Obtain validation set $\mathscr{V}_k = \{v^{(1)}, \ldots, v^{(M_k)}\}$ drawing M_k i.i.d. validation samples from $\mathscr{W}$ according to the probability $\Pr_{\mathscr{W}}$.

(iv) If $\sum_{\ell=1}^{M_k} g(\hat{\theta}_k, v^{(\ell)}) \leq m_k$, then $\hat{\theta}_k$ is a probabilistic solution. The failure function is $g(\hat{\theta}_k, v^{(\ell)})$

$$g(\hat{\theta}_k, v^{(\ell)}) := \\ \begin{cases} 0 & \text{if } |y(w) - \gamma_d^T \varphi_d(w)| \leq \lambda_d^T |\varphi_d(w)| \\ 1 & \text{otherwise.} \end{cases}$$

(v) Exit if the exit condition is satisfied.

(vi) $k = k + 1$. $\mathscr{W}_k = \mathscr{W}_k \bigcup V_k$. Goto (i).

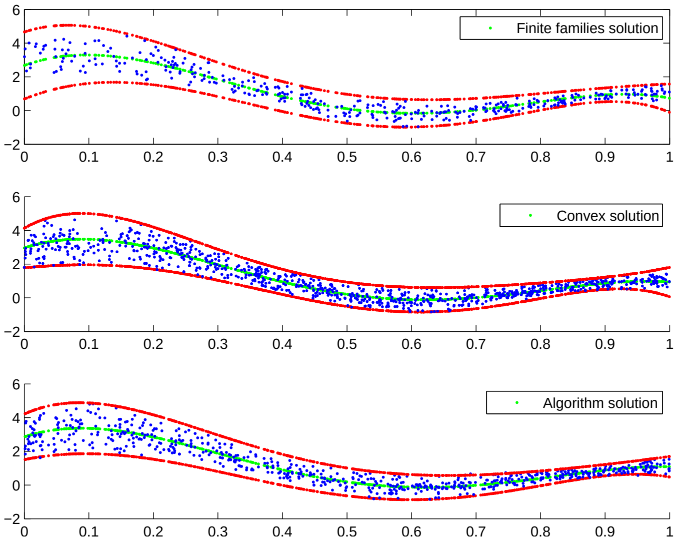


Figure 3.1: Initial data set and envelope of the set of solutions.

Figure (3.1) shows the initial data set generated using the procedure described above, and the envelope that contains all the solution polynomials. Using this strategy, 686 data are required. We obtained a performance index of 0.8877, slightly larger than that obtained by the finite families strategy.

The number of failures obtained in the last step of the algorithm is $m = 5$, being the empirical probability of failure $\eta = \frac{m}{M_k} = 0 < 0.05$.

We remark that if we set $a = 0$ in the algorithm, there are no allowed failures and this coincides with the approach studied in (Oishi, 2007). In this case, the algorithm did not find a solution for $\eta = 0.05$ and $M_k < 30000$.This is consistent with the results on Section 3.5.

In Table 1 the results of the three approaches are compared for different values of η.

η	N_{finite}	N_{convex}	N_{SPV}
0.2	190	212	582
0.1	378	423	543
0.05	754	845	686
0.02	1884	2113	2230
0.01	2366	4225	4061

3.8 Conclusions

In this chapter we presented a general class of randomized algorithms based on probabilistic validation. We provided a strategy to adjust the cardinality of the validation sets to guarantee that the obtained solutions meet the probabilistic specifications. The proposed strategy is compared with other schemes from the literature and it has been shown that a strict validation strategy in which the design parameter has to satisfy the constraints for all the elements of the validation set might not be appropriate in some situations. Finally we proved that the proposed approach does not suffer from this limitation because it allows the use of non strict validation tests.

A randomized sequential algorithm that permits approaching optimization problems subject to uncertainty has been introduced. This algorithm is based on a strategy that iteratively adjusts the sample size of the training and validation sets. The main advantage of this proposal is that the algorithm leads to significant improvements in terms of the required sample size. The results allow us to address non-convex optimization problems with uncertainties, which is of great relevance in the context of robust control design.

3.9 Appendix

Property 3.10 *Suppose that N is a positive integer and that s is a strictly positive scalar. Then,*

$$\sum_{k=1}^{N} \frac{1}{k^s} \leq \Phi(s, \lceil \log_2 N \rceil)$$

where, given $s \geq 0$ and the integer $t \geq 0$,

$$\Phi(s,t) := \begin{cases} \dfrac{1-2^{(1-s)(t+1)}}{1-2^{1-s}} & \textit{if } s \neq 1 \\ t+1 & \textit{otherwise}. \end{cases}$$

Property 3.11 *Suppose that N is a positive integer and that s is a strictly positive scalar. Then,*

$$\sum_{k=1}^{N} \frac{1}{k^s} \leq \Phi(s, \lceil \log_2 N \rceil)$$

where, given $s \geq 0$ and the integer $t \geq 0$,

$$\Phi(s,t) := \begin{cases} \dfrac{1-2^{(1-s)(t+1)}}{1-2^{1-s}} & \textit{if } s \neq 1 \\ t+1 & \textit{otherwise}. \end{cases}$$

Proof: Given $N > 0$ and $s > 0$, define $t := \lceil \log_2(N) \rceil$ and $S(t) := \sum_{k=1}^{2^t} \frac{1}{k^s}$. With these definition we have

$$\sum_{k=1}^{N} \frac{1}{k^s} \leq \sum_{k=1}^{2^t} \frac{1}{k^s} = S(t).$$

In what follows we show that $S(t) \leq 1 + 2^{1-s}S(t-1)$ for every integer t greater than 0. Since $S(0) = 1$ and $S(1) = 1 + 2^{-s}$, the inequality is clearly satisfied for $t = 1$. We now prove the inequality for t greater than 1.

$$\begin{aligned}
S(t) &= \sum_{k=1}^{2^t} \frac{1}{k^s} = \sum_{k=1}^{2^{t-1}} \left[\frac{1}{(2k)^s} + \frac{1}{(2k-1)^s} \right] \\
&= 2^{-s} \sum_{k=1}^{2^{t-1}} \frac{1}{k^s} + \sum_{k=1}^{2^{t-1}} \frac{1}{(2k-1)^s} \\
&= 2^{-s}S(t-1) + 1 + \sum_{k=2}^{2^{t-1}} \frac{1}{(2k-1)^s} \\
&\leq 2^{-s}S(t-1) + 1 + \sum_{k=2}^{2^{t-1}} \frac{1}{(2k-2)^s} \\
&= 2^{-s}S(t-1) + 1 + 2^{-s} \sum_{k=2}^{2^{t-1}} \frac{1}{(k-1)^s} \\
&= 2^{-s}S(t-1) + 1 + 2^{-s} \sum_{k=1}^{2^{t-1}-1} \frac{1}{k^s} \\
&\leq 2^{-s}S(t-1) + 1 + 2^{-s} \sum_{k=1}^{2^{t-1}} \frac{1}{k^s} \\
&= 2^{-s}S(t-1) + 1 + 2^{-s}S(t-1) \\
&= 1 + 2^{1-s}S(t-1).
\end{aligned}$$

We have therefore proved the inequality $S(t) \leq 1 + 2^{1-s}S(t-1)$ for every integer t greater than 0. Using this inequality in a recursive way with $S(0) = 1$ we obtain

$$S(t) \leq \sum_{k=0}^{t} 2^{(1-s)k} = \Phi(s,t).$$

This proves the result. ■

Chapter 4

A sequentially optimal R.A. for robust LMI feasibility problems

4.1 Introduction

This chapter proposes a randomized algorithm for feasibility of uncertain LMIs (Álamo et al., 2013). The algorithm is based on the solution of a sequence of semidefinite optimization problems involving a reduced number of constraints. A bound of the maximum number of iterations required by the algorithm is given. Analogies and differences with the gradient and localization methods are discussed. Finally, the performance and behaviour of the algorithm are illustrated by means of a numerical example.

The use of randomized algorithms (see (Tempo et al., 2005)) has attracted the attention of the control community in the last few years. One of the reasons for this widespread interest is that randomization can be used to circumvent the NP-hard nature of a large number of robust control problems (Nemirovskii, 1993),(Poljak and Rohn, 1993). Randomization allows one to obtain a solution that satisfies the constraints of a given robustness problem for most of the possible realizations of the uncertainty. This concept of approximate feasibility has been introduced in the context of robust control in (Barmish and Scherbakov, 2002). Under relatively mild assumptions, the randomized methods are able to compute (in polynomial time) an approximate solution to a robust problem. The measure of the set of original constraints that are violated by the approximate solution can be made smaller than any pre-specified quantity.

The randomized gradient approach presented in (Polyak and Tempo, 2001),(Calafiore and Polyak, 2001),(Fujisaki et al., 2003) and (Liberzon and Tempo, 2004) finds a solution to a robust problem involving linear matrix inequalities in a finite number of iterations with probability one, if a strong feasibility condition holds. These gradients algorithms are based on an iterative scheme where the current solution is updated towards a descent direction obtained by a random gradient of a suitable feasibility violation function.

Another important class or randomized methods are based on probabilistic versions of standard localization methods. These localization methods have better theoretical convergence properties than the gradient ones. Among them one finds the probabilistic ellipsoid method (Kanev et al., 2003),(Oishi, 2003) and the probabilistic analytic center cutting plane method (Calafiore and Dabbene, 2006).

In the context of robust optimization, the scenario approach also plays a relevant role. It is shown in (Calafiore and Campi, 2005) and (Calafiore and Campi, 2006) that by appropriate sampling of the constraints one obtains a standard convex optimization problem (the scenario) whose solution is approximately feasible for the original (usually infinite) set of constraints, i.e., the measure of the set of original constraints that are violated by the scenario solution decreases to zero as the number of samples is increased.

In some sense, the scenario approach, the gradient and the ellipsoid methods have a very different nature. The scenario approach obtains an approximate solution to a robust optimization problem solving a simple optimization problem with a large number of constraints. On the other hand, the gradient method and the ellipsoid algorithm obtain an approximate solution to a robust feasibility problem in a sequential way, by means of a considerable number of iterations in which the candidate solution is updated by means of a simple rule so that no optimization is really required.

In this chapter we present a randomized algorithm that addresses the problem of obtaining a feasible robust solution to a possibly uncountable number of linear matrix inequalities. The presented algorithm does not belong to any of the aforementioned strategies. The algorithm requires a finite number of iterations to converge. It addresses the robust feasibility problem by means of the solution of a sequence of relatively simple optimization problems. The proposed algorithm, as the gradient method and the ellipsoid algorithm, has a sequential nature. However, instead of using a simple updating rule, each candidate solution is updated by means of an optimization problem involving a reduced number of constraints obtained from the original constraints of the problem. One of the advantages of the proposed algorithm is that it is capable of determining the non feasibility of a given robust feasibility problem. Our numerical experience shows that the algorithm performs satisfactorily: in an affordable number of iterations it obtains an (approximately) feasible solution in case of feasibility, or it detects that the problem is not feasible.

The chapter is organized as follows: Section 4.2 presents some notations. Section 4.3 details the class of feasibility problems under consideration. In Section 4.4 it is shown that the feasibility problem can be rewritten as an optimization problem. The notion of normalized valid cuts is introduced in Section 4.5. The proposed algorithm is presented in Section

4.6. Section 4.7 addresses the problem of checking the robust feasibility of a given candidate solution. The relationship with other randomized algorithms is discussed in Section 4.8. Some numerical results are presented in Section 4.9. The chapter draws to a close with a concluding section.

4.2 Notation

- Given vector $\mathbf{x}$, x_i denotes its i-th component.
- Given symmetric matrix A, $\bar{\lambda}(A)$ denotes its largest eigenvalue; $A < 0$ denotes that A is negative definite; given symmetric matrices A and B, $A < B$ denotes that $A - B$ is negative definite.
- The Euclidean norm is denoted as $\|\cdot\|_2$.
- For $x \in \mathbb{R}$, $x > 0$, $\lceil x \rceil$ denotes the minimum integer greater than or equal to x.

4.3 Problem statement

In this chapter we address the solution of the following robust LMI problem: find $\mathbf{z} \in \mathbb{R}^m$ such that

$$A(\mathbf{z}, w) < 0, \ \ \forall w \in W \tag{4.1}$$

where W is a compact set, $A(\mathbf{z}, w) = \sum\limits_{i=1}^{m} z_i A_i(w)$, and $A_i(w) = A_i^\top(w) \in \mathbb{R}^{q \times q}$, $i = 1, \ldots, m$, $\forall w \in W$. It will be assumed that each of the entries of the matrices $A_i(w)$, $i = 1, \ldots, m$ are bounded for every $w \in W$. If $\mathbf{z}$ satisfies $A(\mathbf{z}, w) < 0$, $\forall w \in W$ then $\mathbf{z}$ is said to be a robust feasible solution.

The set of feasible solutions to the robust LMI problem (4.1) will be denoted as $\mathscr{D}$,

$$\mathscr{D} = \{ \, \mathbf{z} \in \mathbb{R}^m \ : \ A(\mathbf{z}, w) < 0, \ \ \forall w \in W \, \}.$$

Note that the dependence of $A(\mathbf{z}, w)$ with respect to $\mathbf{z}$ is linear. This implies that $A(\mu\mathbf{z}, w) = \mu A(\mathbf{z}, w)$, $\forall \mu \in \mathbb{R}$. From this it is clear that if $\mathbf{z}$ is a robust feasible solution then $\bar{\mathbf{z}} = \frac{\mathbf{z}}{\sqrt{\mathbf{z}^\top \mathbf{z}}}$ is also a robust feasible solution and $\bar{\mathbf{z}}^\top \bar{\mathbf{z}} = 1$. This means that in order to analyze the robust feasibility of problem (4.1), it suffices to analyze if there exists a robust feasible solution in the unit sphere $\{\, \mathbf{z} \in \mathbb{R}^m \,:\, \mathbf{z}^\top \mathbf{z} \leq 1 \,\}$. With this in mind, it is pertinent to introduce the notion of ε-feasibility.

Definition 4.1 *Given $\varepsilon > 0$, robust feasibility problem (4.1) is said to be ε-feasible if there is $\mathbf{z} \in \mathbb{R}^m$ such that*

$$\begin{array}{rcl} A(\mathbf{z},w) & \leq & -\varepsilon \mathbf{I}, \ \forall w \in W \\ \mathbf{z}^\top \mathbf{z} & \leq & 1. \end{array}$$

Next it is shown that a robust feasibility problem involving matrix inequalities in which the dependence with respect to the decision variable is affine can be reformulated as a robust LMI problem. Consider the problem of finding $\mathbf{x} \in \mathbb{R}^n$ such that

$$F(\mathbf{x},w) < 0, \ \ \forall w \in W \tag{4.2}$$

with $F(\mathbf{x},w) = F_0(w) + \sum\limits_{i=1}^{n} x_i F_i(w)$ and where $F_i(w) = F_i^\top(w)$, $i = 0,\ldots,n$, $\forall w \in W$. Note that this class of robust feasibility problems appears very often in the context of robust control (see e.g. (Boyd et al., 1994)).

Suppose that $\mu > 0$. Then, the constraints given in equation (4.2) are equivalent to

$$\mu F_0(w) + \sum_{i=1}^{n} \mu x_i F_i(w) < 0, \ \forall w \in W.$$

Denote now $\mu x_i = z_i$, $i = 1,\ldots,n$ and $\mu = z_{n+1}$ and consider the following robust LMI feasibility problem: find $\mathbf{z} \in \mathbb{R}^{n+1}$ such that

$$A(\mathbf{z},w) = \begin{bmatrix} z_{n+1} F_0(w) + \sum\limits_{i=1}^{n} z_i F_i(w) & 0 \\ 0 & -z_{n+1} \end{bmatrix} < 0, \tag{4.3}$$
$$\forall w \in W.$$

Note that feasibility problems (4.2) and (4.3) are equivalent: if $\mathbf{x} \in \mathbb{R}^n$ is a feasible solution for problem (4.2) then $\mathbf{z} = [\mathbf{x}^\top \ 1]^\top$ is a feasible solution for problem (4.3). Conversely, if

$\mathbf{z}$ is a feasible solution to problem (4.3) then $z_{n+1} > 0$ and $\mathbf{x} = \frac{1}{z_{n+1}}[z_1\ z_2\ \ldots\ z_n]^\top$ is a feasible solution to problem (4.2). It is clear that feasibility problem (4.3) belongs to the class of feasibility problems given by (4.1). This proves that any robust feasibility problem in which the dependence with respect the decision variable is affine can be rewritten as a robust feasibility problem in which the dependence is linear.

4.4 A related optimization problem

As it is shown in the following theorem, the feasibility problem (4.1) can be cast as an optimization problem.

Theorem 4.2 *Denote γ^* the solution to the following minimization problem*

$$\begin{array}{rcl} \gamma^* = & \min\limits_{\mathbf{z},\gamma} & \gamma \\ & s.t. & A(\mathbf{z},w) \leq \gamma \mathrm{I}, \ \forall w \in W \\ & & \mathbf{z}^\top \mathbf{z} \leq 1. \end{array} \tag{4.4}$$

Then

(i) The robust LMI problem (4.1) is feasible if and only if $\gamma^* \neq 0$.

(ii) The robust LMI problem is not ε-feasible for every $\varepsilon > -\gamma^*$.

Proof:

(i) First, it will be shown that $\gamma^* \neq 0$ implies that problem (4.1) is feasible. Note that optimization problem (4.4) is always feasible ($\mathbf{z} = 0$, $\gamma = 0$ is a feasible solution). This means that $\gamma^* \leq 0$. From this and the assumption $\gamma^* \neq 0$ it is inferred that γ^* is strictly smaller than zero. That is, if the minimum of the minimization problem is attained at $(\mathbf{z}^*, \gamma^*)$ then $\gamma^* < 0$ and

$$A(\mathbf{z}^*, w) \leq \gamma^* \mathrm{I} < 0, \ \forall w \in W.$$

From this it is concluded that $\mathbf{z}^*$ is a feasible solution to problem (4.1).

To finish the proof of this first claim, we need to show that $\gamma^* = 0$ implies that problem (4.1) is unfeasible. This fact will be proved showing that feasibility of problem (4.1) implies $\gamma^* < 0$.

Suppose that there is $\bar{\mathbf{z}}$ that satisfies the robust constraints of problem (4.1). That is, $\bar{\mathbf{z}}$ belongs to $\mathscr{D}$. The strict inequality of problem (4.1) implies that $\bar{\mathbf{z}} \neq 0$. Bearing in mind the linear dependence with respect $\mathbf{z}$, it results that $\rho\bar{\mathbf{z}} \in D$ for all $\rho > 0$, $\rho \in \mathbb{R}$. In particular, $\hat{\mathbf{z}} = \left(\frac{1}{\sqrt{\bar{\mathbf{z}}^\top \bar{\mathbf{z}}}}\right)\bar{\mathbf{z}} \in \mathscr{D}$. Note that $\hat{\mathbf{z}} \in \mathscr{D}$ and $\hat{\mathbf{z}}^\top\hat{\mathbf{z}} = 1$. From this and the compactness of W it is inferred that there exits $\hat{\gamma} < 0$ such that

$$\begin{aligned} A(\hat{\mathbf{z}}, w) &\leq \hat{\gamma}\mathrm{I}, \ \forall w \in W \\ \hat{\mathbf{z}}^\top\hat{\mathbf{z}} &\leq 1. \end{aligned}$$

From this we conclude that $\gamma^* < 0$. ■

(ii) This fact stems directly from the definition of ε-feasibility. Suppose that $\varepsilon > -\gamma^*$ and that the problem is ε-feasible. Then there is $\hat{\mathbf{z}}$ such that $A(\hat{\mathbf{z}}, w) \leq -\varepsilon\mathrm{I}$, $\forall w \in W$ and $\hat{\mathbf{z}}^\top\hat{\mathbf{z}} \leq 1$. Thus, $\gamma^* \leq -\varepsilon$. This contradicts the assumption $\varepsilon > -\gamma^*$.

4.5 Normalized valid cuts

In this section the notion of normalized valid cuts is introduced. The results of this section play a fundamental role when analyzing the convergence of the proposed algorithm.

Given $w \in W$,

$$\begin{aligned} \bar{\lambda}(A(\mathbf{z}, w)) &= \max_{v \in \mathbb{R}^q, v^\top v = 1} v^\top A(\mathbf{z}, w) v \\ &= \max_{v \in \mathbb{R}^q, v^\top v = 1} \sum_{i=1}^{m} (v^\top A_i(w) v) z_i. \end{aligned}$$

This proves the following property.

Property 4.3 *Denote*

$$C(w) = \{\mathbf{c} \in \mathbb{R}^m \ : \ \textit{there is} \ \ v \in \mathbb{R}^q \ \textit{such that}$$

$$v^\top v = 1 \ \textit{and} \ c_i = v^\top A_i(w) v, \ i = 1, \ldots, m\}.$$

With this notation, it results that $A(\mathbf{z}, w) \leq \gamma\mathrm{I}$ *if and only if* $\max_{\mathbf{c} \in C(w)} \mathbf{c}^\top \mathbf{z} \leq \gamma$.

The union of all sets $C(w)$ will be denoted $\mathscr{C}$

$$\mathscr{C} = \bigcup_{w \in W} C(w).$$

$\mathscr{C}$ will be called the set of all normalized valid cuts. From this definition, and Property 4.3 it results that problem (4.4) can be rewritten as

$$\begin{array}{rll} \gamma^* = & \min\limits_{\mathbf{z},\gamma} & \gamma \\ & s.t. & \mathbf{c}^\top \mathbf{z} \leq \gamma, \ \forall \mathbf{c} \in \mathscr{C} \\ & & \mathbf{z}^\top \mathbf{z} \leq 1. \end{array} \tag{4.5}$$

As it is shown in the following property, the feasibility of a given vector $\mathbf{z}$ can be determined by means of the set of all the normalized valid cuts.

Property 4.4 *Vector* $\mathbf{z}$ *is a feasible solution to problem (4.1) if and only*

$$\mathbf{c}^\top \mathbf{z} < 0, \ \forall \mathbf{c} \in \mathscr{C}.$$

Proof: The proof stems directly from the fact that $A(\mathbf{z}, w)$ is negative definite for every $w \in W$ if and only if $v^\top \left(\sum\limits_{i=1}^{m} A_i(w) z_i \right) v < 0$ for every $w \in W$ and every v, $v^\top v = 1$. From the definition of $\mathscr{C}$, it is inferred that the last inequality is equivalent to $\mathbf{c}^\top \mathbf{z} < 0$, $\forall \mathbf{c} \in \mathscr{C}$. ■

Definition 4.5 *Vector* $\mathbf{h}$ *belongs to the convex hull of set* $\mathscr{C}$*, (denoted* $\mathrm{Co}\{\mathscr{C}\}$ *) if and only if there exists* $\mathbf{c}_1, \ldots, \mathbf{c}_p$ *and* $\lambda_1, \ldots, \lambda_p$ *such that*

$$\begin{array}{rcl} \mathbf{h} & = & \sum\limits_{i=1}^{p} \lambda_i \mathbf{c}_i \\ 1 & = & \sum\limits_{i=1}^{p} \lambda_i \\ \lambda_i & \geq & 0, \ i = 1, \ldots, p \\ \mathbf{c}_i & \in & \mathscr{C}, \ i = 1, \ldots, p. \end{array}$$

The following property states that a lower bound of γ^* is directly inferred from every element in $\text{Co}\{\mathscr{C}\}$.

Property 4.6 *We have*

$$\gamma^* \geq -\|\mathbf{h}\|_2, \quad \forall \mathbf{h} \in \text{Co}\{\mathscr{C}\}.$$

Proof:

Suppose that $\bar{\mathbf{h}}$ belongs to $\text{Co}\{\mathscr{C}\}$. It will be proved that $\gamma^* \geq -\|\bar{\mathbf{h}}\|_2$. From $\bar{\mathbf{h}} \in \text{Co}\{\mathscr{C}\}$ it is inferred that there exists $\mathbf{c}_1, \ldots, \mathbf{c}_p$ and $\lambda_1, \ldots, \lambda_p$ such that: $\bar{\mathbf{h}} = \sum\limits_{i=1}^{p} \lambda_i \mathbf{c}_i$, $1 = \sum\limits_{i=1}^{p} \lambda_i$, $\lambda_i \geq 0$, $\mathbf{c}_i \in \mathscr{C}$, $i = 1, \ldots, p$. From (4.5) it is inferred that

$$\begin{array}{rll} \gamma^* \geq \gamma_c^* = & \min\limits_{\mathbf{z},\gamma} & \gamma \\ & s.t. & \mathbf{c}_i^\top \mathbf{z} \leq \gamma, \ i = 1, \ldots, p \\ & & \mathbf{z}^\top \mathbf{z} \leq 1. \end{array}$$

Note that the inequalities $\mathbf{c}_i^\top \mathbf{z} \leq \gamma$, $i = 1, \ldots, p$ imply

$$\sum_{i=1}^{p} \lambda_i \mathbf{c}_i^\top \mathbf{z} \leq \sum_{i=1}^{p} \lambda_i \gamma = \gamma$$

$$\left(\sum_{i=1}^{p} \lambda_i \mathbf{c}_i \right)^\top \mathbf{z} = \bar{\mathbf{h}}^\top \mathbf{z} \leq \gamma.$$

Therefore,

$$\begin{array}{rll} \gamma^* \geq \gamma_c^* \geq \gamma_d^* = & \min\limits_{\mathbf{z},\gamma} & \gamma \\ & s.t. & \bar{\mathbf{h}}^\top \mathbf{z} \leq \gamma \\ & & \mathbf{z}^\top \mathbf{z} \leq 1. \end{array}$$

Note that the solution to the previous optimization problem is

$$\mathbf{z}_d^* = \left\{ \begin{array}{ll} \frac{-\bar{\mathbf{h}}}{\sqrt{\bar{\mathbf{h}}^\top \bar{\mathbf{h}}}} & \text{if } \ \bar{\mathbf{h}} \neq 0 \\ 0 & \text{if } \ \bar{\mathbf{h}} = 0 \end{array} \right. , \ \gamma_d^* = \bar{\mathbf{h}}^\top \bar{\mathbf{z}}_d^* = -\|\bar{\mathbf{h}}\|_2.$$

That is, it has been proved that

$$\gamma^* \geq \gamma_c^* \geq \gamma_d^* = -\|\bar{\mathbf{h}}\|_2.$$

■

4.6 Randomized algorithm

In this section, an algorithm to solve the robust feasibility problem (4.1) is presented. This algorithm constitutes the main contribution of the chapter. Consider a given integer $N_{max} \geq 1$ and a real number $\varepsilon \in (0,1)$ (the effect of N_{max} and ε in the performance of the algorithm will be clarified later). The proposed algorithm is detailed below.

Algorithm 1 *Feasibility problem.*

1. *Pick an element of W, denote it w_0 and solve the minimization problem*

$$\begin{array}{rcll} \gamma_0^* = & \min\limits_{\mathbf{z},\gamma} & \gamma & \\ & s.t. & A(\mathbf{z},w_0) & \leq \gamma \mathrm{I} \\ & & \mathbf{z}^\top \mathbf{z} & \leq 1. \end{array} \tag{4.6}$$

Denote $(\mathbf{z}_0^*, \gamma_0^*)$ the solution to this optimization problem. Make $\mathbf{h}_0 = \gamma_0^* \mathbf{z}_0^*$, $S_0 = \{w_0\}$ and $k = 0$.

2. *Check if $\mathbf{z}_k^*$ is a robust feasible solution to problem (4.1) (this might be done in a probabilistic way as it is shown in next section).*

3. *If $\mathbf{z}_k^*$ is a (probabilistic) robust feasible solution then STOP. Else, obtain $w_{k+1} \in W$ such that $\bar{\lambda}(A(\mathbf{z}_k^*, w_{k+1})) > 0$.*

4. *Make $S_{k+1} = S_k \bigcup w_{k+1}$. If the number of elements of S_{k+1} is greater than N_{max}, then eliminate from S_{k+1} the vector $\hat{w} \in S_{k+1}$ that minimizes $\bar{\lambda}(A(\mathbf{z}_k^*, w))$.*

5. *Solve the optimization problem*

$$\begin{array}{rcl} \gamma_{k+1}^* = & \min\limits_{\mathbf{z},\gamma} & \gamma \\ & s.t. & \mathbf{h}_k^\top \mathbf{z} \leq \gamma, \\ & & A(\mathbf{z},w) \leq \gamma \mathrm{I}, \ \forall w \in S_{k+1} \\ & & \mathbf{z}^\top \mathbf{z} \leq 1. \end{array} \tag{4.7}$$

Denote $(\mathbf{z}_{k+1}^*, \gamma_{k+1}^*)$ its solution. Make $\mathbf{h}_{k+1} = \gamma_{k+1}^* \mathbf{z}_{k+1}^*$.

6. *If $\gamma^*_{k+1} > -\varepsilon$ then the robust feasible problem 4.1 is not ε-feasible. STOP. Else, make $k = k+1$ and go to step 2.*

The following theorem states that in a finite number of iterations, the algorithm finds a (probabilistic) feasible solution or determines the non ε-feasibility of the problem. If a feasible solution is obtained, the nature of such a solution will depend on how step 2 of the algorithm is implemented. This question is addressed in the next section.

Theorem 4.7 *Suppose that $\max\limits_{\mathbf{c}\in\mathscr{C}} \|\mathbf{c}\|_2 < \sigma$ and $N_{max} \geq 1$. Then the proposed algorithm obtains a (probabilistic) robust feasible solution to problem (4.1), or determines that it is not ε-feasible in no more than*

$$k_{max} = \left\lceil \frac{4\sigma^2}{\varepsilon^2} \right\rceil + \left\lceil \left(\frac{4\sigma^2}{\varepsilon^2}\right) \ln\ln \frac{\sigma^2}{\varepsilon^2} \right\rceil$$

iterations.

Proof:

The proof of this theorem can be found in (Álamo, Tempo, Ramírez and Camacho, 2006*b*). The proof has not been included in here.

Next, a sketch of the proof is provided for the sake of completeness. The key point in the proof is that $\mathbf{h}_k \in \mathrm{Co}\{\mathscr{C}\}$, $k = 0,1,\ldots$, (see Theorem 2 in (Álamo et al., 2006*b*)). From this and the fact that $S_k \subseteq W$, $k = 0,1,\ldots$, it is inferred that

$$\gamma^* \geq \gamma^*_k,\ k = 0,1,\ldots$$

Thus, if $\gamma^*_k > -\varepsilon$ then $\gamma^* \geq \gamma^*_k > -\varepsilon$, which (according to Theorem 4.2) implies that the problem is not ε-feasible. This means that if the algorithm classifies the problem as non ε-feasible, then the problem is really not ε-feasible (this proves the correctness of the algorithm).

On the other hand, it is shown in Section IX of (Álamo et al., 2006*b*) that

$$\gamma^*_{k+1} \geq \gamma^*_k \left(1 - \frac{(\gamma^*_k)^2}{8\sigma^2}\right) \geq \gamma^*_k. \tag{4.8}$$

This means that if the algorithm does not find a feasible solution then there exists k_{max} such that $\gamma^*_{k_{max}} > -\varepsilon$. Therefore, the proposed algorithm obtains a (probabilistic) robust feasible solution, or determines that it is not ε-feasible in no more than k_{max} iterations. The bound on the number of iterations given in the theorem is a direct consequence of equation (4.8) and Property 10 of (Álamo et al., 2006*b*). ■

Remark 4.8 *Note than Theorem 4.7 holds for every $N_{max} \geq 1$. Unfortunately, the provided worst case number of iterations does not reflect the potential benefit of choosing $N_{max} > 1$. Our numerical experience shows that the number of iterations required by the algorithm strongly decreases with increasing values of N_{max} (see the numerical results presented in Section 4.9).*

4.6.1 Bounding the value of σ

Recall that each set $C(w)$ is compact ($C(w)$ is defined in Property 4.3). As W is also compact, it results that $\mathscr{C} = \sum\limits_{w \in W} C(w)$ is a compact set. From this and the assumption that the components of each matrix $A_i(w)$, $i = 1, \ldots, m$ are bounded for every $w \in W$ we conclude that there is a constant σ such that $\|\mathbf{c}\|_2 \leq \sigma$, for every $\mathbf{c} \in \mathscr{C}$. Thus, the assumption of the existence of σ is not restrictive.

In order to implement the proposed algorithm it is not necessary to obtain a bound of $\max\limits_{\mathbf{c} \in \mathscr{C}} \|\mathbf{c}\|_2$. However, as the worst-case number of iterations required by the algorithm depends on such a bound, it might be interesting in some applications to compute σ satisfying $\max\limits_{\mathbf{c} \in \mathscr{C}} \|\mathbf{c}\|_2 < \sigma$. In this way, the worst-case number of iterations required by the algorithm can be computed in advance.

Note that

$$\begin{aligned} |c_i| &\leq \max_{w \in W} \max_{v^\top v = 1} |v^\top A_i(w) v| \\ &= \max_{w \in W} \sqrt{\bar{\lambda}(A_i^2(w))} \\ &= \max_{w \in W} \bar{\sigma}(A_i(w)), \;\; i = 1 \ldots, m \;\; \forall \mathbf{c} \in \mathscr{C} \end{aligned}$$

where $\bar{\sigma}(A_i(w))$ denotes the largest singular value of the symmetric matrix $A_i(w)$.

If the dependence of matrices $A_i(w)$ with respect to w is affine (or linear-fractional) and W is an structured norm-bounded set ($W = \{ \Delta \in \blacksquare \; : \; \|\Delta\| \leq \rho \; \}$), then it is possible to use results from μ-theory (see (Zhou, Doyle and Glover, 1996)) or to introduce relaxations (Ben-Tal and Nemirovski, 2001) to obtain $\sigma_1, \sigma_2, \ldots, \sigma_m$ such that

$$\max_{w \in W} \bar{\sigma}(A_i(w)) < \sigma_i, \;\; i = 1, \ldots, m.$$

Thus, the following (conservative) bound is obtained

$$\max_{\mathbf{c} \in \mathscr{C}} \|\mathbf{c}\|_2 < \sqrt{\sum_{i=1}^{m} \sigma_i^2}.$$

4.7 Checking robust feasibility

It is clear that the results of μ-theory (see (Zhou et al., 1996)), and some techniques based on relaxations (Ben-Tal and Nemirovski, 2001) could be used (under some assumptions) to determine in a deterministic way if a given candidate solution is robustly feasible. However, these results and techniques can be very conservative. In some important cases (for example, affine dependence with respect to w), we can utilize vertex results to prove that a given candidate solution is a robust feasible solution (Barmish, 1994),(Boyd et al., 1994). In these cases, it suffices to check a finite number of elements of W, denoted W_F. If the number of elements of W_F is affordable, step 2 of the proposed algorithm can be implemented in a deterministic way. There exists different ways of visiting each one of the elements of W_F, see for example the "scheduling function" presented in (Liberzon and Tempo, 2004).

However, in most vertex results, the number of elements required to prove robust stability grows exponentially with the dimension of W. This is not a surprise since checking robust feasibility is in many cases an NP-hard problem (see (Nemirovskii, 1993),(Poljak and Rohn, 1993)). Moreover, there are many robust feasibility problems in which it is not possible to use any of the aforementioned vertex results. One of the ways of circumventing these difficulties consists in the use of a relaxed notion of feasibility (see, for example, (Barmish and Scherbakov, 2002)).

Definition 4.9 *Consider a probability measure Prob over the uncertain set W. Vector $\mathbf{z} \in \mathbb{R}^m$ is said to be a δ-level robust feasible solution if*

$$\text{Prob}\,\{\, w \in W \;:\; \bar{\lambda}(A(\mathbf{z}, w)) \geq 0 \,\} \leq \delta.$$

As it is stated in the following theorem, the algorithm proposed in Section 4.6 can be used to obtain a δ-level robust feasible solution with a pre-specified probability of failure β.

Theorem 4.10 *Suppose that given $0 < \beta < 1$ and $0 < \delta < 1$, steps 2 and 3 of the proposed algorithm are substituted by*

2. Obtain the smallest integer M_k that satisfies $(1-\delta)^{M_k} < \frac{6\beta}{\pi^2(1+k)^2}$. Pick, according to probability measure Prob, M_k elements of W: $\bar{w}_1,\ldots,\bar{w}_{M_k}$.

3. If $\bar{\lambda}(A(\mathbf{z}_k^*,\bar{w}_i)) < 0$, $i = 1\ldots,M_k$ then classify $\mathbf{z}_k^*$ as a (δ-level) robust feasible solution. Stop. Else, make w_{k+1} equal to one of the elements of $\{\ \bar{w}_i\ :\ \bar{\lambda}(A(\mathbf{z}_k^*,\bar{w}_i)) \geq 0, i = 1\ldots,M_k\ \}$.

With these modifications it results that if the algorithm stops classifying a given $\mathbf{z}_k^*$ as a (δ-level) robust feasible solution, then, $\mathbf{z}_k^*$ is a (δ-level) robust feasible solution with probability greater or equal to $1-\beta$.

Proof:

This proof follows arguments very similar to those presented in (Oishi, 2003). Suppose that at iteration k, $\mathbf{z}_k^*$ is not a (δ-level) robust feasible solution. That is, Prob $\{\ w \in W\ :\ \bar{\lambda}(A(\mathbf{z}_k^*,w)) \geq 0\ \} > \delta$. Then, the probability of classifying (erroneously) $\mathbf{z}_k^*$ as a (δ-level) robust feasible solution is smaller than

$$(1-\delta)^{M_k} \leq \frac{6\beta}{\pi^2(k+1)^2}.$$

Thus, the probability that the algorithm stops at an erroneously classified (δ-level) robust feasible solution is smaller or equal than

$$\sum_{k=0}^{k_{max}} \frac{6\beta}{\pi^2(k+1)^2} < \left(\frac{6\beta}{\pi^2}\right) \sum_{k=1}^{\infty} \frac{1}{k^2} = \left(\frac{6\beta}{\pi^2}\right)\left(\frac{\pi^2}{6}\right) = \beta.$$

■

4.8 Relationships with other randomized algorithms

4.8.1 Gradient algorithms

In what follows, the analogies and differences of the proposed approach with respect to the gradient algorithms (see (Calafiore and Polyak, 2001),(Liberzon and Tempo, 2004),(Tempo et al., 2005)) are now discussed. Given a candidate solution to a robust feasible problem, a (randomized) gradient algorithm tries to obtain (by means of a randomized method) an element belonging to the uncertainty set that contradicts the feasibility of the candidate solution. If such an element is found, the candidate solution is updated using a sub-gradient obtained from such an element of the uncertainty.

Consider now the proposed algorithm with $N_{max} = 1$. In this case, the candidate solution is denoted $\mathbf{z}_k^*$. If $\mathbf{z}_k^*$ is shown to be a non feasible solution then it is updated using $\mathbf{h}_k = \gamma_k^* \mathbf{z}_k^*$ and an element of W that contradicts the feasibility of $\mathbf{z}_k^*$. The analogies with (randomized) gradient algorithms are apparent. The main difference is that in the gradient algorithms the update is done by means of a simple explicit formula while in the proposed algorithm this is performed by means of an optimization problem. In order to stress those similarities, we will say that the proposed algorithm runs in gradient mode if N_{max} is small when compared to the number of decision variables of the problem.

To better illustrate the analogies with the gradient algorithms, the following property is presented.

Property 4.11 *Suppose that robust feasible problem (4.1) is ε-feasible. Suppose also that* $\max\limits_{\mathbf{c}\in\mathscr{C}} \|\mathbf{c}\|_2 < \sigma$. *Then, there is* $\bar{\mathbf{z}}$, $\bar{\mathbf{z}}^\top \bar{\mathbf{z}} \leq 1$ *such that*

$$A(\mathbf{z},w) \leq \frac{-\varepsilon}{2}\mathrm{I},\ \ \forall w \in W,\ \forall \mathbf{z} \in \{\ \mathbf{z}\ :\ \|\mathbf{z}-\bar{\mathbf{z}}\|_2 \leq \frac{\varepsilon}{2\sigma}\ \}.$$

Proof:

As the problem is assumed to be ε-feasible, there exists $\bar{\mathbf{z}}$ such that $\bar{\mathbf{z}}^\top \bar{\mathbf{z}} \leq 1$ and $\bar{\lambda}(A(\bar{\mathbf{z}},w)) \leq -\varepsilon$, $\forall w \in W$. Suppose now that $\mathbf{z} \in \{\ \mathbf{z}\ :\ \|\mathbf{z}-\bar{\mathbf{z}}\|_2 \leq \frac{\varepsilon}{2\sigma}\ \}$ and that $w \in W$. Property 4.3 guarantees that

$$\begin{aligned}
\bar{\lambda}(A(\mathbf{z},w)) &= \max_{\mathbf{c}\in C(w)} \mathbf{c}^\top \mathbf{z} \\
&= \max_{\mathbf{c}\in C(w)} \mathbf{c}^\top (\bar{\mathbf{z}} + (\mathbf{z}-\bar{\mathbf{z}})) \\
&\leq \max_{\mathbf{c}\in C(w)} \mathbf{c}^\top \bar{\mathbf{z}} + \max_{\mathbf{c}\in C(w)} \mathbf{c}^\top (\mathbf{z}-\bar{\mathbf{z}}) \\
&\leq \bar{\lambda}(A(\bar{\mathbf{z}},w)) + \max_{\mathbf{c}\in C(w)} \|\mathbf{c}\|_2 \|\mathbf{z}-\bar{\mathbf{z}}\|_2 \\
&\leq -\varepsilon + \max_{\mathbf{c}\in\mathscr{C}} \|\mathbf{c}\|_2 \|\mathbf{z}-\bar{\mathbf{z}}\|_2 \\
&\leq -\varepsilon + \sigma\frac{\varepsilon}{2\sigma} = -\frac{\varepsilon}{2}.
\end{aligned}$$

■

Gradient algorithms are normally based on a strong feasibility condition: it is assumed that the robust feasible set contains a ball or radius r. Under this assumption, the worst case number of iterations required by such gradient algorithms is bounded by a quantity proportional to $\left\lceil \frac{L}{r^2} \right\rceil$ (where L is proportional to the distance of the initial guess of the solution to the robust feasibility set (see (Calafiore and Polyak, 2001),(Liberzon and Tempo, 2004))).

Suppose that problem (4.1) is ε-feasible and consider the feasibility problem: Find $\mathbf{z}$ such that $\mathbf{z}^\top \mathbf{z} \leq 1$, $A(\mathbf{z}, w)) + \frac{\varepsilon}{2}\mathrm{I} \leq 0$. Property 4.11 guarantees that the feasibility radius of this problem is not smaller than $\frac{\varepsilon}{2\sigma}$. Hence, a gradient algorithm could be used to find a robust feasible solution to the aforementioned feasibility problem with a number of iterations bounded by $\left\lceil \frac{4\sigma^2 L}{\varepsilon^2} \right\rceil$. Recall now the worst case number of iterations of the proposed algorithm (see Theorem 4.7)

$$\left\lceil \frac{4\sigma^2}{\varepsilon^2} \right\rceil + \left\lceil \left(\frac{4\sigma^2}{\varepsilon^2}\right) \ln \ln \frac{\sigma^2}{\varepsilon^2} \right\rceil .$$

The analogies are clear: the number of iterations is basically determined by $\frac{\sigma^2}{\varepsilon^2}$. This means that the worst case number of iterations quickly degrades for both the gradient algorithms and the proposed algorithm, when ε is too small. This degradation of the performance with decreasing ε is observed in our numerical experience only for small values of N_{max} (gradient mode). The numerical results show that when N_{max} is chosen large enough (a sensible choice is $N_{max} = m$) the number of iterations grows in a benign way with decreasing ε (see Section 4.9).

4.8.2 Localization methods

The localization methods keep at each iteration an updated set that is guaranteed to contain the original feasible set of the problem. Cutting plane methods, as the ellipsoid algorithm, belong to this class of optimization algorithms. The ellipsoid algorithm has been used in a randomized scheme to address the robust feasibility problem in (Kanev et al., 2003) and (Oishi, 2003). A probabilistic analytic center cutting plane method is also presented in (Calafiore and Dabbene, 2006) to solve robust feasibility problems. Normally, the localization methods exhibit better theoretical worst-case performance than the gradient methods. This is mainly due to the fact that for some methods (as the ellipsoid method) it is possible to show that the volume of the region that bounds the feasibility region decreases in an exponential way with the number of iterations.

In what follows it is shown that the proposed algorithm is a localization method if N_{max} is greater than k_{max} (or equivalently, when no element is eliminated from set S_k). Note that k_{max} is the worst case number of iterations required by the proposed algorithm. With this choice of N_{max} it results that $S_0, S_1, \ldots, S_k$ satisfy the relations $S_0 \subset S_1 \subset \ldots \subset S_k$.

Denote now $D_k = \{ \mathbf{z} \in \mathbb{R}^m \, : \, A(\mathbf{z}, w) < 0, \forall w \in S_k \}$. It follows that each set D_k contains the feasibility region. From $S_k \subset S_{k+1}$, it is inferred that $D_{k+1} \subseteq D_k$. Note now that from the definition of z_k^* it is clear that z_k^* belongs to the interior of D_k. Moreover, by construction, $z_k^* \notin D_{k+1}$. This means that D_{k+1} is strictly included in D_k. The similarities with the localization methods are thus clear.

4.9 A numerical example

Consider the following system

$$\left\{ \begin{array}{rcl} \dot{x} & = & Ax + Bu + Ew \\ y & = & Cx + Du \end{array} \right.$$

where $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $E \in \mathbb{R}^{n_x \times n_w}$, $C \in \mathbb{R}^{n_y \times n_x}$ and $D \in \mathbb{R}^{n_y \times n_u}$ are affected by interval uncertainty. That is,

$$\begin{array}{rcl} A \in \mathscr{A}_I & = & \{ A \, : \, |A_{i,j} - \tilde{A}_{i,j}| \le 0.02, \forall i, \forall j \} \\ B \in \mathscr{B}_I & = & \{ B \, : \, |B_{i,j} - \tilde{B}_{i,j}| \le 0.02, \forall i, \forall j \} \\ E \in \mathscr{E}_I & = & \{ E \, : \, |E_{i,j} - \tilde{E}_{i,j}| \le 0.02, \forall i, \forall j \} \\ C \in \mathscr{C}_I & = & \{ C \, : \, |C_{i,j} - \tilde{C}_{i,j}| \le 0.02, \forall i, \forall j \} \\ D \in \mathscr{D}_I & = & \{ D \, : \, |D_{i,j} - \tilde{D}_{i,j}| \le 0.02, \forall i, \forall j \} \end{array}$$

where

$$\tilde{A} = \begin{bmatrix} 0.083 & 0.650 & -0.627 & -0.624 & 0.091 \\ -0.832 & -0.261 & -0.023 & 0.275 & 0.658 \\ -0.985 & -0.447 & -0.007 & 0.731 & -0.942 \\ 0.179 & -0.957 & 0.649 & -0.873 & -0.350 \\ -0.305 & 0.239 & 0.610 & -0.752 & -0.624 \end{bmatrix},$$

$$\tilde{B} = \begin{bmatrix} 0.316 & 0.545 \\ 0.819 & -0.355 \\ 0.303 & 0.120 \\ -0.518 & -0.490 \\ -0.889 & -0.955 \end{bmatrix}, \tilde{E} = \begin{bmatrix} -0.935 & 0.282 \\ 0.023 & -0.753 \\ -0.597 & 0.329 \\ 0.152 & -0.010 \\ -0.680 & 0.858 \end{bmatrix},$$

$$\tilde{C} = \begin{bmatrix} -0.433 & 0.131 & 0.877 & 0.986 & 0.179 \\ 0.823 & 0.551 & -0.664 & -0.930 & -0.747 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\tilde{D} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0.876 & -0.368 \\ 0.317 & 0.773 \end{bmatrix}.$$

The total number of uncertain parameters is $n_x(n_x + n_u + n_w) + n_y(n_x + n_u) = 73$. This means that the uncertainty can be parameterized by means of a vector of 73 components: $W = \{ w \in \mathbb{R}^{73} \ : \ \|w\|_\infty \leq 0.02 \}$.

The objective of the synthesis problem is to find a state feedback gain K such that the L_2 gain of the system is minimized. It is well known (see, for example (Boyd et al., 1994)) that the L_2 gain of the closed loop uncertain system is bounded by φ if there is $P > 0$ such that

$$\frac{d}{dt}(x^\top P x) + y^\top y - \varphi^2 w^\top w \leq 0.$$

Denoting $Q = P^{-1}$ and $Y = KQ$, the synthesis problem can be rewritten as

$$\begin{gathered} \min_{Q,Y,\varphi} \ \varphi^2 \\ s.t. \\ \begin{bmatrix} AQ + QA^\top + BY + Y^\top B^\top & * & * & * \\ E^\top & -\varphi^2 \mathrm{I} & * & * \\ CQ + DY & 0 & -\mathrm{I} & * \\ 0 & 0 & 0 & -Q \end{bmatrix} < 0 \\ \forall A \in \mathscr{A}_I, \forall B \in \mathscr{B}_I, \forall C \in \mathscr{C}_I, \forall D \in \mathscr{D}_I. \end{gathered} \tag{4.9}$$

It is well known that due to the interval nature of the considered uncertainty and the affine parametric dependence, it suffices to check all the extreme realizations of the uncertainty. That is, the $2^{73} > 10^{21}$ vertices of the hypercube W (see (Horisberger and Belanger, 1976)).

In order to reduce the number of vertices, a recent vertex result (Álamo, Tempo, Ramírez and Camacho, 2006*a*) can be used. This result reformulates the full interval matrix uncertainty into an equivalent diagonal uncertainty and, when applied to this example, reduces the number of required vertices to $2^{18} = 262,144$.

Using a bisection approach for the decision variable φ, this optimization problem can be solved by means of a sequence of robust feasibility problems of the form: given $\varphi > 0$, obtain (if possible) Q and Y such that the robust constraint (4.9) is satisfied for every possible realization of the interval uncertainty (that is, for every $w \in W$). Taking into account the symmetry of Q, it is possible to parameterize matrices $Q \in \mathbb{R}^{n_x \times n_x}$ and $Y \in \mathbb{R}^{n_u \times n_x}$ by means of a vector of 25 components. Thus, the previous feasibility problem can be rewritten as a feasibility problem of the following form: find $\mathbf{x} \in \mathbb{R}^{25}$ such that

$$F_0(w) + \sum_{i=1}^{25} x_i F_i(w) < 0, \ \ \forall w \in W.$$

Using the strategy presented in Section 4.3, this optimization problem can be recast as a feasibility problem of the form: find $\mathbf{z} \in \mathbb{R}^{26}$ such that

$$A(\mathbf{z}, w) < 0, \ \ \forall w \in W$$

where $A(\mathbf{z}, w) = \sum_{i=1}^{26} z_i A_i(w)$.

As discussed before, in this example we can check in an affordable time if a given pair (Q, Y) is a robust feasible solution (for a given φ). This means that given $\mathbf{z}$, we can check in a deterministic way if $A(\mathbf{z}, w) < 0$, $\forall w \in W$. Therefore, each of the feasibility problems required to obtain an optimal value of φ can be solved using the proposed algorithm without resorting to the notion of δ-level feasible solution (that is, step 2 of the algorithm has been implemented in a deterministic way).

Note that when the bisection algorithm is close to the optimal value of φ, the associated feasibility problem (if feasible) will be ε-feasible only for small values of ε. This has two consequences

1. The obtained optimal value of φ will depend on the choice of ε. The smaller ε is, the closer the final obtained value of φ will be to the actual solution of the robust L_2 optimization problem.

2. In the bisection process aimed to the computation of the smallest value of φ for which the associated robust feasible problem is ε-feasible, the proposed algorithm will encounter the following two situations when it approaches the final value of φ:

- The problem is ε-feasible, but as φ is close to the final solution, the feasibility radius of the problem can be very small. In this case, the reduced feasibility radius increases the number of iterations required to obtain a feasible solution.
- The problem is not ε-feasible, but there is $\tilde{\varepsilon}$, only slightly smaller than ε, for which the problem is $\tilde{\varepsilon}$-feasible. In this case, the algorithm has to determine the non ε-feasibility of a problem that is "almost" ε-feasible. This is also a worst-case situation that increases the number of iterations required to determine the non ε-feasibility.

Thus, an optimization problem in which the final solution is obtained using a sequence of feasibility problems is, in our opinion, the best scenario to test the properties of an algorithm designed for the feasibility problem.

In order to illustrate the effect of both ε and N_{max} in the performance of the proposed algorithm, the L_2 robust optimization problem has been solved for different values of the pair (N_{max}, ε). The numerical results presented in this section have been obtained with an Intel Pentium 4 at 1.8 GHz using the LMI toolbox of Matlab. Table 4.1 shows the numerical results for $\varepsilon = 0.01$ and different values of N_{max}. As discussed before, in order to solve the optimization problem, different feasibility problems are solved. In Table 4.1, *max iterations* has the following meaning: all the feasibility problems (corresponding to a given pair (N_{max}, ε)) have been solved with a number of iterations smaller or equal to max iterations. The entry *total CPU time* is the total time in seconds required to solve the optimization problem (the computation of φ). The entry *obtained* φ denotes the minimal value of φ for which a feasible solution has been found.

Table 4.2 shows the numerical results for $\varepsilon = 0.001$ and different values of N_{max}. It results apparent from the comparison of this table with Table 4.1 that for small values of N_{max}, the entry *max iterations* and *total CPU time* clearly degrades with decreasing ε. The best computational times are obtained for $N_{max} = 10$ (several order of magnitudes smaller than the one corresponding to $N_{max} = 1$). It is also important to note that the obtained value of φ corresponding to $\varepsilon = 0.001$ is significantly smaller than the one corresponding to $\varepsilon = 0.01$.

In Table 4.3, the numerical results for $N_{max} = 10$ and different values of ε are shown. In contrast to the case in which N_{max} is very small, the behavior of the algorithm does not degrade with decreasing ε. It can be observed from the table that the obtained value of φ does not improve significantly beyond $\varepsilon = 10^{-5}$. This means that the optimal robust solution to the robust L_2 gain problem is obtained with $\varepsilon = 10^{-5}$ and $N_{max} = 10$ in less than two minutes.

N_{max}	1	2	5	10
max iterations	255	16	8	9
total CPU time	533.4	67.9	21.7	36.2
obtained φ	2.04	2.09	2.03	2.01

Table 4.1: Numerical results for different values of N_{max} with ε equal to 0.01.

N_{max}	1	2	5	10
max iterations	6021	2726	405	9
total CPU time	$2.4 \cdot 10^4$	$1.2 \cdot 10^4$	$2.2 \cdot 10^3$	63.5
obtained φ	1.64	1.60	1.59	1.59

Table 4.2: Numerical results for different values of N_{max} with ε equal to 0.001.

ε	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
max iterations	9	9	12	9	16
total CPU time	36.2	63.5	103.0	102.3	128.1
obtained φ	2.01	1.59	1.56	1.55	1.55

Table 4.3: Numerical results for different values of ε with N_{max} equal to 10.

4.10 Conclusions

In this chapter a new randomized algorithm that addresses the robust feasibility problem under uncertain LMIs is presented. The proposed algorithm has clear differences with randomized gradient and localization methods. The algorithm is guaranteed to obtain a δ-level feasible solution if the problem is ε-feasible. Additionally, if the problem is not ε-feasible the algorithm detects this non feasibility in a finite number of iterations. A bound on the maximal number of iterations required has been obtained. Moreover, the analogies and differences with other existing randomized methods have been discussed. A numerical example that illustrates the merits of the algorithm has been provided.

Chapter 5

Fault detection with probabilistic validation

5.1 Introduction

We present here a general strategy for the design of a fault-detection block with probabilistic validation (PCV- Processing, Classification, Validation) (Blesa et al., 2013). A general scheme of PCV is proposed, that allows to design a fault detection block with probabilistic validation in the maximum percentage of non detected faults (set as design condition) and in the percentage of false alarms (obtained a posteriori). In each iteration of the sequential algorithm, a candidate solution is probabilistically validated by a set of samples randomly generated. A general framework is presented in which the candidate solution can violate the constraints for a limited number of elements of the validation set. This generalized scheme shows significant advantages, in particular in terms of the obtention of the probabilistic solution.

A fault is defined as any change in the behavior of some of the system components (not allowed deviation of the one of characteristic parameters or properties) so that it and can not fulfill the function for which it was designed (Blanke, 1999). Besides faults, there exist other factors that alter the normal behavior of the system, as disturbances and noise. Disturbances are non known entries that can occur in the system at any time but they have been taken into account when designing the conventional control loop. Any disturbance which had not been into account in this design will be considered a fault. Noise is also a non known input manifested in the system but, unlike disturbance, has zero mean, and it is possible to have a priori knowledge of which is its amplitude. A fault detection system must react to faults and be immune (robust), so far as possible, to the other factors in the system that create uncertainty. Furthermore, many of the fault detection methods are based on a model (mathematical or quantitative) of the monitored system which can never accurately describe the real behavior of the system and therefore it will present a modeling error that must be considered.

The goal of a fault detection block is, once a fault has occurred in an instant T_F, to detect it in a time range less or equal than $T_{D_{max}}$, set in advance. Depending on the magnitude and incidence of the faults desired to be detected and the possible presence of other uncertainty factors in the system, it may not be possible to design a detection block detecting all faults without false alarms in situations in which there are no faults. So there is always a compromise between the proportion of undetected faults (*MF* ”Missed Faults”) and the proportion of times that the detection block is activated without the presence of faults due to the uncertainties present in the system (*FA* ”False alarms”). This compromise, that should be taken into account in the design process of the detector block, it is logical to prioritize minimizing not detected faults respect to the minimization of false alarms.

The random nature of faults and uncertainties inherent in the system makes the design problem of the detection block a robustness problem.

Typically, for a robustness problem, the design parameters, and different auxiliary variables, are described in terms of a vector of decision variables θ, denoted as ”design parameter” and is restricted to the set Θ. Moreover, the uncertainty w is bounded on the set $\mathscr{W}$. That is, each element $w \in \mathscr{W}$ represents one of the admissible realizations of uncertainty, with probability $\mathrm{Pr}_{\mathscr{W}}$. In our context of fault detection, θ corresponds to the decision variables that determine the fault detection block. This block allows us to determine if there is a fault or not in a given scenario, so there will be two uncertainty sets $\mathscr{W}_F$ and $\mathscr{W}_N$ consisting in all possible scenarios of the system to be monitored operation, with faults and without faults respectively. Furthermore, w_F and w_N represent a realization of a scenario with and without fault. $\mathscr{W}_F$ and $\mathscr{W}_N$ have spaces probability Pr_F and Pr_N respectively.

We also consider two measurable binary functions:

$$g(\theta, w) := \begin{cases} 0 & \text{if } \theta \text{ detects a fault} \\ 1 & \text{in other case.} \end{cases}$$

$$h(\theta, w) := \begin{cases} 0 & \text{if } \theta \text{ doesn't detect a fault} \\ 1 & \text{in other case.} \end{cases}$$

Applying these two functions on the spaces $\mathscr{W}_F$ and $\mathscr{W}_N$ we obtain the following expected values:

$$
\begin{aligned}
E_g(\theta) &:= \mathrm{Pr}_F\{w_F \in \mathscr{W}_F : g(\theta, w_F) = 1\} \\
E_h(\theta) &:= \mathrm{Pr}_N\{w_N \in \mathscr{W}_N : h(\theta, w_N) = 1\}.
\end{aligned}
$$

Where $E_g(\theta)$ and $E_h(\theta)$ are the proportion of undetected faults (MF) and false alarms (FA) respectively.

The utility of randomized algorithms arises when being able to treat the following design problem

$$
\min_{\theta\in\Theta} E_h(\theta) \ \text{ subject to } E_g(\theta) \leq \eta_F \tag{5.1}
$$

where η_F is the maximum proportion of undetected faults imposed as a constraint of the detector block.

In this context, we can extract N_N and N_F i.i.d. samples (independent and identically distributed) $\{w_N^{(1)}, \dots, w_N^{(N_N)}\}$ of $\mathscr{W}_N$ and $\{w_F^{(1)}, \dots, w_F^{(N_F)}\}$ of $\mathscr{W}_F$ according to the probability Pr_N and Pr_F respectively, and with a ratio between scenarios with fault and without fault $F_N = \frac{N_F}{N_N}$ determined by the probability of failure of the system to be monitored. This way can solve the following sampled optimization problem

$$
\begin{gathered}
\min_{\theta\in\Theta} \sum_{\ell_N=1}^{N_N} h(\theta, w_N^{(\ell_N)}) \\
\text{subject to} \sum_{\ell_F=1}^{N_F} g(\theta, w_F^{(\ell_F)}) \leq \eta_F N_F
\end{gathered}
\tag{5.2}
$$

In this chapter we propose a design method of a fault detector block based on the use of historical or real simulations with and without faults, avoiding the difficulty of analysis, that is not always possible, due to the complexity of problem.

The result thus obtained, through a probabilistic validation test, guarantees that the proposed solution behaves the desired way with a certain probability, fixed a priori. It also guarantees the satisfaction of probabilistic constraints. This technique is well suited for addressing complex problems.

Finally, this chapter illustrates the proposed methodology with the presentation of an application, in this case, the design of a fault detector with probabilistic guarantee in a virtual deposit that models the behavior of a collector of a network of sewer as detailed in (Puig and Blesa, 2013).

5.2 Proposed design scheme

As previously discussed, the objective of this work is to propose a scheme that allows a fault detection block design with probabilistic validation in the maximum percentage of undetected faults (imposed as a design condition) and the false alarm rate (obtained posteriori). The operation of this failure detection block is described in the diagram of Figure 5.1. From input / output data of the system to monitor, each time instant j some attributes or indicators $(r_j^{(1)}, r_j^{(2)}, ..., r_j^{(n_r)})$ are extracted from it by a processing block which acts on sliding time horizon. These indicators, that can be of different types, are sensitive to the presence of failures but also to noise, disturbances and other factors such as modeling errors. For this reason they are introduced in a classifier that determines whether or not a failure in the system has occurred.

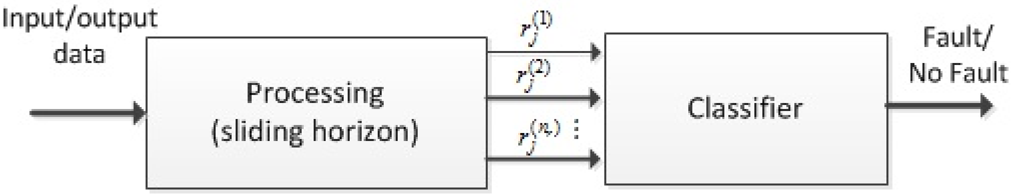


Figure 5.1: Fault detector scheme on line

Furthermore, the off-line algorithm of the process to the design detector block is described in the diagram of Figure 5.2.

At the beginning the processing block is designed or, what is the same, it is determined which indicators will be useful in order to detect faults present in the system. For this step the knowledge of the plant and basic techniques for fault detection (Isermann, 2006), (Isermann, 2011) must be used. The nature of the used indicators can be very diverse: residuals between measurements and estimates, certain signals to detect faults such as vibration, sound signals, and so on; magnitudes to determine a change of operation of the plant such as temperature, humidity, etc.

After determining the indicators to be extracted from the information available of the plant, the probabilistic requirements bound in percentage of undetected faults (MF) and guaranteed minimum probabilistic compliance must be specified. Then we proceed to design a classifier to which plant data are provided in situations of operation with and without

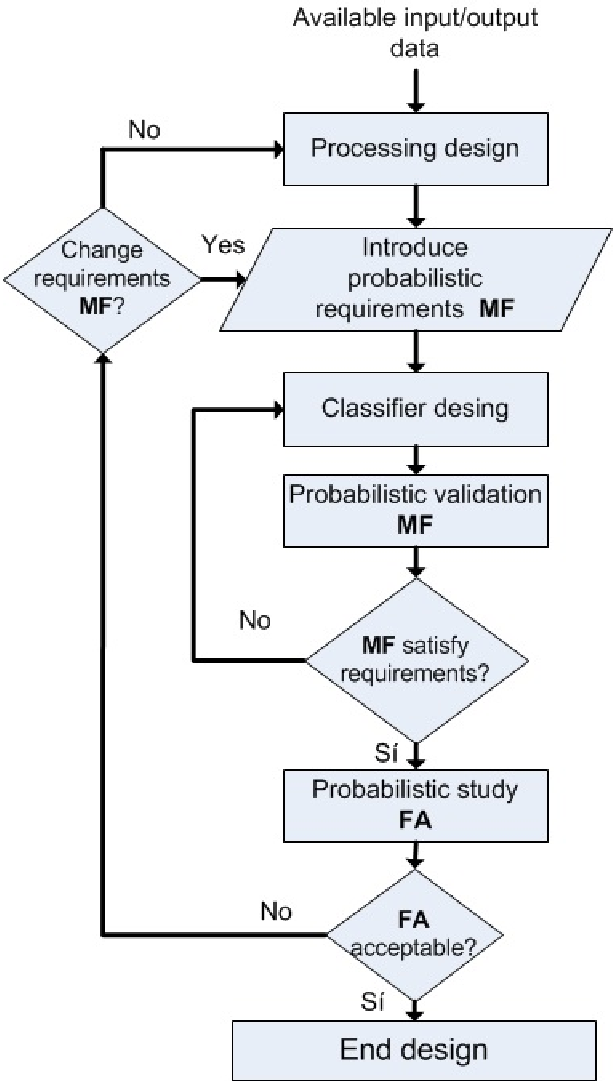


Figure 5.2: Scheme of design of the fault detection block offline

faults (real or simulated). By and optimization process the number of false alarms (*FA*) is minimized and we impose (as a constraint) a maximum number of non detected faults (*MF*) and the classifier is calibrated properly. The classifier is then evaluated using a probabilistic validation test using different data than those used in the design of the classifier. If the probabilistic validation test is not passed, come back to the design stage of the classifier. If the validation test is passed, the classifier meets the specifications for *MF* with the imposed probabilistic constraints and we proceed to a probabilistic study of the number of false alarms. If the result is not satisfactory (high number of false alarms) there are two options: change (relax) the probabilistic requirements *MF* or improve the processing design (better indicators).

5.3 Processing design

As discussed above, the indicators extracted from the monitored system may be of different nature. In this section two examples of possible indicators used in fault detection methods based on mathematical models are shown. In this type of methods consistency or inconsistency of the model of the system to monitor is checked with the obtained measurements of this system.

For example, if we consider that the monitored system can be described by the regression model in discrete time

$$y(j) = \varphi^T(j)\beta_0 + e(j),\ j = 1,\ldots,M \tag{5.3}$$

where

- $y(j)$ is the output measurement
- $\varphi(j)$ is the regression vector of dimension n_β function of inputs $u(j)$ and outputs $y(j)$
- β_0 is the nominal parameters vector of dimension n_β
- $e(j)$ is the additive error that contains measurements noise and modeling error.

Then this consistence can be evaluated computing in each instant j the residual $\Delta(j)$ between the measured output and the estimation given by the model

$$\Delta(j) = y(j) - \varphi^T(j)\beta_0. \tag{5.4}$$

In an ideal case, the residual should be different from zero just in the case that a there was a fault in the system. However, due to the presence of noise and modeling error in the system, the residual can be different from zero when there is no fault and the detection method should be robust. One way to approach this problem is applying the Set-Membership techniques that consider the error *e(j)* unknown but bounded (Milanese, Norton, Lahanier and Walter, 1996), that is

$|e(j)| \leq \sigma\ j = 1,\ldots,M.$

This way the following detection fault test can be applied

$$\begin{cases} \text{If } |\Delta(j)| \leq \sigma \Rightarrow \text{NoFault} \\ \text{If } |\Delta(j)| > \sigma \Rightarrow \text{Fault.} \end{cases}$$

And therefore the design of the diagnostician consists on choosing a proper bound σ. This test is called direct test.

Another way to look at the consistency of the model with the measurements in the parameter space, using a time window of N samples, can be made by a parameter estimation

$$\hat{\beta}(j) = \left(\Phi^T(j)\Phi(j)\right)^{-1}\Phi^T(j)Y(j) \tag{5.5}$$

where $\Phi(j) = \begin{pmatrix} \varphi^T(j-N) \\ \vdots \\ \varphi^T(j) \end{pmatrix}$ y

$$Y(j) = \begin{pmatrix} y(j-N) \\ \vdots \\ y(j) \end{pmatrix}$$

and in the same way than with the temporal residual, a residual of parameters can be calculated respect to a nominal model

$$\Delta\beta(j) = \hat{\beta}(j) - \beta_0$$

and like this define the following detection test

$$\begin{cases} \text{If}\,\Delta\beta(j) \in \text{B} \Rightarrow \text{NoFault} \\ \text{If}\,\Delta\beta(j) \notin \text{B} \Rightarrow \text{Fault,} \end{cases}$$

where B is the paramenters uncertainty set due to the additive error $e(j)$ and the small richness of the data used in the identification.

Both direct and inverse tests can be included in the general scheme 5.1 choosing $r_j^{(1)} = \Delta(j)$ for the direct test and $r_j^{(2)} = \Delta\beta(j)$ for the inverse test. In addition to this, basic tests such as maximum values, fixed or maximum variations in the obtained data by the sensor can be implemented choosing the correct indicators.

5.4 Classification design

With the purpose of achieving a good discrimination between scenarios with and without faults, a two stages classifiers is proposed: a static and a dinamic stage, as shown in 5.3.

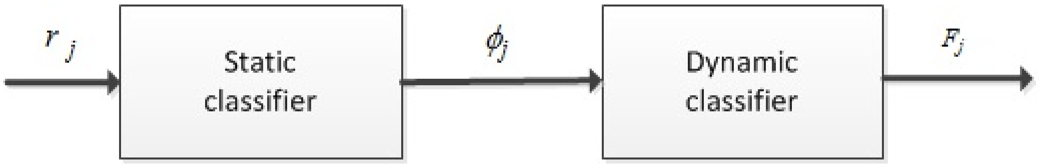


Figure 5.3: Scheme of the proposed classifier

The objective of the static classifier is to determine in a time instant j, with the indicators vector in this same time instant r_j, if the situation is symptom of fault or not by a binary signal ϕ_j. If the situation suggests a fault $\phi_j = 1$ and if not $\phi_j = 0$. This operation could be done by a vectorial analogical input function and binary output h_{est}

$$\phi_j = h_{est}\left(r_j\right)$$

where $r_j^T = (r_j^{(1)}, r_j^{(2)}, ..., r_j^{(n_r)})^T$.

The objective of the dynamic classifier of order T_p is to determine in the time instant j if a fault has occurred or not with the last T_p symptomatic signals. That is:

$$F_j = h_{din}\left(\phi_j, \phi_{j-1}, \cdots, \phi_{j-T_P+1}\right).$$

Where h_{din} is a function of binary inputs and outputs and, for the definition of detectability, it holds that $T_P \leq T_{D_{\max}}$. The reason for dividing in two parts the classifier is to make it the most robust as possible respect to false alarms without worsen the detectability performances (non detected faults).

To design these two classifiers there will be scenarios with and without faults $\{w_N^{(1)}, \ldots, w_N^{(N_N)}\}$ from $\mathscr{W}_N$ and $\{w_F^{(1)}, \ldots, w_F^{(N_F)}\}$ from $\mathscr{W}_F$ respectively. Each scenario consists on a sequence of indicators $r_1{}^{(i)}, r_2{}^{(i)}, \cdots, r_{T_{Si}}{}^{(i)}$ where T_{Si} is the number of time instants that lasts scenario i. The set of all the indicators of all the scenarios can be divided in two subsets: the set of the indicators that have a fault and the set of indicators free of faults $\mathfrak{R}_F$ and $\mathfrak{R}_N$ respectively. It holds that all the indicators that belong to an scenario free of faults belong to the set $\mathfrak{R}_N$, while some of the indicators belonging to a faulty scenario are affected for the effect of the fault (and therefore belong to $\mathfrak{R}_F$) and some of them are free of the effects of the fault (and therefore belong to $\mathfrak{R}_N$).

The purpose of the static classifier is to distinguish as much as possible between the indicators belonging to the two sets ($\Re_F$ and $\Re_N$). This can be achieved by the correct election of a vector λ satisfying the best as possible the following constraints

$$\begin{array}{l} f^T(r)\lambda > 0 \Rightarrow r \in \Re_F \\ f^T(r)\lambda \leq 0 \Rightarrow r \in \Re_N \end{array},$$

where $f(r)$ is a vectorial expression of r. The value of λ can be found by the following convex optimization problem

$$\min_{\lambda} \left(\sum_{r\in\Re_F} e^{-\tau f^T(r)\lambda} + \sum_{r\in\Re_N} e^{f^T(r)\lambda} \right) \tag{5.6}$$

where $\tau \in (0,\infty)$ is a constant determined before solving the optimization problem and allows penalizing to a greater or lesser extent the classification errors of indicators $r \in \Re_F$ respect to the classification errors of indicators $r \in \Re_N$ and therefore prioritize more or less the behavior of the system respect the non detected faults or respect the false alarms. Specifically, the greater τ the more penalization will be applied to the indicators classified as normal situation, that is to say, non detected faults, and therefore the false alarms will have less weight.

As mentioned before, in the design process of the fault detector we will impose a maximum value of non detected faults (MF defined by η_F) and this will be achieved choosing a proper τ. The problem of the static classifier is that taking the decision depending on the indicators in a certain time instant, imposing a low number of faults may imply a high sensitivity to the system uncertainties and this could be translated in a high number of false alarms. The second classifier's objective is to filter the effect of these uncertainties and allow a diagnosis as robust as possible using the last T_P outputs of the static diagnostician. As the diagnostician function h_{din} is a function of binary input and binary output with 2^{T_P} combinations, these can be proved exhaustively and choose which one of them presents the best performance.

Therefore the design of the classifier is reduced to finding the minimum value of τ satisfying the maximum number of allowed false alarms. The obtained block of fault detection will be called θ.

5.5 Validation

5.5.1 Sequential algorithms with probabilistic validation in MF

In this section a general family of randomized algortihms is presented, called SPV algorithms ("**S**equential **P**robabilistic **V**alidation algorithms"), see (Álamo et al., 2012).

The main characteristic of this type of algorithms is that they are based on a probabilistic validation step.

Each iteration of an SPV algorithm is composed of the computation of a candidate solution for the problem and a validation step. The results are basically independent of the particular strategy chosen to obtain candidate solutions.

The objective of using these algorithms in this context is to guarantee a maximum level of undetected faults (MF) defined by η_F with a minimum probabilistic guarantee of achievement of $1-\delta_F$.

The accuracy $\eta_F \in (0,1)$ and confidence $\delta_F \in (0,1)$ required for the probabilistic solution play a relevant role when determining the sample size of each validation step. The main purpose of this chapter is to provide a validation scheme such that it guarantees that for given accuracy η_F and confidence δ_F, all the probabilistic solutions obtained running the SPV algorithm have a probability of violation no larger than η_F with probability no smaller than $1-\delta_F$.

We enumerate each iteration of the algorithm by means of integer k. We denote m_k the number of violations that are allowed at the validation step of iteration k. We assume that m_k is given by a function of k, that is, $m_k = m(k)$ where the function $m : \mathbb{N} \rightarrow \mathbb{N}$ is given. We also denote M_k the sample size of the validation step of iteration k. We assume that M_k is given by a function of k, ε and δ. That is, $M_k = M(k,\varepsilon,\delta)$ where $M : \mathbb{N}\times\mathbb{R}\times\mathbb{R} \rightarrow \mathbb{N}$ has to be appropriately designed in order to guarantee the probabilistic properties of the algorithm (the main contribution of (Oishi, 2007) was to provide this function for the particular case $m_k = 0$ for every $k \geq 1$). For future references we denote the functions $m(\cdot)$ and $M(\cdot,\cdot,\cdot)$ as *level function* and *cardinality function* respectively.

Structure of an SPV algorithm

(i) Set $\eta_F \in (0,1)$ and $\delta_F \in (0,1)$ equal to the desired levels. Set k equal to 1.

(ii) Obtain a candidate solution $\hat{\theta}_{F_k}$ to the robust optimization problem (A.4.1).

(iii) Set $m_k = m(k)$ and $M_k = M(k, \eta_F, \delta_F)$.

(iv) Obtain validation set $\mathcal{V}_k = \{v^{(1)}, \ldots, v^{(M_k)}\}$ drawing M_k i.i.d validation samples from $\mathcal{W}$ according to probability $\Pr_{\mathcal{W}}$.

(v) If $\sum_{\ell=1}^{M_k} g(\hat{\theta}_{F_k}, v^{(\ell)}) \leq m_k$, then $\hat{\theta}_{F_k}$ is a probabilistic solution.

(vi) Exit if the exit condition is satisfied.

(vii) k=k+1. Goto (ii).

Figure 5.4 shows the part of the proposed general designed structure that corresponds with the SPV algorithm, where it is indicated which step of the algorithm corresponds with each block.

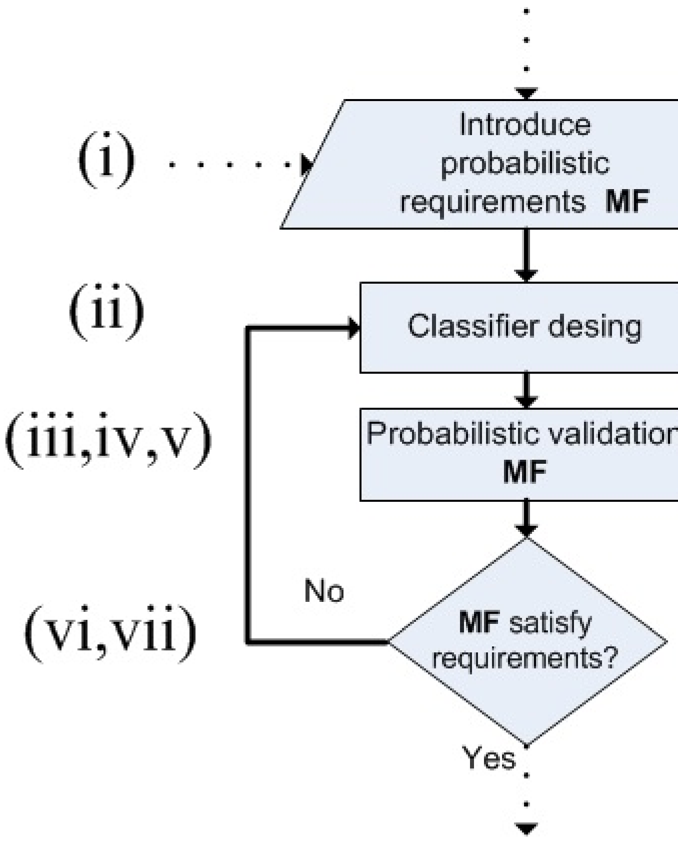


Figure 5.4: Scheme of probabilistic validation of non detected faults

Although the algorithm exit condition can be quite general, a reasonable strategy is to exit after a given number of candidate solutions have been classified as probabilistic solutions or if it exceeds a certain computational time since the beginning of the algorithm. After exiting you can choose from probabilistic solutions the one that optimizes a given performance index. In the next subsection we propose a strategy for choosing the cardinality of the validation set in iteration k so that, with probability not less than $1-\delta_F$ all solutions classified as probabilistic solutions for the algorithm satisfy the accuracy η_F.

5.5.2 Sample size

Let's consider an SPV algorithm with given accuracy $\eta_F \in (0,1)$, confidence $\delta_F \in (0,1)$ and level function $m(\cdot)$. Then, the cardinality function

$$M(k,\eta_F,\delta_F) =$$

$$\left\lceil \frac{1}{\eta_F}\left(m(k) + \ln \frac{1}{\delta_F \mu(k)} + \sqrt{2m(k) \ln \frac{1}{\delta_F \mu(k)}} \right) \right\rceil,$$

where

$$\mu(k) = \frac{1}{\xi(\alpha)k^{\alpha}},$$

where $\xi(\cdot)$ is the zeta Riemann function, and $\alpha > 1$,

guarantees that, with probability greater than $1-\delta_F$ all the probabilistic solutions obtained by the SPV algorithm have a probability of violation (non detected faults) no greater than η_F (Álamo et al., 2012). Function $\mu(k)$ can adopt other expressions, see (Álamo et al., 2012).

5.5.3 False alarms

Once a classifier satisfying the probabilistic constraints respecting non detected faults has been designed, the result obtained in (Álamo et al., 2010*a*) can be used, where the sample complexity given as an empirical mean converge in probability to the real probability of violation (in this case false alarms), to determine the accuracy $\eta_N \in (0,1)$ and confidence $\delta_N \in (0,1)$ in terms of false alarms (*FA*).

Given $\hat{\theta} \in \Theta$ obtained in the design process of the detection block and a set of N_N scenarios free of fault, it holds that

$$N_N \geq \frac{\ln \frac{1}{\delta_N}}{(\sqrt{\eta_N} - \sqrt{\rho_N})^2}$$

with $0 \leq \rho_N < \eta_N < 1$ where ρ_N is the proportion of false alarms obtained applying the designed detection block to the N_N available scenarios free of faults.

$$\rho_N = \frac{\text{Number of false alarms}}{N_N}.$$

Then

$$\mathrm{Pr}_{\mathscr{W}^N}\{\mathrm{w} \in \mathscr{W}^N \,:\, \hat{E}(\hat{\theta}, \mathrm{w}) \leq \rho_N \text{ and } E(\hat{\theta}) > \eta_N\} \leq \delta_N.$$

Therefore, if our estimation of false alarms $\hat{E}(\hat{\theta}, \mathrm{w})$ is given by the value ρ_N, it is guaranteed that the discrepancy between this value and the probability that the real number of false alarms $E(\hat{\theta})$ is greater than other value $\eta_N > \rho_N$ fixed a priori, is bounded by δ_N.

If a desired δ_N is imposed, the statistical guarantee η_N is determined by the previous inequations. Specifically

$$\eta_N = \left(\left(\frac{\ln(\frac{1}{\delta_N})}{N_N} \right)^{\frac{1}{2}} + \sqrt{\rho_N} \right)^2 \tag{5.7}$$

5.6 Results

To prove the effectiveness of the presented methodology it has been applied to the design of a fault detection block with probabilistic guarantee in the virtual deposit shown in Figure 5.5 modeling the behaviour of a collector of a network of sewers as detailed in (Puig and Blesa, 2013).

We have taken into account faults in the input and output sensors ($f_u(t)$, $f_y(t)$) and parametric faults ($f_a(t)$, $f_b(t)$) as indicated in Figure 5.6. The behavior of the real system considering these faults can be described by the following discrete time model

$$\tilde{y}(j) = (\tilde{a} + f_a(j-1))\,\tilde{y}(j-1) + \left(\tilde{b} + f_b(j-1)\right)\tilde{u}(j-1) + e_d(j)$$

where

- $\tilde{a}$, $\tilde{b}$ are the real parameters of the system

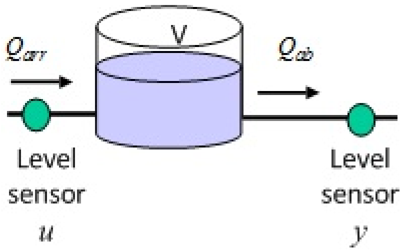


Figure 5.5: Virtual Deposit

- $\tilde{u}(j)$, $\tilde{y}(j)$ are the real input and output of the system
- $e_d(j)$ is the discretization error that will depend on the sampling time
- $e_u(j)$, $e_y(j)$ son los errores aditivos introducidos por los sensores

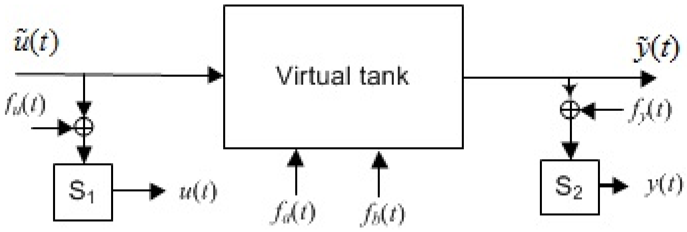


Figure 5.6: Posible faults in the virtual deposit system

Thus, with the available measurements

$y(j) = \tilde{y}(j) + f_y(j) + e_y(j)$

$u(j) = \tilde{u}(j) + f_u(j) + e_u(j)$

the following model can be used to describe the behavior of the real system

$$y(j) = a_0 y(j-1) + b_0 u(j-1) + e(j) \tag{5.8}$$

where

- a_0, b_0 are the parameters of the model obtained with input and output data of the system without fault
- $e(j)$ is the modeling error taking into account the error $e_d(j)$, $e_u(j)$, $e_y(j)$ and the possible discrepance between the real parameters ($\tilde{a}$,$\tilde{b}$) and those of the model (a_0,b_0)

With the purpose of obtaining the data of the normal behavior and the behavior with faults a simulator in Matlab-Simulink has been implemented, where scenarios with and without faults, based in real situations, have been introduced, having possible errors between the model and the real system. The objective is to design a fault detector satisfying $MF < 1\%$ with probabilitty no smaller of $1 - 10^{-6}$ ($\eta_F = 0.01$ $\delta_F = 10^{-6}$) with a maximum delay in fault detection of $T_{D_{max}} = 5$ samples.

First, using the obtained input/output data of the procces we have studied which of the indicators that will allow to distinguish between the situations with fault and those of normal performance. Figures 5.7 and 5.8 show the error indicators in the parameters identification $r_j^{(1)} = (\Delta a(j), \Delta b(j))^T$ ($\Delta a(j) = \hat{a}(j) - a_0$ y $\Delta b(j) = \hat{b}(j) - b_0$) taking a time window of 50 samples, for scenarios with and without faults respectively. Moreover, in this figures it is also shown the area $f^T(r)\lambda \leq 0$ obtained designing an optimal classifier. Only with this indicator the 65% of the faults can be detected with a very low level of false alarms(0.001%).

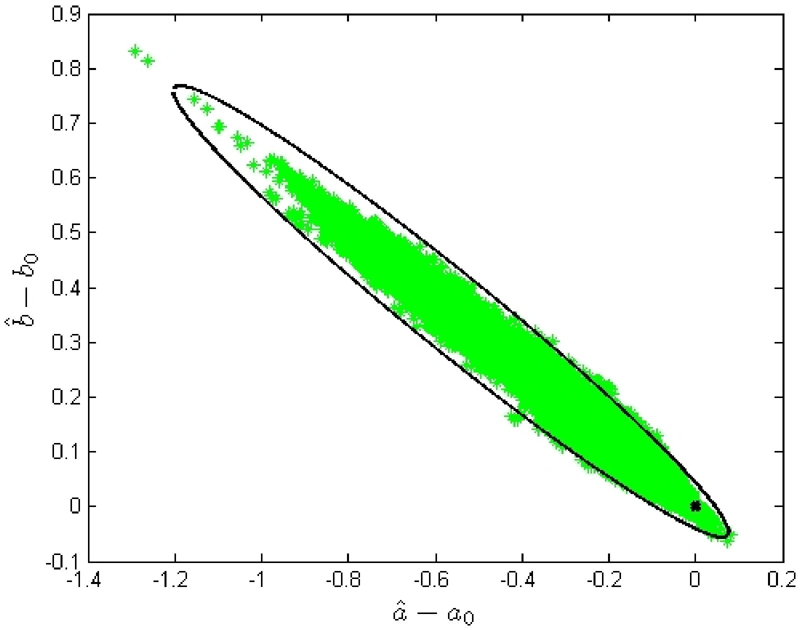


Figure 5.7: Residual indicator of parameters in scenarios without faults and set $f^T(r)\lambda \leq 0$

One advantage of the use of classifiers is that, in order to improve discrimination between scenarios with and without fault, other indicators can be added. In this case, in addition to the

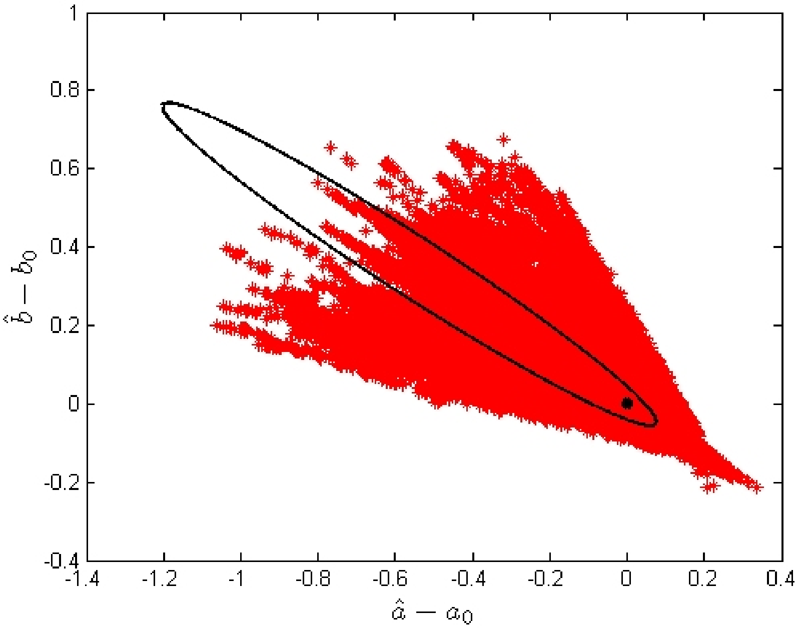


Figure 5.8: Residual indicator of parameters in scenarios with faults and set $f^T(r)\lambda \leq 0$

indicator described above, other indicators have been added: the same error in the estimation of parameters but with windows of 15 and 100 samples, the temporal residual (5.3) using the model given by (5.6) and accumulating this residual with windows of 15, 50 and 100 samples, and maximum values in the measures determined by the maximum allowed draft in the collector.

Having decided the indicators to be used by the classifier, the algorithm 5.2 has been used to find the classifier that meets the probabilistic constraints MF defined above (a priori) obtaining $\tau = 7.1$. Applying the designed classifier to the set of scenarios without failure of the system it is obtained, a posteriori, one level of false alarms $FA < 4.5\%$ with a probability not less $1 - 10^{-6}$ ($\eta_N = 0.045$ $\delta_N = 10^{-6}$)

5.7 Conclusions

In this chapter we have proposed a general methodology for the design of fault detectors with probabilistic guarantee. The great advantage of the proposed methodology is, on the one hand its flexibility to introduce different tools of fault detection and on the other its certified probabilistic guarantee of the proposed detector. The operation of this methodology has been illustrated with an application example to a virtual deposit. As future work, following the philosophy of the proposed scheme, it would be interesting to address the problem of designing a diagnostician to determine, once a fault is detected, which type of failure has occurred with a certain probabilistic guarantee.

Chapter 6

Application to frequential identification

6.1 Introduction

The main objective is the identification of a fuel cell applying randomized algorithms (Ponce, 2013). It aims to develop a testing tool in Matlab to evaluate the batteries in the laboratory and obtain an optimal signal stimulus for modeling these, in order to characterize them in the future as quickly and accurately as possible . This algorithm is based on the results presented in (Álamo et al., 2010*a*) . Batteries are very complex devices that have many components, and their performance depend on variables that can be easily measured , such as voltage and temperature , and others that are not so easy to measure, such as age, manufacturing tolerances and variations between cells within the battery, which can have a big impact on performance at the end. In all manufacturing processes (specially chemical processes), no matter how good they are, something unexpected can always occur. There are several reasons why one should test battery systems: to ensure that the equipment is fine to prevent unexpected failures by checking the battery status, to warn or prevent depletion and to answer three basic questions: What is the capacity and battery status now?; When to replace it?; What can you do to improve its life? For all these reasons, tests should always be performed for proper maintenance. One important thing to know is the state in which the battery is, are provided by the capacity and the electrical impedance model.

The approach of this chapter is useful for this modeling, as it deals with finding the signal which can get the best approximation of the electric battery model . During execution of the experiments the batteries will be excited with piecewise constant signals chosen randomly. Randomized algorithms are many times faster than the traditional algorithms and other solutions , solutions that are not possible within the domain of traditional algorithms. In common practice, the algorithms approximate random number generator with pseudo-random number generation. In this chapter a brief introduction to randomized algorithms is made, and the current state of the storage systems on the market and new lines of research are also presented.

6.2 Fuel cells

Several properties make fuel cells attractive for electricity production, but the strongest incentives are high efficiency and high energy density. The fuel cell provides a highly efficient conversion of the chemical energy in hydrogen, natural gas, or hydrocarbons into electrical energy. This is extremely beneficial for automotive applications and small-scale energy production for stationary uses. And because of their high energy density (energy per unit weight of the power source), fuel cells are superior to batteries in portable equipment. Fuel cells may have their greatest environmental impact in motor vehicles. In most automotive engines, gasoline is converted into mechanical energy through heat produced during combustion. The efficacy of this process is limited by the efficiency formula for the Carnot cycle, which describes the thermomechanical operation that takes place in a gasoline engine. The operating temperature of the engine determines its efficiency, which is about 20 per cent for an automobile. In a fuel-cell-powered vehicle, the chemical energy in the fuel is converted to electrical energy and then to mechanical energy by an electric motor. The process bypasses the limitations of the Carnot cycle, such as the 20 per cent engine efficiency. As a result, the theoretical efficiency of fuel cells is substantially higher than that of the combustion engine (around 90 per cent). In practice, values reach about 50 per cent.

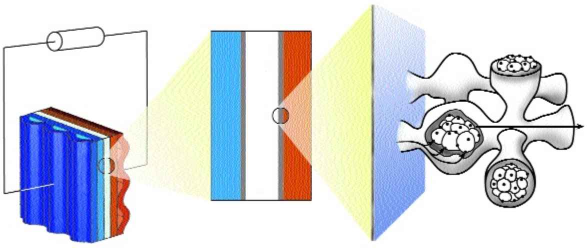


Figure 6.1: Fuel Cell

This higher efficiency implies that fuel-cell-powered automobiles can travel more than twice as far as conventional cars using the same amount of fuel. Consequently, carbon dioxide emissions are lower. The electronics industry is pursuing miniature fuel cells. Motorola has demonstrated a cellular telephone powered by a fuel cell whose operating time with one fuel cartridge is five times longer than that of a conventional battery with one recharge. Other likely applications include power supplies in laptop computers and portable video, audio, and entertainment equipment. Fuel cells, which generated electrical power for the Apollo spacecraft during the U.S. lunar-landing program, have great potential for generating power in places without an electrical infrastructure. They can also serve as power sources in boats, portable construction tools, temporary traffic-control stations, mobile life-support units, and military applications. In batteries, energy is stored as chemical energy in the battery itself. In fuel cells, chemical energy is stored in the fuel tank and electricity is produced on demand. This separation eliminates downtime for recharging (add a source of hydrogen, and the fuel cell can generate electricity in less than 1 minute). Fuel cells also provide an uninterrupted power supply because there is no self- discharge, as occurs with batteries, an advantage that minimizes maintenance and maximizes the reliability of the system. In addition, fuel cells have a higher energy density and therefore a higher power capacity per unit of weight. The development of fuel-cell-powered equipment and vehicles has accelerated during the past years (see The Industrial Physicist, February 1999). Competition among companies is growing, and the fight for a share of a potentially huge market has already started. Technology development is an important weapon at this stage, and small companies with technical skills in fuel-cell processes have become important partners to the large electronic and automotive companies. In this rapidly developing and highly competitive market, the time from idea to prototype has shrunk. Therefore, tools for developing virtual prototypes have become exceptionally important. Optimizing performance of a fuel cell in combination with its auxiliary equipment and operation of the electric motor requires a lot of mathematical puzzling. Thus, mathematical modeling, the basis of virtual prototyping, is a vital tool in the development of fuel cells.

A combination of modeling and experimentation has reduced the cost and accelerated the pace of building and understanding prototype systems. Modeling provides valuable insights into the electrochemistry of the fuel cell and the processes that take place in the heart of the fuel cell system (the electrodes and electrolyte in the fuel cell stack). These processes are described at the microlevel: single catalyst agglomerates; a unit cell consisting of an anode, a cathode, and the electrolyte between them; and as reactor models of the fuel processor in fuel cells for cars. Other important aspects include the design of the bipolar plates of the fuel cell, their influence on ohmic losses in the fuel cell stack, and the use of modeling to optimize the design of fuel cell systems.

6.3 Frequential analysis

Frequency response is the quantitative measure of the output spectrum of a system or device in response to a stimulus, and is used to characterize the dynamics of the system. It is a measure of magnitude and phase of the output as a function of frequency, in comparison to the input. In simplest terms, if a sine wave is injected into a system at a given frequency, a linear system will respond at that same frequency with a certain magnitude and a certain phase angle relative to the input. Also for a linear system, doubling the amplitude of the input will double the amplitude of the output. In addition, if the system is time-invariant, then the frequency response also will not vary with time.

One application of frequency response analysis is to give the closed-loop system improved response as compared to the uncompensated system. The feedback generally needs to respond to system dynamics within a very small number of cycles of oscillation (usually less than one full cycle), and with a definite phase angle relative to the commanded control input. For feedback of sufficient amplification, getting the phase angle wrong can lead to instability for an open-loop stable system, or failure to stabilize a system that is open-loop unstable.

6.4 Estimation and plotting. Bode diagram

Estimating the frequency response for a physical system generally involves exciting the system with an input signal, measuring both input and output time histories, and comparing the two through a process such as the Fast Fourier Transform (FFT). One thing to keep in mind for the analysis is that the frequency content of the input signal must cover the frequency range of interest or the results will not be valid for the portion of the frequency range not covered.

The frequency response of a system can be measured by applying a test signal, for example: applying an impulse to the system and measuring its response; sweeping a constant-amplitude pure tone through the bandwidth of interest and measuring the output level and phase shift relative to the input; applying a signal with a wide frequency spectrum (for example digitally-generated maximum length sequence noise, or analog filtered white noise equivalent, like pink noise); and calculating the impulse response by deconvolution of this input signal and the output signal of the system.

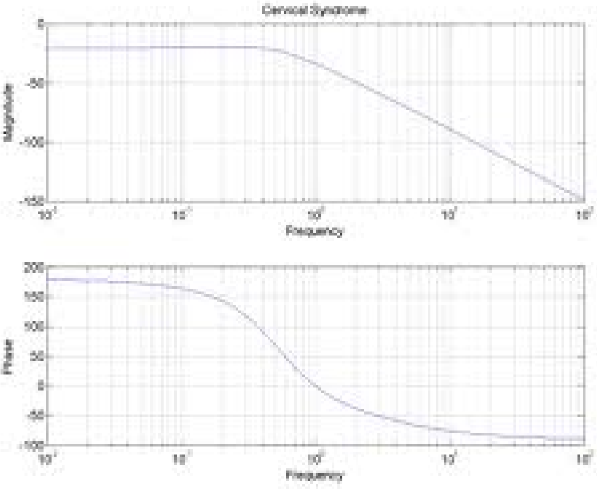


Figure 6.2: Bode Diagram

The frequency response is characterized by the magnitude of the system's response, typically measured in decibels (dB) or as a decimal, and the phase, measured in radians or degrees, versus frequency in radians/sec or Hertz (Hz).

These response measurements can be plotted in three ways: by plotting the magnitude and phase measurements on two rectangular plots as functions of frequency to obtain a Bode plot; by plotting the magnitude and phase angle on a single polar plot with frequency as a parameter to obtain a Nyquist plot; or by plotting magnitude and phase on a single rectangular plot with frequency

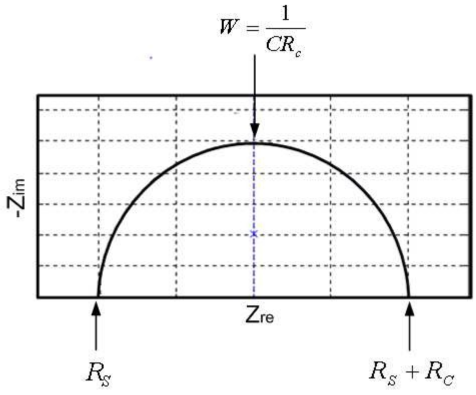


Figure 6.3: Nyquist Diagram

For design of control systems, any of the three types of plots [Bode, Nyquist, Nichols] can be used to infer closed-loop stability and stability margins (gain and phase margins) from the open-loop frequency response, provided that for the Bode analysis the phase-versus-frequency plot is included.

6.5 Randomized algorithms

We consider N experiments consisting in the random i.i.d. generation of N sequences (of in principle unlimited length) numbers in the interval $[-1,1]$.

$$\begin{aligned}
w^{(i)} &= \{w_1(1), w_1(2), \ldots, w_1(k), \ldots, \\
w^{(2)} &= \{w_2(1), w_2(2), \ldots, w_2(k), \ldots, \\
\vdots &= \vdots \\
w^{(N)} &= \{w_N(1), w_N(2), \ldots, w_N(k), \ldots,
\end{aligned}$$

We assume a strategy to generate a random signal to excite the battery. The strategy depends of four parameters:

1. Number of samples per second: n_s.
2. Number of seconds per period: n_p.
3. Number of periods per experiment: n_e.
4. Excitation factor: A.

The signal corresponding to experiment $w^{(i)}$ and parameters n_s, n_p, n_e and A is such that it is piecewise constant (during $\frac{1}{n_s}$ seconds). The value corresponding to the k-th constant period is $Aw_i(k)$. This signal is periodic with period n_p. That is, only the first n_pn_s random numbers of the sequence $w^{(i)}$ are used. This signal is repeated n_e times. That is, the total duration of the experiment is then n_pn_e.

Therefore, each experiment $w^{(i)}$ provides a temporal signal $u^{(i)}(n_s,n_p,n_e,A,t)$ that depends on the particular choice for n_s, n_p, n_e and A. An experimental Bode can be obtained from this signal and we can assess the quality of the obtained experimental Bode comparing it with the real one. We say that the parameters n_s, n_p, n_e and A are appropriate for the particular experiment $w^{(i)}$ if a given performance index is satisfied or not (this index depends on the quadratic error between the results and the theoretical bode). In order to simplify the notation, we define θ as the design vector composed by the four parameters. That is,

$$\theta = \begin{bmatrix} n_s \\ n_p \\ n_e \\ A \end{bmatrix}.$$

We assume therefore that one has the function $g(\cdot,\cdot)$ that provides a value of 0 if the specifications are met and 1 otherwise. That is,

$$g(w^{(i)},\theta) = \begin{cases} 0 & \text{if} \quad \theta = \begin{bmatrix} n_s \\ n_p \\ n_e \\ A \end{bmatrix} \text{meets the specifications for experiment } w^{(i)} \\ 1 & \text{otherwise.} \end{cases}$$

We assume that every possible parameter vector θ belongs to parameter design set Θ, which given the integers m_s^+, m_p^+, m_e^+ and m_A^+ is given by:

$$\Theta = \{ \begin{bmatrix} n_s \\ n_p \\ n_e \\ A \end{bmatrix} = \begin{bmatrix} 2^{m_s} \\ 10^{m_p} \\ n_e \\ \frac{m_A}{m_A^+} \end{bmatrix} : 1 \le m_s \le m_s^+, -m_p^+ \le m_p \le m_p^+, 1 \le n_e \le m_e^+, 1 \le m_A \le m_A^+ \}$$

In the previous definition, m_s, m_s, n_e and m_A are integers. Therefore, the cardinality of Θ is $n_C = m_s^+(2m_p^+ + 1)m_e^+ m_A^+$. Between the possible values for θ, one should take the one minimizing a given performance criteria $J(\theta)$ subject to the constraint that the specifications are met for a given percentage of the experiments. That is, the following problem should addressed

$$\begin{array}{ll} \min\limits_{\theta \in \Theta} & J(\theta) \\ s.t. & \sum\limits_{i=1}^{N} g(w^{(i)}, \theta) \le m \end{array}$$

It has been proved in (Álamo et al., 2010*a*) that if

$$N \ge \frac{1}{\eta}\left(m + \ln \frac{n_C}{\delta} + \sqrt{2m \ln \frac{n_C}{\delta}} \right)$$

experiments are generated and a feasible (maybe suboptimal) solution $\hat{\theta}$ is obtained for the proposed optimization problem then this solution satisfies with a probability no smaller than $1-\delta$ that for any experiment w, the probability of having $g(w, \hat{\theta}) = 1$ is no larger than η.

6.6 Application. Modeling and identification of fuel cells

Any of the mentioned factors can have a huge impact on the final performance, it is convenient to carry out a monitoring of the state of the battery. The best way to do it is obtaining its electrical model and use this data to establish tendencies about changes. It is interesting to model this kind of systems from the electrical point of view, because the real behavior is approximated, allowing the use of the more important experimental measurements. The equivalent circuit of a battery is mainly a impedance. By definition, a impedance is the opposite of a current measurement. If there exists any way to quantify precisely the degradation of the battery, it would be measuring the internal impedance. The value of internal impedance has a direct relationship with the degradation of its internal components. The impedance is a modern, quick and low-cost test of the battery (Ponce, 2013).

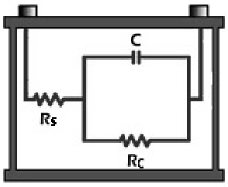


Figure 6.4: Electrical model of the battery

Figure 6.4 represents the equivalent circuit for an energy cell. R_s corresponds with the metallic resistance. C is the capacity of the parallel plates that constitute the electrodes of the cell and R_c is the resistance of charge transfer.

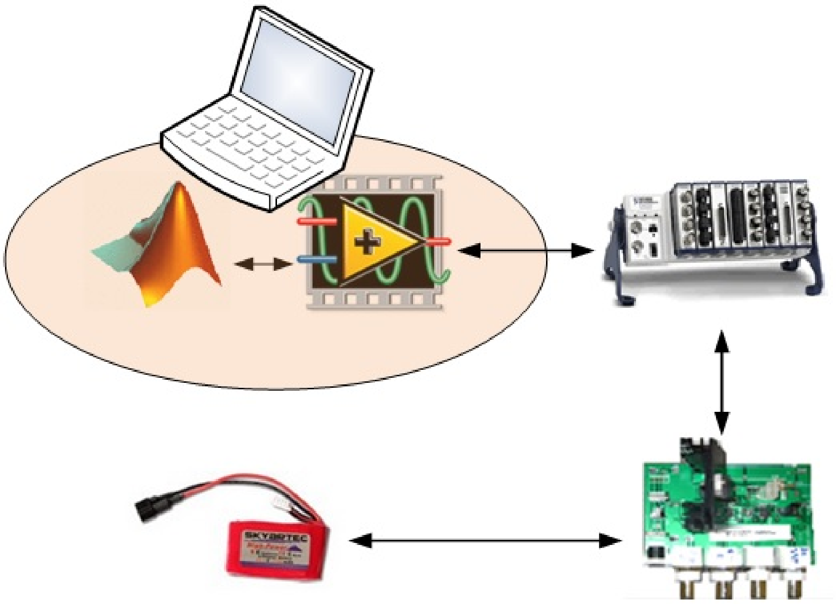


Figure 6.5: Methodology

To obtain the frequential response of a system, it can be stimulated with any kind of signal, to compare input and output and obtain the relationship in magnitude and phase, and then the experimental Bode. Some inconveniences can appear, for example, if the system has a very slow dynamics the test signal have to be of very low frequency, and the test time would be very long. For the optimization of the response analysis in frequency, an algorithm is implemented with the purpose of obtaining the optimal stimulus signal. N experiments based on the generation of N numerical random sequences i.i.d. are considered, in the interval [-1,1], as explained in previous sections.

6.7 Methodology and design of the tool

The battery is stimulated with random signals, to measure voltage and current, process these measurements and obtain data of the impedance in frequency domain. From the internal impedance of the battery the characteristical parameters of the electrical model (R_s, R_c y C) of the battery can be computed by EIS (Electrochemical Impedance Spectroscopy) tests. With these R_s,R_c and C values precision of the electrical model is valorated, comparing values with the theoretical ones. The goodness of the stimulus signal could be obtained this way. Stimulus signals are constant signals defined by four parameters:

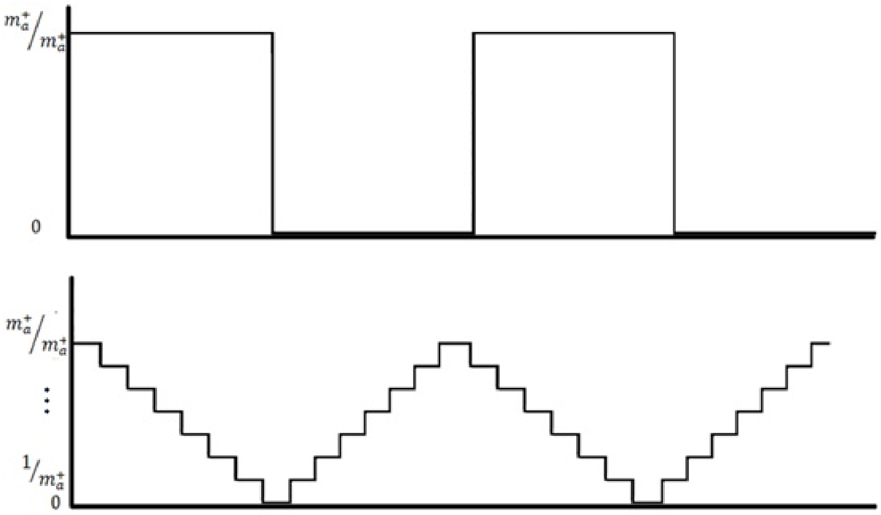


Figure 6.6: Normalization

- Maximum amplitude that the signal can have. The values of the steps is in the interval $[0, m_a^+]$. Then the signal is normalized in amplitude, bounding it in the interval [0,1] with a minimum gap of $1/(m_a^+)$. With a higher maximum amplitude higher precision will be obtained.
- Number of periods in the signal (n_e).
- Time length of the period (n_p), an entire number.
- Sampling frequency, defined as the number of samples per time unit, taken from a continuous signal to produce a discrete signal. The used unit is the Hertz. It would be defined by 2^{m_s}, being m_s an entire number.

For the random generation of the stimulus, a function is created. This function define the stimulus from random numbers. As a result, different signals defined with the same

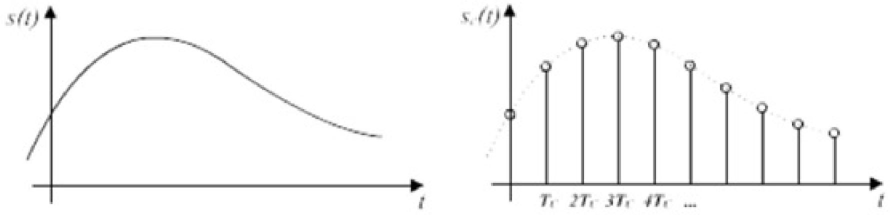


Figure 6.7: Sampling

parameters will be completely different.The set of signals defined with the same parameters is called family of signals. To study the battery a single complex and extensive experiment is done. This experiment is composed by many test or small sub-experiments, each of them corresponding to a different stimulus signal for the battery. Among all these stimulus signals we will select the one with the closest results to the theoretical ones.

To do the experiments, it is necessary to define a limit for each parameter that determines the stimulus signal. The total length of the experiment depends on this choice. They will be introduced by the user at the beginning of the experiment by a graphical interface, created to make the tool more intuitive. The length of the experiment can therefore be modified depending on the battery to study. All the experiments done for the same battery have to be identical in length, that is to say, the same limits should be introduced in all the experiments, in order to avoid affecting the obtained results. Once the limits are introduced, the experiment is configured.

During the execution of the experiments, all the possible signals in the limits should be evaluated. It is necessary to have a main loop, with several loops in it, to go over the different parameters.

For each set of four parameters the main function or program of the tool is run. Each execution of the main function corresponds with a sub-experiment. During it, a stimulus signal to excite the battery is generated, data are read and processed, minimizing the noise as far as possible, using different filters and approximations. At last, energy of the system and characteristical values of the battery are computed.

Having small period time and small sampling period can cause a precision not enough to make a suitable frequential analysis, not having enough points. The sampling frequency also determines the complexity of the process. More acquired samples are more samples to be processed.

$$(N_{points})/period = n_s \times n_p$$

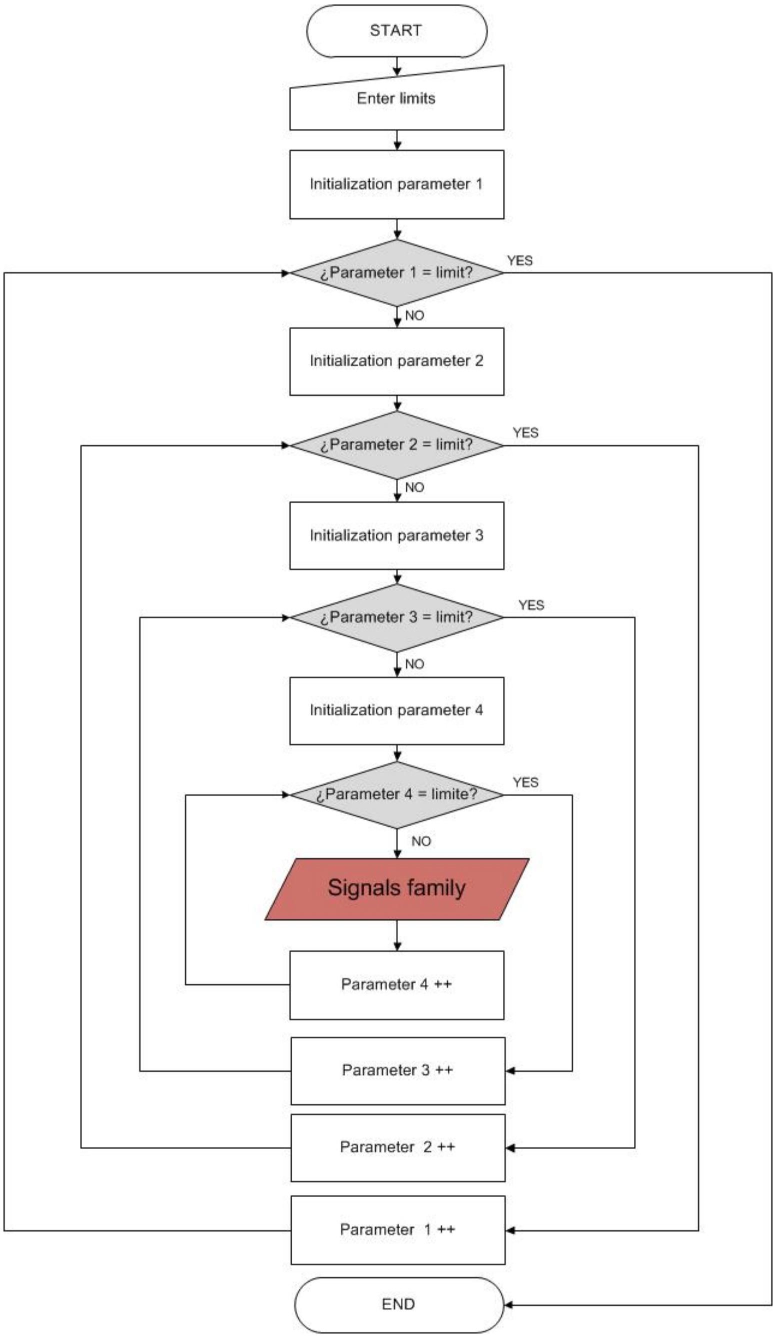


Figure 6.8: Flow diagram

Only signal with at least two points per period are considered ($n_s \times n_p \geq 2$).

An experiment is build from the execution of the main program or function with each of the different stimulus signals. The inputs for the main program are the characteristics of the signal, and the outputs are the characteristical values of the electrical model of the battery (R_s, R_c y C) and the energy (J). The closest to the theory results are chosen.

The validation of the obtained results consists on a comparison between the experimental values of the electrical model and the theoretical ones. If the theoretical values of the battery are not available, they will be previously computed by a frequencial sine-wave scanning as shown in figure 6.9. This process has to be done only the first time the battery is studied, because data are saved for future uses.

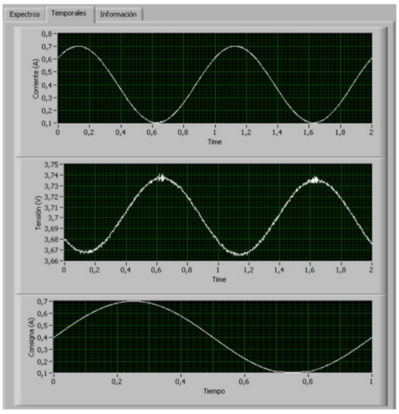


Figure 6.9: Frequencial sine-wave scanning

If the error between the characteristical parameters of the battery computed by the tool and the theoretical ones is higher than 10 per cent, the signal will be considered a fail and the analysis will continue with more signals of the same family. The error is defined as:

$$error = (ExperimentalValue)/(TheoreticalValue) \times 100$$

If the number of failures in the same signal families is higher than the maximum allowed one, defined as *m* for the total of *N* experiments done for each family, the family is discarded. If the error is smaller of 10 per cent, energy of the system (J) is computed. The solution will be chosen minimizing the energy, among all the valid families satisfying the constraint of allowed failures.

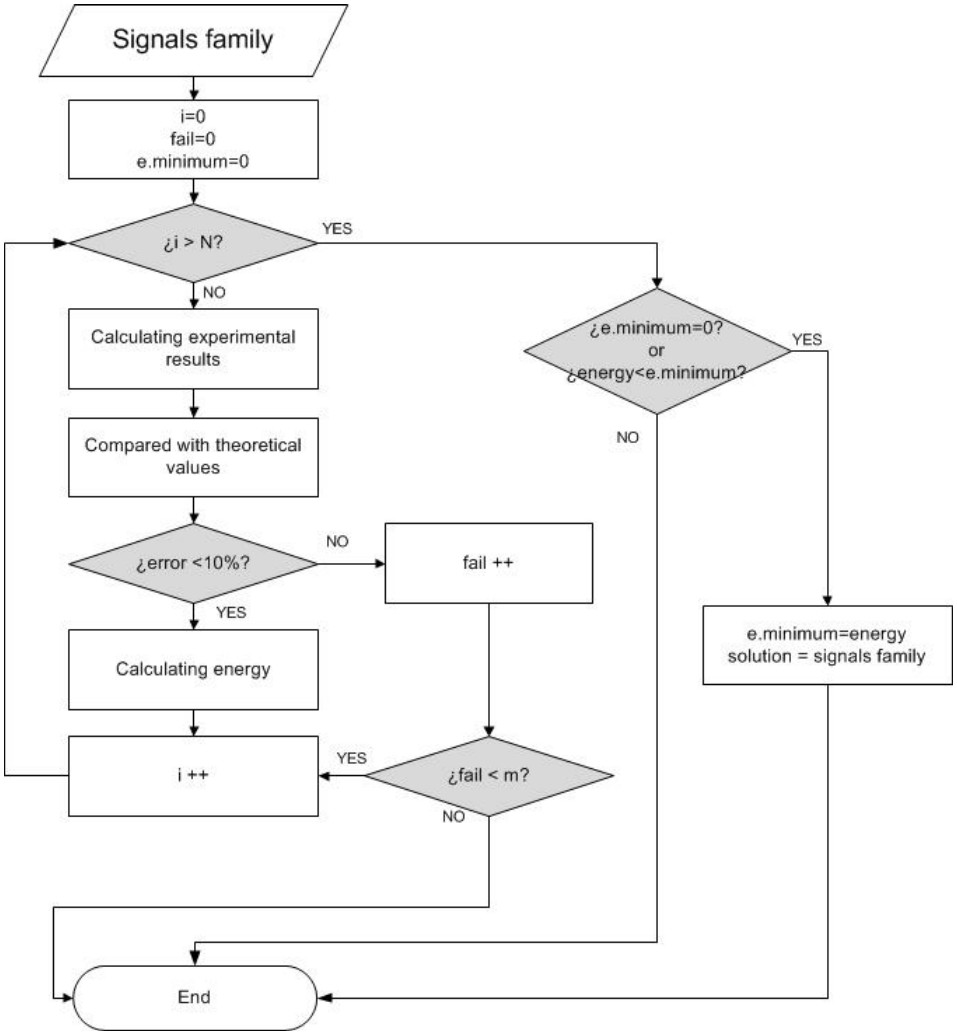


Figure 6.10: Flow diagram

6.8 Results

Some tests done with the aim to show the tool's power are presented. All the experiments done with the same fuel cell have the same limit parameters. A maximum frequency of 10 kHz for the stimulus signal has been chosen, because for higher frequencies data can not be acquired properly. Results of a first experiment are shown.

An optimal signal to be used as stimulus is obtained. Its sampling frequency is 2048 points per second, 7 cycles or steps, 1 second long each of them, and with an amplitude of 10 A.

Once the parameters that define the stimulus signal are obtained, we proceed to verify the validity of this signal. A signal with the same characteristics is generated, and one single

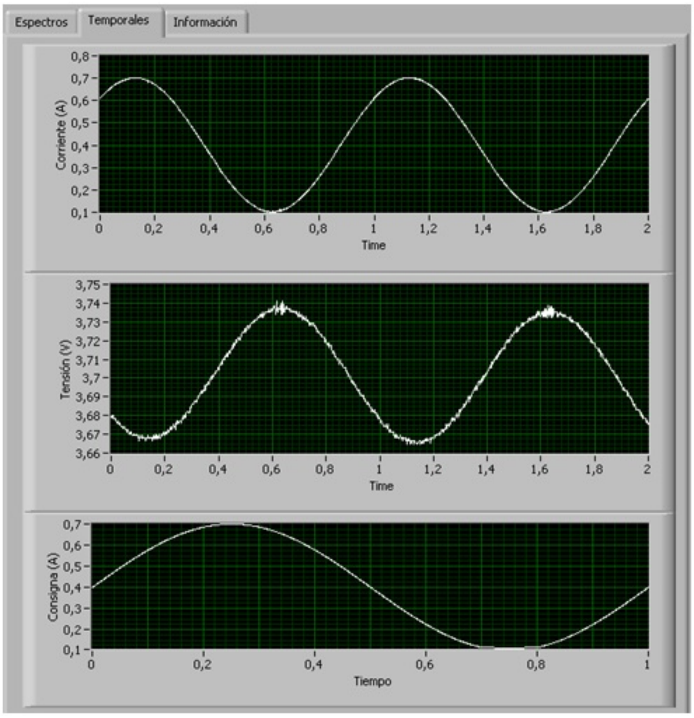
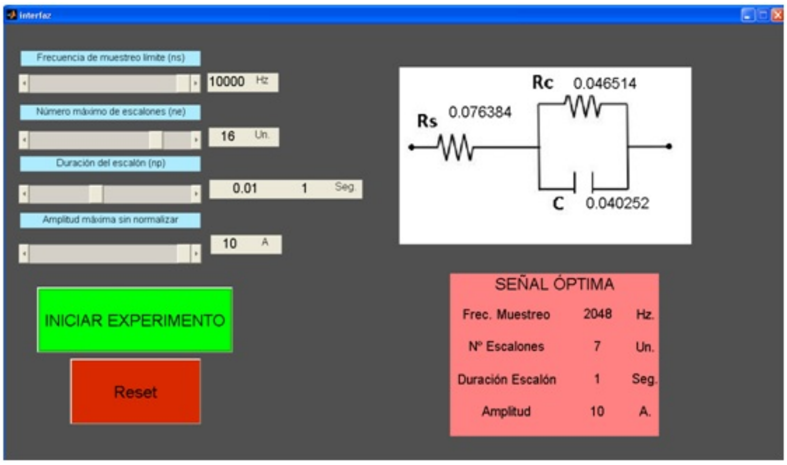


Figure 6.11: Overflowing



Frecuencia de muestreo límite (ns) 10000 Hz

Número máximo de escalones (ne) 16 Un.

Duración del escalón (np) 0.01 1 Seg.

Amplitud máxima sin normalizar 10 A

Rs 0.076384

Rc 0.046514

C 0.040252

SEÑAL ÓPTIMA		
Frec. Muestreo	2048	Hz.
Nº Escalones	7	Un.
Duración Escalón	1	Seg.
Amplitud	10	A.

Figure 6.12: Results of a first experiment

test is done, to obtain the frequential response of the battery and draw Bode and Nyquist diagrams.

Having all the measurements, they are processed to eliminate all the noise, obtaining useful data for interpretation.

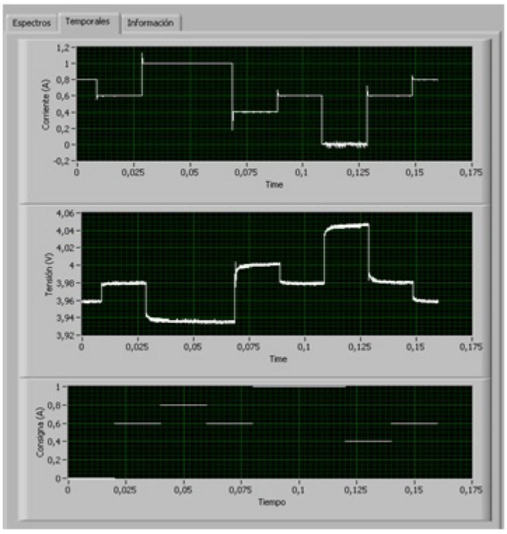


Figure 6.13: Data acquisition

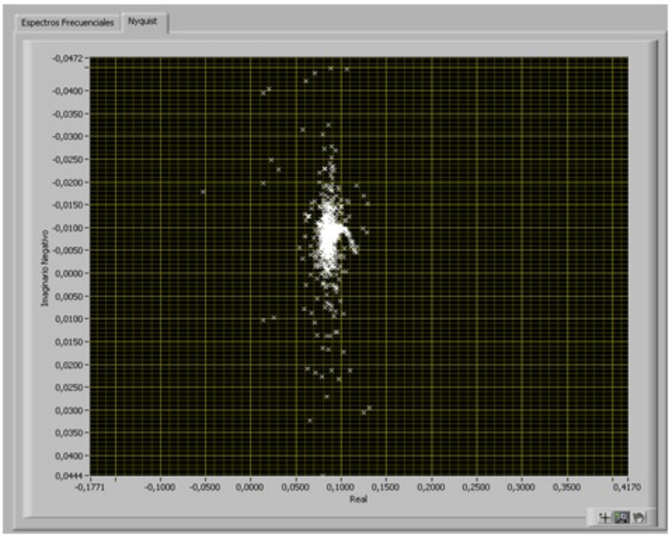


Figure 6.14: Frequency response

Results obtained using the stimulus signal resulting from the experiments are quite good. The obtained characteristical values of the equivalent electrical model are

$$R_s = 0.0729$$
$$R_c = 0.0481$$
$$C = 0.0398$$

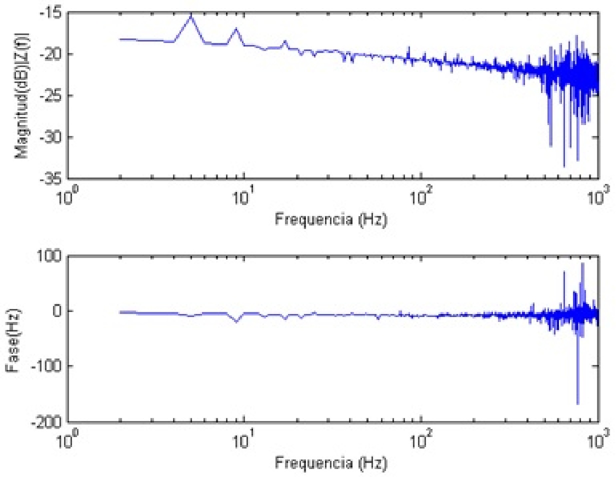


Figure 6.15: Experimental Bode

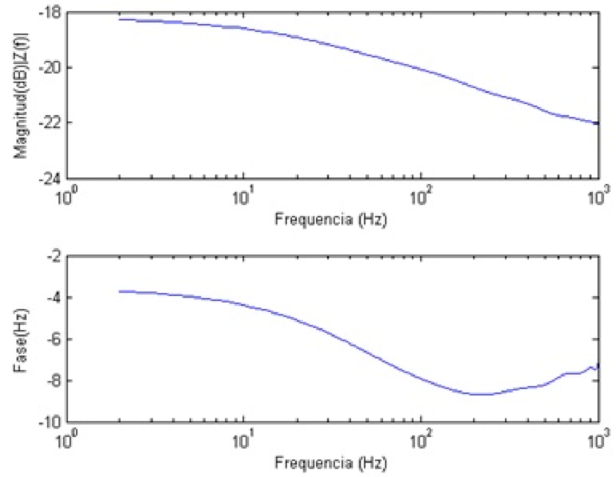


Figure 6.16: Processed Bode

To confirm the results, the experiment is repeated, obtaining almost the same result. After the verification of the results achieved for the first battery, some experiments are done with the other battery available in the laboratory. The results are shown in the following figure.

As both battery have the same specifications, the results is very similar. In this case it seems that the battery is in a better condition, due to the fact that in the equivalent electrical model the capacity is a bit higher, while the internal resistance is slightly smaller.

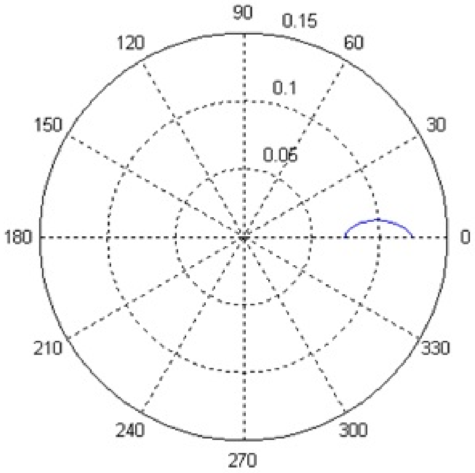
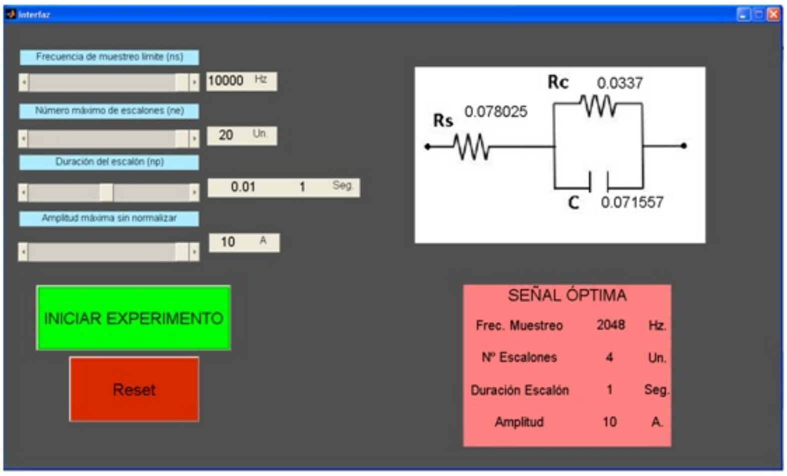


Figure 6.17: Experimental Nyquist



Frecuencia de muestreo limite (ns) 10000 Hz

Número máximo de escalones (ne) 20 Un.

Duración del escalón (np) 0.01 1 Seg.

Amplitud máxima sin normalizar 10 A

Rs 0.078025

Rc 0.0337

C 0.071557

SEÑAL ÓPTIMA

Frec. Muestreo	2048	Hz.
Nº Escalones	4	Un.
Duración Escalón	1	Seg.
Amplitud	10	A.

Figure 6.18: Results of the second experiment

6.9 Conclusions

The aim of this work has been developing a tool able to obtain an optimal signal to characterize fuel cells applying randomized algorithms. To do it, fuel cells have been stimulated by a randomized input signal and their frequential response has been studied. Variations in magnitude and phase have been translated to a Nyquist diagram.

To build the tool, Matlab program has been used, as well as for the connection between the fuel cell and Labview. Communication between these two programs has been a challenge. The solution requires the exchange of the text files with the data wanted for each of the programs, depending on the sense of the communications. Concluded the design of the tool, some tests have been done in order to confirm its proper performance. This tests have been a success, and many results checks have been done. The tool is able to generate a random signal, interact with Labview to stimulate the battery, make the acquisition and measuring, and compute the frequential response and characteristical parameters. Once obtained a signal useful to characterize a battery, it is no longer necessary to perform long experiments in every frequency. Not only the optimal signal is obtained, but also all the parameters of the equivalent models. Comparing these values with old ones it is possible to diagnose the current state of the battery.

The main advantages for this method that can be mentioned are:

- The saving of time.
- It can be run in any computer, having Matlab, Labview and the acquisition tools
- Any kind of battery can be studied, changing the limits of the experiments to improve performance.
- The visualization of the electrical equivalent model.

As future work, a data base could be added to the tool. This way a complete historical archive of all studied cells could be used. When studying a battery its optimal stimulus signal could be found in the historical archive. Another possible future work is integrating the available Labview program in the main Matlab program and to make communications more fluent, making the tool more robust and composed of a unique block.

Chapter 7

Application of R.A. to feedback controllers testing and congestion control

7.1 Introduction

In computer networks congestion appears when there are too many sources sending data too fast for the network to handle. Techniques to reduce congestion are of great interest. This chapter concentrates on congestion control methodologies where feedback control techniques provide efficient solutions ((Jacobson, 1988), (S. Ryu and Qiao, 2004), (Hollot et al., 2002), (Sun et al., 2007), (Floyd and Jacobson, 1993)).

A central problem in designing controllers for these systems is the difficulty of ensuring adequate performance in all possible conditions, as these systems operate under a very wide range of conditions, are inherently nonlinear and suffer from significant time-varying delays. Thus, designers frequently have to show the effectiveness of their proposal by extensive simulations, which is a time-consuming methodology, and does not offer a definite guarantee of performance: simulation results in most of the references show only specific cases and scenarios. Prompted by this problem, the chapter concentrates on the following issue: given a required degree of confidence, how many simulations are needed to check the adequate performance of the controllers?

Thus, we develop a randomized approach based on some ideas in ((Su-Woon et al., 2012),(Álamo et al., 2009), (Álamo et al., 2010*b*), (Fujisaki and Kozawa, 2006)), to test whether a controller robustly satisfies a set of specifications with a given probabilistic error margin. The results presented in this chapter are stated in an implicit way, that is, the number of experiments required is obtained from a simple numerical procedure. The theoretical framework proposed in (Álamo et al., 2009) and (Álamo et al., 2010*b*) is conveniently tailored for this particular application. Hence this chapter constitutes a proof of concept of the methodology proposed in the aforementioned references. The main idea is to test the controller under a finite set of different scenarios. When the controller satisfies the specifications for a sufficient number of these scenarios, then certain properties can be concluded with a given degree of confidence, and no more simulations are needed.

One of the main characteristics of the technique is that it is independent of the family of controllers (PI, PID, predictive, robust, etc). The methodology is applied in this chapter to the active queue management (AQM) scheme, which complements the end-to-end Transmission Control Protocol (TCP), at the routers' transport layer. The AQM objectives (S. Ryu and Qiao, 2004), (Hollot et al., 2002), (Sun et al., 2007), are efficient queue utilization, queuing delay and robustness. Numerous AQM algorithms have been proposed (see (S. Ryu and Qiao, 2004) for a good survey on the subject), with Random Early Detection (RED) (Floyd and Jacobson, 1993) being the most widely used algorithm, as it can detect and respond to long-term traffic patterns. This chapter uses the AQM mathematical models published in (Hollot et al., 2002), and extensively used in the literature ((Jacobson, 1988), (S. Ryu and Qiao, 2004), (Hollot et al., 2002), (Floyd and Jacobson, 1993), (Vidyasagar, 2001)) and the references therein) for controller design and testing. The main metrics proposed to determine controller performance are: router queue size (real value and standard deviation), link utilization and the probability of packet losses.

As a demonstration, the proposed technique is applied to a problem of two routers connected in a Dumbbell topology, which represents a single bottleneck scenario. The length of their queues is controlled with a PID ((Aström and Hägglund, 2006), (nez et al., 2011)) whose probabilistic properties are guaranteed following the results presented in the chapter. The simulations are done using the software ns-2, which is a discrete event simulator targeted at networking research, providing substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless networks. It must be pointed out that although the proposed methodology was prompted by a congestion control problem, and is demonstrated on this problem, it can be directly applied to other control testing problems, as plants to be controlled are frequently nonlinear, uncertain and subject to parameter variations ((nez et al., 2011)). This chapter is organized as follows: Section 2 introduces the theory behind the proposed randomized test method. Results and an example of application are given in section 3. Finally, some conclusions are presented (Maestre et al., 2012).

7.2 Randomized test method

This section develops the mathematical tools needed to establish a bound on the number of simulations needed to guarantee certain properties of the feedback controller with a prespecified level of confidence.

7.2.1 Notation

$\theta \in \Theta$ is a vector representing the parameters that characterize the controller (as the controller is a PID, there are three parameters: the proportional, the integral and the derivative terms). $w \in W$ is a vector that contains the parameters that characterize any possible scenario where the controller will work (parameters of the model, states and inputs).

7.2.2 Assumptions

The following assumptions are needed to develop the proposed methodology:

1. The controller belongs to a finite family Θ that contains m elements: $\Theta = \theta_1, \theta_2, \ldots, \theta_m$. **Remark 1**: This assumption is frequent in robustness problems, where the controller design parameters, along with different auxiliary variables, are parameterized by means of the decision variable vector $\theta \in \Theta$. If the set of parameters is infinite, finite cardinality can be forced by gridding (Álamo et al., 2009).

2. The network and its possible inputs can both be fully determined in terms of a set when the parameters are contained in the vector $w \in W$. This assumption relies on the fact that for these particular systems there exist well-known parameterized models (see for example (Fujisaki and Kozawa, 2006)) that can be used to simulate the network. The set W represents a reasonable range of variation of not only the parameters that characterize the given network but also of additional terms of probabilistic nature.

3. Given $t_2 > t_1$, a scenario $\widehat{w} \in W$ and a controller $\widehat{\theta} \in \Theta$, there exists a **procedure** to evaluate whether the controller $\widehat{\theta}$ fulfils the design specifications for a particular scenario during the time interval $[t_1, t_2]$ for the scenario $\widehat{w}$. Mathematically, the procedure is a function g such that:
$$g : \Theta \times W \longrightarrow 0, 1.$$
Thus, the procedure g gives 0 if the specifications are satisfied; otherwise it gives 1. This is a very mild assumption. We just assume that we can check if a given controller satisfies the design specifications for a given scenario w.

   Remark 2: For congestion control, this procedure is based on the result of a network simulation that makes it possible to check whether a given controller fulfils the specifications.

4. There is a probability distribution Pr_W defined over the set W and an algorithm that provides elements of W according to Pr_W. In other words, it is possible to generate a sequence $w_1, w_2, \ldots, w_k$ of characteristic independent and identically distributed elements of W obtained according to Pr_W. **Remark 3:** This assumption can be relaxed

if it is possible to generate the sequence of characteristic elements $w_1, w_2, \ldots, w_k \in W$ without knowing the exact probability distribution of the parameters.

7.2.3 Problem formulation

The objective is to test whether a controller satisfies the design specifications described by the function g with a probability greater than $1 - \varepsilon$ that is:

$$E_g(\widehat{\theta}) = Prw \in W : g(\widehat{\theta}, w) = 1 < \varepsilon,$$

where ε is the probability that the given controller $\widehat{\theta}$ does not work properly in a scenario w. In general, it is not possible to test the controller in all possible scenarios, so we propose a probabilistic approach. To this end, the set of N scenarios $w_1, w_2, \ldots, w_N \in W$, that fulfils Assumption IV, is used. These scenarios test if the controller satisfies the specifications and allow the probability of failure to be estimated:

$$\widehat{E}_g(\widehat{\theta}) = \frac{1}{N} \sum_{i=1}^{N} g(\widehat{\theta}, w_i) \leq \rho$$

where ρ is a bound of the probability of failure (level parameter) for the controllers being tested (empirically estimated). In other words, ρ sets the maximum acceptable probability of failure, so it can be used as a design parameter. Additionally, we will require $\rho < \varepsilon$, that is, the estimated probability of failure must be lower than the real probability of failure ε . This constraint means that in practice more restrictive conditions are applied and provides a margin to assure that the real behavior will fail with a probability smaller than ε. As a probabilistic methodology is applied, it is important to consider the probability of false positives, that is, that (2) is empirically satisfied, but (1) is not (these false positives correspond to situations in which the choice of the characteristic sequence $w_1, w_2, \ldots, w_N \in W$ does not represent W in an appropriate way). This will be considered by estimating the probability β of the failure of the methodology, defined as:

$$\beta = Pr(E_g(\widehat{\theta}) > \varepsilon | \widehat{E_g}(\widehat{\theta}) < \rho)$$

Note that, given a controller $\widehat{\theta} \in \Theta$, this probability is a function of ε, ρ and N. Logically, the greater the number of different scenarios used to test the controller, the better the results provided by the proposed methodology. It can be proved that the function of the probability of failure of the methodology verifies:

$$\lim_{N\to\infty} \beta(\varepsilon,\rho,N) = 0.$$

As it is not possible to make infinite simulations to guarantee that the probability of false positives of the methodology is 0, we will set a bound that represents the maximum rate allowed. In other words, we require $\beta < \widehat{\beta}$. Based on these ideas, we now present a method for determining a bound on the number N of scenarios needed to test controllers so that it can be guaranteed with a level of confidence $\widehat{\beta}$ that the chosen controller works properly according to the design parameters; that is, $\beta(\varepsilon,\rho,N) \leq \widehat{\beta}$.

7.2.4 Testing a single controller

Some confidence bounds are derived now to ensure that a given controller fulfils the design specifications for a given number of scenarios N. This confidence bounds can then be used directly to derive a condition on the number of scenarios needed for a specific controller to satisfy those bounds.

Theorem 1: The minimum number N of representative scenarios needed to guarantee that the controller fulfils the design specifications described by the function g, with a probability greater than $1-\varepsilon$, and a confidence level $\widehat{\beta}$ can be calculated solving the following equation:

$$\sum_{k=0}^{\lfloor \rho\cdot N \rfloor} \left(\begin{array}{c} N \\ k \end{array} \right) \varepsilon^k (1-\varepsilon)^{N-k} \leq \widehat{\beta}$$

where $\lfloor N\cdot\rho \rfloor$ is $N\cdot\rho$ floor rounded, and ρ the level parameter.

Proof: As a single controller is tested then $\Theta = \widehat{\theta}$. Without loss of generality, it can be temporarily assumed that

$$E(\widehat{\theta}) = Pr\{w \in W : g(\widehat{\theta},w) = 1\} = \widehat{\varepsilon}, \widehat{\varepsilon} \geq \varepsilon$$

that is, the chosen controller does not fulfil the required specifications for all the scenarios $w \in W$. With this assumption, we can calculate the probability that this controller passes the tests. It is obvious that a controller that fails with $\widehat{\varepsilon} = \varepsilon$ has a bigger chance of passing the tests and therefore this case, the most restrictive one, will be used to establish the bound.

When the controller is tested with the N scenarios the goal is to verify (2), which can be transformed into:

$$\sum_{i=1}^{N} g(\widehat{\theta}, w_1) \leq \lfloor \rho \cdot N \rfloor,$$

Note that the last inequality is a sufficient condition to guarantee that (2) holds. The probability for this event can be calculated with the aid of the binomial distribution, which is defined as the sum of the probabilities of all the cases where the number of failed scenarios is lower than or equal to $\lfloor \rho \cdot N \rfloor$.

$$Pr(\sum_{i=1}^{N} g(\widehat{\theta}, w_1) \leq \lfloor \rho \cdot N \rfloor) = \sum_{k=0}^{\lfloor \rho \cdot N \rfloor} \left(\begin{array}{c} N \\ k \end{array} \right) \varepsilon^k (1-\varepsilon)^{N-k}$$

This is the probability that sets the bound of confidence in the methodology, so N has to be chosen such that

$$\sum_{k=0}^{\lfloor \rho \cdot N \rfloor} \left(\begin{array}{c} N \\ k \end{array} \right) \varepsilon^k (1-\varepsilon)^{N-k} \leq \widehat{\beta}$$

which completes the proof.

Thus, given $\widehat{\beta}$, ε and ρ , it is possible to establish a number N of representative scenarios that are enough to guarantee that the controller works properly with a confidence level of $\widehat{\beta}$. There are several ways to find the value of N that validates (4) (see (Álamo et al., 2009) for a more detailed description), but a simple numerical bisection method is enough.

7.2.5 Selecting one controller

In the general case, several controllers will be tested until one is found that satisfies the specifications with the required degree of confidence (that includes the possibility of choosing a set of non-relevant scenarios for a given controller). Therefore, we assume that the set of tested controllers has a finite cardinality $m > 1$.

Corollary 1: The minimum number N of representative scenarios needed to guarantee that a set of controllers $\Theta = \theta_1, \theta_2, \ldots, \theta_m$ fulfils the design specifications described by the function g, with a probability greater than $1-\varepsilon$, and a confidence level $\widehat{\beta}$ is

$$m \cdot \sum_{k=0}^{\lfloor \rho \cdot N \rfloor} \left(\begin{array}{c} N \\ k \end{array} \right) \varepsilon^k (1-\varepsilon)^{N-k} \leq \widehat{\beta}$$

where $\lfloor N \cdot \rho \rfloor$ is $N \cdot \rho$ floor rounded, and ρ the level parameter.

Proof: The proof is direct, by considering Theorem 1 for each of the controllers and considering the worst-case.

Remark 4: Notice that (5) admits other possible applications. For instance, given a set N of possible simulations and controllers to be tested, it provides probability bounds.

Remark 5: The finite cardinality assumption holds for cases such as when there are a random number of samples in the space of design parameters according to a given probability ((Álamo et al., 2009), (Álamo et al., 2010*b*),(Fujisaki and Kozawa, 2006)).

7.3 TCP/IP network problem

This section presents the application of the method developed in section 2 to the congestion control problem. The main objective is to select from a set of possible congestion controllers one that fulfils certain performance specifications. This selection is based on testing the controllers on a number of detailed simulations determined by the proposed technique. To this end, Corollary 1 is used to establish the desired properties for the controllers. As a demonstration, the proposed technique is applied to a problem of two routers connected in a Dumbbell topology, which represents a single bottleneck scenario (see Figure 1). The length of their queues controlled with a PID (Maestre, Álvarez, Salim and Álamo, 2010) whose properties are guaranteed following the results presented in the chapter. The simulations needed for the scenarios were done using the software ns-2 (Vidyasagar, 2001). The main metrics proposed to determine controller performance were: the router queue size (real value and standard deviation); the link utilization; and the probability of packet losses. In this case, Corollary 1 is used to establish the number of required simulations for each controller, so that the desired properties can be statistically guaranteed. As we are interested in finding the best possible controller within a predefined set of controllers, all of them are put to the test and the one that offers the best results (in terms of confidence in fulfilling the specifications) is chosen. In the case that none of the controllers passes the simulation tests, the specifications would have to be relaxed or the control scheme re-evaluated. Of course, with an increment in the number of simulations, the probability bounds could also be improved if needed, at the cost of additional testing time.

7.3.1 Parameters selection

Most of the parameters in the proposed technique are given by statistical considerations. The following set of parameters has been found to give good results in practical applications:

- $\varepsilon = 0.05$: in 5
- $\rho = 0.01$: for each 100 scenarios, only one failure is tolerated that is, the controller under test has not achieved the required performance in one scenario.
- $\widehat{\beta} = 0.1$, this value gives a controller that fulfills the probabilistic constraints imposed with a probability of 90
- $m = 54$ (number of controllers to be tested).

How was m selected? To select m, some preliminary simulations were carried out on a linearized model to check the range of valid parameters for the controller. For this, the working scenario defined in (Wang, Ji and Zhu, 2009) was used: nominal number of TCP sessions $N_{TCP} = 40$ TCP sessions, nominal link capacity C=250 packets/sec., and nominal transportation delay $T_p = 0.3$ sec., with the following expected changes in the conditions of the experiments: the number of TCP sessions can fluctuate between 20 and 180, the link capacity between 100 and 1000 packets/sec and the transportation delay T_p, between 0.1 and 0.6 seconds. The set of PID parameters was restricted to the following ranges: $K_p = -0.0004, -0.0005, \ldots, -0.0009$, $T_i = 1, 1.5, 2$ and $T_d = 2, 2.75, 3$, where the parameters correspond to the standard PID controller:

$$u(t) = K_p[e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{de(t)}{dt}],$$

The control signal is the probability of marking a packet (p) and the error is evaluated from the difference between the desired bottleneck queue length and the real one (q). Thus, 54 different controllers are considered to be tested, which makes possible to fix the last parameter needed for evaluation of Corollary 1.

7.3.2 Results

Evaluating Corollary 1 with the selected parameters, the resulting number of different scenarios needed to obtain a controller that guarantees the specifications was 168 simulations. A simulation is considered to be satisfactory (i.e., g=0) if the queue length q, whose reference will be at 120 packets, has a mean value in the interval [116,124] and its mean deviation is smaller than 17. These are the bounds that define the function g introduced above.

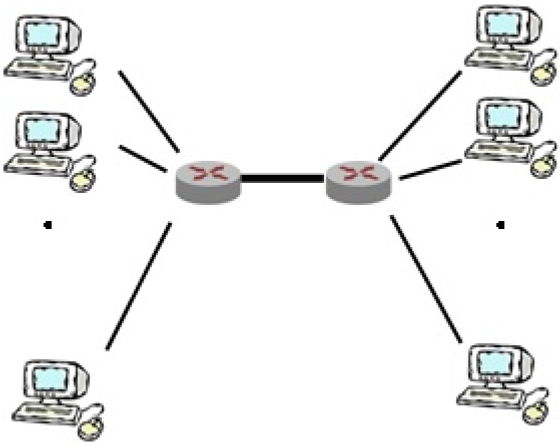


Figure 7.1: Dumbbell topology

Once N has been determined, the procedure consisted in testing, using ns-2 simulations, one controller after another for the N simulations until one is found that fulfills the design specifications. Following (Maestre et al., 2010), the equations used to implement the PID controller in ns-2 are given by

$$p(t_s) = K_p e(t_k),$$

$$d(t_k) = \frac{T_d}{T_d + N_{TCP} \cdot T_s}(d(t_{k-1}) - K_p N_{TCP}(y(t_k) - y(t_{k-1}))),$$

$$u(t_k) = sat(p(t_k) + d(t_k) + i(t_k)),$$

$$i(t_{k+1}) = i(t_k) + \frac{K_p}{T_i} e(t_k),$$

where T_s is the sampling period.

After all the scenarios were tested for the set of selected controllers, it was concluded that the controller with $K_p = -0.0009$, $T_i = 1.5$ and $T_d = 3$ successfully passed the tests: therefore its performance is statistically guaranteed within the range established. Figs.7.2 and 7.3 show, respectively, all the scenarios and the ones that passed the specifications, for the selected controller. To show that the proposed controller really achieves the desired

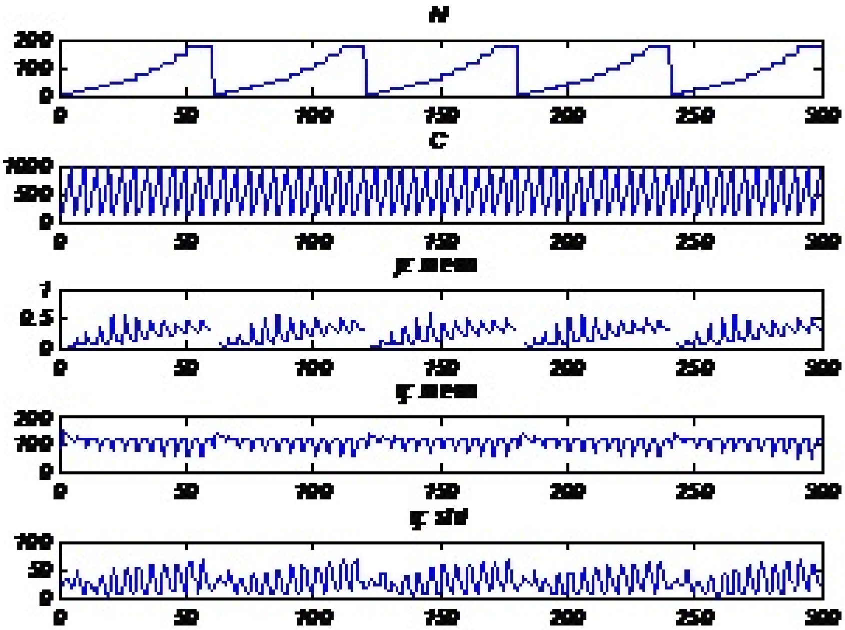


Figure 7.2: All experiments for the selected controller ($K_p = -0.0009, T_i = 1.5, T_d = 3$)

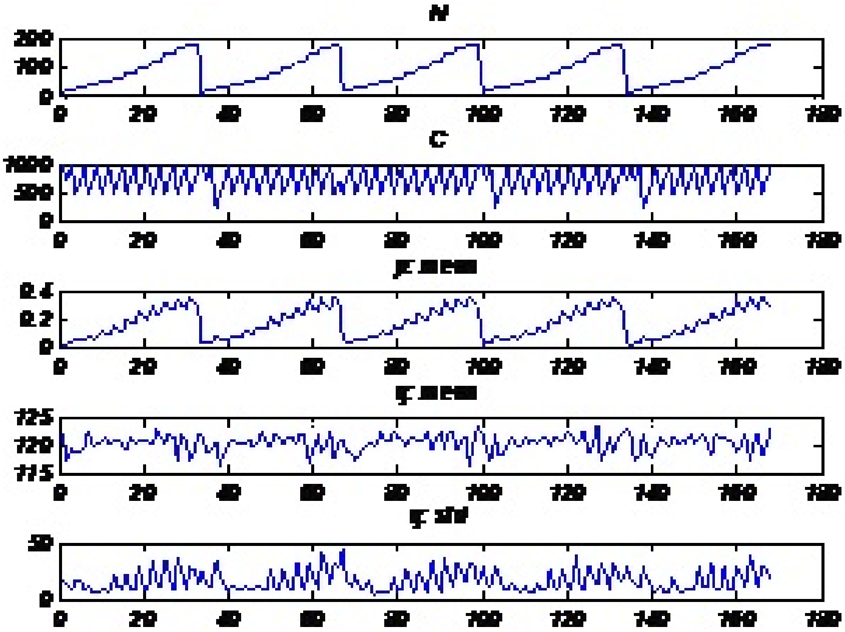


Figure 7.3: Valid experiments for selected controller ($K_p = -0.0009, T_i = 1.5, T_d = 3$)

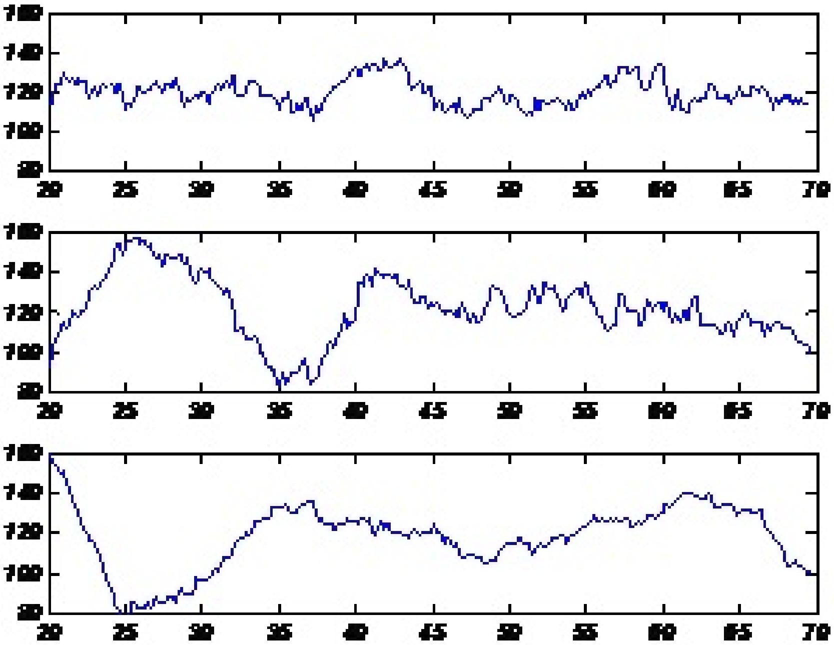
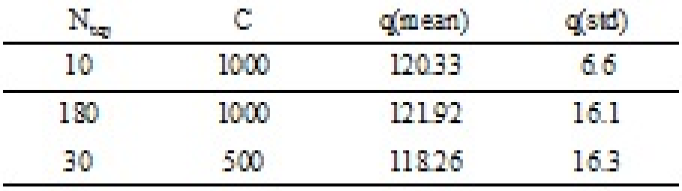


Figure 7.4: Final validation of the selected controller for the scenarios in Table 7.5. Evolution of the queue size in packets



N_{tcp}	C	q(mean)	q(std)
10	1000	120.33	6.6
180	1000	121.92	16.1
30	500	118.26	16.3

Figure 7.5: Data for scenarios in Fig.4

performance, validation was carried out with some additional simulations using parameters that might not be in the testing set. Some simulation results are shown in Fig. 7.4, that correspond to the three different scenarios presented in Table 7.5. It can be seen that for those extreme scenarios the designed controller performs correctly.

7.4 Conclusions

Prompted by a congestion control problem in computer networks, a methodology has been presented in this chapter that guarantees statistical properties of a family of controllers when applied to a certain set of plants. The proposed method does not depend on the family of controllers considered, as it is very flexible. For instance, it can be used to determine probability bounds or establish a minimum number of simulations required to accept or reject a controller. The importance of this result can be seen in the fact that it allows some properties to be guaranteed, with a given probability level, in cases where there is a great difficulty, or impossibility, to demonstrate those properties. This is especially useful in congestion control, as it is common to find design procedures in the literature that are only tested in a few cases with no guarantee that the controller behavior will be similar in other scenarios. The methodology has been tested on a single bottleneck problem: two routers connected in a Dumbbell topology, using TCP/AQM protocol, controlled by a PID selected using the proposed method. The results confirm the applicability of the method proposed to certifying properties of congestion controllers.

Chapter 8

Conclusions and future work

This chapter summarizes the contributions made by this thesis, as well as the main conclusions that can be drawn from it. It also gives insights on some possible directions for future research in the continuation of this work.

8.1 Conclusions

Sample complexity results for various analysis and design problems related to uncertain systems have been derived. In particular we provided new results which guarantee that a binomial distribution expression is smaller than a pre-specified value. These results are subsequently exploited for the analysis of worst case performance and constraint violation. With regard to design problems we considered the case of finite cardinality of controller families and the special case when the design problem can be recast as a robust convex optimization problem.

A general class of randomized algorithms based on probabilistic validation has been presented. We provided a strategy to adjust the cardinality of the validation sets to guarantee that the obtained solutions meet the probabilistic specifications. The proposed strategy is compared with other schemes from the literature and it has been shown that a strict validation strategy in which the design parameter has to satisfy the constraints for all the elements of the validation set might not be appropriate in some situations. We also proved that the proposed approach does not suffer from this limitation because it allows the use of non strict validation tests.

A randomized sequential algorithm that permits approaching optimization problems subject to uncertainty has been introduced. This algorithm is based on a strategy that iteratively adjusts the sample size of the training and validation sets. The main advantage of this proposal is that the algorithm leads to significant improvements in terms of the required sample size. The results allow us to address non-convex optimization problems with uncertainties, which is of great relevance in the context of robust control design.

A new randomized algorithm that addresses the robust feasibility problem under uncertain LMIs is presented. The proposed algorithm has clear differences with randomized gradient and localization methods. The algorithm is guaranteed to obtain a δ-level feasible solution if the problem is ε-feasible. Additionally, if the problem is not ε-feasible the algorithm detects this non feasibility in a finite number of iterations. A bound on the maximal number of iterations required has been obtained. Moreover, the analogies and differences with other existing randomized methods have been discussed. A numerical example that illustrates the merits of the algorithm has been provided.

A general methodology for the design of fault detectors with probabilistic guarantee has been presented. The great advantage of the proposed methodology is, on the one hand its flexibility to introduce different tools of fault detection and on the other its certified probabilistic guarantee of the proposed detector. The operation of this methodology has been illustrated with an application example to a virtual deposit. As future work, following the philosophy of the proposed scheme, it would be interesting to address the problem of designing a diagnostician to determine, once a fault is detected, which type of failure has occurred with a certain probabilistic guarantee.

Prompted by a congestion control problem in computer networks, a methodology has been presented that guarantees statistical properties of a family of controllers when applied to a certain set of plants. The proposed method does not depend on the family of controllers considered, as it is very flexible. For instance, it can be used to determine probability bounds or establish a minimum number of simulations required to accept or reject a controller. The importance of this result can be seen in the fact that it allows some properties to be guaranteed, with a given probability level, in cases where there is a great difficulty, or impossibility, to demonstrate those properties. This is especially useful in congestion control, as it is common to find design procedures in the literature that are only tested in a few cases with no guarantee that the controller behavior will be similar in other scenarios. The methodology has been tested on a single bottleneck problem: two routers connected in a Dumbbell topology, using TCP/AQM protocol, controlled by a PID selected using the proposed method. The results confirm the applicability of the method proposed to certifying properties of congestion controllers.

A tool able to obtain an optimal signal to characterize fuel cells applying randomized algorithms has also been developed. To do it, fuel cells have been stimulated by a randomized input signal and their frequential response has been studied. Variations in magnitude and phase have been translated to a Nyquist diagram.

To build the tool, Matlab program has been used, as well as for the connection between the fuel cell and Labview. Communication between these two programs has been a challenge. The solution requires the exchange of the text files with the data wanted for each of the programs, depending on the sense of the communications. Concluded the design of the tool, some tests have been done in order to confirm its proper performance. This tests have been a success, and many results checks have been done. The tool is able to generate a random signal, interact with Labview to stimulate the battery, make the acquisition and measuring, and compute the frequential response and characteristical parameters. Once obtained a signal useful to characterize a battery, it is no longer necessary to perform long experiments in every frequency. Not only the optimal signal is obtained, but also all the parameters of the equivalent models. Comparing these values with old ones it is possible to diagnose the current state of the battery.

The main advantages for this method that can be mentioned are:

- The saving of time.
- It can be run in any computer, having Matlab, Labview and the acquisition tools
- Any kind of battery can be studied, changing the limits of the experiments to improve performance.
- The visualization of the electrical equivalent model.

8.2 Future work

The possible areas that have been tackled by the thesis and could be considered for further study are listed below:

- Application to CUDA and parallel processing.
- Application to identification (for example, to stock market) (Arahal, Soria and Diaz, 2006). Functionals to be minimized can be associated to an economical criteria. A senior project on this topic has been developed (Molleja, 2013).
- Application to MPC. Model Predictive Control (MPC) can use iterative optimization to obtain a controller for a given system.
- Application to renewable energies (Berenguel, Arahal and Camacho, 1998), as shown in (Luque, 2010).

- Improvements in the identification tool. A data base could be added. The available Labview program could be integrated in the main Matlab program.

Appendix A

Resumen en castellano

A.1 Enfoques aleatorios para análisis y diseño de sistemas de control

En los últimos años, la investigación en análisis probabilístico y métodos de diseño para sistemas y control ha progresado significativamente. Áreas específicas en las que se han visto desarrollos convincentes incluyen los sistemas híbridos e inciertos (Tempo et al., 2005), (Vidyasagar, 1997). Un ingrediente técnico clave de este enfoque es el uso de la teoría de eventos raros y desigualdades de grandes desviaciones que son susceptibles de acotar la cola de la distribución de probabilidad. Éstas desigualdades son cruciales en el área de Statistical Learning Theory (Vapnik, 1998), (Vidyasagar, 1997). El uso de esta teoría para el diseño realimentado sistemas inciertos fue iniciado por (Vidyasagar, 1997). Recientemente, se han proporcionado mejoras significativas en relación a la complejidad muestral en (Álamo et al., 2009). Para el caso especial de problemas de optimización convexa, el enfoque del escenario ha sido introducido por (Calafiore and Campi, 2006) para el diseño de controladores probabilísticos.

La utilidad de los algoritmos aleatorios se apoya en el hecho de que pueden sortear los problemas de complejidad del diseño no convexo.

En este marco, se pueden extraer N muestras i.i.d. $\{w^{(1)},\ldots,w^{(N)}\}$ de $\mathscr{W}$ de acuerdo con la probabilidad $\Pr_{\mathscr{W}}$ y resolver el problema de complejidad muestral.

Dado q obtener una solución global al problema anterior es difícil en el caso general, analizamos en este trabajo las propiedades probabilísticas de las soluciones factibles subóptimas (si se permiten como máximo m violaciones de las N restricciones). La idea de permitir alguna violación de las restricciones no es nueva y puede encontrarse, por ejemplo, en el contexto de identificación (Bai et al., 2002). Las estrategias aleatorias han sido recientemente estudiadas en (Álamo et al., 2009), ver también (Tempo et al., 2005; Vidyasagar, 1997).

A.2 Esquemas de validación aleatoria

El diseño en presencia de incertidumbre es de máxima relevancia en diferentes campos. Desafortunadamente, los problemas de optimización semi-infinita relacionados a menudo son de tipo NP-duros, lo que compromete seriamente su solución en un tiempo computacional razonable (Blondel and Tsitsiklis, 2000). Existen dos modos de sortear este asunto. Una opción consiste en relajar las resticciones del problema original que normalemnte se resuelven en tiempo polinomial pero que pueden llevar a soluciones demasiado conservadoras (Scherer, 2006). Otro enfoque es asumir que la planta incierta es descrita probabilísticamente de modo que se puede derivar un algoritmo aleatorio para obtener, normalmente en tiempo polinomial, una solución con algunas propiedades dadas, normalmente en términos de la probabilidad de error (Tempo et al., 2005), (Vidyasagar, 1997).

El campo de los algoritmos aleatorios ha evolucionado significativamente en los últimos años. Un estudio reciente de este tema puede encontrarse en (Calafiore et al., 2011). Se han propuesto dos enfoques complementarios, métodos secuenciales y no-secuenciales. El enfoque clásico para los métodos no-secuenciales se basa en la 'statistical learning theory' (Vapnik, 1998). En particular, el uso de esta teoría para el diseño realimentado de sistemas de control para sistemas inciertos ha sido iniciado en (Vidyasagar, 1997); más trabajos en esta dirección incluyen (Koltchinskii et al., 2000), (Vidyasagar, 2001), (Vidyasagar and Blondel, 2001), (Álamo et al., 2009). En (Álamo et al., 2010*a*) y (Luedtke and Ahmed, 2008) se analiza el caso particular en el que el conjunto de parámetros de diseño tiene cardinalidad finita.

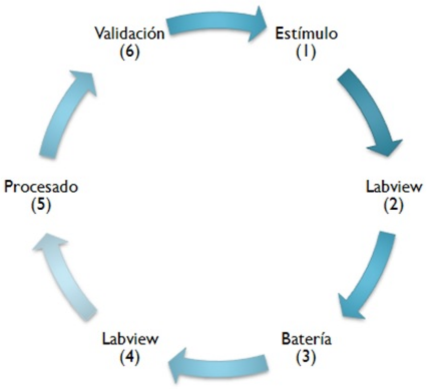
La ventaja de estos métodos es que el problema a abordar puede ser no-convexo. Para los problemas de optimización convexa, se ha introducido un exitoso paradigma no-secuencial, denotado como escenario, en (Calafiore and Campi, 2005) y (Calafiore and Campi, 2006). Ver también (Campi and Garatti, 2008), (Campi and Garatti, 2011), (Calafiore, 2010) y (Álamo et al., 2010*a*) para resultados relacionados.

En los métodos no secuenciales el problema original de control robusto es reformulado en términos de un único problema de optimización con restricciones muestreadas que son generadas aleatoriamente. Un tema relevante de estos enfoques es que no requieren ningún paso de validación. El número de muestras requerido para garantizar que la solución obtenida cumple algunas especificaciones debe tener en cuenta la naturaleza específica del problema en cuestión. El resultado principal de esta línea de investigación es derivar cotas inferiores explícitas para este tamaño muestral requerido. Recientemente, se han proporcionado mejoras en esta complejidad muestral en (Álamo et al., 2009). En cualquier caso, las cotas explícitas obtenidas para la complejidad muestral pueden ser muy conservadoras, porque se basan en un análisis del peor caso posible y crecen (al menos linealmente) con el número de variables de decisión.

Para los métodos secuenciales, los algoritmos iterativos resultantes se basan en el gradiente estocástico (Calafiore and Polyak, 2001), (Polyak and Tempo, 2001), iteraciones elipsoidales (Kanev et al., 2003), (Oishi, 2007), o métodos de plano de corte y centro analítico (Calafiore and Dabbene, 2007), ver también (Álamo, Tempo, Ramírez and Camacho, 2007) para otras clases de algoritmos secuenciales. Las propiedades de convergencia en tiempo finito son de hecho uno de los puntos de interés de estas publicaciones. Varios problemas de control se han resuelto utilizando estos algoritmos secuenciales, incluyendo reguladores LQ robustos, sistemas multi-estado y desigualdades matriciales lineales inciertas (LMIs). Los métodos secuenciales se usan mayormente para problemas inciertos convexos porque el esfuerzo computacional de cada iteración es asumible. De todos modos, pueden ser aplicados en principio a cualquier tipo de problema de diseño robusto. Por ejemplo, un algoritmo secuencial que puede ser aplicado a una generalidad de tipos de problemas se presenta en (Álamo et al., 2009).

El principal punto en común de todos estos algoritmos secuenciales es el uso de una estrategia de validación presentada en (Oishi, 2003) (ver (Oishi, 2007) para una versión de revista). Las soluciones candidatas proporcionadas en cada iteración de estos algoritmos son validadas usando un conjunto de validación que se extrae de acuerdo a una medida probabilística definida en el conjunto incierto. Si la solución candidata satisface las especificaciones de diseño para cada elemento de este conjunto de validación entonces es clasificada como solucion probabilística y el algoritmo termina. El principal punto de este esquema de validación es que la cardinalidad del conjunto de validación se incrementa con cada iteración del algoritmo. La estrategia garantiza que si se obtiene una solucion probabilística, entonces esta cumple algunas especificaciones probabilísticas. Un enfoque similar, introducido en (Dabbene et al., 2010), se presenta en (Calafiore et al., 2011) en el contexto de algoritmos secuenciales. La contribución es una reducción de la cardinalidad requerida para los conjuntos de validación.

La principal contribución de este trabajo es proponer un esquema de validación relajado en el que se permita a las soluciones candidatas violar las especificaciones de diseño para uno o más de los miembros del conjunto de validación. La idea de permitir algunas violaciones de las restricciones no es nueva y puede ser encontrada, por ejemplo, en el contexto de identificacion (Bai et al., 2002), optimización con restricciones (Campi and Garatti, 2011) y statistical learning theory (Álamo et al., 2009). Este esquema tiene sentido en presencia de restricciones suaves o cuando no es posible encontrar una solución que satisfaga las especificaciones para todas las realizaciones admisibles de la incertidumbre.



Este esquema nos permite reducir, en algunos casos drásticamente, el número de iteraciones requeridas por el algoritmo secuencial. Otra ventaja del esquema propuesto es que no se basa en la existencia de una solución robusta determinista. La estrategia presentada es bastante general y no se basa en asumir convexidad.

A.3 Algoritmos aleatorios secuencialmente óptimos para problemas factibles LMI

En los últimos años, los algoritmos aleatorios ((Tempo et al., 2005), (Calafiore and Dabbene, 2007), (Calafiore et al., 2011), (Álamo et al., 2009)) han despertado el interés de los investigadores en control, fundamentalmente por la posibilidad de usarlos para evitar la naturaleza NP-dura de muchos problemas que aparecen en control robusto ((Nemirovskii, 1993)). Estos algoritmos permiten obtener en tiempo polinomial una solución aproximada que satisface la mayor parte de las restricciones[1] asociadas a posibles valores de la incertidumbre de un problema de robustez. Además, el número de restricciones del problema original que son violadas por la solución aproximada se puede hacer tan pequeño como se desee.

Una clase destacada en las estrategias aleatorias son las basadas en métodos gradenciales ((Polyak and Tempo, 2001; Calafiore and Polyak, 2001; Fujisaki et al., 2003; Liberzon and Tempo, 2004)). Estas estrategias son capaces de encontrar, con probabilidad uno, una solución a un problema de control robusto basado en LMIs en un número finito de iteraciones, si el problema es factible. Estos métodos están basados en un esquema iterativo, en el que la solución actual se actualiza en la dirección obtenida del gradiente aleatorio de una función de factibilidad apropiada.

Otro tipo importante de algoritmos aleatorios son los basados en versiones probabilísticas de los métodos de localización. La convergencia a la solución en estos métodos es, teóricamente, mejor que los gradenciales. Entre estos métodos cabe citar el algoritmo probabilístico del elipsoide ((Oishi, 2003; Kanev et al., 2003)) y la versión probabilística del método del plano de corte por el centro analítico de un conjunto de LMIs ((Calafiore and Dabbene, 2006)).

La estrategia del escenario juega también un papel importante en la resolución de problemas de optimización robusta. Tal y como se demuestra en (Calafiore and Campi, 2006), muestreando de manera apropiada el conjunto de restricciones, se obtiene un problema convexo (el escenario) cuya solución es aproximadamente factible para el problema original (el cual implica un número mucho mayor, posiblemente infinito, de restricciones). Al incrementar el número de muestras, el número de restricciones del problema original que se violan tiende a cero.

La estrategia de escenario, los métodos gradenciales y el método del elipsoide tienen una naturaleza muy diferente. La primera obtiene una solución aproximada mediante la resolución de un único problema convexo con un número alto de restricciones. Por otra parte los métodos gradenciales y el método del elipsoide obtienen una solución aproximada de manera secuencial, empleando un número considerable de iteraciones, en cada una de

[1] Es decir la solución encontrada gozaría de una factibilidad aproximada según la definición presentada en (Barmish, 1994).

las cuales se actualiza una solución candidata mediante una sencilla regla sin optimización alguna.

En este trabajo se presenta un algoritmo aleatorio que resuelve el problema de obtener una solución robusta factible para un conjunto posiblemente infinito de LMIs. Este algoritmo no pertenece a ninguna de las clases anteriores y converge en un número finito de iteraciones, siendo además capaz de determinar la no factibilidad del problema. La solución se alcanza mediante una secuencia de problemas de optimización relativamente simples. El método propuesto comparte con los métodos gradenciales y de elipsoide su naturaleza secuencial. A diferencia de éstos utiliza una regla de actualización de la solución candidata basada en un problema de optimización en el que se emplea un reducido número de restricciones obtenidas del problema original. Una de las ventajas del método propuesto es que es capaz de determinar la no factibilidad del problema de factibilidad robusta. En la práctica el algoritmo funciona de manera muy satisfactoria, obteniendo una solución aproximadamente factible o detectando la no factibilidad en un número razonable de iteraciones.

A.4 Aplicaciones de los algoritmos aleatorios

Se han desarrollado metodos de diseño y algoritmos aleatorios para varias aplicaciones relacionadas con sistemas y control (Tempo et al., 2013).

A.4.1 Detección de fallos

Se entiende como fallo todo cambio en el comportamiento de alguno de los componentes del sistema (desviación no permitida de alguna de sus propiedades o parámetros característicos) de manera que éste ya no puede satisfacer la función para la cual ha sido diseñado (Blanke, 1999). Además de los fallos, existen otros factores que alteran el comportamiento normal del sistema, como las perturbaciones y el ruido. Las perturbaciones son entradas no conocidas que pueden manifestarse en el sistema en cualquier momento pero que se han tenido en cuenta a la hora de diseñar el lazo de control convencional. Cualquier perturbación que no se haya tenido en cuenta en este diseño será considerada como un fallo. El ruido también es una entrada no conocida que se manifiesta en el sistema pero, a diferencia de las perturbaciones, tiene media nula y, además, a priori se puede tener conocimiento de cual es su amplitud. Un sistema de detección de fallos ha de reaccionar frente a los fallos y ser inmune (robusto), en la medida de lo posible, a los otros factores presentes en el sistema que generan incertidumbre. Por otro lado, muchos de los métodos de detección de fallos se basan en un modelo (matemático o cuantitativo) del sistema a monitorizar que nunca podrá describir de manera exacta el comportamiento del sistema real y por lo tanto presentará un error de modelado que también se deberá tener en cuenta.

El objetivo de un bloque de detección de fallos es, una vez se ha producido un fallo en un instante T_F, detectarlo en un intervalo de tiempo menor o igual a $T_{D_{max}}$ fijado previamente. Dependiendo de la magnitud e incidencia de los fallos que se deseen detectar y de la presencia de otros factores de incertidumbre en el sistema, no siempre será posible diseñar un bloque de detección que detecte todos los fallos sin que en situaciones de no fallo se activen falsas alarmas. Así que siempre existirá un compromiso entre la proporción de fallos que no se detecten (*MF* ”Missed Faults”) y la proporción de veces que se active el bloque detector sin la presencia de fallos debido a los factores de incertidumbre presentes en el sistema (*FA* ”False alarms”). En este compromiso que se deberá tener en cuenta en el proceso de diseño del bloque detector de fallos es l’ogico priorizar la minimización de fallos no detectados respecto a la minimización de falsas alarmas.

La naturaleza aleatoria de los fallos y las incertidumbres inherentes del sistema convierten el problema de diseño del bloque de detección en un problema de robustez.

Tipicamente, para un problema de robustez, los parámetros de diseño, asi como diferentes variables auxiliares, son descritos en terminos de un vector de variables de decision θ, que se denota como parametro de diseño, y es restringido al conjunto Θ. Por otro lado, la incertidumbre w está acotada en el conjunto $\mathscr{W}$. Es decir, cada elemento $w \in \mathscr{W}$ representa una de las realizaciones admisibles de la incertidumbre, con probabilidad $\Pr_{\mathscr{W}}$. En nuestro contexto de detección de fallos, θ corresponde a las variables de decisión que determinan el bloque de detección de fallos. Dicho bloque permite determinar si hay un fallo o no en un determinado escenario, por lo tanto tendremos dos conjuntos de incertidumbre $\mathscr{W}_F$ y $\mathscr{W}_N$ que consisten en todos los posibles escenarios de funcionamiento del sistema a monitorizar con fallo y sin fallo respectivamente. Por otro lado, w_F y w_N representan una realización de un escenario con fallo y sin fallo. $\mathscr{W}_F$ y $\mathscr{W}_N$ tienen asociados unos espacios de probabilidad $\Pr_F$ y $\Pr_N$ respectivamente.

Además consideramos también dos funciones binarias medibles:

$$g(\theta,w) := \begin{cases} 0 & \text{si } \theta \text{ detecta fallo} \\ 1 & \text{en otro caso.} \end{cases}$$

$$h(\theta,w) := \begin{cases} 0 & \text{si } \theta \text{ no detecta fallo} \\ 1 & \text{en otro caso.} \end{cases}$$

Al aplicar estas dos funciones sobre los espacios $\mathscr{W}_F$ y $\mathscr{W}_N$ se obtienen las siguientes esperanzas

$$E_g(\theta) := \Pr_F\{w_F \in \mathscr{W}_F : g(\theta,w_F) = 1\}$$

$$E_h(\theta) := \mathrm{Pr}_N\{w_N \in \mathscr{W}_N : h(\theta, w_N) = 1\}.$$

Donde $E_g(\theta)$ y $E_h(\theta)$ son el tanto por uno de fallos no detectados (MF) y falsas alarmas (FA) respectivamente. La utilidad de los algoritmos aleatorios surge del hecho de poder tratar el siguiente problema de diseño

$$\min_{\theta \in \Theta} E_h(\theta) \ \text{ sujeto a } E_g(\theta) \leq \eta_F \tag{A.1}$$

donde η_F es el tanto por uno máximo de fallos no detectados impuesto como requerimiento del bloque detector.

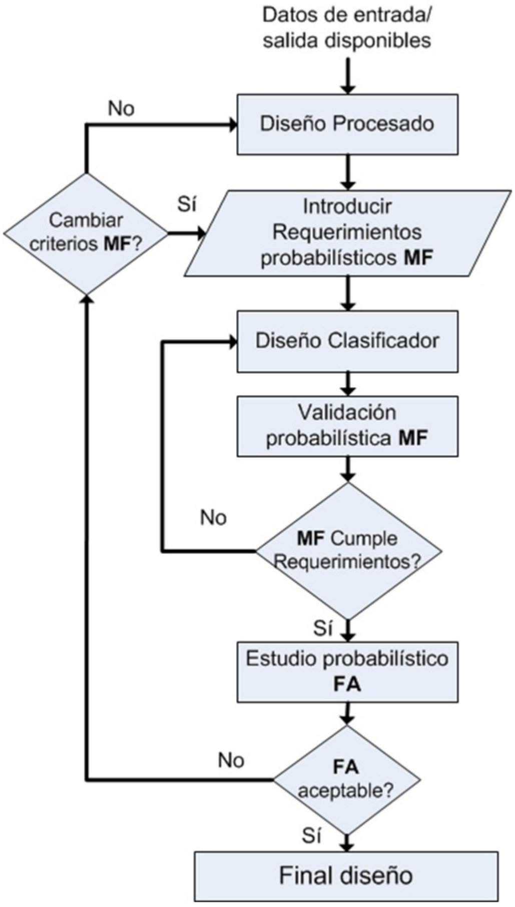
En este marco, se pueden extraer N_N y N_F i.i.d. muestras (independientes e identicamente distribuidas) $\{w_N^{(1)}, \ldots, w_N^{(N_N)}\}$ de $\mathscr{W}_N$ y $\{w_F^{(1)}, \ldots, w_F^{(N_F)}\}$ de $\mathscr{W}_F$ de acuerdo a la probabilidad Pr_N y Pr_F respectivamente y con una proporción entre escenarios de fallo y no fallo $F_N = \frac{N_F}{N_N}$ determinada por la probabilidad de fallo del sistema a monitorizar. De esta manera se puede resolver el siguiente problema de optimización muestreado

$$\min_{\theta \in \Theta} \sum_{\ell_N = 1}^{N_N} h(\theta, w_N^{(\ell_N)}) \tag{A.2}$$

$$\text{sujeto a } \sum_{\ell_F = 1}^{N_F} g(\theta, w_F^{(\ell_F)}) \leq \eta_F N_F$$

La idea de permitir algunas violaciones de las restricciones no es nueva y puede encontrarse, por ejemplo, en el contexto de identificación (Bai et al., 2002).

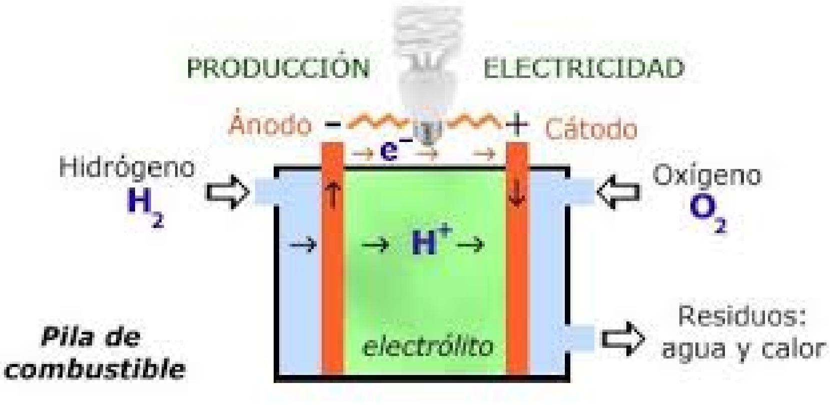
En este artículo se propone un método de diseño del bloque detector de fallos basado en la utilización de históricos o simulaciones de episodios reales con fallo y sin fallo evitando la dificultad del análisis, que no siempre es posible, debido a la complejidad del problema.



El resultado así obtenido, mediante un test de validacion probabilística, garantiza que la solución propuesta se comporta de la manera deseada con una cierta probabilidad, fijada a priori. Se garantiza asimismo la satisfacción probabilística de las restricciones. Esta técnica resulta muy adecuada para el abordaje de problemas complejos.

A.4.2 Identificación de una pila de combustible

El objetivo principal es la identificación de una pila de combustible mediante la aplicación de algoritmos aleatorios. Se pretende desarrollar una herramienta de ensayos en Matlab con la que evaluar baterías en el laboratorio y obtener una señal de estímulo óptima para el modelado de estas, con el fin de caracterizarlas en un futuro de la manera más rápida y precisa posible. Para ello se apoya en el algoritmo presentado en (Álamo et al., 2010*b*). Las baterías son dispositivos muy complejos que tienen numerosos componentes, y su rendimiento depende de variables que se pueden medir de forma sencilla, como el voltaje y la temperatura, y otras que no son tan sencillas de medir, como la edad, las tolerancias de fabricación y las variaciones entre celdas dentro de la batería, que pueden tener un gran impacto en el rendimiento final.

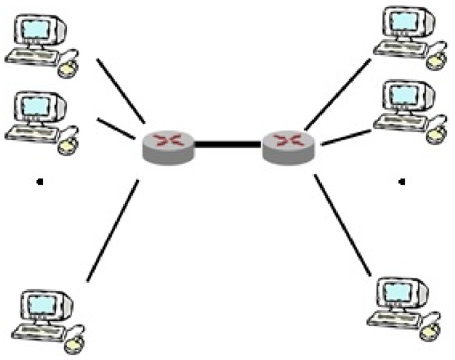


Por todo esto siempre se debe realizar pruebas para un mantenimiento adecuado, inspeccionando su estado hasta que sea necesario reemplazarla. Unas pruebas importantes para conocer el estado en el que se encuentra la batería son aquellas que permiten conocer la capacidad e impedancias de su modelo eléctrico. El planteamiento de este trabajo es de utilidad en este aspecto, ya que se ocupa de encontrar la señal con la que se pueda conseguir la mejor aproximación del modelo eléctrico de la batería. Durante la ejecución de los experimentos se excitarán las baterías con señales constantes definidas a trozos elegidas de forma aleatorias. Los algoritmos aleatorios, presentan, en muchas ocasiones, soluciones más veloces que los algoritmos tradicionales y en otras, soluciones que no serían posibles dentro del dominio de los algoritmos tradicionales. En la práctica común, los algoritmos aleatorios son una aproximación con un generador de números pseudo-aleatorios.

A.4.3 Pruebas de controladores realimentados

En las redes informaticas la congestión aparece cuando hay demasiadas fuentes enviando datos demasiado rápido como para que la red los maneje. Las técnicas para reducir la congestión son de gran interés. Este trabajo se concentra en las metodologías donde las técnicas de control realimentado proporcionan soluciones eficientes ((Jacobson, 1988), (S. Ryu and Qiao, 2004), (Hollot et al., 2002), (Sun et al., 2007), (Floyd and Jacobson, 1993)).

Un problema central al diseñar controladores para estos sistemas es la dificultad de asegurar desempeño adecuado en todas las condiciones posibles, porque estos sistemas operan bajo un rango muy amplio de condiciones, son inherentemente no lineales y sufren de retrasos que varían muy significativamente. Por ello, los diseñadores a menudo tienen que mostrar la efectividad de su propuesta mediante simulaciones extensivas, lo cual es una metolodogía muy costosa en tiempo, y no ofrece garantía definitiva de desempeño: los resultados de simulación en la mayoría de las referencias muestran solo casos y escenarios específicos.



Enfrentados a este problema, nos concentramos en lo siguiente: dado un grado de confianza requerida, ¿cuántas simulaciones se necesitan para comprobar el desempeño adecuado de los controladores? Así, desarrollamos un enfoque aleatorio, basado en algunas ideas en (Su-Woon et al., 2012),(Álamo et al., 2009), (Álamo et al., 2010*b*), (Fujisaki and Kozawa, 2006), para comprobar si un controlador satisface de manera robusta un conjunto de especificaciones con un margen de error probabilístico.

Los resultados se muestran de modo implícito, es decir, el número de experimentos requerido es obtenido por un simple procedimiento numérico. El marco teórico propuesto en (Álamo et al., 2009) y (Álamo et al., 2010*b*) es convenientemente retocado para esta aplicación. Así este trabajo constituye una prueba de concepto de la metodología propuesta en las referencias mencionadas.

La idea principal es probar el controlador bajo un conjunto finito de diferentes escenarios. Cuando el controlador satisfaga las restricciones para un número suficiente de estos escenarios, entonces ciertas propiedades se pueden concluir con un grado de confianza dado, y no se necesitan más simulaciones. Una de las características principales de esta técnica es que es independiente de la familia de controladores (PI, PID, predictivo, robusto, etc).

Debe señalarse que aunque la metodología propuesta ha sido enfrentada a un problema de control, y ha sido demostrada en este problema, puede ser directamente aplicada a otros problemas de control y pruebas, dado que las plantas a ser controladas son frecuentemente no lineales, tienen incertidumbres y variaciones de parametros ((nez et al., 2011)).

A.5 Esquema de la tesis y contribuciones

Lo que sigue es un esquema de la tesis y sus contribuciones:

- **Capítulo 3**: Cotas explícitas para la complejidad muestral requerida. En este capítulo se estudia la complejidad muestral de métodos probabilísticos para el control de sistemas inciertos. Se aborda también el caso particular en el que el problema de diseño puede ser formulado como un problema incierto de optimización convexa. Se proporcionan cotas simples y explícitas para garantizar que las soluciones obtenidas cumplen algunas especificaciones probabilisticas pre-especificadas.

- **Capítulo 4**: Esquemas de validación aleatoria. Se presenta una estrategia para el diseño bajo incertidumbre. Se proporciona una clase general de algoritmos secuenciales que satisfacen las especificaciones requeridas usando validación probabilística. En cada iteración del algoritmo secuencial se válida una solución candidata, en términos de un conjunto de muestras inciertas generadas aleatoriamente.

- **Capítulo 5**: Se propone un algoritmo aleatorio secuencialmente óptimo para problemas de factibilidad robusta de LMIs. El algoritmo se basa en la solución de una secuencia de problemas de optimización semidefinidos que involucran a un pequeño número de restricciones. Se da una cota para el máximo número de iteraciones requeridas por el algoritmo. Se discuten analogías y diferencias con los métodos del gradiente y de localización. El desempeño y comportamiento del algoritmo son ilustrados en términos de un ejemplo numérico.

- **Capítulo 6**: Detección de fallos con validación probabilística. Presentamos una estrategia general para el diseño de un bloque de detección de fallos con validación probabilística (PCV- Procesado, clasificación, validación). Se propone un esquema general de PCV, que permite diseñar un bloque de detección de fallos con validación probabilística en el porcentaje máximo de fallos no detectados (impuesto como condición de diseño) y en el porcentaje de falsas alarmas (obtenido a posteriori). En cada iteración del algoritmo secuencial, una solución candidata se valida probabilísticamente mediante un conjunto de muestras generadas aleatoriamente. Presentamos un marco general en el que la solución candidata puede violar las restricciones para un reducido número de elementos del conjunto de validación. Este esquema generalizado muestra significativas ventajas, en particular en términos de la obtención de la solución probabilística.

- **Capítulo 7**: Aplicación a la identificación frecuencial. Identificación de una pila de combustible mediante la aplicación de algoritmos aleatorios. Se desarrolla una herramienta de ensayos en Matlab con la que evaluar baterías en el laboratorio y obtener una señal de estímulo óptima para el modelado de éstas, con el fin de caracterizarlas en un futuro de la manera más rápida y precisa posible, ya que se ocupa de encontrar la señal con la que se pueda conseguir la mejor aproximación del modelo eléctrico de la batería. Durante la ejecución de los experimentos se excitarán las baterías con señales constantes definidas a trozos elegidas de forma aleatoria.

- **Capítulo 8**: Un enfoque probabilístico para probar controladores realimentados. Se presenta un enfoque probabilístico para probar si un controlador satiface de forma robusta un conjunto de especificaciones con un cierto margen probabilístico de error. Los resultados se expresan de modo implícito, esto es, el número de experimentos requeridos se obtiene de un simple procedimiento numérico. Este capítulo constituye una prueba de concepto de la metodología propuesta en capítulos anteriores. La idea principal es probar controladores bajo un número finito de posibles escenarios. Cuando el controlador satisface las especificaciones para un número suficiente de estos escenarios, entonces ciertas propiedades pueden concluirse con un nivel de confianza dado, y no se necesitan más simulaciones.

A.6 Publicaciones

Los siguientes artículos se han presentado o han sido enviados para su publicación durante la elaboración de esta tesis:

CAPÍTULOS DE LIBRO:

1. ”On the sample complexity of probabilistic analysis and design methods” T. Alamo, R. Tempo, A. Luque. Perspectives in mathematical system theory, control and signal processing. Lecture notes in control and information series 398. Springer. USA. 2010.

REVISTAS:

1. A probabilistic approach for testing feedback controllers, with application to congestion control. José M. Maestre, Teresa Alvarez, Teodoro Alamo, Anuar Salim and Amalia Luque. Technical Notes .International Journal of Control, Automation, and Systems.Volume 10, Number 4, August 2012.
2. Un algoritmo secuencial, aleatorio y óptimo para problemas de factibilidad robusta. T. Álamo, R. Tempo, D.R. Ramírez, A. Luque, E.F. Camacho. RIAI. 2013.
3. The Sample Complexity of Randomized Methods for Analysis and Design of Uncertain Systems. T. Alamo, R. Tempo b, A. Luque, D.R. Ramírez. Submitted to Automatica.

   Trabajo en curso:

   - Detección de fallos usando validación probabilística. Para ser enviado a Automatica.
   - Validación de controladores. Para ser enviado a una revista.
   - Publicación sobre identificación frecuencial.

CONGRESOS:

1. ”Dynamic model of the relationships between technology and employment.” A. Luque, A. Conseglieri, T. Álamo. Proceedings of the European Control Conference 2009. Budapest-Hungría. 23-26 Agosto 2009. ISBN 978-963-311-369-1
2. ”Algoritmos aleatorios”. A. Luque, T. Álamo, R. Tempo. XXX Jornadas de Automática. Valladolid. 2-4 Septiembre 2009. ISBN 13-978-84-692-2387-1
3. ”Modelado de sistemas híbridos de energias renovables y su aplicación a una planta termosolar de agua caliente sanitaria (A.C.S.)” . A. Luque, A. Quintero, T. Álamo, D. Limón, M. R. Arahal, A. Conseglieri. XXX Jornadas de Automática. Valladolid. 2-4 Septiembre 2009. ISBN 13-978-84-692-2387-1.
4. ”Randomized Algorithms and their application to renewable energy systems”. A. Luque, T. Álamo. IFAC - Conference on control methodologies and technology for energy efficiency. Marzo 2010.
5. ”On the Sample Complexity of Randomized Approaches to the Analysis and Design under Uncertainty”. T. Alamo, R. Tempo, A.Luque. Proceedings of the 2010 American Control Conference (ACC10), ISBN: 978-1-4244-7425-7. Junio 2010.
6. ”Algoritmo para el diseño robusto de sistemas complejos”. A. Luque, T. Alamo. XXXI Jornadas de Automatica. Jaén. 2-4 Septiembre 2010. ISBN: 978-84-693-0715-1
7. ”Modeling of a hybrid renewable/fossil hot water production system”. A. Luque T. Alamo M. R. Arahal, D. Limón. 2010 IEEE International Conference on emerging technologies and factory automation (ETFA’ 2010), ISBN: 978-1-4244-6849-2. Septiembre 2010.
8. ”Hybrid modeling of renewable energy systems and its application to a hot water solar plant”. A.Luque, A. Quintero, T. Álamo, D. Limón, M. R. Arahal, A. Conseglieri, E. F. Camacho. IFAC Conference on control methodologies and technology for energy efficiency. Marzo 2010.
9. T. Alamo, A. Luque, D. R. Ramírez and R. Tempo, ”Randomized Control Design through Probabilistic Validation,” Proc. of the American Control Conference, Montreal, Canada, June 2012.
10. Detección de fallos con validación probabilística. Blesa, Joaquin ; Luque-Sendra, Amalia; Álamo-Cantarero, Teodoro; Dabbene, F. XXXIV Jornadas de Automatica. 2013. Terrassa. Barcelona.

A.7 Conclusiones

Se han derivado resultados para varios problemas de análisis y diseño relacionados con sistemas inciertos. En particular, hemos proporcionado nuevos resultados que garantizan que una expresión de distribución binomial es menos que un valor pre-especificado. Estos resultados son explotados para el análisis del peor caso y la violación de restricciones. En relación a los problemas de diseño consideramos el caso de cardinalidad finita de familias de controladores y el caso especial cuando el problema de diseño puede ser reescrito como un problema de optimización robusta convexo.

Se ha presentado una clase general de algoritmos aleatorios basados en validación probabilística. Proporcionamos una estrategia para ajustar la cardinalidad del conjunto de validación para garantizar que las soluciones obtenidas cumplen las especificaciones probabilísticas. La estrategia propuesta es comparada con otros esquemas de la literatura y se muestra que una estrategia de validación estricta, en la que el parametro de diseño tenga que satisfacer las restricciones para todos los elementos del conjunto de validación, puede no ser apropiada en algunas situaciones. También probamos que el enfoque propuesto no sufre de esta limitación porque permite el uso de pruebas de validación no estrictas.

Se propone un algoritmo secuencial aleatorio que permite abordar los problemas de optimización sujetos a incertidumbres. Este algoritmo está basado en una estrategia que iterativamente ajusta el tamaño muestral de los conjuntos de entrenamiento y validación. La principal ventaja de esta propuesta es que el algoritmo lleva a mejoras significativas en términos del tamaño muestral requerido. El resultado permite abordar problemas de optimización no convexos con incertidumbres, que es de gran relevancia en el contexto de diseño de control robusto.

Se ha presentado un algoritmo aleatorio que aborda en problema de factibilidad robusta bajo LMIs con incertidumbres. El algoritmo propuesto tiene diferencias claras con los métodos del gradiente aleatorio y de localización. El algoritmo garatiza obtener una solucion factible de nivel δ si el problema es ε-factible. Adicionalmente, si el problema no es ε-factible el algoritmo detecta esta no-factibilidad en un número finito de iteraciones. Se obtiene una cota para el número máximo de iteraciones requeridas. Se discuten ademas las analogías y diferencias con otros métodos aleatorios existentes. Se proporciona un ejemplo numérico que ilustra el mérito del algortimo propuesto.

Se presenta una metodología para el diseño de detectores de fallo con garantía probabilística. La gran ventaja de la metodología propuesta es, por un lado, su flexibilidad para introducir diferentes herramientas de detección de fallos y otra su garantía probabilística certificada del detector propuesto. La operación de esta metodología se ha ilustrado con un ejemplo de aplicación para un depósito virtual. Como trabajo futuro, siguiendo la metolodogía del esquema propuesto, podría ser interesante abordar el problema de diseño que determine, una vez que se detecta un fallo, qué tipo de fallo ha ocurrido con una cierta garantía probabilística.

Enfrentados a un problema de congestión y control en redes de ordenadores, se presenta una metodología que garantiza propiedades estadísticas de una familia de controladores cuando se aplica a un cierto conjunto de plantas. El método propuesto no depende de la familia de controladores considerada, así que es muy flexible. Por ejemplo, puede ser usada para determinar las cotas probabilísticas o para establecer un número mínimo de simulaciones requeridas para aceptar o rechazar un controlador. La importacia de este resultado se puede ver en el hecho de que permite garantizar algunas propiedades, con un cierto nivel de probabilidad, en casos en los que hay un alto nivel de dificultad, o imposibilidad, de demostrar esas propiedades. Esto es especialmente útil en control de congestión, porque es común encontrar procedimientos de diseño en la literatura que sólo son probados en unos cuántos casos sin garantía de que el comportamiento del controlador sea similar en otros escenarios. La metodología ha sido probada en un problema de cuello de botella: dos routers conectados en una topología Dumbbell, usando un protocolo TCP/AQM, controlados por un PID seleccionado usando el método propuesto. Los resultados confirman la aplicabilidad del método propuesto para certificar propiedades de los controladores de congestión.

Las posibles áreas que han sido abordadas en la tesis y que se pueden considerar para trabajo futuro son:

- Aplicacion a CUDA y procesamiento paralelo.
- Aplicaciones de identificación (por ejemplo, a la bolsa de valores). Los funcionales a minimizar pueden ser asociados a un criterio económico. Un proyecto fin de carrera sobre esto puede ser encontrado en (Molleja, 2013).
- Aplicación a MPC. Model Predictive Control (MPC) puede usar la optimización iterativa para obtener un controlador para un determinado sistema.
- Aplicación a las energías renovables, como se muestra en (Luque, 2010).
- Mejoras en la herramienta de identificación. Puede añadirse una base de datos. El programa de Labview disponible puede ser integrado en el programa principal de Matlab.

Bibliography

Aho, A., Hopcrof, J. and Ullman, J. (1974), *The design and analysis of computer algorithms*, Addison-Wesley, Reading.

Álamo, T., Fiacchini, M., Cepeda, A., Limón, D. and Camacho, E. (2007), *On the Computation of Robust Control Invariant Sets for Piecewise Affine Systems*, Springer-Verlag, NewYork, chapter Assessment and Future Directions of Nonlinear Model Predictive Control (Lncis), pp. 131–140.

Álamo, T., Luque, A., Ramírez, D. R. and Tempo, R. (2012), Randomized control design through probabilistic validation, *in* 'Proceedings of the American Control Conference', Montreal, Canada.

Álamo, T., Tempo, R. and Camacho, E. (2009), 'Randomized strategies for probabilistic solutions of uncertain feasibility and optimization problems', *IEEE Transactions on Automatic Control* **54**(11), 2545–2559.

Álamo, T., Tempo, R. and Luque, A. (2010*a*), On the sample complexity of randomized approaches to the analysis and design under uncertainty, *in* 'Proceedings of the American Control Conference', Baltimore, USA.

Álamo, T., Tempo, R. and Luque, A. (2010*b*), *Perspectives in Mathematical System Theory, Control, and Signal Processing*, Springer-Verlag, London, chapter On the sample complexity of probabilistic analysis and design methods, pp. 39–50.

Álamo, T., Tempo, R., Ramírez, D. and Camacho, E. (2006*a*), A new vertex result for robustness problems with interval matrix uncertainty, *in* 'Submitted to the ECC-07 Conference. Downloadable from http://telefonica.net/web2/teodoroalamo'.

Álamo, T., Tempo, R., Ramírez, D. and Camacho, E. (2006*b*), A sequentially optimal randomized algorithm for robust LMI feasibility problems, *in* 'Downloadable from http://telefonica.net/web2/teodoroalamo'.

Álamo, T., Tempo, R., Ramírez, D., Luque, A. and Camacho, E. (2013), 'Un algoritmo secuencial, aleatorio y óptimo para problemas de factibilidad robusta', *RIAI* .

Álamo, T., Tempo, R., Ramírez, D. R. and Camacho, E. F. (2007), A sequentially optimal randomized algorithm for robust lmi feasibility problems, *in* 'Proceedings of the European Control Conference', Kos, Greece.

Álvarez, T. and Martínez, D. (2013), An efficient maximization algorithm with implications in min-max predictive control, *in* 'Chapter in IAENG Transactions on Engineering Technologies (Special Volume of the World Congress on Engineering 2012)'.

Arahal, M. R., Soria, M. B. and Diaz, F. R. (2006), *Técnicas de Predicción con Aplicaciones en Ingeniería*, Servicio de Publicaciones de la Universidad de Sevilla, Sevilla, Spain.

Aström, K. J. and Hägglund, T. (2006), 'Advanced pid control', *Aström Instrument Society of America, Research Triangle Park, NC* .

Athans, M. (1971), 'Editorial on the lqg problem', *IEEE Transactions on Automatic Control* **6**, 528.

Bai, E., Cho, H., Tempo, R. and Ye, Y. (2002), 'Optimization with few violated constraints for linear bounded error parameter estimation', *IEEE Transactions on Automatic Control* **47**(4), 1067–1077.

Barmish, B. (1994), *New Tools for Robustness of Linear Systems*, MacMillan Publishing Company, New York, USA.

Barmish, B. and Scherbakov, P. (2002), 'On avoiding vertexization of robustness problems: The approximate feasibility concept', *IEEE Transactions on Automatic Control* **47**(5), 819–824.

Ben-Tal, A. and Nemirovski, A. (2001), *Lectures on Modern Convex Optimization. Analysis, Algorithms, and Engineering Applications*, MPS/SIAM Series on Optimization, Philadelphia, PA.

Berenguel, M., Arahal, M. R. and Camacho, E. (1998), 'Modelling the free response of a solar plant for predictive control', *Control engineering practice* **6**(10), 1257–1266.

Bhattacharyya, S., Chapellat, H. and Keel, L. (1995), *Robust control: the parametric approach*, Prentice Hall, Upper Saddle River.

Blanke, M. (1999), *Fault-tolerant control systems*, New Trends in Advanced Control. Springer-Verlag.

Blesa, J., Luque, A., Álamo, T. and Dabbene, F. (2013), Detección de fallos con validación probabilística, *in* 'Proceedings of the XXXIV Jornadas de Automática', Terrassa. Barcelona.

Blondel, V. and Tsitsiklis, J. N. (2000), 'A survey of computational complexity results in system and control', *Automatica* **36**, 1249–1274.

Boyd, S., Ghaoui, L. E., Feron, E. and Balakrishnan, V. (1994), *Linear Matrix Inequalities in Systems and Control Theory*, SIAM, Philadelphia, PA.

Bravo, J., Álamo, T. and Camacho, E. (2004), Robust nonlinear mpc based on guaranteed prediction, *in* 'Proceedings of CONTROLO', Faro, Portugal.

Bravo, J., Álamo, T., Redondo, M. and Camacho, E. (2008), 'An algorithm for bounded-error identification of nonlinear systems based on dc functions', *Automatica* **44**, 437–444.

Bravo, J., Limón, D., Álamo, T. and Camacho, E. (2005), *Aplicación del Análisis Intervalar al Control Predictivo Basado en Modelo. El Análisis de Intervalos en España: Desarrollos, Herramientas y Aplicaciones*, Documenta Universitaria, Gerona, Spain.

Calafiore, G. (2010), 'Random convex programs', *SIAM Journal of Optimization* **20**, 3427–3464.

Calafiore, G. and Campi, M. (2005), 'Uncertain convex programs: randomized solutions and confidence levels', *Math. Program.* **102**, 25–46.

Calafiore, G. and Campi, M. (2006), 'The scenario approach to robust control design', *IEEE Transactions on Automatic Control* **51**(5), 742–753.

Calafiore, G. and Dabbene, F. (2006), 'A probabilistic analytic center cutting plane method for feasibility of uncertain LMIs', *Technical Report* pp. 1–28.

Calafiore, G. and Dabbene, F. (2007), 'A probabilistic analytic center cutting plane method for feasibility of uncertain lmis', *Automatica* **43**, 2022–2033.

Calafiore, G., Dabbene, F. and Tempo, R. (2011), 'Research on probabilistic methods for control system design', *Automatica* **47**, 1279–1293.

Calafiore, G. and Polyak, B. (2001), 'Stochastic algorithms for exact and approximate feasibility of robust lmis', *IEEE Transactions on Automatic Control* **46**(11), 1755–1759.

Camacho, E., Álamo, T., Limón, D., Bravo, J., Ramírez, D., noz, D. M. and Ruiz Arahal, M. (2005), Robust nonlinear mpc for processes with bounded uncertainties, *in* 'NMPC', Freudenstadt-Lauterbad, Germany.

Campi, M. C. and Garatti, S. (2011), 'A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality', *Journal of Optimization Theory and Applications* **148**, 257–280.

Campi, M. and Garatti, S. (2008), 'The exact feasibility of randomized solutions of robust convex programs', *SIAM Journal of Optimization* **19**, 1211—-1230.

Chernoff, H. (1952), 'A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations', *Annals of Mathematical Statistics* **23**, 493–507.

Copeland and Jack, B. (2008), ‘The church-turing thesis’, *The Stanford Encyclopedia of Philosophy* .

Dabbene, F., Shcherbakov, P. S. and Polyak, B. T. (2010), ‘A randomized cutting plane method with probabilistic geometric convergence’, *SIAM Journal of Optimization* **20**, 3185–3207.

Daleh, M. and Díaz Bobillo, I. (1995), *Control of linear systems: a linear programming approach*, Prentice Hall, Englewood Cliffs.

Doyle, J. (1978), ‘Guaranteed margins for lqg regulators’, *IEEE Transactions on Automatic Control* (23), 756–757.

Floyd, S. and Jacobson, V. (1993), ‘Random early detection gateways for congestion avoidance’, *IEEE/ACM Transactions on Networking* **1**, 397–413.

Fujisaki, Y., Dabbene, F. and Tempo, R. (2003), ‘Probabilistic design of LPV control systems’, *Automatica* **39**, 1323–1337.

Fujisaki, Y. and Kozawa, Y. (2006), *Probabilistic robust controller design: probable near minimax value and randomized algorithms*, In G. Calafiore and F. Dabbene, editors, Probabilistic and Randomized Methods for Design under Uncertainty, Springer, London.

Hansen, L. and Sargent, T. (2008), *Robustness*, Princeton University Press, Princeton.

Hollot, C., Misra, V., Towsley, D. and Gong, W. (2002), ‘Analysis and design of controllers for aqm routers supporting tcp flows’, *IEEE Transactions on Automatic Control* **47**, 945–959.

Horisberger, H. and Belanger, P. (1976), ‘Regulators for linear, time invariant plants with uncertain parameters’, *IEEE Transactions on Automatic Control* **21**(5), 705–708.

Horowitz, I. (1991), ‘Survey of quantitative feedback theory (qft)’, *Int J Control* **53**, 255–291.

Houpis, C. and Rasmussen, S. (1999), *Quantitative Feedback Theory*, Marcel Dekker, New York.

Isermann, R. (2006), *Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*, Berlin, Germany. Springer.

Isermann, R. (2011), *Fault-Diagnosis Applications*, Springer-Verlag, Berlin, Heidelberg.

Jacobson, V. (1988), Congestion avoidance and control, *in* ‘Proc. of the ACM SIGCOMM’.

Kanev, S., Schutter, B. D. and Verhaegen, M. (2003), ‘An ellipsoid algorithm for probabilistic robust controller design’, *Systems and Control Letters* **49**, 365–375.

Koltchinskii, V., Abdallah, C., Ariola, M., Dorato, P. and Panchenko, D. (2000), 'Improved sample complexity estimates for statistical learning control of uncertain systems', *IEEE Transactions on Automatic Control* **12**(45), 2383–2388.

Le, V. H., Stoica, C., Álamo, T., Camacho, E. and Dumur, D. (October 2013), *Zonotopes: From Guaranteed State-estimation to Control*, Wiley-ISTE.

Liberzon, D. and Tempo, R. (2004), 'Common Lyapunov functions and gradient algorithms', *IEEE Transactions on Automatic Control* **49**(6), 990–994.

Luedtke, J. and Ahmed, S. (2008), 'A sample approximation approach for optimization with probabilistic constraints', *SIAM Journal of Optimization* **2**(19), 674–699.

Luque, A. (2010), Randomized algorithms and their application to renewable energy systems, PhD thesis, Universidad de Sevilla.

Maestre, J., Álvarez, T., Álamo, T., Salim, A. and Luque, A. (2012), 'A probabilistic approach for testing feedback controllers, with application to congestion control', *Technical Notes. International Journal of Control, Automation, and Systems* **10**.

Maestre, J., Álvarez, T., Salim, A. and Álamo, T. (2010), A method for testing aqm controllers with probability guaranteed properties, *in* 'Proceedings of the World Congress on Engineering', pp. 1679–1685.

Mesquine, F., Tadeo, F. and Álvarez, T. (2011), Feedback stabilization of infinite-dimensional linear systems with constraints on control and its rate, *in* 'Proceedings of the European Journal of Control', p. 191 – 193.

Milanese, M., Norton, J., Lahanier, H. P. and Walter, E. (1996), *Bounding approaches to system identification*, Plenum Press.New York, USA, London, UK.

Mitzenmacher, M. and Upfal, E. (2005), *Probability and computing: randomized algorithms and probabilistic analysis*, Cambridge Univertsity Press, Cambridge.

Molleja, E. (2013), Análisis de tendencias en Bolsa mediante algoritmos de clasificación, PhD thesis, Universidad de Sevilla.

Morari, M. and Zafiriou, E. (1989), *Robust Process Control*, Prentice Hall.

Motwani, R. and Raghavan, P. (1995), *Randomized algorithms*, Cambridge Univertsity Press, Cambridge.

Mulmuley, K. (1994), *Computational geometry: an introduction through randomization algorithms*, Prentice Hall, Englewood Cliffs.

Nemirovskii, A. (1993), 'Several NP-hard problems arising in robust stability analysis', *Mathematics of Control, Signal and Systems* **6**(2), 99–105.

nez, V. S., Camarillo, K., Moreno-Valenzuela, J. and Campa, R. (2011), 'A practical pid regulator with bounded torques for robot manipulators', *International Journal of Control, Automation and Systems* **8**, 544–555.

Oishi, Y. (2003), Probabilistic design of a robust state-feedback controller based on parameter-dependent Lyapunov functions, *in* 'Proceedings of the 42nd IEEE Conference on Decision and Control', Maui, Hawaii USA, pp. 1920–1925.

Oishi, Y. (2007), 'Polynomial-time algorithms for probabilistic solutions of parameter-dependent linear matrix inequalities', *Automatica* **43**, 538–545.

Okamoto, M. (1959), 'Some inequalities relating to the partial sum of binomial probabilities', *Annals of the Institute of Statistical Mathematics* **1**(10), 29–35.

Packard, A. and Doyle, J. (1993), 'The complex structured singular value', *Automatica* **29**, 71–109.

Papadimitriou, C. (1994), *Computational Complexity*, Addison-Wesley, Reading.

Paz, A. (1981), 'Non deterministic polynomial optimization problems and their approximations', *Theoretical Computer Science* **15**, 251–277.

Poljak, S. and Rohn, J. (1993), 'Checking robust nonsingularity is NP-hard', *Mathematics of Control, Signal and Systems* **6**(1), 1–9.

Polyak, B. and Tempo, R. (2001), 'Probabilistic robust design with linear quadratic regulators', *Systems and Control Letters* **43**(5), 343–353.

Ponce, S. (2013), Identificación frecuencial de baterías mediante algoritmos aleatorios, PhD thesis, Universidad de Sevilla.

Puig, V. and Blesa, J. (2013), Limnimeter and rain gauge fdi in sewer networks using an interval parity equations based detection approach and an enhanced isolation scheme, *in* 'Control engineering practice', Vol. 21, pp. 146–170.

S. Ryu, C. R. and Qiao, C. (2004), 'Advances in active queue management (aqm) based tcp congestion control', *Telecommunication Systems* **25**, 317–351.

Safonov, M. (2012), Origins of robust control: early history and future speculations, *in* 'Proceedings of the 7th IFAC symposium on robust control design'.

Scherer, C. (2006), 'Lmi relaxations in robust control', *European Journal of Control* **12**, 3–29.

Schweppe, F. (1973), *Uncertain dynamic Systems*, Prentice-Hall, Englewood Cliffs.

Su-Woon, K., Chang-Jin, B., Sin, K. and Ho-Chan, K. (2012), 'Stable controller design of mimo systems in real grassmann space', *International Journal of Control, Automation and Systems* **10**, 213–226.

Sun, J., Chan, S., Ko, K.-T., Chen, G. and Zukerman, M. (2007), 'Instability effects of two-way traffic in a tcp/aqm system', *Computer Communications* **30**, 2172–2179.

Tempo, R., Bai, E. and Dabbene, F. (1997), 'Probabilistic robustness analysis: explicit bounds for the minimum number of samples', *Systems & Control Letters* **30**, 237–242.

Tempo, R., Calafiore, G. and Dabbene, F. (2005), *Randomized Algorithms for Analysis and Control of Uncertain Systems*, Communications and Control Engineering Series. Springer-Verlag, London.

Tempo, R., Calafiore, G. and Dabbene, F. (2013), *Randomized Algorithms for Analysis and Control of Uncertain Systems, with Applications*, second edn, Springer-Verlag, London.

Vapnik, V. (1998), *Statistical Learning Theory*, John Wiley and Sons, New York.

Vidyasagar, M. (1997), *A Theory of Learning and Generalization: with Applications to Neural Networks and Control Systems*, Springer, London.

Vidyasagar, M. (2001), 'Randomized algorithms for robust controller synthesis using statistical learning theory', *Automatica* **37**, 1515–1528.

Vidyasagar, M. and Blondel, V. D. (2001), 'Probabilistic solutions to some np-hard matrix problems', *Automatica* **37**, 1397–1405.

Wang, B., Ji, H. and Zhu, J. (2009), 'Robust control design of a class of nonlinear systems in polynomial lower-triangular form', *International Journal of Control, Automation and Systems* **7**, 41–48.

Zames, G. (1981), 'Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms and approximate inverses', *IEEE Transactions on Automatic Control* **26**, 301–320.

Zhou, K., Doyle, J. and Glover, K. (1996), *Robust and Optimal Control*, Prentice-Hall, Englewood Cliff, NJ.