

Robust Static Output Feedback Design with Deterministic and Probabilistic Certificates

Denis Arzelier, F. Dabbene, S. Formentin, Dimitri Peaucelle, Luca Zaccarian

▶ **To cite this version:**

Denis Arzelier, F. Dabbene, S. Formentin, Dimitri Peaucelle, Luca Zaccarian. Robust Static Output Feedback Design with Deterministic and Probabilistic Certificates. Uncertainty in Networked Systems: In Honor of Roberto Tempo, Springer, pp.121-148, 2018, 10.1007/978-3-030-04630-9_3 . hal-01970884

HAL Id: hal-01970884
https://laas.hal.science/hal-01970884

Submitted on 15 Jan 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Robust static output feedback design with deterministic and probabilistic certificates

D. Arzelier, F. Dabbene, S. Formentin, D. Peaucelle, and L. Zaccarian

Abstract Static output feedback design for linear plants is well known to be a challenging non-convex problem. The presence of plant uncertainty makes this challenge even harder. In this chapter, we propose a new BMI formulation with S-variables which includes an interesting link between state feedback, output injection, state injection and static output feedback gains in a unified framework. Based on this formulation, the robust design problem is suitably addressed by iterative optimization procedures with either deterministic or probabilistic viewpoints exploiting the fact that Lyapunov certificates are separated from the control gain design variables. The deterministic approach is for affine polytopic systems. The probabilistic approach requires no assumption on the uncertain system and is based on the Scenario with Certificates (SwC) method which was recently proposed to address certain static anti-windup design problems. Numerical results illustrate the effectiveness of the approach in both deterministic and stochastic cases.

1 Introduction

Static output feedback (SOF) represents probably the simplest and most intuitive way to design a feedback control law: the plant's output is measured and fed-back to the input, multiplied by a specifically designed static gain. Its straightforward

D. Arzelier and D. Peaucelle
LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France e-mail: {arzelier, peaucelle}@laas.fr
F. Dabbene
CNR-IEIIT, Politecnico di Torino, Italy e-mail: fabrizio.dabbene@ieiit.cnr.it
S. Formentin
Politecnico di Milano, Italy e-mail: simone.formentin@polimi.it
L. Zaccarian
LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France and Industrial Engineering Dept, University of Trento, Italy e-mail: zaccarian@laas.fr

implementation and the fact that the full state vector is usually not accessible, and only a partial information is available via the measured output, render it particularly attractive to control designers and practitioners. Moreover, it is known that different problems related to the design of dynamic controllers, such as e.g. fixed/low-order control, can be recast as a SOF design problem by introducing a suitable re-parameterization [10].

However, it has been known since many years that the SOF implementation simplicity is counteracted by an intrinsic complexity in obtaining strong theoretical results: the problem is extremely difficult, and no systematic constructive numerical solutions exist guaranteeing SOF design, or allowing to determine whether such a feedback does not exist. Even its exact theoretical complexity is not known. Indeed, it is easy to see that the problem is immediately rewritten in terms of a bilinear matrix inequality (BMI), whose solution is known to be NP-hard [16]. The interested reader can refer to the 1997 survey [19], or to the most recent overview [18]. In particular, in [18], the different possible solutions proposed in the literature for tackling the SOF problem are discussed and classified according to the specific approach adopted: i) Methods based on the numerical solution of the ensuing BMI problems: these techniques directly tackle the bilinear problem by making recourse to specific optimization solvers, such as e.g. the PENBMI and PENLAB toolboxes [14, 11]. ii) Methods based on Lyapunov theory: most of these methods present an iterative algorithm in which a set of linear matrix inequalites (LMIs) is iteratively repeated until some termination criteria are met. iii) Non-Lyapunov-based static output feedback control strategies, such as those based on the direct solution of the non-smooth optimization problem of minimizing the spectral abscissa of the closed loop system, (see for instance the free package HIFOO [13], [2] or the MATLAB© macro `HINFSTRUCT`).

Among the previously introduced classes, the Lyapunov-based one presents some very interesting features, as pointed out in [18]. First, these techniques allow keeping a clear insight into the original problems, insight that is usually lost when directly tackling the problem through an optimization-based approach. Secondly, the formulation in many cases immediately extends to *uncertain* problems, i.e. problems in which the plant to be controlled is not perfectly known but instead is affected by uncertainty. This is an important characteristic, which is becoming of fundamental importance in modern control design. Hence, iterative LMI methods allow extending the approach to the solution of *robust static output feedback* (R-SOF) problems. Clearly, the presence of plant uncertainty makes this challenging problem even harder. Hence, if on one side one may expect that the solution of the R-SOF problem will enlarge its practical interest, on the other side one should be aware that R-SOF solutions may not exist in many cases.

In this chapter, we follow the Lyapunov-based approach, and we introduce a novel BMI formulation, based on S-variables. An interesting feature of the proposed formulation is that the involved design variables provide a clear link to the state feedback, output injection, state injection and static output feedback gains. In particular, the formulation captures all these sub-problems in a unified framework. Moreover, the formulation immediately extends to the uncertain case, so that vertex

results available for polytopic-type uncertainty can be directly applied. Furthermore, it is observed that a key feature of the derived framework is that it maintains an explicit distinction between *design variables* (i.e. those variables directly involved in the definition of the controller gain, and the *certificates*, i.e. variables whose existence is necessary to prove the existence of a SOF, but that are not involved in the controller construction. This paves the way to the use of recent results on probabilistic robust design, based on the so-called *scenario-with-certificates* (SwC) approach [12]. This approach represents one of the more recent findings in the area of randomized methods for systems and control [4, 20], emerged in the last decade to synergize with the standard deterministic methods for control of systems with uncertainty. Results in this area are based on a combination of probability and random sampling, and their goal is to provide the research engineer with robustness guarantees which hold only with high probability. The payback is a reduction in the computational complexity of classical control algorithms, and in the conservativeness of standard robust control techniques.

The chapter represents the confluence and combination of two different viewpoints to handle uncertainty in systems: the deterministic/robust approach, in which one is interested in obtaining guaranteed results, that hold for every possible instance of the uncertainty, and the so-called probabilistic approach, which characterizes the uncertain parameters as random variables, and then evaluates the system performance in terms of probabilities. This confluence was made possible by the farsightedness and vision of our colleague and friend Roberto Tempo (1956-2017), recently suddenly passed away. Roberto was a strong believer in collaboration and cross-fertilization of research. He always insisted that the two approaches should not be viewed as alternative but rather complementary to each other: one adds to the other.

To describe the philosophy underlying the present work, we use Roberto's words, taken from the proposal of one of the first formal collaborations between our two groups:[1] *"Robustness can be tackled by two means. One, probabilistic, consists in testing a finite number of configurations among the infinitely many admissible ones. This approach is said to be optimistic in the sense that if a level of performance is valid for all tested cases, it may not hold for the actual ones. The second approach, deterministic, provides, using mathematical tools, a guaranteed level of performances for all configurations. It is unfortunately conservative (or pessimistic) in the sense that the guaranteed performance is usually worse than the worst case. The project aims at comparing and hence enriching the optimistic and pessimistic approaches."*

We organize the presentation in three main sections. Section 2 presents the unified S-variable formulation used throughout the chapter. Section 3 illustrates deterministic results and a corresponding heuristic design procedure. Section 4 presents parallel results providing probabilistic guarantees. Finally, we illustrate the effectiveness of the proposed constructions on numerical examples in Section 5.

[1] Bilateral Project "Convex optimization and randomized algorithms for robust control" (CORARC), between IEIIT CNR and LAAS CNRS (2012).

Notation: I stands for the identity matrix. A^T is the transpose of the matrix A. $\{A\}^{\mathscr{S}}$ stands for the symmetric matrix $\{A\}^{\mathscr{S}} = A + A^T$. For a matrix $A \in \mathbb{R}^{n\times m}$ of rank r, $A^{\perp} \in \mathbb{R}^{m\times(m-r)}$ is the matrix of maximal rank such that $AA^{\perp} = 0$, and $A^{\circ} \in \mathbb{R}^{r\times n}$ stands for the full rank matrix such that $A^{\circ}A$ is full rank. $A \succ B$ is the matrix inequality stating that $A - B$ is symmetric positive definite. The terminology *"congruence operation of A on B"* is used to denote $A^T BA$. If A is full rank, and $B \succ 0$, the congruence operation of A on B gives a positive definite matrix: $A^T BA \succ 0$. A matrix inequality of the type $N(X) \succ 0$ is said to be a linear matrix inequality (LMI for short), if $N(X)$ is affine in the decision variables X. In the following, decision variables are highlighted[2] using the blue color. $\Phi_{\bar{v}} = \{\phi_{v=1\ldots\bar{v}} \geq 0,\ \sum_{v=1}^{\bar{v}} \phi_v = 1\}$ is the unitary simplex in $\mathbb{R}^{\bar{v}}$. The elements ϕ of unitary simplexes are used to describe polytopic type uncertainties. In the following, uncertainties ϕ are highlighted using the red color.

2 S-variables formulation of robust stability

We consider an LTI uncertain system that depends on a vector of constant but uncertain parameters q (to our knowledge the proposed results are the first to address the case where all matrices are parameter-dependent):

$$\begin{pmatrix} \dot{x} \\ y \end{pmatrix} = \begin{bmatrix} A(q) & B(q) \\ C(q) & 0 \end{bmatrix} \begin{pmatrix} x \\ u \end{pmatrix} \tag{1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the vector of control inputs, and $y \in \mathbb{R}^p$ is the vector of measured outputs. We assume that the uncertain parameters q take values in a set $\mathbb{Q}$ whose structure will be specified next, depending on whether a deterministic or a probabilistic approach is adopted for the static output feedback stabilizer design.

For the uncertain system (1), our primary goal is to design a robustly stabilizing static output feedback gain, clarified next:

(OF) Goal. Design a static output feedback gain F, such that the closed loop between plant (1) and $u = Fy$, corresponding to $\dot{x} = (A(q) + B(q)FC(q))x$ is robustly stable, namely the matrix $A(q) + B(q)FC(q)$ is Hurwitz for all $q \in \mathbb{Q}$.

A suggestive aspect of the approach presented in the sequel is that a few auxiliary (and arguably simpler) problems will turn out to be instrumental for the solution of the (OF) Goal, and correspond to:

- **(SF) Goal**. Design a State Feedback gain K such that the closed loop between plant (1) and $u = Kx$, corresponding to $\dot{x} = (A(q) + B(q)K)x$ is robustly stable.
- **(OI) Goal**. Design an Output Injection gain L such that the closed loop $\dot{x} = (A(q) + LC(q))x$ is robustly stable.

[2] Not available in the html-only version of the book.

- **(SI) Goal**. Design a State Injection gain J such that the closed loop $\dot{x} = (A(q) + J)x$ is robustly stable.

For the problems above, a necessary condition for the existence of a solution to the (SF) Goal is that the pair $(A(q), B(q))$ is stabilizable for all $q \in \mathbb{Q}$. Moreover, a necessary condition for the existence of a solution to the (OI) Goal is that the pair $(C(q), A(q))$ is detectable for all $q \in \mathbb{Q}$ (indeed L is well understood as the gain of a full order Luenberger observer). Finally, both conditions above are necessary for the existence of a solution to our main goal (OF), whereas goal (SI) is trivial and always feasible as long as $\mathbb{Q}$ is bounded and matrix $A(\cdot)$ is a locally bounded function.

The heuristic approach proposed in this chapter for the solution of the (OF) Goal is based on a main result presented here, wherein we manage to represent all the design goals (OF), (SF), (OI), (SI) listed above within a single matrix inequality depending bilinearly on a set of variables to be (optimally) selected. This matrix inequality arises from the dual calculations associated with [9, Thm 6.8], and involves a Lyapunov certificate $X(q) \succ 0$ and a number of S-variables. It corresponds to:

$$\begin{bmatrix} 0 & 0 & X(q) \\ 0 & 0 & 0 \\ X(q) & 0 & 0 \end{bmatrix} \prec \left\{ \begin{bmatrix} I \\ -\left(\lambda \begin{bmatrix} C(q) \\ 0_{p-n,n} \end{bmatrix} + M(q) \right) \\ -A(q) \end{bmatrix} S_1(q) + \begin{bmatrix} 0 \\ S_2 \\ B(q)Z \end{bmatrix} \begin{bmatrix} 0 & I & -H^T \end{bmatrix} \right\}^{\mathscr{S}} . \tag{2}$$

In particular, the relation between feasibility of (2) for certain selections of the blue variables, and the four design problems (OF), (OI), (SF) and (SI) is clarified in the next main result.

Theorem 1. *Consider system* (1) *and any selection of variables* $X > 0$, λ, M, S_1, S_2, Z, H *satisfying* (2) *for all* $q \in \mathbb{Q}$. *The following holds:*

- **(OF)** *if* $\lambda = 1$, $M(q) = 0$ *for all* $q \in \mathbb{Q}$ *and* S_2 *is nonsingular, then selection* $F = -ZS_2^{-1} \begin{bmatrix} I_p \\ 0_{n-p,p} \end{bmatrix}$ *solves the (OF) goal;*
- **(OI)** *if* $\lambda = 1$ *and* $M(q) = 0$ *for all* $q \in \mathbb{Q}$, *then selection* $L = H \begin{bmatrix} I_p \\ 0_{n-p,p} \end{bmatrix}$ *solves the (OI) goal;*
- **(SF)** *if* $\lambda = 0$, $M(q) = M$ *is common to all* $q \in \mathbb{Q}$, *and* S_2 *is nonsingular, then* $K = -ZS_2^{-1}M$ *solves the (SF) goal;*
- **(SI)** *if* $\lambda = 0$ *and* $M(q) = M$ *is common to all* $q \in \mathbb{Q}$, *then* $J = HM$ *solves the (SI) goal.*

We shall prove the four items of the theorem one by one. In particular, given any $q \in \mathbb{Q}$, for each one of the four items, we show below that the corresponding closed-loop matrix is Hurwitz. But before going into each individual proof, let us

state the following facts. Assuming invertibility of S_2, the congruence operation of $\begin{bmatrix} I & 0 & 0 \\ 0 & B(q)(-ZS_2^{-1}) & I \end{bmatrix}$ on (2) implies

$$\begin{bmatrix} 0 & X(q) \\ X(q) & 0 \end{bmatrix} \prec \left\{ \begin{bmatrix} I \\ -A(q) - B(q)(-ZS_2{}^{-1}) \left(\lambda \begin{bmatrix} C(q) \\ 0_{p-n,n} \end{bmatrix} + M(q) \right) \end{bmatrix} \hat{S}_1(q) \right\}^{\mathscr{S}} \tag{3}$$

where $\hat{S}_1(q) = S_1 \begin{bmatrix} I & 0 & 0 \\ 0 & B(q)(-ZS_2^{-1}) & I \end{bmatrix}^T$, and the congruence operation of $\begin{bmatrix} I & 0 & 0 \\ 0 & H & I \end{bmatrix}$ on (2) implies

$$\begin{bmatrix} 0 & X(q) \\ X(q) & 0 \end{bmatrix} \prec \left\{ \begin{bmatrix} I \\ -A(q) - H \left(\lambda \begin{bmatrix} C(q) \\ 0_{p-n,n} \end{bmatrix} + M(q) \right) \end{bmatrix} \check{S}_1 \right\}^{\mathscr{S}} \tag{4}$$

where $\check{S}_1 = S_1 \begin{bmatrix} I & 0 & 0 \\ 0 & H & I \end{bmatrix}^T$. The uncertainty q will be omitted in most steps of the following proofs to simplify the notations.

Proof of (OF). We need to show that matrix $A(q) + B(q)FC(q) = A + BFC$ is Hurwitz. Using the assumption that $\lambda = 1$, $M(q) = 0$, invertibility of S_2, and the selection $F = -ZS_2^{-1} \begin{bmatrix} I_p \\ 0_{n-p,p} \end{bmatrix}$, inequality (3) implies

$$\begin{bmatrix} 0 & X \\ X & 0 \end{bmatrix} \prec \left\{ \begin{bmatrix} I \\ -A - BFC \end{bmatrix} \hat{S}_1 \right\}^{\mathscr{S}}. \tag{5}$$

This S-variable inequality, together with $X \succ 0$, implies that $A + BFC$ is Hurwitz (see [9]). This is also corroborated by performing a congruence operation of $\begin{bmatrix} A + BFC & I \end{bmatrix}$ on (5) which gives the classical Lyapunov inequality: $(A + BFC)X + X(A + BFC)^T \prec 0$.

Proof of (OI). We need to show that matrix $A(q) + LC(q) = A + LC$ is Hurwitz. Using the assumption that $\lambda = 1$, $M(q) = 0$, and the selection $L = H \begin{bmatrix} I_p \\ 0_{n-p,p} \end{bmatrix}$, inequality (4) implies

$$\begin{bmatrix} 0 & X \\ X & 0 \end{bmatrix} \prec \left\{ \begin{bmatrix} I \\ -A - LC \end{bmatrix} \check{S}_1 \right\}^{\mathscr{S}}.$$

This S-variable inequality, together with $X \succ 0$, implies that $A + LC$ is Hurwitz.

Proof of (SF). We need to show that matrix $A(q) + B(q)K = A + BK$ is Hurwitz. Using the assumption that $\lambda = 0$, invertibility of S_2, and the selection $K = -ZS_2^{-1}M$, inequality (3) implies

$$\begin{bmatrix} 0 & X \\ X & 0 \end{bmatrix} \prec \left\{ \begin{bmatrix} I \\ -A-BK \end{bmatrix} \hat{S}_1 \right\}^{\mathscr{S}}.$$

This S-variable inequality, together with $X \succ 0$, implies that $A+BK$ is Hurwitz.

Proof of (SI). We need to show that matrix $A(q)+J = A+J$ is Hurwitz. Using the assumption that $\lambda = 0$ and the selection $J = HM$, inequality (4) implies

$$\begin{bmatrix} 0 & X \\ X & 0 \end{bmatrix} \prec \left\{ \begin{bmatrix} I \\ -A-J \end{bmatrix} \check{S}_1 \right\}^{\mathscr{S}}.$$

This S-variable inequality, together with $X \succ 0$, implies that $A+J$ is Hurwitz. ■

Remark 1. Theorem 1 establishes conditions for specific selections of variables λ and M. Alternative cases are also of interest. In particular, for the general case when $M \neq 0$ and $\lambda \neq 1$, S-variable conditions (3) and (4) show respectively that matrices $A(q) - B(q)ZS_2^{-1}\left(\lambda \begin{bmatrix} C(q) \\ 0_{n-p,p} \end{bmatrix} + M(q)\right)$ and $A(q) + H\left(\lambda \begin{bmatrix} C(q) \\ 0_{n-p,p} \end{bmatrix} + M(q)\right)$ are Hurwitz. These properties clarify the rationale behind the heuristic algorithm proposed in the next section, which stems from picking an initial "simple" selection such that $A(q)+HM$ be Hurwitz, and then performing iterations aiming at minimizing the norm of $M(q)$ while converging to $\lambda = 1$, so that the first one of the two matrices above corresponds to the closed loop with the static output feedback gain.

3 Robust Deterministic Static Output Feedback Design

3.1 Deterministic robust stability

In the deterministic approach addressed in this section, we shall assume a polytopic uncertainty structure where the matrices in (1) lie in the convex hull of vertices computed at extremal values $q^{[v]}$, $v = 1,\ldots,\bar{v}$, with $\bar{v}$ being the number of vertices of the polytopic representation:

$$\mathbb{Q} = \{q = \sum_{v=1}^{\bar{v}} \phi_v q^{[v]}, \quad \phi = (\phi_1,\ldots,\phi_{\bar{v}}) \in \Phi_{\bar{v}}\}. \tag{6}$$

A model from this uncertain polytopic set is parameterized by the barycentric coordinates $\phi \in \Phi_{\bar{v}}$ in the following form:

$$\begin{pmatrix} \dot{x} \\ y \end{pmatrix} = \begin{bmatrix} A_\phi & B_\phi \\ C_\phi & 0 \end{bmatrix} \begin{pmatrix} x \\ u \end{pmatrix} = \sum_{v=1}^{\bar{v}} \phi_v \begin{bmatrix} A(q^{[v]}) & B(q^{[v]}) \\ C(q^{[v]}) & 0 \end{bmatrix} \begin{pmatrix} x \\ u \end{pmatrix}. \tag{7}$$

Alternatively, in the probabilistic approach addressed in Section 4, we will assume a more general not necessarily convex dependence of the matrices in (1) on q, not requiring convexity of the uncertainty set.

The polytopic representation in (7), together with the peculiar structure of the S-variable characterization in (2), allows providing bilinear conditions imposed at the vertices of the polytope (6). Then, we may apply convex combinations to conclude robust stability in the whole polytope, as long as the dependence on the uncertain parameters is affine. In particular, a problematic term arises from the product between $S_1(q)$ and other uncertain variables in (2). Due to this fact, we propose the use of a more conservative condition, corresponding to

$$\begin{bmatrix} 0 & 0 & X^{[v]} \\ 0 & 0 & 0 \\ X^{[v]} & 0 & 0 \end{bmatrix} \prec \left\{ \begin{bmatrix} I \\ -\left(\lambda \begin{bmatrix} C(q^{[v]}) \\ 0_{p-n,n} \\ -A(q^{[v]}) \end{bmatrix} + M^{[v]}\right) \end{bmatrix} S_1 + \begin{bmatrix} 0 \\ S_2 \\ B(q^{[v]})Z \end{bmatrix} \begin{bmatrix} 0 & I & -H^T \end{bmatrix} \right\}^{\mathscr{S}}, \tag{8}$$

for all $v = 1 \dots \bar{v}$, where we selected a common value S_1 for all the values of $q \in \mathbb{Q}$. We may then prove the following Corollary to Theorem 1.

Corollary 1. *Consider system* (7) *and any selection of variables* $X^{[v]} \succ 0$, λ, $M^{[v]}$, S_1, S_2, Z, H *satisfying* (8) *for all* $v = 1 \dots \bar{v}$. *The following holds:*

- **(OF)** *if* $\lambda = 1$, $M^{[v]} = 0$ *for all* $v = 1 \dots \bar{v}$, *and* S_2 *is nonsingular, then selection* $F = -ZS_2^{-1} \begin{bmatrix} I_p \\ 0_{n-p,p} \end{bmatrix}$ *solves the (OF) goal;*
- **(OI)** *if* $\lambda = 1$ *and* $M^{[v]} = 0$ *for all* $v = 1 \dots \bar{v}$, *then selection* $L = H \begin{bmatrix} I_p \\ 0_{n-p,p} \end{bmatrix}$ *solves the (OI) goal;*
- **(SF)** *if* $\lambda = 0$, $M^{[v]} = M$ *is common to all* $v = 1 \dots \bar{v}$, *and* S_2 *is nonsingular, then* $K = -ZS_2^{-1}M$ *solves the (SF) goal;*
- **(SI)** *if* $\lambda = 0$ *and* $M^{[v]} = M$ *is common to all* $v = 1 \dots \bar{v}$, *then* $J = HM$ *solves the (SI) goal.*

Proof. The proof is based on the selection of the parameter dependent matrix $X_\phi = \sum_{v=1}^{\bar{v}} \phi_v X^{[v]} \succ 0$, which emerges naturally when performing a convex combination, through ϕ of inequalities (8), providing

$$\begin{bmatrix} 0 & 0 & X_\phi \\ 0 & 0 & 0 \\ X_\phi & 0 & 0 \end{bmatrix} \prec \left\{ \begin{bmatrix} I \\ -\left(\lambda \begin{bmatrix} C_\phi \\ 0_{p-n,n} \\ -A_\phi \end{bmatrix} + M_\phi\right) \end{bmatrix} S_1 + \begin{bmatrix} 0 \\ S_2 \\ B_\phi Z \end{bmatrix} \begin{bmatrix} 0 & I & -H^T \end{bmatrix} \right\}^{\mathscr{S}}. \tag{9}$$

Since equation (9) involves the matrices of the uncertain system (7), the proof is completed following steps parallel to those of the proof of Theorem 1, using the polytopic stability certificate X_ϕ. ■

Remark 2. While Theorem 1 provides conditions that are hard to check in practice, Corollary 1 corresponds to a conservative way to obtain a viable practical approach to the problem. In particular, the use of a common value of S_1 for all vertices is key to ensuring the affine nature of the conditions with respect to the polytopic uncertainty in (6), so that the convex combination can be carried over to the uncertainty-dependent conditions in (8). Note that while the common value of S_1 may be a source of conservatism of the tractable conditions of Corollary 1 (as compared to those of Theorem 1), a byproduct is the polytopic nature of the selected Lyapuonv function (see [9, Lemma 3.3] which proves that the search for polytopic Lyapunov certificates is lossless under the constraint that the S-variable is common to all uncertainties). More general parameter-depedent Lyapunov functions may be effective at reducing the conservatism, perhaps at the expense of a higher computational burden. This is one of the goals of the probabilistic approach adopted in Section 4.

3.2 Iterative heuristic for deterministic robust control

In this section we propose a heuristic procedure to design a robust static output feedback exploiting the matrix inequalities (8) and Corollary 1. The proposed approach is an iterative procedure to address the bilinear nature of (8), while starting from a reasonable initial condition. It consists of three fundamental phases:

- an *initialization phase*, which solves the (SI) and (SF) goals, also providing an initial guess of a solution to (8) having promising features in terms of convergence to the condition $\lambda = 1$ and $M(\cdot) \equiv 0$ required in item (OF) of Corollary 1;
- an *iteration phase*, which iterates between two steps aiming at refining the candidate solution to the BMI in the direction of this necessary condition $\lambda = 1$ and $M(\cdot) \equiv 0$;
- a *validation phase*, comprising a semidefinite program solving the (OF) and (OI) goals, thus providing a static output feedback selection if the previous phase converged to a solution sufficiently close to the condition $\lambda = 1$ and $M(\cdot) \equiv 0$.

Let us present the three above-mentioned phases one by one.

3.2.1 Initialization phase

The initialization phase aims at finding an initial selection of $X^{[v]} > 0$, λ, $M^{[v]}$, S_1, S_2, Z, H satisfying (8). A possible strategy is to fix variables λ, $M^{[v]}$, and H, so that optimizing the remaining variables is a convex LMI problem. We then select these variables according to the following straightforward consequence of Corollary 1.

Proposition 1. *For a selection $X^{[v]} \succ 0$, $\lambda = 0$, $M^{[v]} = M$, S_1, S_2, Z, H to be a solution of* (8)*, it is necessary that $A(q) + J = A(q) + HM$ be Hurwitz for all $q \in \mathbb{Q}$.*

Motivated by the proposition above, we propose the following selection:

$$\lambda = \lambda_0 = 0, \quad M^{[v]} = M_0, \quad H = H_0 = J_0 M_0^{-1}, \tag{10}$$

where J_0 ensures that $A(q)+J_0$ be Hurwitz for all $q \in \mathbb{Q}$, and M_0 is some invertible common selection of $M^{[v]}$. More specifically, keep in mind that we aim for the following convergence

$$\lambda \begin{bmatrix} C(q) \\ 0_{n-p,p} \end{bmatrix} + M(q) \to \begin{bmatrix} C(q) \\ 0_{n-p,p} \end{bmatrix}.$$

A natural choice of initial M_0 is hence such that its first p rows mimic $C(q)$. Therefore define $C_m = \frac{1}{\bar{v}} \sum_{v=1}^{\bar{v}} C(q^{[v]})$ the average of all matrices computed at vertices, and choose

$$M_0 = \begin{bmatrix} C_m^{\circ} C_m \\ C_m^{\perp T} \end{bmatrix}. \tag{11}$$

In this way M_0 is square and non-singular and its first rows span the same range as the average of the $C(q)$ matrices.

For the initial guess of the SI matrix J_0 in (10), to ensure that $A(q)+J_0$ be robustly stable, let μ denote the maximum real part of all matrices $A(q^{[v]})$. Then we may select

$$J_0 = (-\mu - h)I, \tag{12}$$

where $h > 0$ is a positive scalar. For a sufficiently large value of h, matrix $A(q)+J_0$ is then robustly stable. Clearly, increasing h provides a natural way to strengthen this robust stability condition, and is the baseline intuition for the initialization algorithm below.

Phase 1. INITIALIZATION (PROVIDES SOLUTIONS TO (SI) AND (SF))

1: **Input**: Select the initial values as per (10)–(12) with $h = 1$.
2: **Iteration**: Solve the following LMI problem for $v = 1 \dots \bar{v}$:

$$\begin{aligned} & X^{[v]} \succ 0, \\ & \begin{bmatrix} 0 & 0 & X^{[v]} \\ 0 & 0 & 0 \\ X^{[v]} & 0 & 0 \end{bmatrix} \prec \left\{ \begin{bmatrix} I \\ -M_0 \\ -A(q^{[v]}) \end{bmatrix} S_1 + \begin{bmatrix} 0 \\ S_2 \\ B(q^{[v]})Z \end{bmatrix} \begin{bmatrix} 0 & I & -H_0^T \end{bmatrix} \right\}^{\mathscr{S}}. \end{aligned} \tag{13}$$

If (13) is feasible, go to the next step. Otherwise, increase h, redefine H_0 according to (10), (12) and repeat step 2. If for larger values of h no solution exists, then stop: the iterative heuristic fails.

3: **Output** If a solution $X^{[v]}$, S_1, S_2, Z to (13) is found, then output $S_{1,0} = S_1$, $\hat{K}_0 = -ZS_2^{-1}$, $K = -ZS_2^{-1}M_0$ and $J = H_0 M_0$. From Corollary 1, K and J are proved to be robustly stabilizing SF and SI gains, respectively.

It is emphasized that there is no guarantee that the algorithm provides a correct solution, and even in the case where there exists a gain K inducing a common quadratic Lyapunov certificate X for the corresponding matrices $A(q^{[v]})+B(q^{[v]})K$, it is unclear how to get a proof of its successful termination. Nevertheless, practical experience revealed that the algorithm is quite effective in finding a feasible solution

to (8). Moreover, there was not a need to iterate on the value of h. If the LMIs were unfeasible for $h = 1$, then they happened to be unfeasible for larger values as well.

3.2.2 Iteration phase

If the initialization phase provides an initial feasible solution to (8), we may proceed with the iteration phase, whose goal is (starting from $\lambda_0 = 0$ and M_0) to iteratively reach a solution where $\lambda = 1$ and $M^{[v]} = 0$. This is done by maximizing $\lambda \in [0\ 1]$ with a constraint on the norm of $M^{[v]}$ of the type $(1-\lambda)I \succ M^T M$ and as formalized next.

Phase 2. ITERATION

1: **Input**: Start from the initial guess $S_{1,0}$ and $\hat{K}_0$ provided by Phase 1 (initialization phase). Initialize $k = 0$.

2: **Step k,1**: Let $k := k+1$. For a fixed $\hat{K}_{k-1}$, $S_{1,k-1}$ coming from the previous step, maximize λ under the following LMI conditions for $v = 1 \dots \bar{v}$

$$X^{[v]} \succ 0, \qquad \begin{bmatrix} (1-\lambda)I & {M^{[v]}}^T \\ M^{[v]} & I \end{bmatrix} \succeq 0, \qquad \lambda \geq 0,$$

$$\begin{bmatrix} 0 & 0 & X^{[v]} \\ 0 & 0 & 0 \\ X^{[v]} & 0 & 0 \end{bmatrix} \prec \left\{ \begin{bmatrix} I \\ -\left(\lambda \begin{bmatrix} C(q^{[v]}) \\ 0_{p-n,n} \end{bmatrix} + M^{[v]} \right) \\ -A(q^{[v]}) \end{bmatrix} S_{1,k-1} + \begin{bmatrix} 0 \\ -I \\ B(q^{[v]})\hat{K}_{k-1} \end{bmatrix} \begin{bmatrix} 0 & -S_2 & Y^T \end{bmatrix} \right\}^{\mathscr{S}}$$

at the optimum set $\lambda_k = \lambda$, $M_k^{[v]} = M^{[v]}$ and $H_k^T = S_2^{-1} Y^T$.

If $1 - \lambda_k$ is smaller than a (small) tolerance, then $F_k = \hat{K}_{k-1} \begin{bmatrix} I_p \\ 0_{n-p,p} \end{bmatrix}$ and $L_k = H_k \begin{bmatrix} I_p \\ 0_{n-p,p} \end{bmatrix}$ are reasonable candidates (OF) and (OI) robustly stabilizing gains, respectively. Therefore, transfer selection H_k to the validation phase. Otherwise, go to Step $k,2$.

3: **Step k,2**: For fixed λ_k, $M_k^{[v]}$ and H_k coming from the previous step, search by bisection the smallest $\alpha \in [0\ 1]$ such that the following LMIs hold for $v = 1 \dots \bar{v}$

$$X^{[v]} \succ 0,$$

$$\begin{bmatrix} 0 & 0 & X^{[v]} \\ 0 & 0 & 0 \\ X^{[v]} & 0 & 0 \end{bmatrix} \prec \left\{ \begin{bmatrix} I \\ -\hat{M}^{[v]}(\alpha) \\ -A(q^{[v]}) \end{bmatrix} S_1 + \begin{bmatrix} 0 \\ S_2 \\ B(q^{[v]})Z \end{bmatrix} \begin{bmatrix} 0 & I & -H_k^T \end{bmatrix} \right\}^{\mathscr{S}}$$

where $\hat{M}^{[v]}(\alpha) = \left((1 + \alpha(\lambda_k - 1)) \begin{bmatrix} C(q^{[v]}) \\ 0_{p-n,n} \end{bmatrix} + \alpha M_k^{[v]} \right)$. At the optimum set $\alpha_k = \alpha$, $\hat{K}_k = -ZS_2^{-1}$ and $S_{1,k} = S_1$.

If α_k is smaller than a (small) tolerance, then $F_k = \hat{K}_k \begin{bmatrix} I_p \\ 0_{n-p,p} \end{bmatrix}$ and $L_k = H_k \begin{bmatrix} I_p \\ 0_{n-p,p} \end{bmatrix}$ are reasonable candidates (OF) and (OI) robustly stabilizing gains, respectively. Therefore, transfer selection H_k to the validation phase. Otherwise, go to Step $k+1,1$.

The key feature enjoyed by the two steps of the procedure listed in Phase 2 above, is that whenever moving from one step to the next one, the quality of the optimized solution (in terms of size of $1-\lambda$ or α) cannot get worse. This is clarified in the next proposition, whose proof is straightforward.

Proposition 2. *For the iterations listed in Phase 2, the following holds:*

- *Given any initial solution provided by Phase 1, the conditions at Step 1,1 are feasible for* $\lambda = 0$*;*
- *Given any solution from Step k,1, Step k,2 is feasible for* $\alpha = 1$*;*
- *Given any solution from Step k,2, Step k+1,1 is feasible for* $\lambda = \lambda_k + (1-\lambda_k)(1-\alpha_k)$.

3.2.3 Validation phase

This heuristic algorithm is completed by a validation phase, which comprises the solution of an LMI, parameterized by matrix H, selected according to the iteration phase of the previous section.

Phase 3. VALIDATION (PROVIDES SOLUTIONS TO (OI) AND (OF))

1: **Input**: Start from matrix H, produced as an output of Phase 2 (iteration).
2: **Validation Step**: Solve the following LMI feasibility problem for $v = 1 \dots \bar{v}$

$$X^{[v]} \succ 0,$$

$$\begin{bmatrix} 0 & 0 & X^{[v]} \\ 0 & 0 & 0 \\ X^{[v]} & 0 & 0 \end{bmatrix} \prec \left\{ \begin{bmatrix} I \\ -\begin{bmatrix} C(q^{[v]}) \\ 0_{p-n,n} \end{bmatrix} \\ -A(q^{[v]}) \end{bmatrix} S_1 + \begin{bmatrix} 0 \\ S_2 \\ B(q^{[v]})Z \end{bmatrix} \begin{bmatrix} 0 & I & -H^T \end{bmatrix} \right\}^{\mathscr{S}}. \tag{14}$$

If a solution is found, then from Corollary 1 $F = -ZS_2^{-1}\begin{bmatrix} I_p \\ 0_{n-p,p} \end{bmatrix}$ and $L = H\begin{bmatrix} I_p \\ 0_{n-p,p} \end{bmatrix}$ are respectively proved to be robustly stabilizing (OF) and (OI) gains. Otherwise the validation phase fails and the algorithm should go back to the iteration phase reducing the tolerance for α and λ.

Note that the LMI conditions in the Validation Step are guaranteed to be feasible whenever matrix H coming from the iteration step was associated with $\lambda = 1$ (equivalently, from Proposition 2, $\alpha = 0$). However, in general one may find it convenient to run the validation step even for cases where these conditions are not exactly met. Due to this fact, and possibly due to numerical errors, it makes sense to possibly come back to the iteration phase (from a failing validation phase) to further improve, via extra iterations, the previous candidate selection of H.

4 Probabilistic Static Output Feedback Design

As emphasized in the previous section, using a polytopic approach to address the design of suitable matrices guaranteeing the conditions in Theorem 1 may be too conservative for the problem at hand. For this reason, in this section we follow an alternative paradigm based on a probabilistic approach, which allows for uncertain dynamics more general than (7) (thereby not requiring convexity with respect to the uncertainty q), enables using multipliers that are not necessarily common among the samples, reaches beyond the use of polytopic Lyapunov certificates, but comes at the expense of providing a probabilistic guarantee of robust stability (rather than a deterministic one), in addition to typically being computationally more expensive.

In particular, throughout this section we do not assume that the uncertainty lies in a convex polytope, but we consider a more general setup, in which the state matrices in (1) may be generic continuous (possibly nonlinear) functions of the uncertainty parameter q. On the other hand, following a classical probabilistic approach [4, 20], we require to have additional probabilistic information on the uncertainty. Formally, we assume that $A(q), B(q), C(q)$ are continuous measurable functions of q, and that q is a random variable with probability distribution Pr with support $\mathbb{Q}$. Such a probability distribution may describe the likelihood of each occurrence of the uncertainty, or a user-defined weight for all possible uncertain situations.

Then, randomized algorithms are applied to design a controller that guarantees performance with a prescribed level of probability. These algorithms are based on the extraction of random samples of the uncertainty

$$q^{(1)} \dots q^{(\bar{r})} \in \mathbb{Q},$$

and the construction of sampled convex programs. The focus of this approach is in the derivation of sample complexity bounds, i.e. bounds on the number of samples to be extracted so as to ensure that the desired probabilistic guarantees are met.

In the next section, we briefly recall the so-called scenario approach originally presented in [3] for dealing with convex optimization problems in the presence of uncertainty.

4.1 The scenario approach

Let us consider a generic class of robust convex optimization problems of the form

$$\begin{aligned} \theta_{\mathrm{RO}} = \arg \ \min_{\theta \in \Theta} \ & c^T \theta \\ \text{s.t. } & f(\theta, q) \leq 0, \ \forall q \in \mathbb{Q}, \end{aligned} \tag{15}$$

where $\theta \in \Theta$ denotes the design variable, bounded in a domain Θ, which is a convex and compact set in $\mathbb{R}^{n_\theta}$, and q is the uncertainty, bounded in the uncertainty set $\mathbb{Q}$, not necessarily compact. For a given $q \in \mathbb{Q}$, $f(\theta, q)$ is a convex function of the op-

timization variable. Furthermore, we assume that $f(\theta, q)$ is a continuous (possibly nonlinear) function of q for any given θ.

To construct a sampled convex program, N independent identically distributed (i.i.d.) samples are extracted according to the probability distribution of q, and the following scenario optimization problem, based on $\bar{r}$ instances (scenarios) of the uncertain constraints

$$\theta_{\mathrm{SO}} = \arg\min_{\theta \in \Theta} c^T \theta \quad \text{s.t. } f(\theta, q^{(r)}) \leq 0,\ r = 1 \dots \bar{r}. \tag{16}$$

Problem (16) represents a sampled relaxation of Problem (15), since it deals only with a subset of the (infinite number of) constraints considered in (15), according to the probability distribution of the uncertainty. However, under rather mild assumptions on Problem (15), by suitably choosing $\bar{r}$, this approximation may in practice become negligible in some probabilistic sense. Specifically, $\bar{r}$ can be selected depending on the level of "risk" of constraint violation that the user is willing to accept. To this end, we define the *violation probability* of a design θ as follows

$$\mathrm{Viol}(\theta) \doteq \Pr\{q \in \mathbb{Q} : f(\theta, q) > 0\} \tag{17}$$

The following result has been proven in [6].

Proposition 3. *[6] Assume that, for any multisample extraction, Problem (16) is feasible and attains a unique optimal solution. Then, given an accuracy level* $\varepsilon \in (0,1)$*, the solution* θ_{SO} *of Problem* (16) *satisfies*

$$\Pr\{\mathrm{Viol}(\theta_{\mathrm{SO}}) > \varepsilon\} \leq \mathrm{B}(\bar{r}, \varepsilon, n_\theta), \tag{18}$$

where

$$\mathrm{B}(\bar{r}, \varepsilon, n_\theta) \doteq \sum_{k=0}^{n_\theta - 1} \binom{\bar{r}}{k} \varepsilon^k (1-\varepsilon)^{(\bar{r}-k)}. \tag{19}$$

We note that non-uniqueness of the optimal solution can be circumvented by imposing additional "tie-break" rules in the problem, see, *e.g.*, Appendix A of [3]. Also, in [5] it is shown that the feasibility assumption can be removed at the expense of substituting $n_\theta - 1$ with n_θ in $\mathrm{B}(\bar{r}, \varepsilon, n_\theta)$.

From (18), explicit bounds on the number of samples necessary to guarantee the "goodness" of the solution have been derived. The bound provided in [1] shows that, if, for given $\varepsilon, \delta \in (0,1)$, the sample complexity $\bar{r}$ is chosen to satisfy the sample-complexity bound

$$\bar{r} \geq \frac{\mathrm{e}}{\varepsilon(\mathrm{e}-1)} \left(\ln \frac{1}{\delta} + n_\theta - 1 \right) \tag{20}$$

(where e denotes the Euler number), then the solution θ_{SO} of Problem (16) satisfies $\mathrm{Viol}(\theta_{\mathrm{SO}}) \leq \varepsilon$ with probability $1 - \delta$. This bound improves by a constant factor upon previous bounds, see e.g. [5], and it shows that Problem (16) exhibits linear dependence in $1/\varepsilon$ and n_θ, and logarithmic dependence on $1/\delta$. Note however that,

from a practical viewpoint, it is always preferable to numerically solve the one-dimensional problem of finding the smallest integer $\bar{r}$ such that $\mathrm{B}(\bar{r},\varepsilon,n_\theta) \leq \delta$.

4.2 Scenario with certificates

The classical scenario approach previously discussed deals with uncertain optimization problems where all variables θ are to be designed. On the other hand, in the design with certificates approach we distinguish between *design variables* θ and *certificates* ξ. The certificates are represented here in green color, and correspond to those variables which are not involved in the construction of the design, but whose existence is necessary for its derivation. A classical example of certificates are Lyapunov functions for proving stability.

Formally, we consider a function $f(\theta,\xi,q)$, *jointly convex* in $\theta \in \Theta$ and $\xi \in \Xi \subseteq \mathbb{R}^{n_\xi}$ for given $q \in \mathbb{Q}$, and study the following robust optimization problem with certificates

$$\theta_{\mathrm{RwC}} = \arg\min_{\theta} c^T\theta \quad \text{s.t. } \theta \in \mathscr{S}(q),\ \forall q \in \mathbb{Q}, \tag{21}$$

where the set $\mathscr{S}(q)$ is defined as

$$\mathscr{S}(q) \doteq \{\theta \in \Theta |\ \exists \xi = \xi(q) \in \Xi \text{ satisfying } f(\theta,\xi,q) \leq 0\}.$$

From the above formulation, the role of certificates is clear: for any value of the uncertainty, the existence of a certificate (possibly depending on the given value of q) is required.

A key observation is that the set $\mathscr{S}(q)$ is convex in θ for any given q, see [12, Theorem 1]. These observations lead to the introduction of the following *scenario with certificates* problem, based again on a sample extraction, to approximate Problem (21), inspired by a similar approach proposed in [17] for the iterative solution of parameter-dependent LMIs:

$$\theta_{\mathrm{SwC}} = \arg \min_{\theta,\xi^{(1)},\ldots,\xi^{(\bar{r})}} c^T\theta \quad \text{s.t. } f(\theta,\xi^{(r)},q^{(r)}) \leq 0,\ r = 1\ldots\bar{r}. \tag{22}$$

Note that, contrary to Problem (16), in this case a new certificate variable $\xi^{(r)}$ is created for every sample $q^{(r)}$, $r = 1,\ldots,\bar{r}$. To analyze the properties of the solution θ_{SwC}, we note that, in the case of SwC, the violation probability of design θ are given by

$$\mathrm{Viol}(\theta) = \Pr\Big\{q \in \mathbb{Q} \,|\, \nexists \xi \in \Xi \text{ satisfying } f(\theta,\xi,q) \leq 0\Big\}.$$

Then, the following theorem can be stated, from [12, Thm 1].

Theorem 2. *[12] Assume that, for a multisample extraction, Problem* (22) *is feasible and attains a unique optimal solution. Then, given an accuracy level* $\varepsilon \in (0,1)$*, the solution* θ_{SwC} *of Problem* (22) *satisfies*

$$\Pr\{\mathrm{Viol}(\theta_{\mathrm{SwC}}) > \varepsilon\} \leq \mathrm{B}(\bar{r}, \varepsilon, n_\theta). \tag{23}$$

We remark that Problem (22) has $\bar{r}$ separate constraints, one for each $q^{(r)}$, and each constraint involves a different certificate. However, notice that the dimension n_ξ of the certificates ξ does not enter into the right-hand side of the probability bound (23) in Theorem 2. Hence, the sample complexity of Problem (22) is smaller than that of the scenario counterpart of the problem with common certificates, in which both θ and ξ play the role of design variables. On the other hand, the complexity of solving Problem (22) is higher, since the number of optimization variables significantly increases, because a different variable $\xi^{(r)}$ is introduced for every sample $q^{(r)}$. This increase in complexity is not surprising, being Problem (22) much more difficult than the robust problem involving common certificates.

4.3 Probabilistic robust stability

In this section, we exploit the SwC setting previously discussed to derive a sample-based heuristic for designing a SOF controller guaranteeing robust stability in probability.

To this end, we revisit the heuristic approach presented in Section 3.2, and observe that both the initialization and the validation phases involve the solution of uncertain LMI problems, where the values of λ, M and H are fixed ($\lambda = 0$, $M = M_0$, $H = H_0$ in Phase 1 and $\lambda = 1$, $M = 0$, $H = H$ in Phase 3). In that case, the necessity of having a convex formulation with respect to q forced us to impose the S-variable $S_1(q)$ to be fixed and independent of q. This limitation can be lifted in the sample-based approach, due to the general result of Theorem 2.

In the following corollary, which is a direct application of Theorems 1 and 2, we show how a sample-based approximation of Problem (2) with fixed values of λ, M, and H, can be derived, together with a precise characterization of its probabilistic properties.

Corollary 2. *Given* $\varepsilon, \delta \in (0,1)$*, extract* $\bar{r}$ *i.i.d. samples* $q^{(1)} \dots q^{(\bar{r})}$ *of the uncertainty* $q \in \mathbb{Q}$*, where* $\bar{r}$ *satisfies*

$$\bar{r} \geq \frac{\mathrm{e}}{\varepsilon(\mathrm{e}-1)} \left(\ln \frac{1}{\delta} + n(n+m) - 1 \right). \tag{24}$$

Consider a selection of $\lambda \in [0,1]$*, M and H. If there exist matrices* S_2*, Z, and certificates* $X^{(r)} \succ 0$*,* $S_1^{(r)}$*, satisfying for* $r = 1 \dots \bar{r}$*,*

$$\begin{bmatrix} 0 & 0 & X^{(r)} \\ 0 & 0 & 0 \\ X^{(r)} & 0 & 0 \end{bmatrix} \prec \left\{ \begin{bmatrix} I \\ -\left(\lambda \begin{bmatrix} C(q^{(r)}) \\ 0_{p-n,n} \end{bmatrix} + M \right) \\ -A(q^{(r)}) \end{bmatrix} S_1^{(r)} + \begin{bmatrix} 0 \\ S_2 \\ B(q^{(r)})Z \end{bmatrix} \begin{bmatrix} 0 & I & -H^T \end{bmatrix} \right\}^{\mathscr{S}}, \tag{25}$$

then, we guarantee with confidence at least $1-\delta$*, a probability of at least* $1-\varepsilon$ *that*

- **(OF)** *if* $\lambda = 1$, $M = 0$ *and* S_2 *is invertible, then selection* $F = -ZS_2^{-1} \begin{bmatrix} I_p \\ 0_{n-p,p} \end{bmatrix}$ *solves the (OF) goal;*
- **(OI)** *if* $\lambda = 1$ *and* $M = 0$*, then selection* $L = H \begin{bmatrix} I_p \\ 0_{n-p,p} \end{bmatrix}$ *solves the (OI) goal;*
- **(SF)** *if* $\lambda = 0$ *and* S_2 *is invertible, then* $K = -ZS_2^{-1}M$ *solves the (SF) goal;*
- **(SI)** *if* $\lambda = 0$*, then* $J = HM$ *solves the (SI) goal.*

4.4 Iterative heuristic for probabilistic robust control

In the sequel, we show how the application of Corollary 2 allows deriving a sample-based version of the heuristic introduced in Section 3.2 for the deterministic case. In particular, the proposed approach involves again three phases:

1. An initialization phase, in which a sample-based SwC problem is solved, leading to the construction of initial candidate variables to be passed to the iteration phase. In particular, this first phase returns, as a side-result, the design of probabilistic solutions to the robust state feedback (SF) and state injection (SI) goals. More specifically, we are able to assess precise probabilistic properties of these solutions, in terms of the measure of the uncertainty set that it may fail to stabilize.
2. An iteration phase, in which a small subset $\bar{r}_\mathrm{d} \leq \bar{r}$ of the samples employed in the first phase is randomly selected (for instance, the first $\bar{r}_d$ ones), and is used in an iterative way to "push" the design obtained in the first phase (wherein $\lambda = 0$) towards a SOF design (wherein we need $\lambda = 1$).
3. A validation phase where, based on matrix H of the previous phase, a sample-based SwC problem is solved for the design of probabilistic solutions to the robust output feedback (OF) and output injection (OI) goals.

It should be noted that if Phase 3 fails, one may investigate more accurate selections of matrix H by repeating phase 2 with a larger number $\bar{r}_\mathrm{d}$ of samples from Phase 1.

4.4.1 Sample-based initialization phase

The initialization phase represents, substantially, the sample-based equivalent of Phase 1 presented in Section 3.2.1.

Phase 1. SAMPLE-BASED INITIALIZATION (PROVIDES PROBABILISTIC SOLUTIONS TO (SI) AND (SF))

1: **Input**: Select the initial values as per (10)–(12) with $h = 1$.
2: **Sample generation**: Given probabilistic levels $\delta, \varepsilon \in [0,1]$, set $\bar{r}$ as per (24), and generate $\bar{r}$ i.i.d. samples $q^{(1)} \dots q^{(\bar{r})}$ according to distribution Pr.
3: **Iteration**: Solve the following sampled feasibility problem for $r = 1 \dots \bar{r}$

$$\begin{array}{l} X^{(r)} \succ 0, \\ \begin{bmatrix} 0 & 0 & X^{(r)} \\ 0 & 0 & 0 \\ X^{(r)} & 0 & 0 \end{bmatrix} \prec \left\{ \begin{bmatrix} I \\ -M_0 \\ -A(q^{(r)}) \end{bmatrix} S_1^{(r)} + \begin{bmatrix} 0 \\ S_2 \\ B(q^{(r)})Z \end{bmatrix} \begin{bmatrix} 0 & I & -H_0^T \end{bmatrix} \right\}^{\mathscr{S}} \end{array} \tag{26}$$

If (26) is feasible, go to the next step. Otherwise, increase h, redefine H_0 according to (10), (12) and repeat step 2. If for larger values of h no solution exists, then stop: the iterative heuristic fails.
4: **Output** If a feasible solution to (26) is found, then output $\hat{K}_0 = -ZS_2^{-1}$, $K = -ZS_2^{-1}M_0$, $J = H_0M_0$, together with samples $q^{(1)} \dots q^{(\bar{r})}$ and the corresponding S-variables $S_1^{(1)} \dots S_1^{(\bar{r})}$, S_2, Z.

Also in this case, there is no guarantee, even in a probabilistic sense, that this step of the algorithm will return a feasible solution. However, if a solution is returned, then by Corollary 2 K and J defined in Step 4 are guaranteed to solve in a probabilistic way the robust state feedback (SF) and state injection (SI) goals, respectively. Moreover, its output constitutes the initialization step of the iteration phase presented next.

4.4.2 Sample-based iteration phase

The objective of this phase is to iteratively "push" the initial solution to (25) provided by Phase 1 (with $\lambda_0 = 0$, $M = M_0$) towards a solution to (25) with $\lambda = 1$ and $M = 0$. This phase represents a completely heuristic procedure, which, if successful, returns a parameter H for the next validation phase, which is instead based on the rigorous results of Corollary 2.

This phase is the one that is computationally most expensive. To alleviate the computational load, a subset of $\bar{r}_{\rm d} \leq \bar{r}$ *design samples* is selected among the samples returned by Phase 1.

Phase 2. SAMPLE-BASED ITERATION

1: **Input**: Start from the initial guess $\hat{K}_0$ provided by Phase 1 (initialization phase). Initialize $k = 0$
2: **Design samples selection** Select a (small) number $\bar{r}_{\rm d} \leq \bar{r}$ of samples $q^{(r)}$ and the corresponding S-variables $S_{1,0}^{(r)} := S_1^{(r)}$, $r = 1 \dots \bar{r}_{\rm d}$ returned by Phase 1.

3: **Step k,1**: Let $k := k+1$. For fixed $\hat{K}_{k-1}$, $S_{1,k-1}^{(r)}$ coming from the previous step, maximize λ under the following conditions for $r = 1 \dots \bar{r}_{\mathrm{d}}$

$$X^{(r)} \succ 0, \qquad \begin{bmatrix} (1-\lambda)I & M^{(r)^T} \\ M^{(r)} & I \end{bmatrix} \succeq 0, \qquad \lambda \geq 0,$$

$$\begin{bmatrix} 0 & 0 & X^{(i)} \\ 0 & 0 & 0 \\ X^{(r)} & 0 & 0 \end{bmatrix} \prec \left\{ \begin{bmatrix} I \\ -\left(\lambda \begin{bmatrix} C(q^{(r)}) \\ 0_{p-n,n} \end{bmatrix} + M^{(r)} \right) \\ -A(q^{(r)}) \end{bmatrix} S_{1,k-1}^{(r)} + \begin{bmatrix} 0 \\ -I \\ B(q^{(r)})\hat{K}_{k-1} \end{bmatrix} \begin{bmatrix} 0 & -S_2 & Y^T \end{bmatrix} \right\}^{\mathscr{S}}$$

at the optimum set $\lambda_k = \lambda$, $M_k^{(r)} = M^{(r)}$ and $H_k^T = S_2^{-1} Y^T$.

If $1-\lambda_k$ is smaller than a (small) tolerance level, then $F_k = \hat{K}_{k-1} \begin{bmatrix} I_p \\ 0_{n-p,p} \end{bmatrix}$ and $L_k = H_k \begin{bmatrix} I_p \\ 0_{n-p,p} \end{bmatrix}$ are reasonable candidates (OF) and (OI) robustly stabilizing gains, respectively. Therefore, transfer selection H_k to the validation phase. Otherwise, go to Step $k,2$.

4: **Step k,2**: For fixed λ_k, $M_k^{(r)}$ and H_k coming from the previous step, search by bisection the smallest $\alpha \in [0\ 1]$ such that the following inequalities hold for $r = 1 \dots \bar{r}_{\mathrm{d}}$

$$X^{(r)} \succ 0,$$
$$\begin{bmatrix} 0 & 0 & X^{(r)} \\ 0 & 0 & 0 \\ X^{(r)} & 0 & 0 \end{bmatrix} \prec \left\{ \begin{bmatrix} I \\ -\hat{M}(q^{(r)}, \alpha) \\ -A(q^{(r)}) \end{bmatrix} S_1^{(r)} + \begin{bmatrix} 0 \\ S_2 \\ B(q^{(r)})Z \end{bmatrix} \begin{bmatrix} 0 & I & -H_k^T \end{bmatrix} \right\}^{\mathscr{S}}$$

where $\hat{M}(q^{(r)}, \alpha) = \left((1 + \alpha(\lambda_k - 1)) \begin{bmatrix} C(q^{(r)}) \\ 0_{p-n,n} \end{bmatrix} + \alpha M_k^{(r)} \right)$. At the optimum set $\alpha_k = \alpha$, $\hat{K}_k = -ZS_2^{-1}$ and $S_{1,k}^{(r)} = S_1^{(r)}$.

If α_k is smaller than a (small) tolerance level, then $F_k = \hat{K}_k \begin{bmatrix} I_p \\ 0_{n-p,p} \end{bmatrix}$ and $L_k = H_k \begin{bmatrix} I_p \\ 0_{n-p,p} \end{bmatrix}$ are reasonable candidates OF and OI robustly stabilizing gains, respectively. Therefore, transfer selection H_k to the validation phase. Otherwise, go to Step $k+1,1$.

Similar to its deterministic counterpart, the iterations in Phase 2 guarantee that the quality of the optimized solution does not get worse. In particular, the results of Proposition 2 still hold.

4.4.3 Sample-based validation phase

The validation phase, uses matrix H returned by the iterations in Phase 2 to construct a scenario with certificates problem.

Phase 3. SAMPLE-BASED VALIDATION (PROVIDES PROBABILISTIC SOLUTIONS TO (OI) AND (OF))

1: **Input**: Start from matrix H, produced as an output of Phase 2 (iteration).

2: **Sample generation**: Given probabilistic levels $\delta, \varepsilon \in [0,1]$, set $\bar{r}$ as per (24), and generate $\bar{r}$ i.i.d. samples $q^{(1)} \dots q^{(\bar{r})}$ according to distribution Pr.
3: **Validation Step**: Solve the following sampled problem for $r = 1 \dots \bar{r}$:

$$X^{(r)} \succ 0,$$
$$\begin{bmatrix} 0 & 0 & X^{(r)} \\ 0 & 0 & 0 \\ X^{(r)} & 0 & 0 \end{bmatrix} \prec \left\{ \begin{bmatrix} I \\ -\begin{bmatrix} C(q^{(r)}) \\ 0_{p-n,n} \end{bmatrix} \\ -A(q^{(r)}) \end{bmatrix} S_1^{(r)} + \begin{bmatrix} 0 \\ S_2 \\ B(q^{(r)})Z \end{bmatrix} \begin{bmatrix} 0 & I & -H^T \end{bmatrix} \right\}^{\mathscr{S}} \tag{27}$$

If a solution is found, then from Corollary 2 selections $F = -ZS_2^{-1} \begin{bmatrix} I_p \\ 0_{n-p,p} \end{bmatrix}$ and $L = H \begin{bmatrix} I_p \\ 0_{n-p,p} \end{bmatrix}$ are probabilistic solutions to the output feedback (OF) and the Output Injection (OI) goals, respectively. Otherwise, the validation phase fails and the algorithm should go back to the iteration phase, increasing the number of selected samples by choosing a larger number $\bar{r}_{\mathrm{d}}$.

Note that the sample-based validation step is by nature less conservative than the corresponding deterministic one (14) for two main reasons: i) it does not require the solution to be feasible for all possible values of the uncertainty, but it requires feasibility only for a suitably selected number of samples, ii) it does not require a common S-variable S_1, but it allows for parameter-dependent certificates. This is done at the expense of giving up deterministic robustness, but instead allowing for a (typically small) probability of failure.

However, if one is indeed interested in robustly guaranteed results, it should be pointed out that nothing prevents us from testing the output of the probabilistic Phase 2 by means to the corresponding deterministic Validation Step (14).

5 Numerical Examples

5.1 OF design without uncertainties

Although the results are intended for robust stability, the heuristic algorithm can also be applied to systems without uncertainties. In this case, there is only one sample and the deterministic and probabilistic algorithms coincide. The CompLeib library provides a collection of such systems. We have tested the heuristic on some of these examples (of low order). The results are as follows ($h = 1$ in all cases). The algorithm finds a stabilizing output feedback gain

- at iteration $k = 1$ for examples AC1-AC5, AC12, AC15-AC17, HE2
- at iteration $k = 2$ for examples AC6, AC7, AC9, HE4
- at iteration $k = 4$ for example AC8
- at iteration $k = 7$ for example HE3
- does not converge for AC11, AC18, HE1 and HE5.

These results are quite encouraging because some of these examples were proved to be hard when tested with similar tools in [9].

5.2 Deterministic OF design with uncertainties

The next example is borrowed from [8] with slight modifications to ensure that all system matrices A, B and C are uncertain. These uncertain matrices belong to a polytope with two vertices:

$$\begin{aligned} A(q^{[1]}) &= \begin{bmatrix} -1 & 4 & 0 \\ 0 & 0 & 1 \\ \underline{a} & 6 & -1 \end{bmatrix}, \quad B(q^{[1]}) = \begin{bmatrix} 0 \\ 0 \\ 0.5 \end{bmatrix}, C(q^{[1]}) = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \\ A(q^{[2]}) &= \begin{bmatrix} -1 & 1 & 0 \\ 0 & -5 & 1 \\ \overline{a} & 1 & -1 \end{bmatrix}, B(q^{[2]}) = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}, \quad C(q^{[2]}) = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \end{aligned} \tag{28}$$

The peculiarity of this numerical example is that the uncertain input matrices $C(q)$ are of full row rank except at one of the vertex of the polytope. This rank deficiency corresponds to a failure of one of the sensors of the system. This uncertain system may be robustly stabilized via static output feedback for different ranges of the parameter $a \in [\underline{a}\ ,\ \overline{a}]$ as indicated in Table 1. For each integer value of $\underline{a} \in [0\ ,10]$ we search for the maximal integer $\overline{a}$ such that the algorithm finds a solution to the (OF) goal. K_0 is the state feedback (SF) gain found at the initialization Phase 1. $\bar{k}$ is the number of iterations in Phase 2 of the algorithm. The maximal number of iterations was set to 10, therefore $\bar{k} = 10$ indicates that Phase 2 terminated because this maximal number is reached. Otherwise the iterations stop when $1 - \lambda < 10^{-7}$. The column $F_{\bar{k}}$ gives the value of the output-feedback gain when the algorithm succeeds in finding a robustly stabilizing one. Results are given in Table 1. They outperform significantly those of [8]. Note that in many cases the number of iterations is very low (typically 2) and hence the computation time is not prohibitive. The last row of the table is a test of the method's capability to find robustly stabilizing state-feedback gains. For this last test we have set $h = 10$ in the initialization Phase 1. In all other tests $h = 1$.

5.3 A comparison between the deterministic and probabilistic approaches

The following example is taken from [7]. The system is given by

$$A(q) = \begin{bmatrix} 0 & -0.5+q_1 \\ 0.5+q_2 & 0 \end{bmatrix}, \quad B(q) = \begin{bmatrix} 0.5+q_1 \\ 0.5-q_2 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

$[\underline{a}\ ,\ \bar{a}]$	K_0	$\bar{k}$	$F_{\bar{k}}$	time
[0 , 9]	$\begin{bmatrix}-1.5939 & -17.5869 & -7.1516\end{bmatrix}$	2	$\begin{bmatrix}-1.3792 & -27.4388\end{bmatrix}$	6.5s
[1 , 14]	$\begin{bmatrix}-3.0824 & -20.0953 & -7.4500\end{bmatrix}$	2	$\begin{bmatrix}-2.6507 & -29.2585\end{bmatrix}$	7.4s
[2 , 23]	$\begin{bmatrix}-5.1340 & -24.8380 & -8.4098\end{bmatrix}$	2	$\begin{bmatrix}-4.9091 & -37.1852\end{bmatrix}$	7.4s
[3 , 29]	$\begin{bmatrix}-7.1790 & -32.6351 & -10.1779\end{bmatrix}$	7	$\begin{bmatrix}-6.8056 & -48.0987\end{bmatrix}$	33.8s
[4 , 49]	$\begin{bmatrix}-16.1234 & -48.9351 & -13.5936\end{bmatrix}$	10	$\begin{bmatrix}-13.0042 & -52.9144\end{bmatrix}$	51.3s
[5 , 58]	$\begin{bmatrix}-21.4924 & -57.7095 & -14.8185\end{bmatrix}$	5	$\begin{bmatrix}-14.4984 & -47.2955\end{bmatrix}$	28.1s
[6 , 72]	$\begin{bmatrix}-27.1872 & -59.7972 & -14.1779\end{bmatrix}$	7	$\begin{bmatrix}-18.9969 & -45.5168\end{bmatrix}$	45.6s
[7 , 77]	$\begin{bmatrix}-31.3509 & -62.1565 & -14.7895\end{bmatrix}$	10	$\begin{bmatrix}-21.6073 & -42.3976\end{bmatrix}$	45.6s
[8 , 80]	$\begin{bmatrix}-34.4863 & -63.0824 & -14.8762\end{bmatrix}$	2	$\begin{bmatrix}-23.2575 & -38.5662\end{bmatrix}$	7.5s
[9 , 83]	$\begin{bmatrix}-37.6496 & -63.8179 & -14.9218\end{bmatrix}$	2	$\begin{bmatrix}-25.2797 & -35.1410\end{bmatrix}$	8.0s
[10 , 86]	$\begin{bmatrix}-40.7372 & -64.4806 & -14.9106\end{bmatrix}$	2	$\begin{bmatrix}-26.9999 & -31.4823\end{bmatrix}$	6.8s
[0 , 1000]	$\begin{bmatrix}-1305.7 & -695.5 & -73.5\end{bmatrix}$	10	fail	59.3s

Table 1 Robust stabilizing SOF gains for numerical example (28).

To compare the deterministic and the probabilistic approaches, we let q_1 and q_2 be defined as

$$q_1 = \bar{q}\cos(2\theta), \quad q_2 = \bar{q}\sin(\theta), \quad \theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right],$$

where $\bar{q}$ is a fixed parameter determining the upper bound on the absolute value of the uncertain parameters for any value of θ.

To perform a robust deterministic design, we need to assume two *independent* uncertainties $|q_1| \leq \bar{q}$ and $|q_2| \leq \bar{q}$, without explicitly considering their (nonlinear) dependence on the common parameter θ. With this assumption, the parameterization becomes convex and we can run the procedure in Section 3.2. Such an overparameterization of the uncertainty results in a larger uncertainty set: see the grey-shaded box in Figure 1 as compared to the real set indicated by the black solid line. Clearly, the corresponding design is convex but more conservative.

For this simple example, one can compute by hand the robustly stabilizing OF gains and these are exactly such that $\frac{\bar{q}-0.5}{\bar{q}+0.5} < F < 0$. Moreover for $\bar{q} > 0.36$ there is no state-feedback that may quadratically stabilize the system (stability may not be proved with a common to all uncertainties Lyapunov matrix). For $\bar{q} > 0.36$ only parameter-dependent Lyapunov certificates may be used to prove robust stability of the closed-loop.

The heuristic algorithms are applied to the example for various values of $\bar{q}$. The results are given in Tables 2, 3 and 4. K_0 is the state feedback (SF) gain found at the initialization Phase 1. $\bar{k}$ is the number of iterations in Phase 2 of the algorithm. The maximal number of iterations is set to 10, therefore $\bar{k} = 10$ indicates that Phase 2 terminated because this maximal number is reached. Otherwise the iterations stop when $1 - \lambda < 10^{-7}$. The column $1 - \lambda_{\bar{k}}$ shows how close λ is to the value 1 when the iterations stop. The column $F_{\bar{k}}$ gives the value of the OF gain when the algorithm stops. The column named 'Validation' indicates whether the validation Phase 3 is successful (OK) or not (-). The computation time is the total time including initial-

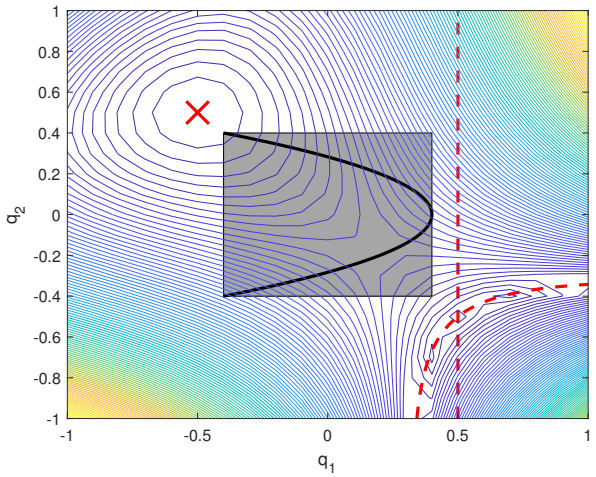


Fig. 1 Deterministic uncertainty set (grey-shaded box) and real uncertainty set (black solid line) for $\bar{q} = 0.4$, namely the limit value for a successful deterministic OF design. In the background, the contour plot of the absolute value of the determinant of the reachability matrix of the system in [7] is also illustrated. This is null corresponding to the red cross and along the red curve in the right bottom part of the figure. Also, a stabilizing SOF cannot be found if the system is unobservable, that is for all points on the dashed line $q_1 = 0.5$.

ization and termination phases. For each tested value of $\bar{q}$ the theoretical limit of the stabilizing output feedback gains is recalled.

The algorithms are applied for three cases. The first one (results of Table 2) corresponds to the purely deterministic case described in Section 3. The last one (results of Table 4) corresponds to the purely probabilistic case described in Section 4. The probabilistic designs are done considering $\bar{r}_{\mathrm{d}} = 10$ samples for the iterations in Phase 2. The number of samples in Phases 1 and 3 is set to $\bar{r} = 450$. This value is computed from (20) with $\varepsilon = 0.05$ and $\delta = 10^{-4}$. The results of Table 3 correspond to an intermediate case where the probabilistic approach is applied to the deterministic model, that is when considering only the four vertices as samples. This case does not allow to conclude robustness (no deterministic nor probabilistic robustness can be deduced) but corresponds to simultaneous stabilization of the four vertices. The goal is to illustrate the degrees of freedom obtained when relaxing S_1 from being common to all vertices/samples.

Some conclusions about these results:

- The deterministic (OF) design is successful up to $\bar{q} = 0.4$ but when looking at the results for larger bounds on the uncertainties it seems that the algorithm is close to converging to valid values. One reason for this non convergence is due to the heuristic nature of the algorithm. The other possible reason is that there might not be any solution to the BMIs for $\bar{q} > 0.4$. The true bound on $\bar{q}$ for the

$\bar{q}$	$\frac{\bar{q}-0.5}{\bar{q}+0.5}$	K_0	$\bar{k}$	$1-\lambda_{\bar{k}}$	$F_{\bar{k}}$	Validation	time
0.1	-0.6667	$\begin{bmatrix} -0.6589 & -0.9263 \end{bmatrix}$	2	$7.5\cdot10^{-9}$	-0.5082	OK	5.3s
0.2	-0.4286	$\begin{bmatrix} -0.4012 & -1.0939 \end{bmatrix}$	3	$2.9\cdot10^{-9}$	-0.3166	OK	8.6s
0.3	-0.2500	$\begin{bmatrix} -0.1776 & -1.0576 \end{bmatrix}$	3	$1.4\cdot10^{-8}$	-0.1185	OK	8.8s
0.4	-0.1111	$\begin{bmatrix} -0.0471 & -0.9240 \end{bmatrix}$	7	$3.5\cdot10^{-8}$	-0.1104	OK	26.7s
0.41	-0.0989	$\begin{bmatrix} -0.0427 & -0.9057 \end{bmatrix}$	10	$8.4\cdot10^{-7}$	-0.1030	-	45.0s
0.42	-0.0870	$\begin{bmatrix} -0.0387 & -0.8886 \end{bmatrix}$	10	$2.7\cdot10^{-6}$	-0.0928	-	53.3s
0.43	-0.0753	$\begin{bmatrix} -0.0345 & -0.8728 \end{bmatrix}$	10	$7.7\cdot10^{-6}$	-0.0827	-	47.0s
0.44	-0.0638	$\begin{bmatrix} -0.0319 & -0.8654 \end{bmatrix}$	10	$1.5\cdot10^{-5}$	-0.0712	-	54.0s
0.45	-0.0526	$\begin{bmatrix} -0.0337 & -0.8739 \end{bmatrix}$	10	$3.8\cdot10^{-5}$	-0.0623	-	50.1s
0.455	-0.0471	$\begin{bmatrix} -0.0353 & -0.8758 \end{bmatrix}$	10	$8.3\cdot10^{-5}$	-0.0546	-	49.2s
0.46	-0.0417	-					1.2s

Table 2 Numerical results for the deterministic approach.

$\bar{q}$	$\frac{\bar{q}-0.5}{\bar{q}+0.5}$	K_0	$\bar{k}$	$1-\lambda_{\bar{k}}$	$F_{\bar{k}}$	Validation	time
0.1	-0.6667	$\begin{bmatrix} -0.6763 & -0.8962 \end{bmatrix}$	2	$9.4\cdot10^{-9}$	-0.5409	OK	4.5s
0.2	-0.4286	$\begin{bmatrix} -0.4403 & -0.9932 \end{bmatrix}$	3	$3.4\cdot10^{-9}$	-0.3657	OK	8.4s
0.3	-0.2500	$\begin{bmatrix} -0.2174 & -0.9742 \end{bmatrix}$	4	$1.1\cdot10^{-8}$	-0.2371	OK	11.7s
0.4	-0.1111	$\begin{bmatrix} -0.0569 & -0.8682 \end{bmatrix}$	4	$1.8\cdot10^{-9}$	-0.1012	OK	12.9s
0.41	-0.0989	$\begin{bmatrix} -0.0522 & -0.8492 \end{bmatrix}$	4	$1.2\cdot10^{-8}$	-0.0927	OK	17.0s
0.42	-0.0870	$\begin{bmatrix} -0.0480 & -0.8297 \end{bmatrix}$	3	$1.6\cdot10^{-8}$	-0.0792	OK	9.4s
0.43	-0.0753	$\begin{bmatrix} -0.0443 & -0.8095 \end{bmatrix}$	2	$2.9\cdot10^{-8}$	-0.0638	OK	5.1s
0.44	-0.0638	$\begin{bmatrix} -0.0406 & -0.7889 \end{bmatrix}$	5	$1.9\cdot10^{-9}$	-0.0554	OK	22.2s
0.45	-0.0526	$\begin{bmatrix} -0.0392 & -0.7867 \end{bmatrix}$	10	$5.6\cdot10^{-6}$	0.0950	-	50.8s
0.455	-0.0471	$\begin{bmatrix} -0.0413 & -0.7968 \end{bmatrix}$	5	$1.5\cdot10^{-8}$	-0.0337	OK	22.4s
0.46	-0.0417	-					1.2s

Table 3 Numerical results for the probabilistic method applied to the deterministic model.

existence of a stabilizing output feedback gain is $\bar{q} < 0.5$, but we cannot say that this bound can be approached by the proposed conservative BMI conditions.

- When relaxing the constraints on having common variables S_1, the (OF) goal is almost always attained whatever $\bar{q} \leq 0.455$. The fact that it fails for $\bar{q} = 0.45$ can be due to numerical errors at some stage of the iteration, or because the heuristic fails by going in an inappropriate direction. The improvements when comparing Tables 2 and 3 illustrate the potential reduction of conservatism that can be achieved by probabilistic methods.
- The approach allows achieving the robustly (SF) goal up to $\bar{q} = 0.455$. We know that such a goal cannot be achieved when imposing common Lyapunov matrices to all uncertainties. This illustrates the fact that the new LMIs of the initialization Phase 1 have quite some potential for the robust state-feedback design problem that remains open.
- The probabilistic approach allows going further in terms of the (SF) goal. This is not surprising since, compared to the deterministic approach, there is some (small) tolerance on stability violation. Typically, during Phase 1 of the probabilistic approach, the extremal values of the uncertainties (which happen in this example to be the worst case values) have low probability to be drawn. Phase

$\bar{q}$	$\frac{\bar{q}-0.5}{\bar{q}+0.5}$	K_0	$\bar{k}$	$1-\lambda_{\bar{k}}$	$F_{\bar{k}}$	Validation	time
0.1	-0.6667	$\begin{bmatrix}-0.7028 & -0.8958\end{bmatrix}$	2	$5.2\cdot10^{-9}$	-0.6125	OK	31.2s
0.2	-0.4286	$\begin{bmatrix}-0.4885 & -1.0711\end{bmatrix}$	3	$4.0\cdot10^{-9}$	-0.4047	OK	43.1s
0.3	-0.2500	$\begin{bmatrix}-0.2597 & -1.1284\end{bmatrix}$	4	$1.1\cdot10^{-8}$	-0.2480	OK	53.4s
0.4	-0.1111	$\begin{bmatrix}-0.0855 & -1.0275\end{bmatrix}$	6	$2.8\cdot10^{-8}$	-0.1296	-	79.0s
0.4	-0.1111	$\begin{bmatrix}-0.0750 & -1.0250\end{bmatrix}$	3	$2.7\cdot10^{-8}$	-0.1501	-	43.0s
0.41	-0.0989	$\begin{bmatrix}-0.0619 & -1.0026\end{bmatrix}$	4	$2.8\cdot10^{-8}$	-0.1275	-	55.2s
0.42	-0.0870	$\begin{bmatrix}-0.0464 & -1.0069\end{bmatrix}$	3	$5.0\cdot10^{-9}$	-0.0547	-	37.5s
0.43	-0.0753	$\begin{bmatrix}-0.0365 & -0.9873\end{bmatrix}$	2	$1.5\cdot10^{-8}$	-0.0942	-	35.0s
0.44	-0.0638	$\begin{bmatrix}-0.0241 & -0.9828\end{bmatrix}$	2	$2.7\cdot10^{-8}$	-0.0619	-	52.0s
0.45	-0.0526	$\begin{bmatrix}-0.0109 & -0.9624\end{bmatrix}$	3	$4.7\cdot10^{-8}$	-0.0293	-	42.25s
0.455	-0.0471	$\begin{bmatrix}-0.0052 & -0.9607\end{bmatrix}$	2	$1.2\cdot10^{-8}$	-0.0173	-	38.2s
0.46	-0.0417	$\begin{bmatrix}-0.0045 & -0.9433\end{bmatrix}$	2	$3.6\cdot10^{-8}$	-0.0174	-	39.1s
0.465	-0.0363	$\begin{bmatrix}-0.0019 & -0.9434\end{bmatrix}$	3	$2.1\cdot10^{-8}$	-0.0027	-	54.8s
0.47	-0.0309	$\begin{bmatrix}0.0006 & -0.9293\end{bmatrix}$	-	num pb		-	88.6s
0.48	-0.0204	$\begin{bmatrix}0.0053 & -0.9237\end{bmatrix}$	5	$6.4\cdot10^{-8}$	$8.98\cdot10^{-5}$	-	84.9s
0.49	-0.0101	$\begin{bmatrix}0.0010 & -0.8937\end{bmatrix}$	-	num pb		-	54.4s
0.495	-0.0050	$\begin{bmatrix}0.0024 & -0.8724\end{bmatrix}$	4	$1.2\cdot10^{-8}$	-0.0127	-	80.0s
0.50	0	-					12.4s
0.505	no	-					15.2s

Table 4 Numerical results for the randomized approach.

1 is hence applied considering a large scenario of samples ($N = 450$ in our case) but there might be no value close the critical samples $|q_1| = |q_2| = \bar{q}$. The relaxed (SF) goal is hence feasible (in probability).

- Since the approach is dependent on the samples that have been drawn, there is no possible monotonicity in the results. This is illustrated for the case where we applied the method for two different scenarios, and the results are inevitably different. For $\bar{q}$ close to 0.5 the problem becomes very constrained and for some cases (depending on the samples) we noticed numerical errors in the algorithm (the LMIs become unfeasible during the iterations although the sequence λ_k is proved to be theoretically decreasing monotonously).
- The iterations of Phase 2 are done on a sub-set of the scenario. When it converges, and this is quite often the case as illustrated by the value of $1-\lambda$, the conclusion is that we have good candidates for stabilization of the few systems ($\bar{r}_{\mathrm{d}} = 10$) used during this phase. There is no guarantee of robustness, not even probabilistic. This is the reason why the termination Phase 3 usually fails (except for $\bar{q} \leq 0.4$). It fails even-though the computed value of the (OF) gain actually does solve the problem. This result illustrates the fact that, even for the scenario situation, the BMIs are conservative. Conservatism comes from the fact that the S-variable S_2 is imposed to be the same for all samples.

The computations were done on a MacBook Pro 2.9 GHz Intel Core i5 with Matlab2016b. The LMIs were coded using YALMIP (release R20141030) by [15] and solved using SDPT3 (version 4.0) by [21].

6 Conclusions

In this chapter we proposed a new robust static output feedback design method stemming from an S-variable description of the set of feasible solutions. The proposed approach leads to both deterministic and probabilistic designs, the first one providing worst-case guarantees (pessimistic approach) requiring polytopic uncertainty sets and specific multipliers structures, and the second one removing these assumptions at the cost of extra computational burden and probabilistic guarantees (optimistic approach). The derived conditions are coded as bilinear matrix inequalities for both cases, so that a heuristic procedure is proposed for their solution. Interestingly, the heuristic approach starts from solving a robustly stabilizing state injection gain and a robust state feedback stabilizer, which is then refined into a robust output feedback stabilizer and a robustly stabilizing output injection gain. Numerical tests on some examples taken from the literature have shown good performance of the heuristic, first in finding nominally stabilizing output feedback gains, and then addressing robust stabilization problems.

Future works involve better characterizing the properties of the suggested heuristic algorithms, possibly providing more sophisticated solution methods for the proposed BMI problems, in addition to further characterizations of the merits of the proposed approach on relevant case studies.

References

1. T. Alamo, R. Tempo, A. Luque, and D.R. Ramirez. Randomized methods for design of uncertain systems: Sample complexity and sequential algorithms. *Automatica*, 52:160–172, 2015.
2. D. Arzelier, G. Deaconu, S. Gumussoy, and D. Henrion. H_2 for HIFOO. In *International Conference on Control and Optimization with Industrial Applications*, Ankara, Turkey, 2011.
3. G. Calafiore and M.C. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006.
4. G. Calafiore, F. Dabbene, and R. Tempo. Research on probabilistic methods for control system design. *Automatica*, 47:1279–1293, 2011.
5. G.C. Calafiore. Random convex programs. *SIAM Journal on Optimization*, 20(6):3427–3464, 2010.
6. M.C. Campi and S. Garatti. The exact feasibility of randomized solutions of robust convex programs. *SIAM Journal on Optimization*, 19(3):1211–1230, 2008.
7. P Colaneri, J.C Geromel, and A. Locatelli. *Control Theory and Design: an RH_2 and RH_∞ viewpoint.* Academic Press, 1997.
8. J. Dong and G-H. Yang. Robust static output feedback control synthesis for linear continuous systems with polytopic uncertainties. *Automatica*, 49:1821–1829, 2013.
9. Y. Ebihara, D. Peaucelle, and D. Arzelier. *S-Variable Approach to LMI-based Robust Control*. Communications and Control Engineering. Springer, 2015.
10. L. El Ghaoui, F. Oustry, and M. AitRami. A cone complementary linearization algorithm for static output-feedback and related problems. *IEEE Transactions on Automatic Control*, 42:1171–1176, 1997.
11. J. Fiala, M. Kocvara, and M. Stingl. PENLAB - a solver for nonlinear semidefinite programming. Technical report, 2013.

12. S. Formentin, F. Dabbene, R. Tempo, L. Zaccarian, and S.M. Savaresi. Robust linear static anti-windup with probabilistic certificates. *IEEE Transactions on Automatic Control*, 62(4):1575–1589, April 2017.
13. S. Gumussoy, D. Henrion, M. Millstone, and M.L. Overton. Multiobjective robust control with HIFOO 2.0. In *IFAC Symposium on Robust Control Design*, Haifa, Israel, 2009.
14. D. Henrion, J. Lofberg, M. Kocvara, and M. Stingl. Solving polynomial static output feedback problems with PENBMI. In *44th IEEE Conference on Decision and Control (CDC) and the European Control Conference (ECC),*, pages 7581–7586, 2005.
15. J. Löfberg. YALMIP, 2014.
16. Toker O and H. Ozbay. On the NP-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback. In *IEEE American Control Conference*, pages 2525–2526), Seattle, 1995.
17. Y. Oishi. Probabilistic design of a robust controller using a parameter-dependent Lyapunov function. In Giuseppe Calafiore and Fabrizio Dabbene, editors, *Probabilistic and Randomized Methods for Design under Uncertainty*, pages 303–316. Springer London, 2006.
18. M.S. Sadabadi and D. Peaucelle. From static output feedback to structured robust static output feedback: A survey. *Annual Reviews in Control*, 42:11–26, 2016.
19. V.L. Syrmos, C.T. Abdallah, P. Dorato, and K. Grigoriadis. Static output feedback–a survey. *Automatica*, 33(2):125–137, 1997.
20. R. Tempo, G.C. Calafiore, and F. Dabbene. *Randomized Algorithms for Analysis and Control of Uncertain Systems: With Applications*. Springer, 2nd edition, 2013.
21. T.C. Toh, M.J. Todd, and R.H. Tutuncu. SDPT3 - a MATLAB software package for semidefinite programming. *Optimization Methods and Software*, 11:545–581, 1999.