Counter-example guided inductive synthesis of control Lyapunov functions for uncertain systems

Daniele Masti, Filippo Fabiani, Giorgio Gnecco, and Alberto Bemporad

Abstract—We propose a counter-example guided inductive synthesis (CEGIS) scheme for the design of control Lyapunov functions and associated state-feedback controllers for linear systems affected by parametric uncertainty with arbitrary shape. In the CEGIS framework, a learner iteratively proposes a candidate control Lyapunov function and a tailored controller by solving a linear matrix inequality (LMI) feasibility problem, while a verifier either falsifies the current candidate by producing a counter-example to be considered at the next iteration, or it certifies that the tentative control Lyapunov function actually enjoys such feature. We investigate the Lipschitz continuity of the objective function of the global optimization problem solved by the verifier, which is key to establish the convergence of our method in a finite number of iterations. Numerical simulations confirm the effectiveness of the proposed approach.

Index Terms—Uncertain systems, Computer-aided control design, Lyapunov methods.

I. INTRODUCTION

The design of effective control laws guaranteeing closed-loop stability and performance is the quintessential problem in control theory. Among the avenues to do so, Lyapunov methods [1] have traditionally attracted interest in the systems-and-control community in view of their flexibility and power. Indeed, several approaches have been developed to synthesize different forms of Lyapunov functions (e.g., quadratic/quartic, piecewise affine/quadratic) for various types of systems, a task that is often accomplished through convex optimization [2]–[5]), although other approaches have been explored based on, e.g., neural networks [6] or formal verification [7]. A recent trend, instead, looks at the design of (control) Lyapunov functions using counter-example guided sampling-based approaches [8], [9], to possibly take advantage of the presence of baseline controllers already in place [10], [11]. Among these, the *counter-example guided inductive synthesis* (CEGIS) family of techniques has found widespread use and success in recent years [12]–[18]. The main idea behind CEGIS consists of designing a function $f : \mathbb{R}^n \to \mathbb{R}$ by exploiting the information contained in a growing dataset of (counter-)examples (hereinafter called $\mathcal{C}_i$ at step i), while making sure that f belongs to some given set of functions Θ. To this end, CEGIS relies on an iterative adversarial procedure with two actors, a *learner* and a *verifier*. In summary, at each iteration $i \in \mathbb{N}^+$:

The authors are with the IMT School for Advanced Studies Lucca, Piazza San Francesco 19, 55100, Lucca, Italy (`{daniele.masti, filippo.fabiani, giorgio.gnecco, alberto.bemporad}@imtlucca.it`). The first author was supported in part by Consorzio Interuniversitario Nazionale per l'Informatica (CINI) through Research Project under Grant CA 01/2021 a.i. 2. The first and third authors are members of INdAM-GNAMPA.

1) The learner takes the dataset $\mathcal{C}_i$ as input and synthesizes a function f_i from a hypothesis space $\mathcal{H}$, according to some criterion, or establishes that such a function can not be synthesized;
2) The selected function f_i is then passed to the verifier, which verifies if $f_i \in \Theta := \{f : \mathbb{R}^n \to \mathbb{R} \mid g(f(z)) \leq 0, \text{ for all } z \in \mathcal{Z}\}$, for some $g : \mathbb{R} \to \mathbb{R}$ and set $\mathcal{Z} \subseteq \mathbb{R}^n$ over which the variable z takes values. The verifier can hence take two different conclusions:
   a) It finds a *counter-example* $z_{i+1} \in \mathcal{Z}$ so that $g(f_i(z_{i+1})) > 0$. In such a case $\mathcal{C}_{i+1} \leftarrow \mathcal{C}_i \cup \{z_{i+1}\}$, and a new iteration step follows;
   b) It certifies that no counter-example exists, yielding the positive conclusion of the procedure.

While CEGIS schemes have been successfully employed to design disparate types of (control) Lyapunov functions for different classes of systems [12]–[17], little attention has been given to how such methods can be used to extend the capabilities of traditional robust control approaches, both in terms of enlarging their applicability and, more prominently, of reducing their computational requirements.

In this paper we make an attempt to fill this gap. Specifically, we start from the classical machinery proposed in [3], and leverage a novel Lipschitz-based verification approach, rooted in perturbation of linear operators [19], to efficiently design control Lyapunov functions, and associated feedback controllers, for linear systems whose parameters are only known to belong to an arbitrary compact set. This is achieved without introducing conservativeness due to the adoption of a convex polytopic outer-approximation of the uncertainty set. Indeed, due to computational reasons, standard methods use it to approximate the convex hull of such a set (which may have a complex shape), possibly based on a few vertices. In summary, we make the following contributions:

- We design a novel CEGIS-based method, in which the learner solves an LMI feasibility problem, while the verifier a Lipschitz continuous nonconvex one;
- We get Lipschitz continuity of the objective function of the verifier's task, which is key to resort on efficient solvers tailored for global Lipschitz optimization;
- We prove that our algorithm converges in finite-time.

In §II we formalize the control problem and recall fundamentals of Lyapunov stability, while in §III we describe the learner' and verifier's tasks, characterize the technical properties of the problems they are asked to solve, and establish the finite-time convergence of our CEGIS-based method. Computational aspects are, instead, analyzed in §IV, while

numerical examples are finally given in §V.

II. Problem definition and preliminaries

In this paper we aim at designing control Lyapunov functions and related state-feedback controllers for discrete-time uncertain linear systems of the form:

$$\Sigma : x(k+1) = A(k)x(k) + B(k)u(k), \qquad (1)$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ denote the state and control vectors, respectively, and x is assumed fully available. The dynamical matrices are so that $(A(k), B(k)) \in \Omega \subseteq \mathcal{A} \times \mathcal{B}$, for all $k \geq 0$, where $\mathcal{A} \subseteq \mathbb{R}^{n \times n}$ and $\mathcal{B} \subseteq \mathbb{R}^{n \times m}$. Next, we introduce the main assumption adopted throughout:

Standing Assumption 1. Ω *is a nonempty compact set.* □

As made evident later in the paper, a distinct feature of our approach will be the possibility to efficiently handle uncertain systems as in (1) i) without necessarily imposing any convex polytopic structure to characterize the uncertainty, and ii) sampling a (possibly large) number of matrices in $\mathcal{A}$ and $\mathcal{B}$. Even though our CEGIS scheme will require the availability of Ω to solve the verifier task, we remark here that it makes no (possibly conservative) outer-approximation of such a set.

We now recall some key notions on Lyapunov stability for discrete-time linear uncertain systems as Σ in (1). In particular, such a system is said to be *stable* in some set $\mathcal{X} \subseteq \mathbb{R}^n$, with $0 \in \mathcal{X}$, if there exist a function $V : \mathbb{R}^n \to \mathbb{R}$ and some control law $u : \mathbb{R}^n \to \mathbb{R}^m$ so that [1, Ch. 2.3]:

$$
\begin{aligned}
V(x) > 0, \ &\forall x \in \mathcal{X} \setminus \{0\}, \\
V(0) = 0 \iff &x = 0, \qquad (2) \\
V(x^+) - V(x) \leq 0, \ &\forall (x, x^+),
\end{aligned}
$$

where (x, x^+) denotes a generic state-successor pair, i.e., $x^+ = Ax + Bu(x)$, for any possible $(A, B) \in \Omega$. In this case, $V(\cdot)$ represents a (common) *control Lyapunov function* with associated state-feedback controller $u(\cdot)$, and ensures closed-loop stability in $\mathcal{X}$ to the trajectories originating from Σ in (1) with matrices $(A(k), B(k))$ varying in Ω. When the third condition holds as strict inequality, one may additionally recover convergence to the origin of the closed-loop trajectories.

To preserve mathematical tractability while designing functions and controllers so that conditions (2) are met, a common choice is to rely on a quadratic form for $V(\cdot)$, i.e., $V(x) = x^\top P x$, with $P \in \mathbb{S}^n_{\succ 0}$, and a linear one for the controller, $u(x) = Kx$, for some gain matrix $K \in \mathbb{R}^{m \times n}$. Such a linear-quadratic parametrization for $V(\cdot)$ and $u(\cdot)$ is the one that will also be employed in the remainder.

We finally recall some facts regarding the well-known convex polytopic case, namely when $\Omega = \text{conv}(\{(A_1, B_1), \ldots, (A_M, B_M)\})$ with M given generator pairs of matrices, that will be instrumental for the learner task. In that case, a control Lyapunov function with associated robust control law for Σ can be designed by solving an LMI feasibility problem [20, Ch. 7.3.3.2]:

$$
\begin{cases}
\min_{X,W} & 0 \\
\text{s.t.} & \begin{bmatrix} X & XA_j^\top + W^\top B_j^\top \\ \star & X \end{bmatrix} \succ 0, \ j = 1, \ldots, M.
\end{cases} \qquad (3)
$$

Given a solution pair $(X^\star, W^\star)$ to (3) one hence obtains:
- A linear controller robustly stabilizing Σ to the origin:

$$u(x) = Kx = W^\star(X^\star)^{-1}x; \qquad (4)$$

- A control Lyapunov function $V(x) = x^\top P x$, $P = (X^\star)^{-1}$, for the system (1) with linear controller (4).

III. Control Lyapunov functions design through counter-examples

A. The learner task

The considerations above regarding (3) directly apply to the CEGIS-based framework we are about to introduce. In fact, let $\mathcal{C}_i$ be the set of *counter-examples* (or simply *samples*) of dynamical matrices $(\hat{A}_j, \hat{B}_j)$ considered at the i-th iteration, $j = 1, \ldots, i$. We rewrite the LMI constraints in (3) to account for the elements of $\mathcal{C}_i$ as follows:

$$
\begin{cases}
\min_{X,W} & 0 \\
\text{s.t.} & \begin{bmatrix} X & X\hat{A}_h^\top + W^\top \hat{B}_h^\top \\ \star & X \end{bmatrix} \succcurlyeq \varepsilon I, \ h = 1, \ldots, M_i, \quad (5) \\
& X \preccurlyeq \eta I, \ W \in \mathcal{W},
\end{cases}
$$

where, in this case, the index h allows us to enumerate the vertex matrices of $\mathcal{C}_i$, with $M_i := |\text{vert}(\text{conv}(\mathcal{C}_i))|$, where clearly $M_i \leq i$ and, for any $h = 1, \ldots, M_i$, $(\hat{A}_h, \hat{B}_h) \in \text{vert}(\text{conv}(\mathcal{C}_i)) \subseteq \Omega$. The hyperparameters $\eta \geq \varepsilon > 0$ can be chosen arbitrarily and are meant to upper (respectively, lower) bound the largest (smallest) eigenvalue of matrix X. Besides imposing some sort of contraction through strict positive-definiteness with ε, we will see that η and ε are hyperparameters of our CEGIS method that strongly affect its performance. Finally, we assume $\mathcal{W} \subset \mathbb{R}^{m \times n}$ to be a nonempty compact set for technical reasons clarified later, whereas we also assume the convexity of $\mathcal{W}$ in order to keep the problem (5) convex.

At each iteration i of our CEGIS scheme, the learner accomplishes two tasks: first, given the current set $\mathcal{C}_i$ of samples of matrix pairs, it identifies those counter-examples that are vertices of its convex hull, for a total of M_i vertices; leveraging this information, it then solves the resulting convex problem (5) with LMI constraints.

Note that the same discussion made in §II also applies here. Specifically, solving (5) produces a linear gain K_i as in (4) that makes $V_i = x^\top(X^\star)^{-1}x$ a control Lyapunov function for the surrogate convex polytopic system identified by generator matrices $(\hat{A}_h, \hat{B}_h) \in \text{vert}(\text{conv}(\mathcal{C}_i))$. On the other hand, since (5) heavily depends on the choice of hyperparameters $\eta \geq \varepsilon > 0$, in case the synthesis process fails, this implies that the parametrization adopted does not allow to design a control Lyapunov function with associated controller for Σ, with the assigned (η, ε). We will discuss possible solutions to this issue later in §III-C.

B. The verifier task

The verifier aims at certifying that a given candidate function and associated controller, designed by exploiting samples in $\mathcal{C}_i$, define a control Lyapunov function for Σ with

$(A(k), B(k)) \in \Omega$. Thus, by specializing the third condition in (2) to a quadratic function and linear controller, we get:

$$V(x^+) - V(x) = (x^+)^\top P x^+ - x^\top P x$$
$$= x^\top ((A^{\mathrm{cl}})^\top P A^{\mathrm{cl}} - P) x \le 0, \ \forall x \in \mathcal{X}, \ (A, B) \in \Omega, \quad (6)$$

where $A^{\mathrm{cl}} := A + BK$. In particular, for a given P and gain matrix K, condition (6) is verified for all $x \in \mathcal{X}$ whenever $(A^{\mathrm{cl}})^\top P A^{\mathrm{cl}} - P$ is negative semi-definite for all $(A, B) \in \Omega$. Thus, fixing $(A, B) \in \Omega$ and applying the Schur complement to (6) yields the symmetric matrix $\Xi(A^{\mathrm{cl}}) := \begin{bmatrix} P & (A^{\mathrm{cl}})^\top P \\ \star & P \end{bmatrix}$, whose eigenvalues are therefore all real. Hence, checking whether a given matrix $P = P^\top \succ 0$ produces a control Lyapunov function for Σ is equivalent to checking non-negativity of the optimal value $\lambda^\star$ of the following optimization problem:

$$\lambda^\star = \min_{(A,B) \in \Omega} \lambda_{\min}(\Xi(A^{\mathrm{cl}})), \quad (7)$$

where $\lambda_{\min}(\Xi(\cdot))$ denotes the smallest eigenvalue of $\Xi(\cdot)$.

Therefore, at the i-th iteration of the procedure, the verifier takes the matrices $P_i = (X^\star)^{-1}$ and $K_i = W^\star(X^\star)^{-1}$ proposed by the learner solving (5), defines $A_i^{\mathrm{cl}} := A + BK_i$, and $\Xi(A_i^{\mathrm{cl}}) = \begin{bmatrix} P_i & (A_i^{\mathrm{cl}})^\top P_i \\ \star & P_i \end{bmatrix}$ as in the definition of $\Xi(A^{\mathrm{cl}})$, where the subscript i makes explicit the dependence on the learner tentative solution, and finally solves (7). If $\lambda^\star < 0$, then a minimizer pair $(A^\star, B^\star) = (\hat{A}_{i+1}, \hat{B}_{i+1})$ is taken as a counter-example. Otherwise, no counter-example exists.

Note that, however, (7) amounts to solving a generally non-convex optimization problem that requires the availability of the set Ω. Moreover, a global minimum must be attained to prove that no counter-example exists. On the other hand, it also features several properties that make finding a globally optimal solution an affordable task. To see this, we need the following result borrowed from matrix perturbation theory:

Lemma 1 ([21]). *Consider two symmetric matrices K and L of the same dimension. Then, it holds that $|\lambda_{\min}(K) - \lambda_{\min}(K + L)| \le \|L\|_{\mathrm{op}}$, where $\|\cdot\|_{\mathrm{op}}$ is the operator norm of a matrix, induced by the l_2-norm.* $\square$

In other words, the minimum eigenvalue $\lambda_{\min}(\cdot)$ of a symmetric matrix is a Lipschitz continuous map of the elements of the matrix. This is a consequence of Weyl's inequalities [21].

It is recalled here that any linear operator between two finite-dimensional normed spaces is bounded (this applies, in particular, to the operator $\Xi(\cdot)$, which maps some A^{cl} to $\Xi(A^{\mathrm{cl}})$). Since bounded linear operators preserve Lipschitz continuity, we are then able to prove the following result:

Theorem 1. *Let $P = (X^\star)^{-1}$ and $K = W^\star(X^\star)^{-1}$ with $(X^\star, W^\star)$ be a solution to (5) with a given $\eta \ge \varepsilon > 0$. Then, there exists some constant $\ell = \ell(\varepsilon) \ge 0$ such that*

$$|\lambda_{\min}(\Xi(A^{\mathrm{cl}})) - \lambda_{\min}(\Xi(A^{\mathrm{cl}} + \Delta A^{\mathrm{cl}}))| \le \ell \|\Delta A^{\mathrm{cl}}\|_{\mathrm{op}},$$

with $A^{\mathrm{cl}} = A + BK$ for $(A, B) \in \Omega$ and perturbation ΔA^{cl} induced by some pair $(\Delta A, \Delta B)$. $\square$

Proof. One gets:

$$|\lambda_{\min}(\Xi(A^{\mathrm{cl}})) - \lambda_{\min}(\Xi(A^{\mathrm{cl}} + \Delta A^{\mathrm{cl}}))| \le \left\| \begin{bmatrix} 0 & (\Delta A^{\mathrm{cl}})^\top P \\ \star & 0 \end{bmatrix} \right\|_{\mathrm{op}}$$

where $\|\cdot\|_{\mathrm{op}}$-norm is the matrix norm induced by the l_2 one. This leads to:

$$\left\| \begin{bmatrix} 0 & (\Delta A^{\mathrm{cl}})^\top P \\ \star & 0 \end{bmatrix} \right\|_{\mathrm{op}} = \sup_{\|x\|_2=1} \left\| \begin{bmatrix} 0 & (\Delta A^{\mathrm{cl}})^\top P \\ \star & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right\|_2$$
$$= \sup_{\|x\|_2=1} \left\| \begin{bmatrix} (\Delta A^{\mathrm{cl}})^\top P x_2 \\ P^\top \Delta A^{\mathrm{cl}} x_1 \end{bmatrix} \right\|_2$$
$$= \sup_{\|x\|_2=1} \sqrt{\|(\Delta A^{\mathrm{cl}})^\top P\|_{\mathrm{op}}^2 \|x_2\|_2^2 + \|P^\top \Delta A^{\mathrm{cl}}\|_{\mathrm{op}}^2 \|x_1\|_2^2}$$
$$= \|(\Delta A^{\mathrm{cl}})^\top P\|_{\mathrm{op}} \le \|\Delta A^{\mathrm{cl}}\|_{\mathrm{op}} \|P\|_{\mathrm{op}} \le \varepsilon^{-1} \|\Delta A^{\mathrm{cl}}\|_{\mathrm{op}},$$

which follows from the fact that $P = P^\top \succ 0$ and $P = (X^\star)^{-1}$, for $X^\star$ solving (5). Thus, setting $\ell = \ell(\varepsilon) := \varepsilon^{-1}$ concludes the proof. ∎

Remark 1. *The bound in Theorem 1 can be expressed also in terms of a Lipschitz constant w.r.t. the pair (A, B), since $\|\Delta A^{\mathrm{cl}}\|_{\mathrm{op}} \le \max\{\|I\|_{\mathrm{op}}, \|K\|_{\mathrm{op}}\} \|\Delta A \ \Delta B\|_{\mathrm{op}}$ as $\Delta A^{\mathrm{cl}} = \Delta A + \Delta B K$. Thus, such a Lipschitz constant is bounded from above by $\varepsilon^{-1} \max\{1, \|K\|_{\mathrm{op}}\}$. A uniform upper bound, i.e., independent from K, could be also obtained by accounting for $K = W^\star(X^\star)^{-1}$, $(X^\star)^{-1} = P$, and $W^\star \in \mathcal{W}$, which is compact. In case B is not uncertain (i.e., $\Delta B = 0$), the upper bound on the Lipschitz constant w.r.t. A reduces to ε^{-1}.* $\square$

We note that Lemma 1 and Theorem 1 suggest that global optimization techniques, tailored for Lipschitz optimization, can be used to verify if a candidate function V_i is a Lyapunov function for Σ. We detail this crucial point in the next section.

C. The proposed CEGIS scheme

Algorithm 1 summarizes the main steps of the proposed CEGIS-based iterative scheme for the design of a control Lyapunov function, accompanied with a suitable linear state-feedback controller, for the linear uncertain system Σ in (1). In it, black-filled bullets refer to tasks performed by the learner, while the empty bullet to the one performed by the verifier.

Algorithm 1: CEGIS-Lyapunov learning method

Initialization: Set $\eta \ge \varepsilon > 0$, fill $\mathcal{C}_1 = \{(\hat{A}_1, \hat{B}_1)\}$ with some $(\hat{A}_1, \hat{B}_1) \in \Omega$.

Iteration $(i \in \mathbb{N}^+)$:
- Identify $\mathrm{vert}(\mathrm{conv}(\mathcal{C}_i))$
- Solve (5), set $P_i = (X^\star)^{-1}$, $K_i = W^\star(X^\star)^{-1}$
  If (5) `infeasible` **then** `exit`
- Solve (7) using a Lipschitz global optimization tool
  If $\lambda^\star < 0$ **then** $\mathcal{C}_{i+1} \leftarrow \mathcal{C}_i \cup \{(A^\star, B^\star)\}$, `repeat`
  If $\lambda^\star \ge 0$ **then** $V = x^\top P_i x$, $K = K_i$, `exit`

Note that Algorithm 1 may terminate in two different points with two possible outcomes. According to the discussion in §III-A, the learner has to solve an LMI feasibility problem, which is not only susceptible to the parametrization adopted and the quality of the samples the verifier produces, but also on the choice of the hyperparameters $\eta \ge \varepsilon > 0$. For these reasons, (5) may not be feasible, and this coincides with the notion of infeasibility mentioned in the next Theorem 2,

namely that a quadratic control Lyapunov function and associated linear controller do not exist for the uncertain system Σ in (1) with the assigned η and ε.

Note that, however, this does not necessarily imply that such a pair does not exist for the underlying system, for other choices of the hyperparameters. In fact, one might want to reinitialize the procedure with, e.g., a smaller ε, a larger η, or a larger $\mathcal{W}$. Moreover, in view of the finite-time convergence guarantees featured by Algorithm 1 (detailed in the following Theorem 2), the choice of $(\hat{A}_1, \hat{B}_1) \in \Omega$ may affect the number of steps needed, and also the specific control Lyapunov function generated, though not the actual outcome of Algorithm 1 (infeasibility or convergence are, indeed, independent from $(\hat{A}_1, \hat{B}_1)$).

Finally, in case the verifier simply terminates without founding any counter-example, Algorithm 1 has converged. Such a convergence property is itself a certification of the correctness of the control Lyapunov function and associated linear controller found. Otherwise, if some counter-example has been found, the set $\mathcal{C}_{i+1}$ includes the new pair $(A^\star, B^\star) = (\hat{A}_{i+1}, \hat{B}_{i+1})$ obtained from (7) and the iterative procedure continues, although not indefinitely, as shown next:

Theorem 2. *Let $\eta \geq \varepsilon > 0$, and some $(\hat{A}_1, \hat{B}_1) \in \Omega$ be given. Then, in a finite number of iterations Algorithm 1 either declares infeasibility or produces a pair $(\bar{P}, \bar{K})$ so that $V = x^\top \bar{P} x$ is a control Lyapunov function for the linear uncertain system Σ in (1) with linear controller $u = \bar{K} x$.* $\qquad\square$

Proof. We split the proof in two parts. First, we show that the verifier can only generate counter-examples through (7) outside the set of matrices considered by the learner to find a solution to (5). This fact is then exploited to prove that the verifier can not produce counter-examples indefinitely, therefore Algorithm 1 either returns infeasibility, or a solution pair $(\bar{P}, \bar{K})$ in a finite number of steps.

i) Assume that (5) is feasible at the generic i-th iteration, yielding a solution pair $(X^\star, W^\star)$. Then, by pre and post-multiplying by $\mathrm{diag}((X^\star)^{-1}, (X^\star)^{-1})$ both sides of the first set of LMI constraints in (5), which is always possible in view of $X^\star \succcurlyeq \varepsilon I$ with $\varepsilon > 0$, we obtain:

$$\begin{bmatrix} (X^\star)^{-1} & (\hat{A}_h + \hat{B}_h W^\star (X^\star)^{-1})^\top (X^\star)^{-1} \\ \star & (X^\star)^{-1} \end{bmatrix}$$
$$\succcurlyeq \varepsilon \mathrm{diag}((X^\star)^{-1}, (X^\star)^{-1})^2 \succcurlyeq \varepsilon \lambda_{\min}^2 ((X^\star)^{-1}) I$$
$$\succcurlyeq (\varepsilon / \lambda_{\max}^2 (X^\star)) I \succcurlyeq (\varepsilon / \eta^2) I,$$

for all $h = 1, \ldots, M_i$. This follows from (5), since $\lambda_{\max}(X^\star) \leq \eta$ in view of $X^\star \preccurlyeq \eta I$, and $\lambda_{\min}(X^\star) \geq \varepsilon$, yielding $\varepsilon / \eta^2 > 0$ as $\eta \geq \varepsilon > 0$. In view of $P_i = (X^\star)^{-1}$ and $K_i = W^\star (X^\star)^{-1}$, from the previous inequalities we have:

$$\Xi(A_{i,h}^{\mathrm{cl}}) \succcurlyeq (\varepsilon / \eta^2) I, \text{ for all } h = 1, \ldots, M_i, \qquad (8)$$

with $A_{i,h}^{\mathrm{cl}} = \hat{A}_h + \hat{B}_h K_i$, which implies $\lambda_{\min}(\Xi(A_{i,h}^{\mathrm{cl}})) \geq \varepsilon / \eta^2 > 0$ for all $h = 1, \ldots, M_i$. However, since each pair $(\hat{A}_h, \hat{B}_h) \in \mathrm{vert}(\mathrm{conv}(\mathcal{C}_i))$, standard arguments in robust control of polytopic systems [20, Chapter 7.3.3.2] imply that this relation not only holds for all vertex matrices, but also for any matrix in their convex hull, yielding $\lambda_{\min}(\Xi(A_i^{\mathrm{cl}})) \geq \varepsilon / \eta^2 > 0$

being A_i^{cl} now constructed by exploiting any pair of matrices $(A, B) \in \mathrm{conv}(\{(\hat{A}_h, \hat{B}_h)\}_{h=1}^{M_i}) = \mathrm{conv}(\mathcal{C}_i)$. Hence, the verifier can only find counter-examples through (7) strictly outside $\mathrm{conv}(\{(\hat{A}_h, \hat{B}_h)\}_{h=1}^{M_i})$, i.e., $(\hat{A}_{i+1}, \hat{B}_{i+1}) \in \Omega \setminus \mathrm{conv}(\mathcal{C}_i)$.

ii) Armed with the previous result, once the learner has obtained from the verifier a new counter-example $(\hat{A}_{i+1}, \hat{B}_{i+1}) \in \Omega \setminus \mathrm{conv}(\mathcal{C}_i)$, either it computes a new tentative solution pair (P_{i+1}, K_{i+1}) considering a larger pool of samples, or Algorithm 1 terminates at the second step of the $(i+1)$-th iteration. If the latter happens, then Algorithm 1 returns infeasibility in a finite number of iterations, as claimed. On the other hand, what could happen instead is that the verifier may produce counter-examples indefinitely. We will show in the following that this can not be the case.

In particular, we start by assuming that (5) is feasible for all the considered iterations $i \in \mathbb{N}^+$, otherwise Algorithm 1 declares infeasibility in a finite number of steps. Then, by relying on Theorem 1, at the generic i-th iteration we can quantify how far from $\mathrm{conv}(\mathcal{C}_i)$ the new counter-example $(\hat{A}_{i+1}, \hat{B}_{i+1})$ is. Precisely, it falls outside the next convex set of matrices generating closed-loop eigenvalues guaranteed to be at most (ε/η^2)-distant from those belonging to $\mathrm{conv}(\mathcal{C}_i)$:

$$\mathcal{S}_{\bar{A}_i^{\mathrm{cl}}} := \{ A^{\mathrm{cl}} \in \mathbb{R}^{n \times n} \mid A^{\mathrm{cl}} = A + BK_i, A \in \mathbb{R}^{n \times n},$$
$$B \in \mathbb{R}^{n \times m}, \|\bar{A}_i^{\mathrm{cl}} - A^{\mathrm{cl}}\|_{\mathrm{op}} \leq \varepsilon^2 / \eta^2,$$
$$\text{for all } \bar{A}_i^{\mathrm{cl}} = A_j + B_j K_i, \ (A_j, B_j) \in \mathrm{conv}(\mathcal{C}_i) \}.$$

For the sake of contradiction, assume that the verifier generates an infinite sequence $\{(\hat{A}_i, \hat{B}_i)\}_{i \in \mathbb{N}^+}$, with $(\hat{A}_i, \hat{B}_i) \in \Omega$. From the first part of the proof, we know that $\mathcal{C}_{i+1} \supset \mathcal{C}_i$, and therefore the infinite, monotonically increasing sequence of sets $\{\mathcal{C}_i\}_{i \in \mathbb{N}^+}$ admits the limit set $\bar{\mathcal{C}} = \cup_{i \in \mathbb{N}^+} \mathcal{C}_i$, which enables us to extract from $\{(\hat{A}_i, \hat{B}_i)\}_{i \in \mathbb{N}^+}$ a subsequence of counter-examples converging to some $(\bar{A}, \bar{B}) \in \mathrm{cl}(\mathrm{conv}(\bar{\mathcal{C}})) \cap \Omega$. Since the learner iteratively generates tentative pairs $\{(P_i, K_i)\}_{i \in \mathbb{N}^+}$, produced by solutions $\{(X^\star, W^\star)\}_{i \in \mathbb{N}^+}$ to (5) living in nonempty compact sets, we can additionally extract a convergent subsequence from $\{(P_i, K_i)\}_{i \in \mathbb{N}^+}$ with limit pair $(\bar{P}, \bar{K})$. Let us now consider a limiting "safe" set $\mathcal{S}_{\bar{A}^{\mathrm{cl}}}$, defined as:

$$\mathcal{S}_{\bar{A}^{\mathrm{cl}}} := \{ A^{\mathrm{cl}} \in \mathbb{R}^{n \times n} \mid A^{\mathrm{cl}} = A + B\bar{K}, A \in \mathbb{R}^{n \times n},$$
$$B \in \mathbb{R}^{n \times m}, \|\bar{A}^{\mathrm{cl}} - A^{\mathrm{cl}}\|_{\mathrm{op}} \leq \varepsilon^2 / \eta^2,$$
$$\text{for all } \bar{A}^{\mathrm{cl}} = A_j + B_j \bar{K}, \ (A_j, B_j) \in \mathrm{conv}(\bar{\mathcal{C}}) \}.$$

Leveraging this limiting set, we can conclude that not all the points of the convergent subsequence extracted from $\{(\hat{A}_i, \hat{B}_i)\}_{i \in \mathbb{N}^+}$ are counter-examples. In fact, by definition of limit, there exists some large enough index i_c associated to the underlying subsequence so that, by starting from $(\hat{A}_{i_c+1}, \hat{B}_{i_c+1})$, all the counter-examples generate closed-loop eigenvalues, obtained with the limit pair $(\bar{P}, \bar{K})$, strictly closer than (ε/η^2) from those generated by $(\hat{A}_{i_c}, \hat{B}_{i_c})$ with the same limit pair $(\bar{P}, \bar{K})$. This contradicts the first part of the proof, i.e., the assumption that they are counter-examples.

Hence, under the feasibility of (5), what can only happen is that the sequence $\{(\hat{A}_i, \hat{B}_i)\}$ converges to a pair $(\bar{A}, \bar{B}) \in \Omega$ in a finite number of iterations, say $N \in \mathbb{N}^+$, entailing that also $\{\mathcal{C}_i\}$ converges to a set $\mathcal{C}_N = \bar{\mathcal{C}}$ in N steps, with $\bar{\mathcal{C}}$ so that $\Omega \setminus \mathcal{S}_{\bar{A}^{\mathrm{cl}}} = \emptyset$. However, after N iterations, the verifier can

not produce any counter-example by evaluating the tentative pair $(P_N, K_N) = (\bar{P}, \bar{K})$ proposed by the learner.

Concluding, one of the two outcomes of Algorithm 1 is returned in a finite number of iterations, ending the proof. ∎

Remark 2. *Compared to ray-shooting [22], which builds a conservative outer-approximation of Ω through geometric considerations, our approach covers the whole set Ω of possible counter-examples by exploiting samples contained in it, along with the "safety" guarantees provided by the sets $\mathcal{S}_{\bar{A}_i^{cl}}$.* □

IV. COMPUTATIONAL ASPECTS

A. Tradeoff induced by η and ε

Besides strongly affecting the learner convex optimization problem (5), we note that η and ε, with $\eta \geq \varepsilon > 0$, assume a key role also in characterizing the performance of both Algorithm 1 and the verification task (7).

In fact, from Theorem 2, specifically the lower bound (8), it is evident that setting η as close as possible to ε, with this latter large enough, the convergence of Algorithm 1 may require a smaller number of steps, as this combination allows to "erode" (by means of the safe set $\mathcal{S}_{\bar{A}_i^{cl}}$) a larger portion of the set $\Omega \backslash \text{conv}(\mathcal{C}_i)$ at each iteration. In addition, this choice is also favourable to improve the performance of the technique employed to solve (7). Indeed, convergence of algorithms for Lipschitz-continuous global optimization problems, as the one in (7), is well-studied in the literature. As a general rule of thumb inferred, the smaller the Lipschitz constant, the better the guarantee on the performance of the underlying algorithm [23]. In view of Theorem 1, this hence suggests to choose larger values for the hypeparameter ε.

However, following the considerations above may undesirably complicate the learner's optimization problem, as the feasible set of (5) would shrink, thus making the design of quadratic control Lyapunov functions and associated linear controllers a challenging task. This naturally induces a tradeoff in the choice of the pair (η, ε), which shall hence be set according to the problem data at hand.

B. Speeding up the verification process

Sensitivity-based methods may lessen the computational requirements of the verification process. Indeed, while in general the partial derivative $\partial\lambda_i(\Xi(A^{cl}))/\partial A_{(h,k)}^{cl}$ of the i-th eigenvalue $\lambda_i(\Xi(A^{cl}))$ with respect to the element in position (h, k) of A^{cl} may be not defined when λ_i changes multiplicity, as noted in [19], such an happenstance is quite uncommon. Exploiting such an information to solve (7) by using a sensitivity-based optimization algorithm like an SQP or interior-point solver might be a valid heuristic to accelerate the process of finding a new counter-example – at the price of loosing, however, any convergence guarantee, since finding a local minimum in (7) may still lead to a counter-example, but if no counter-example can be found by a local solver, then nothing can be concluded. This motivates the next modification of the verifier's task in Algorithm 1. Given some number of trials $N_t \in \mathbb{N}^+$, with $\hat{\lambda}^\star$ an estimate of $\lambda^\star$ (already an overestimate, as (7) amounts to a minimization problem):

Set `counter` $= 0$
While `counter` $< N_t$
  ○ Solve (7) by a sensitivity-based method to get $\lambda^{\star,\text{sens}}$
    If $\lambda^{\star,\text{sens}} < 0$ **then** $\hat{\lambda}^\star \leftarrow \lambda^{\star,\text{sens}}$, `exit while`
    Else `counter` $\leftarrow$ `counter` $+ 1$, set different initial conditions for sensitivity-based method
End
If `counter` $== N_t$ **then** Solve (7) as in Algorithm 1

The same intuition also holds for the *refinement* of the approximate minimizer found by the global optimization procedure, as it is well-known that the *quality* of the solution provided by most global optimization tools is often subpar.

Remark 3. *We remark that numerical issues may produce only a rough estimate $\hat{\lambda}^\star$ of $\lambda^\star$. When $\hat{\lambda}^\star < 0$, this would be not harmful, because the verifier would still be able to generate a counter-example. However, for $\hat{\lambda}^\star \geq 0$, a non-valid Lyapunov function could be generated (as in [18, Example 8]), in case $\lambda^\star < 0$. To avoid this, one may replace the condition $\lambda^\star \geq 0$ in Algorithm 1 with, e.g., $\lambda^\star \geq \frac{\varepsilon}{2}$, and $\frac{\varepsilon}{2}$ larger than the desired precision. In this case, the main change in the proof of Theorem 2 would be the generation of smaller safe sets.* □

V. NUMERICAL EXAMPLES

We test our approach on two numerical examples. All experiments were carried out on a PC with an Intel core i7-1165G7, Ubuntu Linux and MATLAB R2022a. Solution of the optimization problem (5) was obtained by YALMIP [24], while the vertices of polytopes were computed using MPT3 [25] and YALMIP [24] As a Lipschitz global optimization solver, we adopted the `dDirect_GLce` algorithm from the DIRECTGO toolbox [26]. All the results of the CEGIS scheme were obtained in a few tens of seconds with $\varepsilon = 10^{-3}$, $\eta = 10$ and the absolute value of each entry of W smaller than 10^6.

A. Polytopic uncertainty set

We first consider an uncertain system Σ in (1) with four states and one input, where the matrix B is not uncertain and $B(k) = B = [0\ 0\ 0\ 1]^\top$, while each entry of the dynamical matrix A is subject – independently from the other entries – to an interval uncertainty, namely

$$A(k) \geq \begin{bmatrix} -0.6685 & -0.8709 & -0.2028 & -1.5547 \\ 1.1457 & -0.5898 & 0.5688 & 0.8496 \\ -0.7812 & -0.5754 & -0.8774 & -0.2501 \\ -1.1429 & 0.1730 & 0.7763 & 0.1618 \end{bmatrix},$$

$$A(k) \leq \begin{bmatrix} -0.6295 & -0.8202 & -0.1910 & -1.4641 \\ 1.2166 & -0.5555 & 0.6040 & 0.9022 \\ -0.7357 & -0.5419 & -0.8263 & -0.2355 \\ -1.0763 & 0.1837 & 0.8243 & 0.1718 \end{bmatrix},$$

for all $k \in \mathbb{N}$. Although such an uncertain system could be reliably stabilized using polytopic techniques such as the one presented in [3], this would require to solve an optimization problem with $2^{16} = 65536$ LMI constraints, each generated by taking independently either the lower or upper bound for each of the 16 entries of the matrix A. While theoretically possible, this would be hardly doable even with modern hardware.

By using the proposed CEGIS method, instead, we found a quadratic control Lyapunov function with P matrix:

$$\bar{P} = \begin{bmatrix} 162.4930 & 29.5126 & 82.0931 & 176.8625 \\ 29.5126 & 42.9150 & 21.4642 & 50.7574 \\ 82.0931 & 21.4642 & 58.6050 & 99.6742 \\ 176.8625 & 50.7574 & 99.6742 & 232.9383 \end{bmatrix}$$

accompanied with linear feedback policy: $\bar{K} = \begin{bmatrix} 1.7667 & 0.9014 & -0.3555 & 1.0089 \end{bmatrix}$. The soundness of the result was verified a-posteriori by checking the positive definiteness of $\Xi(A_h + B\bar{K})$, built from the obtained $\bar{P}$, on each of the 65536 pair of vertex matrices (A_h, B) defining Ω. Such a result was obtained with a final constraint set $\mathcal{C}_i$ made of five samples (the initial point and four counter-examples). On the other hand, on our reference hardware solving the original LMI feasibility problem (3) was not possible due to limitations on the available computational power. In contrast, the proposed method was able to synthesize a quadratic control Lyapunov function in a few tens of seconds.

We note that the results shown above were obtained excluding from the set of Lyapunov candidates all those functions not satisfying the constraints dictated by our choice of η and ε in (5). While, in our experience, tuning said quantities was not problematic, an unreasonable choice of those hyperparameters (e.g., $\eta = 2 \cdot 10^3$ and $\varepsilon = 10^3$) may indeed cause the proposed scheme to fail. In that case, one can always restart the procedure with more permissive values for η and ε. Finally, when a quadratic control Lyapunov function is found, it may be characterized by a smaller ε than in the verifier task.

B. Spherical uncertainty set

We consider now the case in which the matrix $A(k)$ is subject to spherical uncertainty, i.e., $\Omega = \{A \in \mathbb{R}^{2 \times 2} \mid (\text{vec}(A) - c)^\top Q(\text{vec}(A) - c) - 1 \le 0\}$, where the operator $\text{vec}(\cdot)$ stacks the columns of its argument into a single column vector, $Q = 5I$, while $B(k) = B = \begin{bmatrix} 0 & 1 \end{bmatrix}^\top$. In particular, $c = \text{vec}(A_{\text{centroid}})$, where $A_{\text{centroid}} = \begin{bmatrix} 0.6458 & 0.3852 \\ -1.4651 & 1.1183 \end{bmatrix}$. The synthesis of a control Lyapunov function was obtained with a counter-example set including four points, i.e., Algorithm 1 converged in three iterations, yielding $\bar{P} = \begin{bmatrix} 117.4770 & 60.7593 \\ 60.7593 & 130.6819 \end{bmatrix}$ and $\bar{K} = \begin{bmatrix} 0.9280 & -1.4962 \end{bmatrix}$. We verified the soundness of these results by using a convex polytopic outer-approximation of the spherical set Ω, built using the method in [27]. Specifically, the polytope was built using 1000 randomly generated rays, yielding 6592 vertices. This fact reinforces the findings of the previous example regarding the efficiency of our approach in comparison to purely geometric set approximation alternatives. Moreover, the technique in [27] unavoidably introduces some degree of conservatism, which is avoided using our method. While a spherical Ω theoretically has an infinite number of vertices, Theorem 2 guarantees that, in view of the assumption $\eta \ge \varepsilon > 0$, to find a quadratic control Lyapunov function and associated linear controller is sufficient to examine only a finite number of vertices.

VI. CONCLUSION

We have introduced a CEGIS-based method to design control Lyapunov functions, accompanied by suitable controllers, for linear systems with parametric uncertainties in an arbitrary nonempty compact set. By building upon results in linear operator theory, our technique features appealing Lipschitz-continuity properties that enable the use of efficient solvers for global optimization and guarantee finite-time convergence. Future work will focus on improving the rate of convergence of the proposed algorithm by enlarging the safe sets, as well as estimating the number of iterations needed for convergence. We will also analyze other type of uncertainties, and on a further comparison with available techniques (e.g., those based on SOS methods [5]).

REFERENCES

[1] F. Blanchini and S. Miani, *Set-theoretic methods in control*. Birkhäuser, 2015.
[2] P. Biswas, P. Grieder, J. Löfberg, and M. Morari, "A survey on stability analysis of discrete-time piecewise affine systems," *IFAC Proc. Volumes*, vol. 38, no. 1, pp. 283–294, 2005.
[3] M. V. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.
[4] P. A. Parrilo, *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. Caltech, 2000.
[5] F. Wu and S. Prajna, "A new solution approach to polynomial LPV system analysis and synthesis," in *IEEE ACC*, 2004, pp. 1362–1367.
[6] N. Gaby, F. Zhang, and X. Ye, "Lyapunov-net: A deep neural network architecture for Lyapunov function approximation," in *IEEE CDC*, 2022.
[7] L. Munser, G. Devadze, and S. Streif, "Synthesis of Lyapunov functions using formal verification," *arXiv:2112.01835*, 2021.
[8] H. Ravanbakhsh and S. Sankaranarayanan, "Learning control Lyapunov functions from counterexamples and demonstrations," *Autonomous Robots*, vol. 43, pp. 275–307, 2019.
[9] J. S. McGough, A. W. Christianson, and R. C. Hoover, "Symbolic computation of Lyapunov functions using evolutionary algorithms," in *Proc. of the 12th IASTED Int. Conf.*, vol. 15, 2010, pp. 508–515.
[10] S. M. Khansari-Zadeh and O. Khatib, "Learning potential functions from human demonstrations with encapsulated dynamic and compliant behaviors," *Autonomous Robots*, 2017.
[11] I. Mordatch and E. Todorov, "Combining the benefits of function approximation and trajectory optimization." in *Robotics: Science and Systems*, vol. 4, 2014, p. 23.
[12] A. Solar-Lezama, L. Tancau, R. Bodik, S. Seshia, and V. Saraswat, "Combinatorial sketching for finite programs," in *ACM ASPLOS*, 2006.
[13] A. Abate, I. Bessa, D. Cattaruzza, L. Cordeiro, C. David, P. Kesseli, D. Kroening, and E. Polgreen, "Automated formal synthesis of digital controllers for state-space physical plants," in *Proc. of the Int. Conf. on Computer Aided Verification*. Springer, 2017.
[14] G. O. Berger and S. Sankaranarayanan, "Learning fixed-complexity polyhedral Lyapunov functions from counterexamples," in *IEEE CDC*, 2022.
[15] Y.-C. Chang, N. Roohi, and S. Gao, "Neural Lyapunov control," in *NIPS*, 2019.
[16] H. Dai, B. Landry, M. Pavone, and R. Tedrake, "Counter-example guided synthesis of neural network Lyapunov functions for piecewise linear systems," in *IEEE CDC*, 2020.
[17] A. Abate, A. Edwards, and M. Giacobbe, "Neural abstractions," *arXiv:2301.11683*, 2023.
[18] D. Ahmed, A. Peruffo, and A. Abate, "Automated and sound synthesis of Lyapunov functions with SMT solvers," in *Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2020, pp. 97–114.
[19] T. Kato, *Perturbation theory for linear operators*. Springer, 2013.
[20] G.-R. Duan and H.-H. Yu, *LMIs in control systems: Analysis, design and applications*. CRC Press, 2013.
[21] T. A. Horn and C. R. Johnson, *Topics in matrix analysis*. Cambridge University Press, 1991.
[22] A. Alessio, M. Lazar, A. Bemporad, and W. Heemels, "Squaring the circle: An algorithm for generating polyhedral invariant sets from ellipsoidal ones," *Automatica*, vol. 43, pp. 2096–2103, 2007.
[23] C. Malherbe and N. Vayatis, "Global optimization of Lipschitz functions," in *ACM ICML*, 2017.
[24] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in matlab," in *In Proc. of the CACSD Conf.*, 2004.
[25] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *IEEE ECC*, 2013.
[26] L. Stripinis and R. Paulavičius, "DIRECTGO: A new DIRECT-type MATLAB toolbox for derivative-free global optimization," *ACM Trans. on Mathematical Software*, 2022.
[27] J. Löfberg, "Sample-based robust optimization," 2016. [Online]. Available: https://yalmip.github.io/example/scenariosampling/