LP-generated Control Lyapunov Functions with application to multicopter control

Huu-Thinh Do, Franco Blanchini, Stefano Miani, Ionela Prodan

LP-generated Control Lyapunov Functions with application to multicopter control

Huu-Thinh Do*, Franco Blanchini**, Stefano Miani**, Ionela Prodan*

***Abstract*—In this work, we study a technique of exploiting open-loop generated trajectories for a constrained control problem, using them to shape suitable non-quadratic Control Lyapunov Functions. These trajectories, generated off-line, allow detecting a suitable domain of attraction in which a candidate Lyapunov function has a negative derivative. Given a suitably constructed basis function, our working machinery is based on linear programming, hence, the technique can be applied to problems of non-trivial size in terms of the number of basis functions and points in the state space. For linear systems, we seek convex Lyapunov functions, which are homogeneous polynomials. Simulation and experimental results for drone control are given.**

***Index Terms*—Constrained control, Lyapunov function, open loop trajectories, linear programming, homogeneous polynomials.**

I. Introduction

The Control Lyapunov Function (CLF) notion has proven fundamental in control design, especially when uncertainty and constraints come into play. Determining a CLF is a pivotal step, after which a controller can be immediately deduced using Sontag's universal formula [1]. In view of applications with input constraints, several explicit control laws are proposed, depending on the constraint's characterization (e.g. Minkowski balls [2], their direct products [3] or recently, polytopes [4], [5]). Furthermore, via a Quadratic or Linear Program, implicit continuous control can be introduced with different objectives (min-norm [6] or max-rate [7] control).

However, proposing a CLF a priori to the controller synthesis is not a trivial task, especially when its stabilizing condition (negative time derivative) has to be proven feasible under operating constraints. This task is usually handled on a case-by-case basis, with specific numerical tools and some restrictions on the number of states and inputs. For example, with sum-of-square methods, CLF for polynomial systems can be achieved with some relaxation via semidefinite programming (SDP) [8], [9]. In [10], a composite quadratic function was developed as the CLF for linear systems subject to saturation and state constraints, resulting in the domain of attraction described by a convex hull of ellipsoids. The study for linear piecewise affine systems is conducted in [11]. More generally for nonlinear systems, yet at the price of high computational burden with respect to the dimensionality problem, procedures to numerically compute a continuous piecewise affine CLF were developed in [12], [13] via state space triangulation. In [14], under a known assumed approximation error bound, a neural network structure was proposed to learn the dynamics, the Lyapunov function, and the controller simultaneously. Online synthesis of state-dependent CLF as well as the corresponding control is proposed for discrete systems [15], [16]. For certain low-dimensional systems, the Hamilton-Jacobi-Bellman partial differential equation can be solved to incorporate the notion of optimality [17], [18]. For a more detailed survey of computational CLF synthesis methods, we refer to the review [19].

Meanwhile, one effective strategy to sidestep the a priori design of either the controller in closed form or a CLF is the Model Predictive Control (MPC) approach. In the standard setting, for example, a proper choice of terminal ingredients (cost and constraints) will result in a stabilizing implicit controller, and a cost function that can be interpreted as a piecewise CLF [20]. But then, the real-time implementation will become more time-consuming due to the solving of an online optimization problem. Furthermore, for controllers with limited hardware power, this difficulty may lead to computational unfeasibility or instability due to the so-called *early termination* [21] (i.e., the search for the optimal control is interrupted due to the limited execution time).

With those pros and cons of both CLF-based control and MPC strategy, our objective is to exploit the systematically designed MPC law to seek a corresponding CLF during the offline synthesis. More specifically, in this work, we propose a synthesis procedure for a CLF by parameterizing it as linear combinations of certain candidate CLFs. The idea includes collecting admissible state-input pairs generated by a stabilizing MPC software, then a parameterization is constructed by a linear program. The advantages of the technique are the simplicity of the offline synthesis and the reduction in computational overhead during the online implementation, allowing applications to high-dimensional systems. Then, to validate its applicability and emphasize the computational advantages, the method is experimentally examined over a challenging and timely application, the quadcopter position stabilization. It is well known that it requires small sampling time, online feasibility, and constraint satisfaction. The computational requirements become even more stringent when considering a team of drones. Summarily, in this work, we:

- propose an offline procedure to construct a CLF based on linear programming, exploiting the admissible control generated by an MPC law;
- validate the proposed process for a drone stabilizing problem through experimental tests. The video for the experiment can be found at `https://youtu.be/XPyZAJoN2Ps`.

*Univ. Grenoble Alpes, Grenoble INP†, LCIS, 26000 Valence, France. Email: {huu-thinh.do,ionela.prodan}@lcis.grenoble-inp.fr
†Institute of Engineering and Management Univ. Grenoble Alpes.
**Dipartimento di Matematica e Informatica, Università di Udine, 33100 Udine, Italy. Email: {franco.blanchini,miani.stefano}@uniud.it

The remainder of the paper is structured as follows. Section II presents the preliminaries and the main principles for the CLF construction. Therein, via a linear program (LP), a CLF is constructed with stabilizing conditions imposed at points in the state-input space. Section III presents the formulation for the acquisition of admissible points shaping the desired CLF from an MPC-based admissible point generator. Section IV particularizes the proposed technique for a quadcopter position control problem. The resulted CLF and the associated control will be validated in Section V with various experimental tests. Finally, Section VI concludes and discusses future directions.

Notation: Bold lower-case letters denote vectors. With $\boldsymbol{x} = [x_1, x_2, ..., x_n]^\top \in \mathbb{R}^n$, $\nabla V(\boldsymbol{x}) = [\frac{\partial V(\boldsymbol{x})}{\partial x_1}, \frac{\partial V(\boldsymbol{x})}{\partial x_2}, ..., \frac{\partial V(\boldsymbol{x})}{\partial x_n}]$ denotes the gradient of the value function $V(\boldsymbol{x})$. The standard norms for a vector $\boldsymbol{x}$ are denoted as $\|\boldsymbol{x}\|_2 \triangleq \sqrt{\boldsymbol{x}^\top \boldsymbol{x}}$, $\|\boldsymbol{x}\|_{\boldsymbol{P}} \triangleq \sqrt{\boldsymbol{x}^\top \boldsymbol{P} \boldsymbol{x}}$. $\|\cdot\|_\infty$ is the infinity norm. $\mathbf{0}$ denotes the origin of the vector space with the appropriate dimension. The letter k denotes the discrete step, while t represents the time variable. $\mathcal{N}(n, a, b)$ represents a finite set of n evenly spaced real numbers over the interval $[a, b]$. $\mathrm{co}\{\cdot\}$ denotes the convex hull. $\mathrm{diag}(\cdot)$ returns a matrix with its entries diagonally arranged.

II. Main idea and machinery

Consider the following nonlinear control affine system:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}, \tag{1}$$

with a convex and compact constraint set:

$$\boldsymbol{u} \in \mathcal{U}, \tag{2}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ denotes the state vector, and $\boldsymbol{u} \in \mathbb{R}^m$ is the input vector constrained in $\mathcal{U}$.

Assumption 1: Assume the pair $\boldsymbol{x} = \mathbf{0}$ and $\boldsymbol{u} = \mathbf{0}$ is a steady state, i.e., $\mathcal{U}$ includes $\mathbf{0}$ in its interior and $\boldsymbol{f}(\mathbf{0}) = \mathbf{0}$. □

Assume that we have an open-loop constrained trajectory software tool, capable of driving the system to $\mathbf{0}$. Then we could just use the software in an MPC fashion for control. Still, for systems with a small sampling time or limited hardware resources, such MPC implementation could be challenging.

The underlying idea is that of adopting this open-loop trajectory generator to simply determine candidate state control-pair optimal trajectories. These state control pairs are then used to generate suitable constraints for finding a CLF, within a linearly parameterized family, through linear programming. These generated trajectories will eventually be disregarded, and only the CLF will be retained for actual control. To this aim, some standard definitions are recalled in the following.

Definition 1 (Control Lyapunov Function): A positive definite smooth Lyapunov function, $V(\boldsymbol{x})$, is a CLF for (1) if there exists a function $\boldsymbol{u} = \boldsymbol{\Xi}(\boldsymbol{x})$ with values in $\mathcal{U}$ and

$$\mathrm{d}V(\boldsymbol{x})/\mathrm{d}t = \nabla V(\boldsymbol{x})[\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{\Xi}(\boldsymbol{x})] \leq -\beta V(\boldsymbol{x}), \tag{3}$$

for some $\beta > 0$. □

Definition 2: The state-control pair $(\boldsymbol{x}, \boldsymbol{u})$ for (1), $\boldsymbol{u} \in \mathcal{U}$ is admissible for $V(\boldsymbol{x})$ if it satisfies (3). □

Definition 3 (Domain of Attraction): $\mathcal{D} \subseteq \mathbb{R}^n$ is a domain of attraction (DOA) of system (1) if there exist $\boldsymbol{u}(t) = \boldsymbol{\Xi}(\boldsymbol{x}(t))$, such that any trajectory $\boldsymbol{x}(t)$ with $\boldsymbol{x}(0) \in \mathcal{D}$, satisfies:

$$\boldsymbol{x}(t) \in \mathcal{D}, \boldsymbol{u}(t) \in \mathcal{U}, \text{ and } \lim_{t\to\infty} \boldsymbol{x}(t) = \mathbf{0}. \tag{4}$$

□

Next, in the offline control synthesis, adopting the available software, we generate several admissible pairs and then, formulate the CLF based on these points. This is a simple idea to generate, under appropriate conditions, points that are inside the domain of attraction $\mathcal{D}$, eliminating all points outside, for which, no constrained control can bring the state to $\mathbf{0}$.

Let us assume that the open-loop trajectory generator ensures the existence of constrained trajectories $\boldsymbol{x}(t), \boldsymbol{u}(t)$, from which we can sample M pairs $(\boldsymbol{x}_j, \boldsymbol{u}_j)$, $\boldsymbol{u}_j \in \mathcal{U}, j \in \{1, ..., M\}$. Consider the following candidate CLF:

$$V(\boldsymbol{x}) = \sum_{i=0}^{N} \alpha_i V_i(\boldsymbol{x}), \quad \alpha_i \geq 0, \tag{5}$$

where $V_i(\boldsymbol{x})$ are smooth positive definite Lyapunov functions.

Next, consider the following linear programming problem with N variables, and M inequality constraints formulated from M state-input pairs collected previously:

$$\min_{\alpha_i} \sum_{i=1}^{N} \ell_i \alpha_i \tag{6a}$$

$$\sum_{i=1}^{N} \alpha_i \left[\nabla V_i(\boldsymbol{x}_j)\left[\boldsymbol{f}(\boldsymbol{x}_j) + \boldsymbol{g}(\boldsymbol{x}_j)\boldsymbol{u}_j\right] + \beta V_i(\boldsymbol{x}_j)\right] \leq 0, \tag{6b}$$

$$\sum_{i=1}^{N} \alpha_i = 1, \alpha_i \geq 0, j \in \{1, ..., M\}, \tag{6c}$$

where ℓ_i are weights. These weights give freedom of choice in the computation. As an example, if $V_1(\boldsymbol{x})$ is a "privileged" function, for instance a quadratic function associated with a local control optimizing cost, then ℓ_1 can be adjusted to influence the function dominance within the parameterization (5). Then, as a result of (6), we achieve a parameterized CLF established as a linear combination of $V_i(\boldsymbol{x})$ with the coefficients α_i. An illustration on how the Lyapunov function $V(\boldsymbol{x})$ is shaped by the stabilizing constraint (6b) at each sampled data pairs is given in Fig. 1.

Remark 1: With the intuition of parameterizing the function $V(\boldsymbol{x})$ via the set of $V_i(\boldsymbol{x})$, it is essential to provide a suitable choice for such candidates. This, indeed, can be done via an iterative process where, after one solution of (6), the function $V_i(\boldsymbol{x})$ with $\alpha_i \approx 0$ can be discarded and a new set of functions can be adapted. Note that, this can be accomplished thanks to the simplicity of the LP (6). A discussion on the choice of $V_i(\boldsymbol{x})$ will be given later in the illustrative example. □

Forming the CLF in this manner has some pros and cons:

- by employing an LP, it is possible to use a considerably large number of points and basis functions $V_i(\boldsymbol{x})$, ensuring the existence of α_i for some β in (6), even in high-dimensional systems;
- although the number of generating data pairs $(\boldsymbol{x}_j, \boldsymbol{u}_j)$ can be huge, we are still working on a finite set.

To address a preliminary answer to the second point, let us consider the following propositions. For the set $\mathcal{U}$, denote by $\Psi_{\mathcal{U}}(\boldsymbol{w})$ its support functional:

$$\Psi_{\mathcal{U}}(\boldsymbol{w}) \triangleq \max_{\boldsymbol{u}\in\mathcal{U}} \boldsymbol{w}^\top \boldsymbol{u}. \tag{7}$$

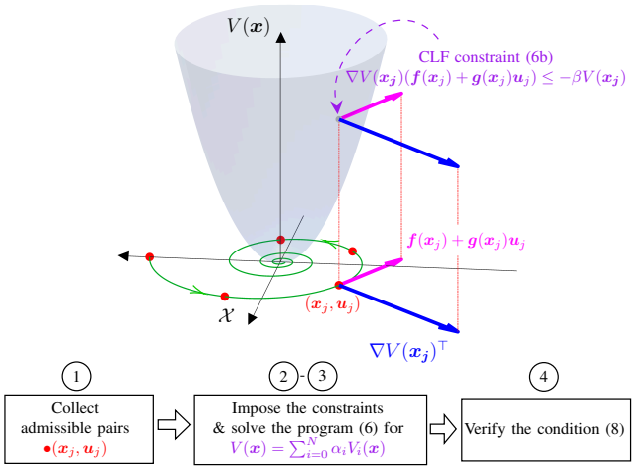

Fig. 1. Geometric interpretation of how the stabilizing condition (6b) is imposed from the sampled pairs (red points) $(\boldsymbol{x}_j, \boldsymbol{u}_j)$ to the surface of $V(\boldsymbol{x})$.

Note that $\Psi_{\mathcal{U}}(\mathbf{0}) = \mathbf{0}$ and that, in view of Assumption 1, $\Psi_{\mathcal{U}}(\boldsymbol{w})$ is positive definite and positively homogeneous of degree one: $\Psi_{\mathcal{U}}(\lambda \boldsymbol{w}) = \lambda \Psi_{\mathcal{U}}(\boldsymbol{w}), \forall\, \lambda > 0$.

Proposition 1: The function $V(\boldsymbol{x})$ is a CLF on a given domain $\mathcal{X}$ containing $\mathbf{0}$ in its interior iff it satisfies the following inequality on $\mathcal{X}$:

$$\nabla V(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}) - \Psi_{\mathcal{U}}(\boldsymbol{g}(\boldsymbol{x})^\top \nabla V(\boldsymbol{x})^\top) + \beta V(\boldsymbol{x}) \leq 0, \tag{8}$$

for some $\beta > 0$ on $\mathcal{X}$. □

Proof: By Definition 1, $V(\boldsymbol{x})$ is a CLF iff $\forall \boldsymbol{x} \in \mathcal{X}, \exists \boldsymbol{u} \in \mathcal{U}$ such that:

$$\nabla V(\boldsymbol{x})[\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}] + \beta V(\boldsymbol{x}) \leq 0, \tag{9}$$

or, equivalently:

$$\min_{\boldsymbol{u}\in\mathcal{U}} \nabla V(\boldsymbol{x})[\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}] + \beta V(\boldsymbol{x}) \leq 0. \tag{10}$$

Condition (10) then can be rewritten as:

$$\max_{\boldsymbol{u}\in\mathcal{U}} \nabla V(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}) - \nabla V(\boldsymbol{x})\boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} + \beta V(\boldsymbol{x}) \leq 0. \tag{11}$$

Then, by the support function definition in (7), one gets to Proposition 1. ■

Remark 2: It is worth noticing that (1) does not refer to a specific control action $\boldsymbol{u} = \boldsymbol{\Xi}(\boldsymbol{x})$ but depends only on the shape of $\mathcal{U}$. For instance, if, as often assumed in the literature, $\mathcal{U}$ is the unit ball $\|\boldsymbol{u}\|_2 \leq 1$, then equation (8) becomes:

$$\nabla V(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}) - \|\boldsymbol{g}(\boldsymbol{x})^\top \nabla V(\boldsymbol{x})^\top\|_2 + \beta V(\boldsymbol{x}) \leq 0 \tag{12}$$

□

For a domain of interest $\mathcal{X}$, the proposition is useful since it allows testing the generated function $V(\boldsymbol{x})$ in (6), by dense gridding and verifying the condition (8) in $\mathcal{X}$. To recap, we propose the following procedure to determine $V(\boldsymbol{x})$.

Procedure 1 Offline CLF generation

1: Collect M admissible pairs $(\boldsymbol{x}_j, \boldsymbol{u}_j)$ from a stabilizing controller for system (1). This can be sampled pairs from the system's trajectories with several random initial conditions (see also Fig. 1);
2: Fix a basis function $V_i(\boldsymbol{x})$ as in (5), $i = 1, ..., N$ of positive definite functions, and $\beta > 0$ as in (3);
3: Formulate and solve the linear program (6);
4: If the problem (6) is feasible, then grid the state space within a region of interest and check (8) for all points in the grid.

Aligning with the common CLF setup, the aforementioned considerations are formalized in the following proposition.

Proposition 2: Suppose that Procedure 1 is successful and that there exists $\kappa > 0$ for which inequality (8) is verified in the sublevel set $\mathcal{D}_\kappa = \{\boldsymbol{x} : V(\boldsymbol{x}) \leq \kappa\}$, then $V(\boldsymbol{x})$ is a CLF for the problem (1) with the DOA $\mathcal{D}_\kappa$. □

Besides intensely gridding the state space inside $\mathcal{D}_\kappa$ to verify (8), one can also achieve an inner estimation as follows.

Proposition 3: Consider an ellipsoid inscribed in the input constraint set $\mathcal{U}$ in (2), which has the form:

$$\mathcal{E} = \{\boldsymbol{u} = \boldsymbol{E}\boldsymbol{w}, \|\boldsymbol{w}\|_2 \leq 1\} \text{ and } \mathcal{E} \subset \mathcal{U}. \tag{13}$$

Let $\Phi(\kappa_e)$ be defined as:

$$\Phi(\kappa_e) \triangleq \max_{V(\boldsymbol{x})\leq\kappa_e} \Big(\nabla V(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}) - \|\boldsymbol{E}\boldsymbol{g}(\boldsymbol{x})^\top \nabla V(\boldsymbol{x})^\top\|_2 + \beta V(\boldsymbol{x})\Big) \tag{14}$$

then, for $\kappa_e > 0$, if $\Phi(\kappa_e) \leq 0$, $\mathcal{D}_{\kappa_e}$ is a DOA. □

Proof: From the convexity and compactness of $\mathcal{E}$ and $\mathcal{U}$, one can state [22]:

$$\mathcal{E} \subset \mathcal{U} \Rightarrow \Psi_{\mathcal{E}}(\boldsymbol{v}) \leq \Psi_{\mathcal{U}}(\boldsymbol{v}), \forall \boldsymbol{v} \in \mathbb{R}^m. \tag{15}$$

Thus,

$$\begin{aligned} &\nabla V(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}) - \Psi_{\mathcal{U}}(\boldsymbol{g}(\boldsymbol{x})^\top \nabla V(\boldsymbol{x})^\top) + \beta V(\boldsymbol{x}) \\ &\leq \nabla V(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}) - \Psi_{\mathcal{E}}(\boldsymbol{g}(\boldsymbol{x})^\top \nabla V(\boldsymbol{x})^\top) + \beta V(\boldsymbol{x}) \\ &= \nabla V(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}) - \|\boldsymbol{E}\boldsymbol{g}(\boldsymbol{x})^\top \nabla V(\boldsymbol{x})^\top\|_2 + \beta V(\boldsymbol{x}). \end{aligned} \tag{16}$$

Consequently, if for $\kappa_e > 0$, $\Phi(\kappa_e) \leq 0$, then, by Proposition 1, $\mathcal{D}_{\kappa_e}$ is a DOA. With this method, we iteratively seek also for the largest κ_e that satisfies $\Phi(\kappa_e) \leq 0$. The trade-off for this is the conservative employment of an ellipsoidal subset $\mathcal{E}$ instead of the original constraint set $\mathcal{U}$. ■

To complete the synthesis, the remaining discussion resides on how to formulate the generator of admissible pairs $(\boldsymbol{x}_j, \boldsymbol{u}_j)$ for (6). Moreover, with the established CLF, the choice of the associated control law is an important question as well. These two issues will be addressed in the next section.

III. Admissible Pair Generator and Control Implementation

In this part, the collection of admissible pairs at step 1 in Procedure 1 is first discussed. Next, the online control policy employed with the generated CLF will be introduced.

A. Admissible pair generation

As mentioned before, we aim to employ an MPC solver to achieve stabilizing trajectories from which admissible pairs are sampled. However, the MPC strategy commonly needs to be implemented in a discretized manner. With such a discretization, the generated pairs do not necessarily imply the stability property or the accurate evolution of the continuous system (1). For this reason, we employ the Euler Auxiliary System (EAS) [23]:

$$\boldsymbol{x}(k+1) = \boldsymbol{x}(k) + \tau\left[\boldsymbol{f}(\boldsymbol{x}(k)) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}(k)\right], \tag{17}$$

where $\tau > 0$ is the discretization constant parameter. Then, with this setup and a CLF generated with (17), we have the following result.

Theorem 1: Any convex function $V(\boldsymbol{x})$ which is a CLF for the EAS in (17) is also a CLF for the associated continuous time system in (1) . □

Proof: The convexity of $V(\boldsymbol{x})$ implies that the difference quotient function is non-decreasing with regard to $h \in \mathbb{R}^+$ and the Dini derivative yields [24]:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}V(\boldsymbol{x}) &= \lim_{h\to 0^+}\sup\frac{V(\boldsymbol{x}+h[\boldsymbol{f}(\boldsymbol{x})+\boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}]) - V(\boldsymbol{x})}{h}\\ &\leq \frac{V(\boldsymbol{x}+\tau[\boldsymbol{f}(\boldsymbol{x})+\boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}]) - V(\boldsymbol{x})}{\tau}, \text{ for } \tau \geq h.\end{aligned} \tag{18}$$

Meanwhile, being a CLF of the EAS implies, $\exists \beta_d > 0$:

$$V(\boldsymbol{x}+\tau[\boldsymbol{f}(\boldsymbol{x})+\boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}]) - V(\boldsymbol{x}) \leq -\beta_d V(\boldsymbol{x}). \tag{19}$$

Thus, from (18), (19) and Definition 1, one can state that :

$$\frac{\mathrm{d}}{\mathrm{d}t}V(\boldsymbol{x}) \leq -\frac{\beta_d}{\tau}V(\boldsymbol{x}), \tag{20}$$

which completes the proof. ■

Therefore, for the offline admissible pair generator in Procedure 1, step 1, we employ the following MPC strategy:

$$\arg\min_{\boldsymbol{u}(\cdot)} \sum_{k=0}^{N_p}\left(\|\boldsymbol{x}(k)\|^2_{\boldsymbol{Q}} + \|\boldsymbol{u}(k)\|^2_{\boldsymbol{R}}\right) + \|\boldsymbol{x}(N_p)\|^2_{\boldsymbol{P}} \tag{21a}$$

$$\text{s.t} \begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{x}(k) + \tau\left[\boldsymbol{f}(\boldsymbol{x}(k)) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}(k)\right], \\ \boldsymbol{u}(k) \in \mathcal{U}, \boldsymbol{x}(k) \in \mathcal{X}, \boldsymbol{x}(N_p) \in \mathcal{X}_f, \end{cases} \tag{21b}$$

where $\boldsymbol{Q}, \boldsymbol{R}$ is the user-defined weightings penalizing the tracking and input, respectively. $\boldsymbol{P}$ and $\mathcal{X}_f$ denote the weighting for the terminal cost and its terminal set, which are associated with a locally optimal control $\boldsymbol{u}_{opt}(\boldsymbol{x})$. This standard penalization and constraints guarantee the recursive feasibility and stability of the system [25]. Then, for a sampled state vector, $\boldsymbol{x}_j$, the admissible control $\boldsymbol{u}_j$ is taken as the first component of the minimizing sequence of (21b) with $\boldsymbol{x}(k = 0) = \boldsymbol{x}_j$. This formulation hence completes the proposed procedure to collect admissible pairs.

B. Associated nonlinear control law

Prior to this point, the ingredients for the offline generation of the CLF have been presented. Yet, a further question is how one can impose a desired performance over such a CLF during online implementation. Following the continuous stabilizer in [26], consider the controller:

$$\boldsymbol{u} = \arg\min_{\boldsymbol{u}} \|\boldsymbol{u} - \boldsymbol{u}_d(\boldsymbol{x})\|_2^2 \tag{22a}$$

$$\text{s.t} \begin{cases} \boldsymbol{u} \in \mathcal{U} \\ \nabla V(\boldsymbol{x})\left[\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}\right] \leq -\beta V(\boldsymbol{x}). \end{cases} \tag{22b}$$

Namely, the constrained control ensures that the time derivative of $V(\boldsymbol{x})$ is negative, while the objective function minimizes the deviation of $\boldsymbol{u}$ from the desired one, $\boldsymbol{u}_d(\boldsymbol{x})$. The above optimization is convex due to the convexity of $\mathcal{U}$ in (2) and the linear constraint in (22b). From a practical viewpoint, the role of $V(\boldsymbol{x})$ is to ensure stability, while choosing $\boldsymbol{u}_d(\boldsymbol{x})$ is also important for performance. This control is typically selected as the locally optimal one.

Remark 3: Suppose that there exists a DOA $\mathcal{D}_\kappa = \{\boldsymbol{x} : V(\boldsymbol{x}) \leq \kappa\}$ deduced from the generated Lyapunov function $V(\boldsymbol{x})$. It follows directly from the differential inequality (3) that the control (22) ensures the exponential convergence with the decay constant β inside $\mathcal{D}_\kappa$. Moreover, note that we can always ensure that such a κ exists if we insert among the basis function a locally optimal quadratic function associated with the standard LQ control.

Next, to demonstrate and highlight the applicability of the procedure, let us proceed with a case study of the quadcopter position control problem (a.k.a. outer loop or high level control) in the next section.

IV. Application on quadcopter position control

In this section, via a feedback linearization law, two different representations of the quadcopter's constrained position control problem are presented. Then, the constructions of the CLF for each representation will follow.

A. Problem formulation for quadcopter position control

Let us provide the outer-loop model for a miniature quadcopter as commonly used in the literature [27]:

$$\ddot{\boldsymbol{\xi}} = T\mathbf{R}_E - g\boldsymbol{e}_3, \tag{23}$$

where $\boldsymbol{\xi} = [x, y, z]^\top \in \mathbb{R}^3$ collects the drone's position. In the Euler angles representation, ϕ, θ, ψ respectively denote the roll, pitch and yaw angles. $\mathbf{R}_E$ denotes the rotation matrix from the body to the global frame, which is computed as: $\mathbf{R}_E = \begin{bmatrix} \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ \cos\phi\cos\theta \end{bmatrix}$. g is the gravitational acceleration and $\boldsymbol{e}_3 = [0, 0, 1]^\top$ is the standard basis vector pointing along the altitude z axis. T denotes the normalized thrust provided by the four propellers. In a quasi-stationary flight settings, we assume that the aerodynamic forces are always in the direction of the z axis. Centrifugal force and disturbance (e.g., drag) are neglected. As the outer loop (see Fig. 9), the control objective for system (23) is to design the position-stabilizing ϕ, θ angles and thrust T. These three signals are regarded as the input for this control layer, which then will be actuated by the low-level controller with a proper

feedback scheme and rotor configuration [28]. In this setup, the input constraint set for $[T, \phi, \theta]^\top$ is described as:

$$\mathcal{C} = \{0 \leq T \leq T_{max}, |\phi| \leq \phi_{max}, |\theta| \leq \theta_{max}\} \tag{24}$$

where $T_{max} > g$, $(\phi_{max}, \theta_{max}) \in (0; \pi/2)^2$ are the inputs' constant bounds. It is well known that system (23) is feedback linearizable. Indeed, consider the input transformation:

$$T = \sqrt{u_x^2 + u_y^2 + (u_z + g)^2}, \tag{25a}$$

$$\phi = \arcsin\left((u_x \sin\psi - u_y \cos\psi)/T\right), \tag{25b}$$

$$\theta = \arctan\left((u_x \cos\psi + u_y \sin\psi)/(u_z + g)\right), \tag{25c}$$

then, under the condition $u_z \geq -g$, the new dynamics can be exactly linearized as a system of three double integrators:

$$\dot{\boldsymbol{p}}_q = \boldsymbol{A}_{2D}\boldsymbol{p}_q + \boldsymbol{B}_{2D}u_q,\ q \in \{x, y, z\}, \tag{26}$$

where $\boldsymbol{p}_q = [q, \dot{q}]^\top$; $\boldsymbol{A}_{2D} = \left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]$, $\boldsymbol{B}_{2D} = [0\ 1]^\top$. Besides, the image of the constraint (24) via (25) can be characterized as a convex set of $\boldsymbol{u}_{pos} \triangleq [u_x, u_y, u_z]^\top \in \mathbb{R}^3$ as in [29]:

$$\boldsymbol{u}_{pos} \in \mathcal{U}_{pos} \triangleq \Big\{\boldsymbol{u}_{pos} : \|\boldsymbol{u}_{pos} + g\boldsymbol{e}_3\|_2^2 \leq T_{max}^2; \sqrt{u_x^2 + u_y^2} \leq (u_z + g)\tan\epsilon_{max}\Big\}. \tag{27}$$

with $\epsilon_{max} \triangleq \min(\phi_{max}, \theta_{max})$. The set $\mathcal{U}_{pos}$ is convex since it is the intersection of two convex sets: a ball of radius T_{max} centered at $-g\boldsymbol{e}_3$ and a convex cone defined with ϵ_{max} and g. A detailed characterization of $\mathcal{U}_{pos}$ as in (27) can be found in [29]. With the dynamics (26), one can consider this linearized system as either three independent double integrators in $\mathbb{R}^2$ tangled by the constraint (27) or one system in $\mathbb{R}^6$ and the input constraint $\boldsymbol{u}_{pos} \in \mathcal{U}_{pos}$. Next, we investigate the dynamics from both points of view with the proposed procedure. To this aim, we consider two common inner approximations of $\mathcal{U}_{pos}$ as in the following.

Case 1: A box-type subset: Since $\mathcal{U}_{pos}$ includes $\mathbf{0}$ as an interior point, there always exist $\bar{u}_i > 0$ such that:

$$\mathcal{U}_b \triangleq \{\boldsymbol{u}_{pos} \in \mathbb{R}^3 : |u_q| \leq \bar{u}_q, q \in \{x, y, z\}\} \subset \mathcal{U}_{pos}. \tag{28}$$

In this setting, system (26) is decoupled as three linear subsystems associated with three input constraints $|u_q| \leq \bar{u}_q$. The maximum volume $\mathcal{U}_b$ can be found by solving a convex optimization problem, maximizing its volume while constraining all the vertices inside $\mathcal{U}_{pos}$.

Case 2: A polytopic subset: In a less conservative manner, by parameterizing the boundary of $\mathcal{U}_{pos}$, we can obtain a polytopic approximation as:

$$\mathcal{U}_p \triangleq \text{co}\left\{\begin{matrix}[0, 0, -g]^\top, [R^*\cos\gamma, R^*\sin\gamma, u_z^*]^\top \\ [r\cos\gamma, r\sin\gamma, \sqrt{T_{max}^2 - r^2} - g]^\top\end{matrix}\right\} \tag{29}$$

with $R^* = T_{max}\sin\epsilon_{max}$, $u_z^* = T_{max}\cos\epsilon_{max} - g$ and for some large integers $S_1, S_2 > 2$, $\gamma \in \mathcal{N}(S_1, 0, 2\pi), r \in \mathcal{N}(S_2, 0, R^*)$. For this scenario, we arrange system (26) as:

$$\dot{\boldsymbol{\zeta}} = \boldsymbol{A}_{6D}\boldsymbol{\zeta} + \boldsymbol{B}_{6D}\boldsymbol{u}_{pos}, \tag{30}$$

where $\boldsymbol{\zeta} \triangleq \left[\begin{smallmatrix} \boldsymbol{p}_x \\ \boldsymbol{p}_y \\ \boldsymbol{p}_z \end{smallmatrix}\right] \in \mathbb{R}^6$ and $\boldsymbol{u}_{pos}$ denote the state and the input vector as in (27), respectively. $\boldsymbol{u}_{pos} \in \mathcal{U}_p$ as in (29), $\boldsymbol{A}_{6D} = \text{diag}(\boldsymbol{A}_{2D}, \boldsymbol{A}_{2D}, \boldsymbol{A}_{2D})$ and $\boldsymbol{B}_{6D} = \text{diag}(\boldsymbol{B}_{2D}, \boldsymbol{B}_{2D}, \boldsymbol{B}_{2D})$.

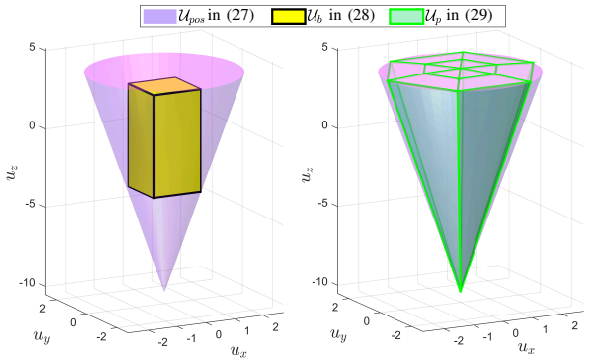

Fig. 2. The inner approximations of $\mathcal{U}_{pos}$ as in (27).

TABLE I
NUMERICAL PARAMETERS

Symbols	Values
T_{max} as in (27)	$1.45g \approx 14.22$ m/s^2
$\theta_{max} = \phi_{max} = \epsilon_{max}$ as in (24),(27)	0.1745 rad
$\bar{u}_x, \bar{u}_y, \bar{u}_z$ as in (28)	$0.8154, 0.8154, 3.27$
S_1, S_2 as in (29)	$7, 3$

An illustration and numerical parameters of the abovementioned sets are depicted in Fig. 2 and TABLE I, respectively.

Remark 4: It is undoubted that, Procedure 1 can be immediately applied for the system (30) in Case 2. However, we believe that exploiting the decoupled representation in Case 1, i.e. examining each double integrator at a time, provides not only illustrative understanding but also fundamental insights for the choice of the basis functions $V_i(\cdot)$ as in (6). Thus, in the sequel, we proceed by investigating first the decoupled representation in Case 1 (or (26)), then the procedure will be extended to Case 2 with the concatenated dynamics (30). □

B. CLF construction

In this subsection, the formulation of the CLFs for (26) and (30) subject to (28) and (29), respectively, will be discussed. With these functions come two ways of controlling the drone, as the following. In the former, three CLFs will be generated for each axis $q \in \{x, y, z\}$. These functions then will be separately combined with the setup (22) to compute the controller u_q. In the latter, one CLF will be generated for the concatenated dynamics (30) with the polytopic constraint (29). Then, the controller $\boldsymbol{u}_{pos} = [u_x, u_y, u_z]^\top$ will be computed similarly with (22) and the generated function. In both cases, the controller $[u_x, u_y, u_z]^\top$ will be transformed back to the inputs T, ϕ, θ by means of (25), then applied to the vehicle.

For Case 1: The idea is to apply Procedure 1 to three subsystems in (26) with $|u_q| \leq \bar{u}_q, q \in \{x, y, z\}$ and to create one CLF for each double integrator. For simplicity, we use $V^q(\cdot)$ to denote the generated CLF for the q axis (e.g., $V^x(\boldsymbol{p}), \boldsymbol{p} \in \mathbb{R}^2$, is the CLF for the first double integrator in (26), or $q = x$). Then, the same procedure is carried out for all three subsystems as follows.

First, the general admissible pair generator (21) will be adapted with the weighting as in TABLE II. The corresponding

TABLE II
PARAMETERS FOR POINT GENERATOR (21) WITH $\tau = 0.1s$, AND THE LP (6), $q \in \{x, y, z\}$.

Representation	Variables $\boldsymbol{x}, \boldsymbol{u}$	$\boldsymbol{Q}$	$\boldsymbol{R}$	$\boldsymbol{P}$	Constraints $\mathcal{X}$,$\mathcal{U}$ for (21)	N_p	β	ℓ_i
Case 1	$\boldsymbol{p}_x, u_x$	$\boldsymbol{Q}_{2D}$	$\boldsymbol{R}_{2D}$	$\boldsymbol{P}_{2D}$	$\lvert\boldsymbol{p}_q\rvert \leq 1.5, \lvert u_q\rvert \leq \bar{u}_q$	50	0.1275	$\ell_1 = 0.05$
	$\boldsymbol{p}_y, u_y$						0.1275	$\ell_{i\neq 1} = 0.1$
	$\boldsymbol{p}_z, u_z$						2.65	
Case 2	$\boldsymbol{\zeta}, \boldsymbol{u}_{pos}$	$\boldsymbol{Q}_{6D}$	$\boldsymbol{R}_{6D}$	$\boldsymbol{P}_{6D}$	$\lvert\boldsymbol{\zeta}\rvert \leq 1.5, \boldsymbol{u}_{pos} \in \mathcal{U}_{pos}$	50	0.15	$\ell_1 = 0.2$
								$\ell_{i\neq 1} = 1$

terminal cost adopts the weighting $\boldsymbol{P}_{2D}$ from the discrete Riccati equation for the EAS of (26) ($\tau = 0.1$s) with the penalty weighting $\boldsymbol{Q}_{2D}, \boldsymbol{R}_{2D}$, while the local controller uses the standard infinite-horizon unconstrained control. The terminal region $\mathcal{X}_f$ is the maximal positive invariant (MPI) set computed with such a local control and input constraint $|u_q| \leq \bar{u}_q$ for each subsystem. Numerical details are given as $\boldsymbol{Q}_{2D} = \left[\begin{smallmatrix} 50 & 0 \\ 0 & 5 \end{smallmatrix}\right]$, $\boldsymbol{R}_{2D} = 5$, $\boldsymbol{P}_{2D} = \left[\begin{smallmatrix} 479.6118 & 181.0469 \\ 181.0469 & 155.5598 \end{smallmatrix}\right]$, $\boldsymbol{K}_{2D} = -[2.7617\ 2.6491]$. The domain of interest for each double integrator is chosen as $\mathcal{X}_q = \{|\boldsymbol{p}_q| \leq 1.5\}, q \in \{x, y, z\}$.

The next step resides on how to choose a collection basis functions $V_i(\cdot)$ as in (5). On one hand, encoding the optimality, at least locally near the origin, is important. For this reason, we select the first basis as the quadratic function:

$$V_1(\boldsymbol{p}) = \|\boldsymbol{p}\|^2_{\boldsymbol{P}^{lqr}_{2D}}, \boldsymbol{p} \in \mathbb{R}^2. \tag{31}$$

where $\boldsymbol{P}^{lqr}_{2D}$ denotes the result of the continuous Riccati equation, with the state and input penalty abusively taken from $\boldsymbol{Q}_{2D}$ and $\boldsymbol{R}_{2D}$. On the other hand, it is necessary that the basis functions are sufficiently rich so that (6) is feasible. At the same time, it was proposed that a suitable Lyapunov function for constrained stabilization is the $2p$-norm $\|\boldsymbol{G}\boldsymbol{p}\|^{2p}_{2p}$ where for $\boldsymbol{p} \in \mathbb{R}^n$, $\boldsymbol{G} \in \mathbb{R}^{n\times r}$ is a full row rank matrix [30]. Thus, the remaining members of the collection of $V_i(\boldsymbol{p})$ are particularly populated with the following polynomials:

$$\begin{aligned} V_{2\leq i\leq 100}(\boldsymbol{p}) \in \Big\{ &\|\eta\boldsymbol{F}\boldsymbol{p}\|^4_4, \eta \in \text{ls}(33, 0.4, 2.0) \\ &\boldsymbol{F} \in \big\{ \begin{bmatrix} 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \end{bmatrix} \big\} \Big\}. \end{aligned} \tag{32}$$

Remark 5: In (32), η plays a role of increasingly enriching the basis with different value scale, while $\boldsymbol{F}$ is responsible for the "orientation" of the surfaces $V_i(\cdot)$. The 2D case of the first double integrator is depicted in Fig. 3. The feasibility of (6) can be managed by meticulously collecting more value of η and vector $\boldsymbol{F}$. □

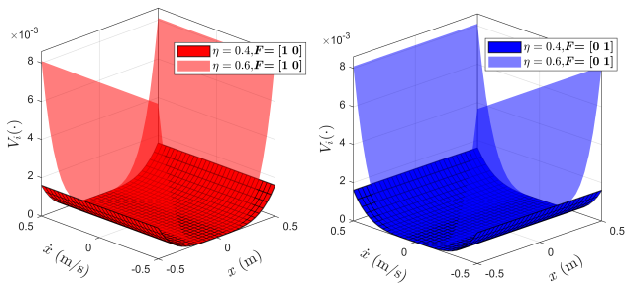

Fig. 3. The surface of $V_i(\cdot)$ with different η and $\boldsymbol{F}$ (Case 1).

Then, by partitioning the domain $\mathcal{X}_q$ by a 25×25 grid and collecting the control u_q given by (21) at those points (See Fig. 6), we achieved the three following CLFs from (6) with the basis (31) and (32):

- For $q \in \{x, y\}$ in (26):

$$\begin{aligned} V^q(\boldsymbol{p}_q) = 0.016\|\boldsymbol{p}_q\|^2_{\boldsymbol{P}^{lqr}_{2D}} + 0.2293(2[1\ 0]\boldsymbol{p}_q)^4 \\ +0.5406(2[0\ 1]\boldsymbol{p}_q)^4 + 0.2141(2[1\ 1]\boldsymbol{p}_q)^4. \end{aligned} \tag{33}$$

- For the z axis in (26):

$$\begin{aligned} V^z(\boldsymbol{p}_z) = 0.0119\|\boldsymbol{p}_z\|^2_{\boldsymbol{P}^{lqr}_{2D}} + 0.3862(2[1\ 0]\boldsymbol{p}_z)^4 \\ +0.6019(2[1\ 1]\boldsymbol{p}_z)^4 \end{aligned} \tag{34}$$

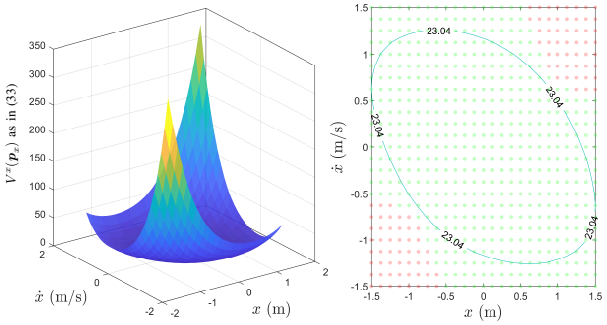

Fig. 4. Generated Lyapunov functions and its DOA for the first subsystem of (26) ($q = x$).

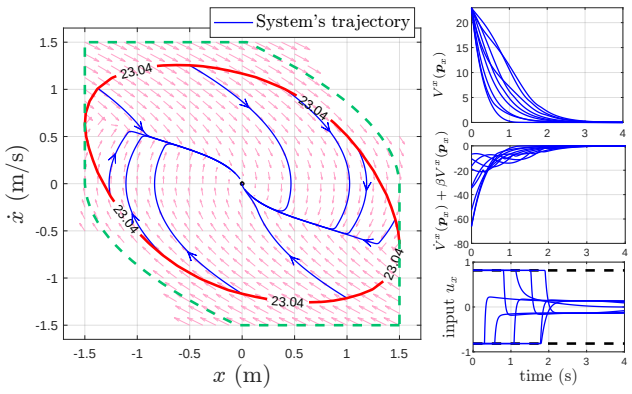

Fig. 5. Set-point tracking simulation with the first subsystem of (26) ($q = x$). The green-dashed line represents the DOA provided by the MPC point generator (21). The red solid curve depicts the sublevel set DOA given by the generated Lyapunov function.

The computation of α_i in (6) is carried out with MATLAB 2021b and its *linprog* solver, while the pair generator (21) and the online controller (22) is computed with *quadprog* solver for the simulation. The illustration for (33) is in Fig. 4. The DOA can be estimated by verifying the state-space with the

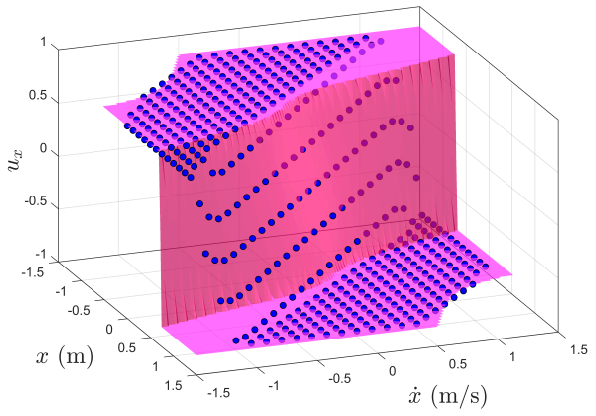

Fig. 6. Comparison between the control sampled from (21) (blue points) and the control (22) (pink surface) for Case 1, $q = x$.

condition (8), which, in this case, is given as the sublevel set $V^x(\boldsymbol{p}_x) \leq 23.04$. The positive invariance of the DOA can also be confirmed via the simulated trajectories and vector field depicted in Fig. 5. Therein, it can be seen again that there always exists $|u_x| \leq \bar{u}_x$ rendering $V^x(\boldsymbol{p}_x)$ decreasing and $\dot{V}^x(\boldsymbol{p}_x) + \beta V^x(\boldsymbol{p}_x) \leq 0$. A comparison between the control policy of the MPC (21) and (22) is given in Fig. 6 .

For Case 2: Similarly, we use the setting (21) as the point generator with the concatenated state $\boldsymbol{\zeta}$ as in (30) and the parameters computed as $\boldsymbol{Q}_{6D} = \text{diag}(\boldsymbol{Q}_{2D}, \boldsymbol{Q}_{2D}, \boldsymbol{Q}_{2D})$, $\boldsymbol{R}_{6D} = \text{diag}(\boldsymbol{R}_{2D}, \boldsymbol{R}_{2D}, \boldsymbol{R}_{2D})$, $\boldsymbol{P}_{6D} = \text{diag}(\boldsymbol{P}_{2D}, \boldsymbol{P}_{2D}, \boldsymbol{P}_{2D})$ and consequently, $\boldsymbol{K}_{6D} = \text{diag}(\boldsymbol{K}_{2D}, \boldsymbol{K}_{2D}, \boldsymbol{K}_{2D})$. Parameters' details are given in TABLE II.

Different from the previous case, for the admissible point collection, random initial conditions in $\{\boldsymbol{\zeta} \in \mathbb{R}^6 : |\boldsymbol{\zeta}| \leq 1.5\}$. The toolbox given in [31] is employed to provide a randomly distributed initial conditions. Then, for each feasible initial condition, we sample along its trajectory to collect state-control pairs since the recursive feasibility of (21) ensures the existence of admissible pairs propagated from that initial state. The projections of those states are provided in Fig. 7.

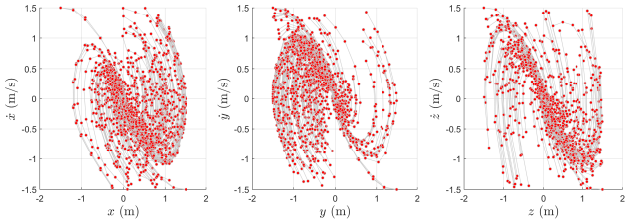

Fig. 7. Admissible pairs (red) collected for Case 2 (projected to each position-velocity axis).

Then the new set of basis functions for this setup is reselected as follows: The first basis function remains as the LQR-based quadratic function in the same choice from (31):

$$V_1(\boldsymbol{\zeta}) = \|\boldsymbol{\zeta}\|^2_{\boldsymbol{P}^{lqr}_{6D}}, \boldsymbol{\zeta} \in \mathbb{R}^6. \tag{35}$$

Moreover, taking advantage of the successfully generated CLFs for each subspace in (33) and (34), we adopt their basis by augmenting the vector $\boldsymbol{F}$ in (32), for $q \in \{x, y, z\}$, with

TABLE III
SPECIFICATIONS OF THE LP (6) FOR THE TWO CASES

	Case 1			Case 2
	$V^x(\boldsymbol{p}_x)$	$V^y(\boldsymbol{p}_y)$	$V^z(\boldsymbol{p}_z)$	$V(\boldsymbol{\zeta})$
No. basis functions N	100	100	100	10
No. admissible pairs M	625	625	625	6330
Computation time of (6)	0.03 (s)	0.03 (s)	0.05 (s)	0.11 (s)

the extended value assigned as 0. For example, for $q = x$, $[1\ 1]$ becomes $[1\ 1\ 0\ 0\ 0\ 0]$.

For the tuning of ℓ_i, as suggested before, it is essential to include the LQR-based quadratic Lyapunov function in the parameterized CLF in order to attain local optimality. Therefore, in both cases, the weighting ℓ_1 associated with the function is progressively decreased so that its corresponding coefficient computed from (6) is non-zero.

With the pairs sampled and the choice of basis functions given, the LP (6) provides the CLF as:

$$\begin{aligned} V(\boldsymbol{\zeta}) &= 0.0244\|\boldsymbol{\zeta}\|^2_{\boldsymbol{P}^{lqr}_{6D}} + 0.4508([0\ 1\ 0\ 0\ 0\ 0]\boldsymbol{\zeta})^4 \\ &+ 0.0674([1\ 1\ 0\ 0\ 0\ 0]\boldsymbol{\zeta})^4 + 0.1631([0\ 0\ 1\ 0\ 0\ 0]\boldsymbol{\zeta})^4 \\ &+ 0.1841([0\ 0\ 1\ 1\ 0\ 0]\boldsymbol{\zeta})^4 + 0.1103([0\ 0\ 0\ 0\ 0\ 1]\boldsymbol{\zeta})^4, \end{aligned} \tag{36}$$

with $V(\boldsymbol{\zeta})$ denoting the generated Lyapunov function for the six-dimensional system (30). The computational details of the procedure are provided in TABLE III. Therein, as expected, it is noticeable that the computation time for the LP (6) is relatively small in both cases, confirming the low complexity of the technique.

To this end, the CLFs were generated for the control problem and examined via simulation tests. In the next section, we verify the constructed CLFs via experimental validation, showing their applicability in practice.

V. EXPERIMENTAL VALIDATION

Herein, the results from the previous section are validated via experimental tests over nano-drone platforms.

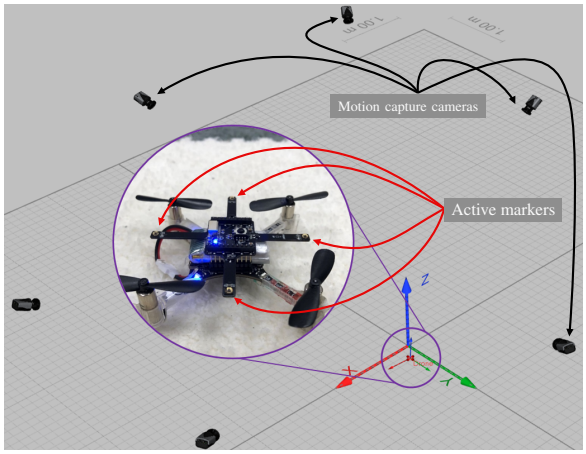

Fig. 8. The Crazyflie's motion restored with a system of cameras capturing the infrared lights from the active markers.

A. Experiment setup

In this section, the open-source quadcopter, Crazyflie 2.1, is used, since it allows customizing the position control within its hierarchical structure (See Fig. 9). We further assume that the integrated inner loop (the green block) is capable of sustaining the rotational dynamics in a properly high bandwidth compared to that of the outer loop, thereby ensuring that the roll and pitch angle commands from the outer loop are tracked with sufficient speed. A detailed description of this layer can be found in [32]. For the outer loop, we substitute our position control, which is designed for system (26) (or equivalently (30)) into the blue block to track the reference position $\boldsymbol{\xi}^{ref}$. The control action $\boldsymbol{u}_{pos}$ then transformed back to the inputs T, ϕ and θ by means of (25). Note that the normalized thrust T needs to be converted into the corresponding value in the digital scale of the nanodrone, ranging from 0 to 65535. The conversion is done via an interpolated curve identified experimentally (see Chapter 3, [33]). These signals are then packed with the value of the desired yaw rate, $\dot{\psi}^{ref} \triangleq 0$. These calculations are carried out in a station computer and and sent to the onboard inner loop via a USB radio dongle. The sampling time is fixed at $t_s = 100$ ms. For all the optimization problems employed in the experiments, the IPOPT solver is used within the Casadi optimization framework in Python 3.8.8, Intel Core, i5-10300H CPU @ 2.50GHz and 16GB RAM. The description of the test cases is given as follows.

On one hand, we use the controller (22) with the two setups *Case 1* and *Case 2* in the previous section with the same tuning given in TABLE II. The desired control $\boldsymbol{u}_d(\cdot)$ is chosen as $\boldsymbol{u}_d(\boldsymbol{p}_q) = -[k_p,\, k_d]\boldsymbol{p}_q$ and $\boldsymbol{u}_d(\boldsymbol{\zeta}) = -\text{diag}([k_p,\, k_d], [k_p,\, k_d], [k_p,\, k_d])\boldsymbol{\zeta}$ for Case 1 and 2, respectively, and $k_p = 1, k_d = 1.45$. Then, for comparison, we stabilize the tracking error dynamics by employing the following optimization problem in a receding horizon fashion.

$$\min \sum_{k=0}^{N_p^*} \left(\|\boldsymbol{\zeta}(k) - \boldsymbol{\zeta}^{ref}(k)\|^2_{\boldsymbol{Q}_{6D}} + \|\boldsymbol{u}_{pos}(k)\|^2_{\boldsymbol{R}_{6D}} \right) + \|\boldsymbol{\zeta}(N_p^*) - \boldsymbol{\zeta}^{ref}(N_p^*)\|^2_{\boldsymbol{P}_{6D}} \tag{37a}$$

$$\begin{cases} \boldsymbol{\zeta}(k+1) = (\boldsymbol{A}_{6D} + t_s \boldsymbol{I})\boldsymbol{\zeta}(k) + t_s \boldsymbol{B}_{6D}\boldsymbol{u}_{pos}(k), \\ \boldsymbol{u}_{pos}(k) \in \mathcal{U}_{pos} \text{ as in (29)}, \\ \boldsymbol{\zeta}(k) - \boldsymbol{\zeta}^{ref}(k) \in \mathcal{Z}, \boldsymbol{\zeta}(N_p^*) - \boldsymbol{\zeta}^{ref}(N_p^*) \in \mathcal{Z}_f, \end{cases} \tag{37b}$$

where $\boldsymbol{A}_{6D}, \boldsymbol{B}_{6D}, \boldsymbol{P}_{6D}, \boldsymbol{Q}_{6D}, \boldsymbol{R}_{6D}$ are given in Section IV-A, $\boldsymbol{\zeta}^{ref}(k)$ denotes the reference for $\boldsymbol{\zeta}(k)$ to follow. $\mathcal{Z} = \{\boldsymbol{\zeta} : \|\boldsymbol{\zeta}\|_\infty \leq 1.5\}$ is the state constraint corresponding to that of the admissible pair generator in Case 1 and 2. $\mathcal{Z}_f$ is the MPI set (as the terminal constraint) associated with the full-state gain $\boldsymbol{K}_{6D}$ given therein. For our setup, the largest prediction horizon N_p^* for (37) is 20, so that the online computation time does not exceed the sampling time t_s.

On the other hand, for the position reference, the examined controllers will first track a set-point $\boldsymbol{\xi}^{ref} = [0.6, 0.6, 0.8]^\top$ (m), noted as Scenario 1 (Sce. 1). Then, to further emphasize the computational advantage, we apply the same control laws for three Crazyflies in a centralized manner. The control then will be sent to each drone via the *Swarm* Python interface [34]. With this Scenario 2 (Sce. 2), a position swapping reference was given to the three drones, which is depicted Fig. 12.

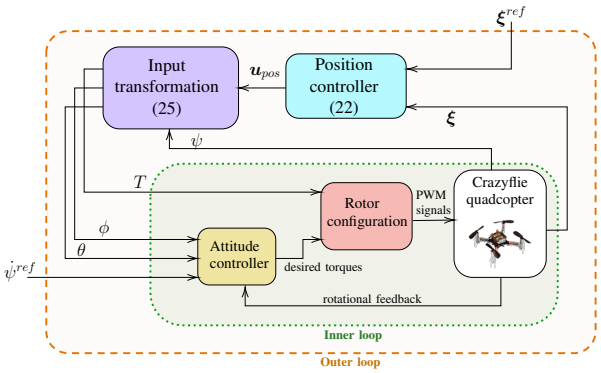

Fig. 9. Hierarchical control architecture of the Crazyflie quadcopter.

B. Experimental results and discussions

The result of the first scenario (Sce. 1) is shown in Fig. 10. It can be seen that, the proposed schemes' tracking performances (Case 1 and 2) are commensurate with the MPC while respecting all the input constraints in (24). However, it is noted from TABLE IV and Fig. 14 that the computation time (CT) of the proposed laws is 5 to 7 times lower than that of the MPC (37). This advantage directly comes from the simple QP formulation of (22) compared to (37) with regard to the complexity of both the cost and constraints.

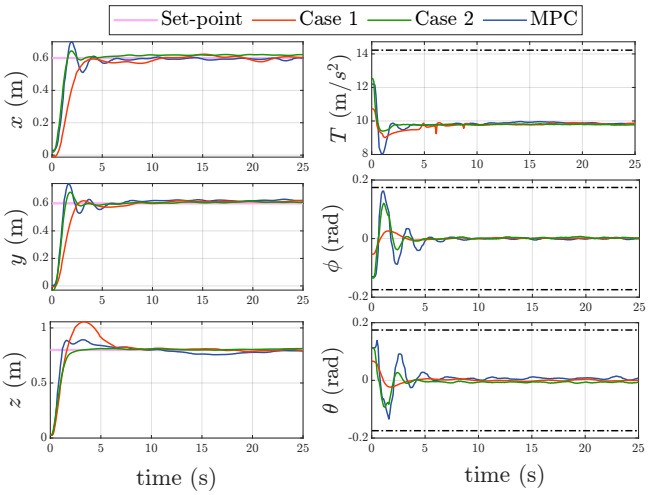

Fig. 10. Left: Set-point reference tracking experimented with the three scenarios. Right: The corresponding real input signals and their amplitude bound (dashed-dotted line).

In Sce. 2, the proposed schemes are also successfully validated. It is noticeable that in this scenario, a centralized MPC for three quadcopters could not be performed since the required calculation time exceeds the sampling time t_s. Meanwhile, even with three quadcopters, the tracking was possible for both Case 1 and 2 with the Avg. CT around 20ms. Fig. 11 depicts the value of the CLF in both cases

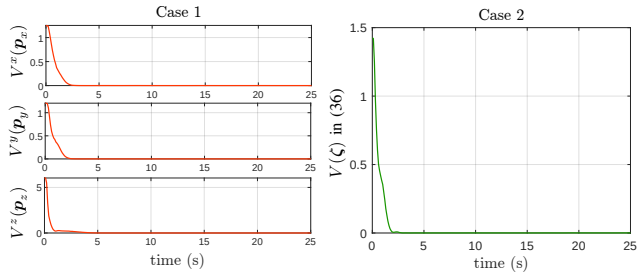

Fig. 11. Lyapunov function for Case 1 and 2 controllers in Scenario 1.

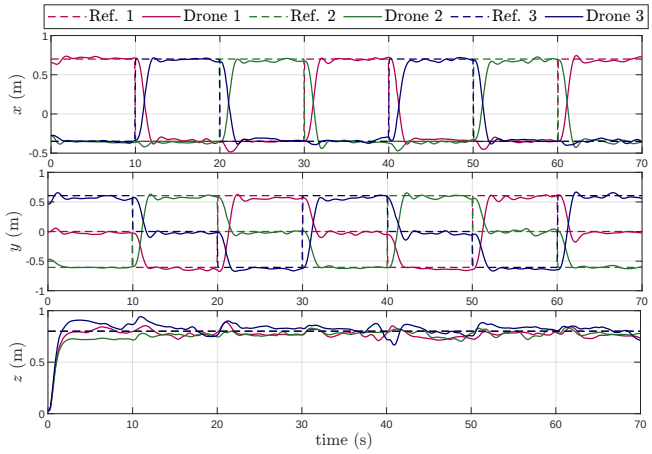

Fig. 12. Position swapping with three Crazyflies using Case 2 controller.

and their monotonicity. This shows that the discussed technique generates effective CLF and is applicable in real-life implementation.

VI. Conclusion

In this work, an offline synthesis procedure for CLF (Control Lyapunov Function) was proposed via linear programming. The idea relied on the stabilizing MPC (Model Predictive Control)-based software to generate admissible control in the state-space and parameterizing a collection of candidate CLFs. The generated CLF's advantage was then validated via quadcopter stabilization experiments. The result showed that the offline calculation of the CLF can be implemented via a linear program, thereby allowing CLF construction for systems with nontrivial dimensions. With the experimental results, the generated non-quadratic CLF, can be employed to reduce the online computational demand while maintaining stability. The future work will investigate how to efficiently choose the sampled data from MPC, hence, providing theoretical guarantees over a finite set of constraints, and allowing scalability.

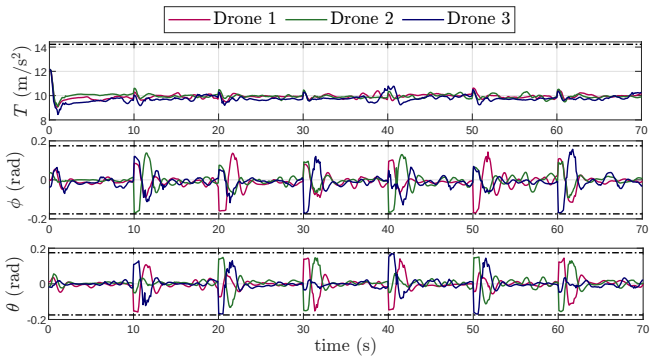

Fig. 13. The outer loop's input signals of the three quadcopters in Sce. 2 with the Case 2 controller.

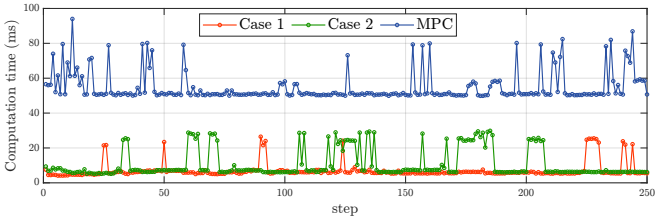

Fig. 14. Computation time in Scenario 1.

TABLE IV
Experiment Results

	Controller	RMS tracking error (cm)			Avg. CT
Sce. 1	Case 1	14.11			7.02 ms
	Case 2	11.39			10.74 ms
	MPC	11.48			54.70 ms
		Drone 1	Drone 2	Drone 3	
Sce. 2	Case 1	24.09	22.20	21.33	24.09 ms
	Case 2	19.14	18.57	18.71	20.51 ms
	MPC	×	×	×	×

References

[1] E. D. Sontag, "A 'universal'construction of Artstein's theorem on nonlinear stabilization," *Systems & control letters*, vol. 13, no. 2, 1989.

[2] M. Malisoff and E. D. Sontag, "Universal formulas for feedback stabilization with respect to minkowski balls," *Systems & Control Letters*, vol. 40, no. 4, pp. 247–260, 2000.

[3] N. Kidane, H. Nakamura, Y. Yamashita, and H. Nishitani, "Controller design for a nonlinear system with inputs restricted to a direct product of minkowski balls," in *2004 43rd IEEE Conference on Decision and Control*, vol. 4. IEEE, 2004, pp. 4069–4074.

[4] Y. Yamashita, R. Matsukizono, and K. Kobayashi, "Asymptotic stabilization of nonlinear systems with convex-polytope input constraints by continuous input," *Automatica*, vol. 138, p. 110032, 2022.

[5] H. Leyva, B. Aguirre-Hernández, and J. F. Espinoza, "Stabilization of affine systems with polytopic control value sets," *Journal of Dynamical and Control Systems*, pp. 1–13, 2023.

[6] R. A. Freeman and P. V. Kokotovic, "Inverse optimality in robust stabilization," *SIAM journal on control and optimization*, vol. 34, no. 4, pp. 1365–1391, 1996.

[7] R. Suarez, J. Solis-Daun, and B. Aguirre, "Global clf stabilization for systems with compact convex control value sets," in *40th IEEE Conf. on Decision and Control*, vol. 4. IEEE, 2001, pp. 3838–3843.

[8] S. Prajna, A. Papachristodoulou, and F. Wu, "Nonlinear control synthesis by sum of squares optimization: A lyapunov-based approach," in *5th Asian control conference*, vol. 1. IEEE, 2004, pp. 157–165.

[9] G. Chesi, "LMI techniques for optimization over polynomials in control: a survey," *IEEE Trans. on Automatic Control*, vol. 55, no. 11, 2010.

[10] T. Hu and Z. Lin, "Composite quadratic lyapunov functions for constrained control systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 3, pp. 440–450, 2003.

[11] M. Lazar and A. Jokić, "On infinity norms as lyapunov functions for piecewise affine systems," in *13th ACM international conference on Hybrid systems: computation and control*, 2010, pp. 131–140.

[12] R. Lavaei and L. J. Bridgeman, "Systematic, lyapunov-based, safe and stabilizing controller synthesis for constrained nonlinear systems," *IEEE Trans. on Automatic Control*, 2023.

[13] T. R. V. Steentjes, M. Lazar, and A. I. Doban, "Construction of continuous and piecewise affine feedback stabilizers for nonlinear systems," *IEEE Trans. on Automatic Control*, vol. 66, no. 9, pp. 4059–4068, 2020.

[14] Y. Min, S. M. Richards, and N. Azizan, "Data-driven control with inherent lyapunov stability," *arXiv preprint arXiv:2303.03157*, 2023.

[15] M. Lazar and A. Jokic, "Synthesis of trajectory-dependent control Lyapunov functions by a single linear program," in *International Workshop on Hybrid Systems: Computation and Control*. Springer, 2009.

[16] M. Lazar, "Flexible control lyapunov functions," in *2009 American Control Conference*. IEEE, 2009, pp. 102–107.

[17] R. W. Beard, G. N. Saridis, and J. T. Wen, "Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation," *Automatica*, vol. 33, no. 12, pp. 2159–2177, 1997.

[18] C. O. Aguilar and A. J. Krener, "Numerical solutions to the Bellman equation of optimal control," *Journal of optimization theory and applications*, vol. 160, pp. 527–552, 2014.

[19] P. Giesl and S. Hafstein, "Review on computational methods for lyapunov functions," *Discrete and Continuous Dynamical Systems-B*, vol. 20, no. 8, pp. 2291–2331, 2015.

[20] F. A. Bayer, F. D. Brunner, M. Lazar, M. Wijnand, and F. Allgöwer, "A tube-based approach to nonlinear explicit mpc," in *IEEE 55th Conference on Decision and Control*. IEEE, 2016, pp. 4059–4064.

[21] M. Hosseinzadeh, B. Sinopoli, I. Kolmanovsky, and S. Baruah, "Robust to early termination model predictive control," *IEEE Transactions on Automatic Control*, 2023.

[22] R. A. Vitale, "L_p metrics for compact, convex sets," *Journal of Approximation Theory*, vol. 45, no. 3, pp. 280–287, 1985.

[23] F. Blanchini, S. Miani, and F. A. Pellegrino, "Suboptimal receding horizon control for continuous-time systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 1081–1086, 2003.

[24] R. T. Rockafellar, *Convex analysis*. Princeton university press, 1997, vol. 11.

[25] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[26] S. Grammatico, F. Blanchini, and A. Caiti, "Control-sharing and merging control lyapunov functions," *IEEE Transactions on Automatic Control*, vol. 59, no. 1, pp. 107–119, 2013.

[27] J. Kim, M.-S. Kang, and S. Park, "Accurate modeling and robust hovering control for a quad-rotor vtol aircraft," in *Selected papers from the 2nd International Symposium on UAVs, Reno, Nevada, USA June 8–10, 2009*. Springer, 2010, pp. 9–26.

[28] N. T. Nguyen, I. Prodan, and L. Lefèvre, "Flat trajectory design and tracking with saturation guarantees: a nano-drone application," *International Journal of Control*, vol. 93, no. 6, pp. 1266–1279, 2020.

[29] H.-T. Do and I. Prodan, "Indoor experimental validation of mpc-based trajectory tracking for a quadcopter via a flat mapping approach," in *2023 European Control Conference (ECC)*, 2023, pp. 1–6.

[30] F. Blanchini and S. Miani, "A new class of universal Lyapunov functions for the control of uncertain linear systems," *IEEE Transactions on Automatic Control*, vol. 44, no. 3, pp. 641–647, 1999.

[31] T. Benham. (2011) Uniform distribution over a convex polytope. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/34208-uniform-distribution-over-a-convex-polytope

[32] Z. Huang, R. Bauer, and Y.-J. Pan, "Closed-loop identification and real-time control of a micro quadcopter," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 3, pp. 2855–2863, 2021.

[33] J. Förster, "System identification of the crazyflie 2.0 nano quadrocopter," B.S. thesis, ETH Zurich, 2015.

[34] "Step-by-step: Swarm interface," (Accessed on 23/08/2023). [Online]. Available: https://www.bitcraze.io/documentation/repository/crazyflie-lib-python/master/user-guides/sbs_swarm_interface/

Huu-Thinh Do received the Engineering degree in Mechatronics from the Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam, in 2021. He is currently pursuing the Ph.D. degree in Control Engineering at the Laboratory of Conception and Integration of Systems (LCIS), Grenoble INP, Univ. Grenoble Alpes, Valence, France. His research interests include differentially flat systems, optimization-based control, and their applications in autonomous systems.

Franco Blanchini was born on 29 December 1959, in Legnano (Italy). He is the Director of the Laboratory of System Dynamics at the University of Udine. He has been involved in the organisation of several international events: in particular, he was Program Vice-Chairman of the conference Joint CDC-ECC 2005, Seville, Spain; Program Vice-Chairman of the Conference CDC 2008, Cancun, Mexico; Program Chairman of the Conference ROCOND, Aalborg, Denmark, June 2012 and Program Vice-Chairman of the Conference CDC 2013, Florence, Italy. He is co-author of the book "Set theoretic methods in control", Birkhäuser. He received the 2001 ASME Oil & Gas Application Committee Best Paper Award as a co-author of the article "Experimental evaluation of a High-Gain Control for Compressor Surge Instability", the 2002 IFAC prize Survey Paper Award as the author of the article "Set Invariance in Control - a survey", Automatica, November 1999, for which he also received the High Impact Paper Award in 2017, and the 2017 NAHS Best Paper Award as a co-author of the article "A switched system approach to dynamic race modelling", Nonlinear Analysis: Hybrid Systems, 2016. He was nominated Senior Member of the IEEE in 2003. He has been an Associate Editor for Automatica (1996–2006 and 2017–2019) and for the IEEE Transactions on Automatic Control (2012–2016), and a Senior Editor for the IEEE Control Systems Letters.

Stefano Miani was born in Parma, Italy, in 1967. He received the electrical engineering Laurea degree and the Ph.D. degree in control engineering from the University of Padova, Italy, in 1993 and 1996, respectively. In 1997 he joined the University of Padova as an Assistant Professor. Since November 1998 he is at the University of Udine, Italy, where he currently holds an Associate Professor position.

He has been member of the editorial board of the "Nonlinear Dynamics and Systems Theory" journal from 2001 to 2002, member of the organizing committees for the "Advanced Manufacturing and System Technology conference" from 1998 to 2001, for the Conference on Decision and Control in 2009 and 2012, chair of the IFAC Workshop on Uncertain Dynamical Systems in 2011.

He is currently member of the IFAC Technical Committee on Robust Control and Associate Editor for IEEE TAC. His research results, which include the areas of constrained control, ℓ_∞ disturbance attenuation problems, gain scheduling control, uncertain production–distribution systems via set-valued techniques, and switching controllers, have been published in more than 80 international papers and an international book.

Ionela Prodan received the B.S. in Automation and Applied Informatics degree from the Univ. Politehnica of Bucharest, Romania in 2009 and her Ph.D. in Control Engineering from Supélec, Gif-sur-Yvette, France in 2012. She continued with a post-doctoral fellowship within the Chair on Systems Science and the Energetic Challenge - EDF, École Centrale Paris, France. Since 2014 she has been an Associate Professor at the Laboratory of Conception and Integration of Systems (LCIS) of Grenoble INP, Univ. Grenoble Alpes, Valence, France. She obtained her HDR (Habilitation to Conduct Research) in 2020. Her research interests are multi-disciplinary with a core expertise in control and applied mathematics. These encompass constrained optimization-based control (Model Predictive Control via distributed and decentralized approaches), mixed-integer programming, differential flatness, set-theoretic methods, and their application to motion planning for autonomous vehicles and energy management in building-scale DC microgrids. Dr. Prodan is a member of the IFAC Technical Committee 6.3. Power and Energy Systems and an Associate Editor for the EUCA European Control Conference.