Optimizing Control using Control Lyapunov Functions and Linear Programming

Sigurdur Freyr Hafstein[*,1], Patrick Schmidt[2], and Stefan Streif[2]

Abstract— **This paper presents an algorithm to optimize a given stabilizing controller by using control Lyapunov functions. The approach consists of two steps. Using linear programming, it first determines a Lyapunov function for the closed-loop system that is obtained by using the initial controller. Second, it determines a controller that is optimal w.r.t. a given objective function, which tightens the input constraints and penalizes large variations of the control. A case study considers a predator-prey model with inputs and shows how the algorithm improves a given controller in the described manner.**

I. Introduction

Control theory provides a variety of techniques to compute a stabilizing controller. To choose a suitable controller for a particular application, a criterion in form of an objective function can often be used. This allows to weight deviations in the state and the input differently to put the focus rather on a fast convergence or small inputs. Many applications also require that the designed controller fulfills constraints on the states and the input. Approaches like model predictive control gained a lot of attention in the last years due to their ability to deal with constraints [2], [5], [10], [17], [40]. However, they yield an optimal decision only at the current state and are computationally expensive.

There exist different approaches and extensions in the literature to satisfy input constraints, such as an extension of Sontag's formula [25], [37] or backstepping [7], [20], [41] and there exists a variety of approaches that require a smooth control Lyapunov function (CLF), such as passivity-based control [8], [31], [42] and backstepping [34], [38]. Further, there exist approaches like nonsmooth backstepping [30], [33] to extend existing methods to nonsmooth CLFs. Besides these adapted methods, also nonsmooth stabilization techniques like steepest descent, Dini aiming [22], [23], optimization-based feedback/control [6], and stabilization via infimum convolutions [32], [36] are based on different subgradients [9] of the nonsmooth CLF to determine a descend direction and to ensure a decay to the origin.

Even if determining a Lyapunov function (LF) for a closed-loop system with a fixed control is a demanding task and determining a CLF is an even more demanding task, in particular for real-world applications and non-academic examples, the effort is often worth it. Approaches to determine either a LF or a CLF cover both analytical [4], [27] as well as numerical approaches [3], [13], [18]. It becomes clear that a CLF helps to derive a stabilizing controller, but some approaches to determine a LF consider systems where a controller is already applied. Therefore, the paper considers this problem by starting with an arbitrary stabilizing controller for an input-affine system. Applying it results in an autonomous system, where a numerical algorithm is used to determine a LF given as a continuous and piecewise affine (CPA) function. As a novel contribution, the CPA algorithm is then used in a similar manner to optimize the given control in the sense that its upper and lower bounds get tightened and that the control varies only slightly between neighboring points.

The rest of the article is organized as follows: Section II introduces the CPA method, which is used in Section III to improve a given controller. The approach is applied to the well-known predator-prey model of Lotka and Volterra in Section IV, before the paper is concluded in Section V.

II. The CPA Method to compute Lyapunov functions

Consider the system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \tag{1}$$

where $\mathbf{f}\colon \mathbb{R}^n \to \mathbb{R}^n$ is a C^2 vector field with $\mathbf{f}(0) = 0$. If the equilibrium at the origin is asymptotically stable, then the system possesses a Lyapunov function V, i.e. a real-valued function from the state-space $\mathbb{R}^n$, that has a minimum at the origin and is decreasing along all solution trajectories in a neighbourhood of the origin [26], [24]. Further, since compact sublevel-sets $\{\mathbf{x} \in \mathbb{R}^n \colon V(\mathbf{x}) \le c\}$, $c > 0$, of V are in the equilibrium's basin of attraction, one is often interested in having the domain of V as large as possible. The analytic generation of a Lyapunov function for a given system is usually very difficult or impossible. Therefore one often resorts to numerical methods to compute a Lyapunov function, see e.g. the review [14]. In this paper we adapt the so-called CPA method, that uses linear programming to parameterize a continuous, piecewise affine Lyapunov function on a triangulation of the state-space, see [12] or [28] and [21] for earlier versions.

The CPA method can be summarized as follows: Given is the system (1) and a triangulation $\mathscr{T} = (\mathfrak{S}_\nu)_{\nu \in I}$, I an index set, of a neighbourhood $\mathscr{D} \subset \mathbb{R}^n$ of the origin, i.e. a subdivision of $\mathscr{D}$ into n-simplices $\mathfrak{S}_\nu = \mathrm{co}(\mathbf{x}_0^\nu, \mathbf{x}_1^\nu, \ldots, \mathbf{x}_n^\nu)$ such that different simplices intersect in a common face

*This research was partially supported by the Icelandic Research Fund in grant number 228725-051.

[1]Sigurdur Freyr Hafstein is with the Science Institute, University of Iceland, Dunhagi 5, 107 Reykjavik, Iceland shafstein@hi.is

[2]Patrick Schmidt and Stefan Streif are with the Chemnitz University of Technology, Professorship Automatic Control and System Dynamics, 09107 Chemnitz {patrick.schmidt, stefan.streif}@etit.tu-chemnitz.de

if they intersect at all. Here $\mathrm{co}(\mathbf{x}_0^\nu, \mathbf{x}_1^\nu, \ldots, \mathbf{x}_n^\nu)$ denotes the convex hull of the affinely independent vectors $\mathbf{x}_0^\nu, \mathbf{x}_1^\nu, \ldots, \mathbf{x}_n^\nu$, which are called the vertices of the simplex $\mathfrak{S}_\nu$. We denote the set of all vertices of all the simplices in $\mathscr{T}$ by $\mathscr{V}$ and we demand that $0 \in \mathscr{V}$, i.e. that the origin is a vertex.

The CPA method constructs a linear programming problem for the system with variables $V_{\mathbf{x}} \in \mathbb{R}$, $\mathbf{x} \in \mathscr{V}$, which are the values of the Lyapunov function V at the vertices, that should fulfill

$$V_0 = 0 \quad \text{and } V_{\mathbf{x}} \geq \|\mathbf{x}\| \text{ for all other } \mathbf{x} \in \mathscr{V} \tag{2}$$

and, for every simplex $\mathfrak{S}_\nu = \mathrm{co}(\mathbf{x}_0^\nu, \mathbf{x}_1^\nu, \ldots, \mathbf{x}_n^\nu) \in \mathscr{T}$ and $i = 0, 1, \ldots, n$,

$$\nabla V_\nu \cdot \mathbf{f}(\mathbf{x}_i^\nu) + \|\nabla V_\nu\|_1 E_i^\nu(\mathbf{f}) \leq -\|\mathbf{x}_i^\nu\|. \tag{3}$$

Here ∇V_ν is the gradient of V on $\mathfrak{S}_\nu$, explained below, $\cdot$ is the scalar product and $\|\cdot\|$ the Euclidian norm, and the system specific constants $E_i^\nu(\mathbf{f})$ are defined using Lemma 1 and equation (6) below.

Given a feasible solution to this linear programming problem, i.e. values $V_{\mathbf{x}} \in \mathbb{R}$, $\mathbf{x} \in \mathscr{V}$, such that the constraints (2) and (3) are fulfilled, we can define a function $V : \mathscr{D} \to \mathbb{R}$ using the following: for every $\mathbf{x} \in \mathscr{D}$ there is a simplex $\mathfrak{S}_\nu$ such that $\mathbf{x} \in \mathfrak{S}_\nu$ and $\mathbf{x} = \sum_{i=0}^n \lambda_i \mathbf{x}_i^\nu$ can be written uniquely as a convex combination of its vertices. We set

$$V(\mathbf{x}) = V\left(\sum_{i=0}^n \lambda_i \mathbf{x}_i^\nu\right) := \sum_{i=0}^n \lambda_i V_{\mathbf{x}_i^\nu},$$

and it can be shown that V defined in this way is continuous. Further, because the values of V at the vertices $\mathscr{V}$ of $\mathscr{T}$ fulfill the constraints (2) and (3), it can be shown that $V(0) = 0$, and for every $\mathbf{x} \in \mathscr{D}^\circ$ (interior of $\mathscr{D}$) we have $V(\mathbf{x}) \geq \|\mathbf{x}\|$, and

$$D^+V(\mathbf{x}) := \limsup_{h \to 0+} \frac{V(\mathbf{x} + h\mathbf{f}(\mathbf{x})) - V(\mathbf{x})}{h} \leq -\|\mathbf{x}\|,$$

see e.g. [11, Thm. 4.6] and [12, Thm. 1] for a proof. The term $D^+V(\mathbf{x})$ is the so-called Dini-derivative of V along the solutions of (1) and assures that V is decreasing along all solution trajectories, see e.g. [39, App. B.I] or [29, Thms. 1.16, 1.17].

For a simplex $\mathfrak{S}_\nu = \mathrm{co}(\mathbf{x}_0^\nu, \mathbf{x}_1^\nu, \ldots, \mathbf{x}_n^\nu)$ we define its shape-matrix

$$X_\nu := \begin{bmatrix} \mathbf{x}_1^\nu - \mathbf{x}_0^\nu & \mathbf{x}_2^\nu - \mathbf{x}_0^\nu & \ldots & \mathbf{x}_n^\nu - \mathbf{x}_0^\nu \end{bmatrix}^T \in \mathbb{R}^{n \times n}.$$

It can be shown that with

$$\nabla V_\nu := X_\nu^{-T} \begin{bmatrix} V_{\mathbf{x}_1^\nu} - V_{\mathbf{x}_0^\nu} \\ V_{\mathbf{x}_2^\nu} - V_{\mathbf{x}_0^\nu} \\ \vdots \\ V_{\mathbf{x}_n^\nu} - V_{\mathbf{x}_0^\nu} \end{bmatrix}$$

we have

$$V(\mathbf{x}) = \nabla V_\nu \cdot (\mathbf{x} - \mathbf{x}_0^\nu) + V_{\mathbf{x}_0^\nu}$$

for all $\mathbf{x} \in \mathfrak{S}_\nu$ and all $\mathfrak{S}_\nu \in \mathscr{T}$. Hence, the constant vector $\nabla V_\nu \in \mathbb{R}^n$ is the gradient of V on $\mathfrak{S}_\nu^\circ$.

The following from [29, Lemma 4.16] is fundamental for the CPA method and we will need it later for our novel approach.

Lemma 1: On an n-simplex $\mathfrak{S} := \mathrm{co}(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n) \subset \mathbb{R}^n$, we have for a function $g \in C^2(\mathfrak{S})$ and every $\mathbf{x} = \sum_{i=0}^n \lambda_i \mathbf{x}_i \in \mathfrak{S}$ and any $d \in \{0, 1, \ldots, n\}$, that

$$\left| g\Big(\sum_{i=0}^n \lambda_i \mathbf{x}_i\Big) - \sum_{i=0}^n \lambda_i g(\mathbf{x}_i) \right| \leq \sum_{i=0}^n \lambda_i E_i,$$

where

$$E_i \geq \sum_{r,s=1}^n \frac{B_{rs}}{2} |[\mathbf{x}_i - \mathbf{x}_d]_r| (|[\mathbf{x}^* - \mathbf{x}_d]_s| + |[\mathbf{x}_i - \mathbf{x}_d]_s|) \tag{4}$$

for every vertex $\mathbf{x}^* \in \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n\}$, $[\mathbf{y}]_i$ is the ith component of the vector $\mathbf{y}$ and

$$B_{rs} := \max_{\mathbf{x} \in \mathfrak{S}} \left| \frac{\partial^2 g}{\partial x_r \partial x_s}(\mathbf{x}) \right|. \tag{5}$$

Recall that since $\mathfrak{S} \subset \mathbb{R}^n$ is a closed set such that $\mathfrak{S} = \overline{\mathfrak{S}^\circ}$, $g \in C^2(\mathfrak{S})$ means that g has continuous second-order derivatives on $\mathfrak{S}^\circ$ that can be extended continuously to $\mathfrak{S}$. $\square$

Motivated by Lemma 1, we define for a C^2 vector field $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ the constants

$$B_{rs}^\nu(\mathbf{f}) := \max_{\substack{\mathbf{x} \in \mathfrak{S}_\nu \\ i = 1, 2, \ldots, n}} \left| \frac{\partial^2 f_i}{\partial x_r \partial x_s}(\mathbf{x}) \right|, \tag{6}$$

for $r, s = 1, 2, \ldots, n$ and $\mathfrak{S}_\nu \in \mathscr{T}$, and the constants $E_i^\nu(\mathbf{f})$ as the E_i in (4) using these $B_{rs}^\nu(\mathbf{f})$ for the B_{rs} and $\mathfrak{S}_\nu$ for $\mathfrak{S}$. Then

$$\left\| \mathbf{f}\Big(\sum_{i=0}^n \lambda_i \mathbf{x}_i^\nu\Big) - \sum_{i=0}^n \lambda_i \mathbf{f}(\mathbf{x}_i^\nu) \right\|_\infty \leq \sum_{i=0}^n \lambda_i E_i^\nu(\mathbf{f})$$

for all $\mathbf{x} = \sum_{i=0}^n \lambda_i \mathbf{x}_i^\nu \in \mathfrak{S}_\nu = \mathrm{co}(\mathbf{x}_0^\nu, \mathbf{x}_1^\nu, \ldots, \mathbf{x}_n^\nu)$. Note that in (4) we have upper bounds on the second order derivatives, which might be conservative, but as these upper bounds are multiplied with numbers that are less than the square of the diameter of the simplex $\mathfrak{S}$, this conservatism is not limiting in theory or practice. For a quite detailed discussion on the $E_i^\nu(\mathbf{f})$ and how to implement them see [16].

We will not discuss further the details of the CPA method here, but refer the interested reader to [12] and [19], where it is shown that the CPA method can parameterize a Lyapunov function for the system (1) whenever the origin is exponentially stable.

III. METHOD

We consider the control affine system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + G(\mathbf{x})\mathbf{u}(\mathbf{x}), \quad G(\mathbf{x}) \in \mathbb{R}^{n \times m}, \tag{7}$$

with the control $\mathbf{u} \colon \mathbb{R}^n \to \mathbb{R}^m$. Let a triangulation $\mathscr{T}$ as in the last section be given and assume $\mathbf{u}$ is given by

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}\Big(\sum_{i=0}^n \lambda_j \mathbf{x}_i^\nu\Big) = \sum_{i=0}^n \lambda_j \mathbf{u}(\mathbf{x}_i^\nu), \tag{8}$$

i.e. $\mathbf{u}$ is a vector-valued CPA function.

A. General idea

In the first step in our approach we use a given control $\mathbf{u}_a$ and use its CPA interpolation $\mathbf{u}$ to obtain a closed-loop system (7), for which we compute a Lyapunov function. That is, we set $\mathbf{u}(\mathbf{x}) = \mathbf{u}_a(\mathbf{x})$ for every $\mathbf{x} \in \mathscr{V}$ and then interpolate as in (8). To compute a Lyapunov function for the closed-loop system (7) using the CPA method we set $\widetilde{\mathbf{f}} = \mathbf{f} + G\mathbf{u}$: The variables of the linear programming problem are $V_{\mathbf{x}} \in \mathbb{R}$ for all vertices $\mathbf{x} \in \mathscr{V}$ and the constraints are

$$\nabla V_\nu \cdot \widetilde{\mathbf{f}}(\mathbf{x}_i^\nu) + \|\nabla V_\nu\|_1 E_i^\nu(\widetilde{\mathbf{f}}) \leq -\|\mathbf{x}_i^\nu\|, \tag{9}$$

where $E_i^\nu(\widetilde{\mathbf{f}})$ is as in (4) with

$$B_{rs} = B_{rs}^\nu(\mathbf{f}) + B_{rs}^\nu(G\mathbf{u}). \tag{10}$$

For the second step we keep the CPA Lyapunov function V computed in the first step constant and compute a new CPA control $\mathbf{u}$, such that V remains a Lyapunov function for the closed-loop system. This is also done by using the CPA method with the following linear programming problem:
For all vertices $\mathbf{x}_i^\nu$ of the simplices in $\mathscr{T}$, the variables of the linear programming problem are $u_k(\mathbf{x}_i^\nu) \in \mathbb{R}$ with $k = 1, 2, \ldots, m$. The constraints of the linear programming problem are given as

$$\nabla V_\nu \cdot \widetilde{\mathbf{f}}(\mathbf{x}_i^\nu) + \|\nabla V_\nu\|_1 E_i^\nu(\widetilde{\mathbf{f}}) \leq -\|\mathbf{x}_i^\nu\|. \tag{11}$$

In both linear programming problems, in the first and the second step, we also include the constraints

$$V_0 = 0 \quad \text{and} \quad V_{\mathbf{x}} \geq \|\mathbf{x}\|$$

for all vertices $\mathbf{x} \in \mathscr{V} \setminus \{0\}$ of the triangulation $\mathscr{T}$, which enforces $V(0) = 0$ and $V(\mathbf{x}) \geq \|\mathbf{x}\|$ for all $\mathbf{x} \in \mathscr{D} \setminus \{0\}$.

That is, in both linear programs we have the same constraints and the only difference is that we have different sets of variables. Further note that in the first step the $B_{rs}^\nu(G\mathbf{u})$ in $E_i^\nu(\widetilde{\mathbf{f}})$ are constant, but in the second step they do depend on the variables $u_k(\mathbf{x}_i^\nu)$. Similarly, the ∇V_ν depend on the variables $V_{\mathbf{x}}$ in the first step, but are constants in the second step.

It follows by Lemma 1 that if the conditions (9) or (11) are fulfilled, then

$$\nabla V_\nu \cdot \widetilde{\mathbf{f}}(\mathbf{x}) \leq -\|\mathbf{x}\|$$

for all $\mathbf{x} \in \mathscr{D}$. Just note that for an $\mathbf{x} \in \mathscr{D}$ and $\mathfrak{S}_\nu \in \mathscr{T}$ such that

$$\mathbf{x} = \sum_{i=0}^n \lambda_i \mathbf{x}_i^\nu \in \mathfrak{S}_\nu$$

is a convex combination of its vertices, we get by Hölder's inequality, Lemma 1, and (11), that

$$\begin{aligned}
\nabla V_\nu \cdot \widetilde{\mathbf{f}}(\mathbf{x}) &= \nabla V_\nu \cdot \left(\widetilde{\mathbf{f}}(\mathbf{x}) - \sum_{i=0}^n \lambda_i \widetilde{\mathbf{f}}(\mathbf{x}_i^\nu) \right) + \sum_{i=0}^n \lambda_i \nabla V_\nu \cdot \widetilde{\mathbf{f}}(\mathbf{x}_i^\nu) \\
&\leq \|\nabla V_\nu\|_1 \left\| \widetilde{\mathbf{f}}(\mathbf{x}) - \sum_{i=0}^n \lambda_i \widetilde{\mathbf{f}}(\mathbf{x}_i^\nu) \right\|_\infty + \sum_{i=0}^n \lambda_i \nabla V_\nu \cdot \widetilde{\mathbf{f}}(\mathbf{x}_i^\nu) \\
&\leq \sum_{i=0}^n \lambda_i \left(\|\nabla V_\nu\|_1 E_i^\nu(\widetilde{\mathbf{f}}) + \nabla V_\nu \cdot \widetilde{\mathbf{f}}(\mathbf{x}_i^\nu) \right) \\
&\leq \sum_{i=0}^n \lambda_i (-\|\mathbf{x}_i^\nu\|) \leq -\left\| \sum_{i=0}^n \lambda_i \mathbf{x}_i^\nu \right\| = -\|\mathbf{x}\|.
\end{aligned}$$

The idea is now to use the objective of the second linear programming problem to obtain a control that is in some sense better than the original control $\mathbf{u}_a$. Our suggestion is to compute a control $\mathbf{u}\colon \mathscr{D} \to \mathbb{R}^m$ in the second step that minimizes

$$\|\mathbf{u}\|_\infty + w \cdot \max_{k=1,2,\ldots,m} \|\nabla u_k\|_\infty \tag{12}$$

for some parameter $w > 0$. The first term is to minimize the control action $\mathbf{u}$. The second term with the $w > 0$ is to obtain a control that does not vary too much; this will become more clear in the example in Section IV.

B. Computing the needed constants

To use our proposed method, we need to compute the constants $B_{rs}^\nu(\mathbf{f})$ and $B_{rs}^\nu(G\mathbf{u})$ in (10). Obtaining bounds for the $B_{rs}^\nu(\mathbf{f})$ is straight forward from formula (6), so we consider the constants $B_{rs}^\nu(G\mathbf{u})$. First note that by (8), on the simplex $\mathfrak{S}_\nu$, the jth component of $\mathbf{u}$ can be written as

$$u_j(\mathbf{x}) = \nabla u_j \cdot (\mathbf{x} - \mathbf{x}_0^\nu) + u_j(\mathbf{x}_0^\nu)$$

and with

$$\widetilde{\mathbf{u}}_j := \left[u_j(\mathbf{x}_1^\nu) - u_j(\mathbf{x}_0^\nu), \quad \ldots \quad , u_j(\mathbf{x}_n^\nu) - u_j(\mathbf{x}_0^\nu) \right]^T \in \mathbb{R}^n$$

we have, similarly as for ∇V_ν, that

$$\nabla u_j = X_\nu^{-T} \widetilde{\mathbf{u}}_j.$$

The jth component $[G\mathbf{u}]_j$ of the vector valued function $G\mathbf{u}$ is

$$[G\mathbf{u}]_j = \sum_{k=1}^m G_{jk} u_k$$

and we compute

$$\begin{aligned}
\frac{\partial^2}{\partial x_r \partial x_s} [G\mathbf{u}]_j &= \frac{\partial^2}{\partial x_r \partial x_s} \sum_{k=1}^m G_{jk} u_k \\
&= \frac{\partial}{\partial x_r} \sum_{k=1}^m \left(\frac{\partial G_{jk}}{\partial x_s} u_k + G_{jk} \frac{\partial u_k}{\partial x_s} \right) \\
&= \sum_{k=1}^m \left(\frac{\partial^2 G_{jk}}{\partial x_r \partial x_s} u_k + \frac{\partial G_{jk}}{\partial x_s} \frac{\partial u_k}{\partial x_r} + \frac{\partial G_{jk}}{\partial x_r} \frac{\partial u_k}{\partial x_s} \right)
\end{aligned}$$

because u_k is linear in $\mathbf{x}$ and therefore $\frac{\partial^2 u_k}{\partial x_r \partial x_s} = 0$.

We now compute upper bounds for $B_{rs}(g)$ with $g(\mathbf{x}) = \nabla V_\nu \cdot G(\mathbf{x})\mathbf{u}(\mathbf{x})$:
With

$$G^*_{\nu,rs} := \max_{\substack{\mathbf{x}\in\mathfrak{S}_\nu \\ k=1,2,\dots,m \\ j=1,2,\dots,n}} \left| \frac{\partial^2 G_{jk}}{\partial x_r \partial x_s}(\mathbf{x}) \right|$$

and

$$G^*_{\nu,r} := \max_{\substack{\mathbf{x}\in\mathfrak{S}_\nu \\ k=1,2,\dots,m \\ j=1,2,\dots,n}} \left| \frac{\partial G_{jk}}{\partial x_r}(\mathbf{x}) \right|$$

we have, because $\max_{\mathbf{x}\in\mathfrak{S}_\nu} \|\mathbf{u}(\mathbf{x})\|_1 = \max_{i=0,1,\dots,n} \|\mathbf{u}(\mathbf{x}_i^\nu)\|_1$,

$$\sum_{j=1}^n \sum_{k=1}^m [\nabla V_\nu]_j \frac{\partial^2 G_{jk}}{\partial x_r \partial x_s} u_k \\ \leq G^*_{\nu,rs} \|\nabla V_\nu\|_1 \max_{i=0,1,\dots,n} \|\mathbf{u}(\mathbf{x}_i^\nu)\|_1,$$

$$\sum_{j=1}^n \sum_{k=1}^m [\nabla V_\nu]_j \frac{\partial G_{jk}}{\partial x_r} \frac{\partial u_k}{\partial x_s} \leq G^*_{\nu,r} \|\nabla V_\nu\|_1 \sum_{k=1}^m \left| \frac{\partial u_k}{\partial x_s} \right|,$$

and

$$\sum_{j=1}^n \sum_{k=1}^m [\nabla V_\nu]_j \frac{\partial G_{jk}}{\partial x_s} \frac{\partial u_k}{\partial x_r} \leq G^*_{\nu,s} \|\nabla V_\nu\|_1 \sum_{k=1}^m \left| \frac{\partial u_k}{\partial x_r} \right|.$$

Hence, by (5) we can set

$$B_{rs}(g) = \|\nabla V_\nu\|_1 \Bigg(G^*_{\nu,rs} \max_{i=0,1,\dots,n} \|\mathbf{u}(\mathbf{x}_i^\nu)\|_1 \\ + G^*_{\nu,r} \sum_{k=1}^m \left| \frac{\partial u_k}{\partial x_s} \right| + G^*_{\nu,s} \sum_{k=1}^m \left| \frac{\partial u_k}{\partial x_r} \right| \Bigg)$$

or, in accordance with (9) and (10),

$$B^\nu_{rs}(G\mathbf{u}) = G^*_{\nu,rs} \max_{i=0,1,\dots,n} \|\mathbf{u}(\mathbf{x}_i^\nu)\|_1 \\ + G^*_{\nu,r} \sum_{k=1}^m \left| \frac{\partial u_k}{\partial x_s} \right| + G^*_{\nu,s} \sum_{k=1}^m \left| \frac{\partial u_k}{\partial x_r} \right|,$$

where we note that $\frac{\partial u_k}{\partial x_r}$ and $\frac{\partial u_k}{\partial x_s}$ are constant numbers on $\mathfrak{S}_\nu$. A simplified sufficient bound for the last term is given with

$$U_\nu := \max_{\mathbf{x}\in\mathfrak{S}_\nu} \|\mathbf{u}(\mathbf{x})\|_\infty = \max_{i=0,1,\dots,n} \|\mathbf{u}(\mathbf{x}_i^\nu)\|_\infty$$

and

$$U^*_\nu := \max_{\substack{\mathbf{x}\in\mathfrak{S}_\nu \\ k=1,2,\dots,m}} \|\nabla u_k(\mathbf{x})\|_\infty = \max_{k=1,2,\dots,m} \|\nabla u_k\|_\infty$$

as

$$B^\nu_{rs}(G\mathbf{u}) = nG^*_{\nu,rs} U_\nu + m(G^*_{\nu,r} + G^*_{\nu,s}) U^*_\nu.$$

These simplified bounds are the bounds we will use in our example in the next section. Note that these sufficient bounds, which might be conservative, are, in the LP problems, multiplied with numbers that are less than the squares of the diameters of the simplices $\mathfrak{S}_\nu$, which can be made arbitrary small.

IV. Example: Predator-prey model

Consider the Lotka-Volterra predator-prey model

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \alpha x_1 - \beta x_1 x_2 \\ -\gamma x_2 + \delta x_1 x_2 \end{bmatrix}}_{=\mathbf{f}(x_1,x_2)} + \underbrace{\begin{bmatrix} x_1 & 0 \\ 0 & x_2 \end{bmatrix}}_{=G(x_1,x_2)} \begin{bmatrix} u_1(x_1,x_2) \\ u_2(x_1,x_2) \end{bmatrix}, \tag{13}$$

from [35], with additional controls u_1, u_2 as an input. The predator-prey model describes the interactions between two species x_1 (the prey) and x_2 (the predator). The parameters in (13) are given as $\alpha = 1.1$, $\beta = 0.4$, $\gamma = 0.4$, and $\delta = 0.1$ and are known as the prey growth rate, prey death rate, predator death rate, and predator growth rate, respectively. We want to stabilize this system at $\mathbf{x}^* = [10,4]^T$. This can be done using the control [35]

$$\mathbf{u}_a(x_1,x_2) = \begin{bmatrix} -\alpha + \beta x_2 + \tanh(-(x_1 - x_1^*)) \\ \gamma - \delta x_1 + \tanh(-(x_2 - x_2^*)) \end{bmatrix}. \tag{14}$$

We map the equilibrium $\mathbf{x}^*$ to the origin using the coordinate transform $[x_1,x_2]^T \mapsto [x_1 - x_1^*, x_2 - x_2^*]^T$, and consider the system (7) with

$$\mathbf{f}(x_1,x_2) = \begin{bmatrix} \alpha(x_1+x_1^*) - \beta(x_1+x_1^*)(x_2+x_2^*) \\ -\gamma(x_2+x_2^*) + \delta(x_1+x_1^*)(x_2+x_2^*) \end{bmatrix} \quad \text{and}$$

$$G(x_1,x_2) = \begin{bmatrix} x_1 + x_1^* & 0 \\ 0 & x_2 + x_2^* \end{bmatrix}.$$

In these coordinates the control (14) is given by

$$\mathbf{u}_a(x_1,x_2) = \begin{bmatrix} -\alpha + \beta(x_2 + x_2^*) + \tanh(-x_1) \\ \gamma - \delta(x_1 + x_1^*) + \tanh(-x_2) \end{bmatrix}. \tag{15}$$

In the example $n = m = 2$ and we can set

$$B^\nu_{12}(\mathbf{f}) = B^\nu_{21}(\mathbf{f}) = \max\{|\beta|,|\delta|\} \text{ and } B^\nu_{11}(\mathbf{f}) = B^\nu_{22}(\mathbf{f}) = 0.$$

Further, we can set

$$G^*_{\nu,rs} = 0 \quad \text{for all } r,s \in \{1,2\}$$

and

$$G^*_{\nu,1} = G^*_{\nu,2} = 1.$$

We used the triangulation $0.2\mathscr{T}^{\mathbf{F}}_{15}$, see Fig. 1; for a detailed discussion of this triangulation see [1], [15]. Note that the domain $\mathscr{D}$ of the triangulation is approximately a circular disc centered at the origin and with radius 3. It is subdivided into 1,800 triangles.

As control we used the CPA interpolation of the analytical control $\mathbf{u}_a$ in (15), see Figs. 2 and 3, and we computed a CPA Lyapunov function (see Fig. 4) as in the first step of the algorithm described in the last section. In the linear program we minimized $\max_{\nu\in I} \|\nabla V_\nu\|_1$ on $\mathscr{D}$. For the analytical control we have

$$\max_{\mathbf{x}\in\mathscr{D}} \|\mathbf{u}(\mathbf{x})\|_\infty = 2.44669 \quad \text{and} \quad \max_{\substack{k=1,2 \\ \mathbf{x}\in\mathscr{D}}} \|\nabla u_k(\mathbf{x})\|_\infty = 0.993386.$$

When we minimize $\max_{\mathbf{x}\in\mathscr{D}} \|\mathbf{u}(\mathbf{x})\|_\infty$ in the second step of the algorithm we obtain the control $\mathbf{u} = [u_1,u_2]^T$ in Figs. 5 and 6. For this function we have

$$\max_{\mathbf{x}\in\mathscr{D}} \|\mathbf{u}(\mathbf{x})\|_\infty = 1.23816.$$

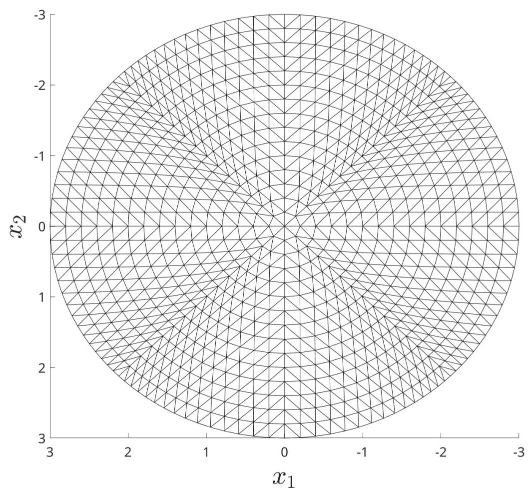

Fig. 1. The triangulation $0.2\mathscr{T}_{15}^{\mathbf{F}}$ used for our example in Section IV.

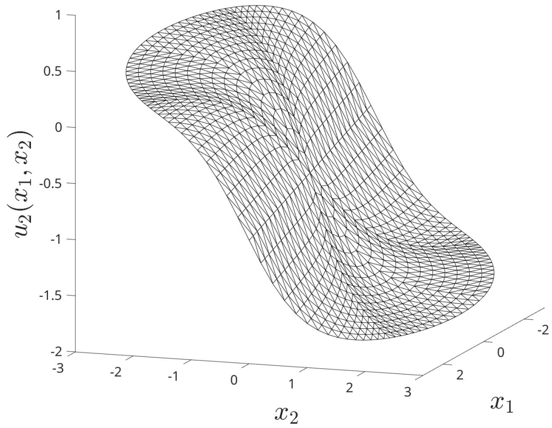

Fig. 3. The second component u_2 of the CPA interpolation of the analytic control $\mathbf{u}_a$ from (15).

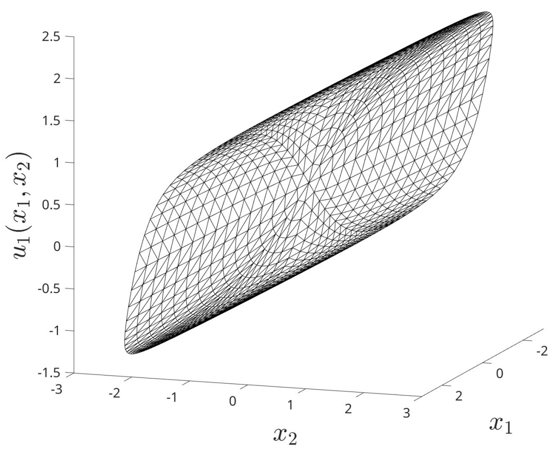

Fig. 2. The first component u_1 of the CPA interpolation of the analytic control $\mathbf{u}_a$ from (15).

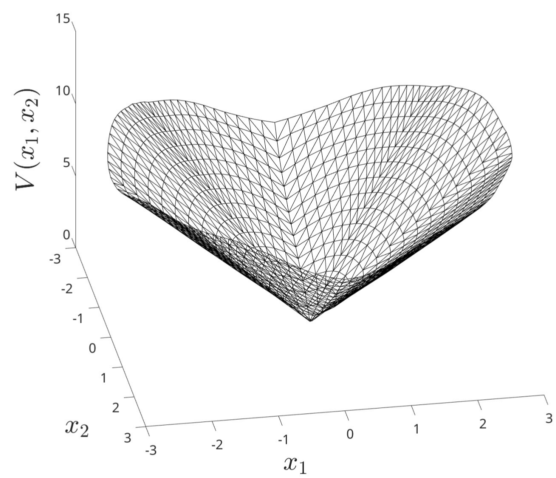

Fig. 4. The Lyapunov function computed for system (13), with the CPA interpolation of (15) as control (closed-loop system), using the CPA method.

Note that the control is very irregular and jumps up and down. This is to be expected, as there is noting in the objective working against this.

If we minimize $\max_{\mathbf{x}\in\mathscr{D}} \|\mathbf{u}(\mathbf{x})\|_\infty + \max_{\substack{k=1,2\\ \mathbf{x}\in\mathscr{D}}} \|\nabla u_k(\mathbf{x})\|_\infty$, i.e. (12) with $w = 1$, we obtain the control in Figs. 7 and 8 with

$$\max_{\mathbf{x}\in\mathscr{D}} \|\mathbf{u}(\mathbf{x})\|_\infty = 1.23816 \quad \text{and} \quad \max_{\substack{k=1,2\\ \mathbf{x}\in\mathscr{D}}} \|\nabla u_k(\mathbf{x})\|_\infty = 0.4123.$$

Interestingly, the value $\max_{\mathbf{x}\in\mathscr{D}} \|\mathbf{u}(\mathbf{x})\|_\infty = 1.23816$ has not grown in comparison to when we only minimized $\max_{\mathbf{x}\in\mathscr{D}} \|\mathbf{u}(\mathbf{x})\|_\infty$, but the control is much more regular.

V. Conclusions

The paper considered a two-step approach with the goal to improve a given controller. In a first step, the CPA algorithm was used to compute a control Lyapunov function (CLF) for the closed-loop system. In a second step, the CPA algorithm was adapted to determine a controller that is optimal w.r.t. a given cost function, which was chosen in order to bound the control input as well as its variation. A case study has shown the effectiveness of the approach.

This approach could be the first step of an iterative procedure to determine a control under given input constraints and thus, it contributes to the existing methods on controller design under constraints.

References

[1] S. Albertsson, P. Giesl, S. Gudmundsson, and S. Hafstein. Simplicial complex with approximate rotational symmetry: A general class of simplicial complexes. *J. Comput. Appl. Math.*, 363:413–425, 2020.

[2] F. Allgöwer, R. Findeisen, and C. Ebenbauer. Nonlinear model predictive control. *Control Systems, Robotics and Automation*, 11:59–84, 2009.

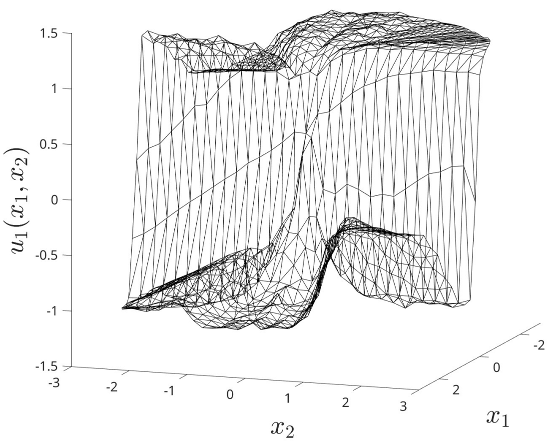

Fig. 5. The first component u_1 of the computed control when minimizing $\max_{\mathbf{x}\in\mathscr{D}} \|\mathbf{u}(\mathbf{x})\|_\infty$.

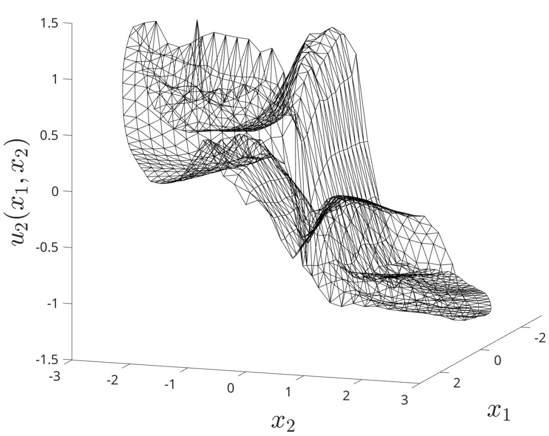

Fig. 6. The second component u_2 of the computed control when minimizing $\max_{\mathbf{x}\in\mathscr{D}} \|\mathbf{u}(\mathbf{x})\|_\infty$.

[3] R. Baier and S.F. Hafstein. Numerical computation of control Lyapunov functions in the sense of generalized gradients. In *Proc. of the 21st International Symposium on Mathematical Theory of Networks and Systems (MTNS)*, pages 1173–1180, 2014.

[4] F. Bianchini, F. Fabiani, and S. Grammatico. On merging constraint and optimal control-Lyapunov functions. In *Proc. of the 57th IEEE Conference on Decision and Control (CDC)*, pages 2328–2333. IEEE, 2018.

[5] A. Boccia, L. Grüne, and K. Worthmann. Stability and feasibility of state constrained mpc without stabilizing terminal constraints. *Systems & Control Letters*, 72:14–21, 2014.

[6] P. Braun, L. Grüne, and C. Kellett. Feedback design using nonsmooth control Lyapunov functions: A numerical case study for the nonholonomic integrator. In *Proc. of the 56th IEEE Conference on Decision and Control (CDC)*, 2017.

[7] Z. Chen, Q. Chen, X. He, and M. Sun. Adaptive backstepping control design for uncertain rigid spacecraft with both input and output constraints. *IEEE Access*, 6:60776–60789, 2018.

[8] N. Chopra and M.W. Spong. Passivity-based control of multi-agent systems. *Advances in robot control: from everyday physics to human-like movements*, pages 107–134, 2006.

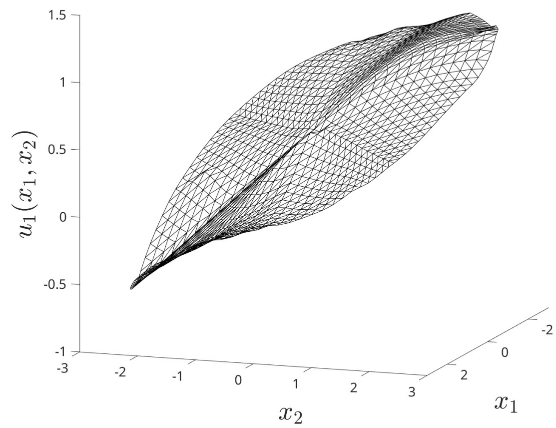

Fig. 7. The first component u_1 of the computed control when minimizing $\max_{\mathbf{x}\in\mathscr{D}} \|\mathbf{u}(\mathbf{x})\|_\infty + \max_{\substack{k=1,2\\ \mathbf{x}\in\mathscr{D}}} \|\nabla u_k(\mathbf{x})\|_\infty$.

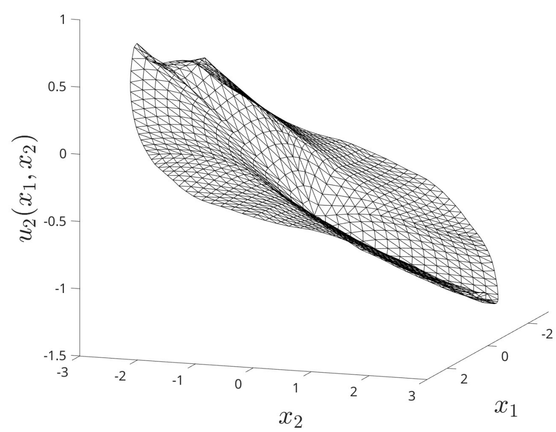

Fig. 8. The second component u_2 of the computed control when minimizing $\max_{\mathbf{x}\in\mathscr{D}} \|\mathbf{u}(\mathbf{x})\|_\infty + \max_{\substack{k=1,2\\ \mathbf{x}\in\mathscr{D}}} \|\nabla u_k(\mathbf{x})\|_\infty$.

[9] F. Clarke, Y. Ledyaev, R. Stern, and P. Wolenski. *Nonsmooth Analysis and Control Theory*, volume 178. Springer Science & Business Media, 2008.

[10] F. A. C. C. Fontes. A general framework to design stabilizing nonlinear model predictive controllers. *Systems & Control Letters*, 42(2):127–143, 2001.

[11] P. Giesl and S. Hafstein. Construction of Lyapunov functions for nonlinear planar systems by linear programming. *J. Math. Anal. Appl.*, 388:463–479, 2012.

[12] P. Giesl and S. Hafstein. Revised CPA method to compute Lyapunov functions for nonlinear systems. *J. Math. Anal. Appl.*, 410(1):292–306, 2014.

[13] P. Giesl and S. Hafstein. Computation and verification of Lyapunov functions. *SIAM J. Appl. Dyn. Syst.*, 14(4):1663–1698, 2015.

[14] P. Giesl and S. Hafstein. Review of computational methods for Lyapunov functions. *Discrete Contin. Dyn. Syst. Ser. B*, 20(8):2291–2331, 2015.

[15] P. Giesl and S. Hafstein. System specific triangulations for the construction of CPA Lyapunov functions. *Discrete Contin. Dyn. Syst Ser. B*, 26(12):6017–6026, 2021.

[16] P. Giesl and S. Hafstein. Triangulations and error estimates for interpolating Lyapunov functions. *SN Comput. Sci.*, 4:Article number 364, 2023.
[17] L. Grüne and J. Pannek. Nonlinear model predictive control. In *Nonlinear Model Predictive Control*, pages 45–69. Springer, 2017.
[18] L. Grüne and M. Sperl. Examples for separable control Lyapunov functions and their neural network approximation. *IFAC-PapersOnLine*, 56(1):19–24, 2023.
[19] S. Hafstein. A constructive converse Lyapunov theorem on exponential stability. *Discrete Contin. Dyn. Syst. Ser. A*, 10(3):657–678, 2004.
[20] T. Jiang, D. Lin, and T. Song. Finite-time backstepping control for quadrotors with disturbances and input constraints. *IEEE Access*, 6:62037–62049, 2018.
[21] P. Julian, J. Guivant, and A. Desages. A parametrization of piecewise linear Lyapunov functions via linear programming. *Int. J. Control*, 72(7-8):702–715, 1999.
[22] C. Kellett and A. Teel. Uniform asymptotic controllability to a set implies locally Lipschitz control-Lyapunov function. In *Proc. of the 39th IEEE Conference on Decision and Control (CDC)*, volume 4, pages 3994–3999, 2000.
[23] C.M. Kellett, H. Shim, and A.R. Teel. Further results on robustness of (possibly discontinuous) sample and hold feedback. *IEEE Transactions on Automatic Control (TAC)*, 49(7):1081–1089, 2004.
[24] J. Kurzweil. On the inversion of Lyapunov's second theorem on stability of motion. *Trans. Amer. Math. Soc.*, (24):19–77, 1963.
[25] Y. Lin and E.D. Sontag. Control-Lyapunov universal formulas for restricted inputs. *Control-Theory and Advanced Technology*, 10(4), 1995.
[26] A. M. Lyapunov. The general problem of the stability of motion. *Internat. J. Control*, 55(3):521–790, 1992. Translated by A. T. Fuller from Édouard Davaux's French translation (1907) of the 1892 Russian original, With an editorial (historical introduction) by Fuller, a biography of Lyapunov by V. I. Smirnov, and the bibliography of Lyapunov's works collected by J. F. Barrett, Lyapunov centenary issue.
[27] M. Malisoff and F. Mazenc. *Constructions of strict Lyapunov functions*. Springer Science & Business Media, 2009.
[28] S. Marinósson. Lyapunov function construction for ordinary differential equations with linear programming. *Dynamical Systems: An International Journal*, 17(2):137–150, 2002.
[29] S. Marinósson. *Stability Analysis of Nonlinear Systems with Linear Programming: A Lyapunov Functions Based Approach*. PhD thesis: Gerhard-Mercator-University Duisburg, Duisburg, Germany, 2002.
[30] H. Nakamura, Y. Yamashita, and H. Nishitani. Minimum projection method for nonsmooth control lyapunov function design on general manifolds. *Systems & Control Letters*, 58(10-11):716–723, 2009.
[31] R. Ortega, Z.P. Jiang, and D.J. Hill. Passivity-based control of nonlinear systems: A tutorial. In *Proc. of the 38th American Control Conference (ACC)*, volume 5, pages 2633–2637. IEEE, 1997.
[32] P. Osinenko, L. Beckenbach, and S. Streif. Practical sample-and-hold stabilization of nonlinear systems under approximate optimizers. *IEEE Control Systems Letters (L-CSS)*, 2(4):569–574, 2018.
[33] P. Osinenko, P. Schmidt, and S. Streif. Nonsmooth stabilization and its computational aspects. *IFAC-PapersOnLine*, 53(2):6370–6377, 2020.
[34] J. Reger and L. Triska. Dynamic extensions for exact backstepping control of systems in pure feedback form. In *Proc. of the 58th Conference on Decision and Control (CDC)*, pages 480–486. IEEE, 2019.
[35] P. Schmidt, P. Osinenko, and S. Streif. Some remarks on practical stabilization via clf-based control under measurement noise. *IEEE Access*, 13:1882–1895, 2024.
[36] P. Schmidt, P. Osinenko, and S. Streif. Some remarks on robustness of sample-and-hold stabilization. *IEEE Control Systems Letters (L-CSS)*, 8:3464–3469, 2025.
[37] E. Sontag. A "universal" construction of Artstein's theorem on nonlinear stabilization. *Systems & Control Letters*, 13(2):117–123, 1989.
[38] S. Vaidyanathan and A. T. Azar. *Backstepping control of nonlinear dynamical systems*. Academic Press, 2020.
[39] W. Walter. *Ordinary Differential Equation*. Springer, 1998.
[40] V. M. Zavala and L. T. Biegler. The advanced-step NMPC controller: Optimality, stability and robustness. *Automatica*, 45(1):86–93, 2009.
[41] J. Zhang, K. Li, and Y. Li. Output-feedback based simplified optimized backstepping control for strict-feedback systems with input and state constraints. *IEEE/CAA Journal of Automatica Sinica*, 8(6):1119–1132, 2021.
[42] M. Zhang, P. Borja, R. Ortega, Z. Liu, and H. Su. Pid passivity-based control of port-hamiltonian systems. *IEEE Transactions on Automatic Control (TAC)*, 63(4):1032–1044, 2017.