

# An Optimization Approach to Verifying and Synthesizing $\mathcal{K}$ -cooperative Systems<sup>\*</sup>

Dimitris Kousoulidis<sup>\*</sup> Fulvio Forni<sup>\*</sup>

<sup>\*</sup> *Department of Engineering, University of Cambridge, UK  
(dk483@eng.cam.ac.uk, f.forni@eng.cam.ac.uk)*

---

**Abstract:** Differential positivity and  $\mathcal{K}$ -cooperativity, a special case of differential positivity, extend differential approaches to control to nonlinear systems with multiple equilibria, such as switches or multi-agent consensus. To apply this theory, we reframe conditions for strict  $\mathcal{K}$ -cooperativity as an optimization problem. Geometrically, the conditions correspond to finding a cone that a set of linear operators leave invariant. Even though solving the optimization problem is hard, we combine the optimization perspective with the geometric intuition to construct a heuristic cone-finding algorithm centered around Linear Programming (LP). The algorithm we obtain is unique in that it modifies existing rays of a candidate cone instead of adding new ones. This enables us to also take a first step in tackling the synthesis problem for  $\mathcal{K}$ -cooperative systems. We demonstrate our approach on some examples, including one in which we repurpose our algorithm to obtain a novel alternative tool for computing polyhedral Lyapunov functions of bounded complexity.

*Keywords:* Nonlinear control, Differential positivity, Linear programming, Monotone systems, Bistability, Consensus

---

## 1. INTRODUCTION

Multistable systems are difficult to analyze with classical system-theoretic methods. The presence of several fixed points limits the use of fundamental methods like Lyapunov theory, constraining its use within the neighborhood of each attractor. Feedback control design is even more challenging. Even the task of tuning the parameters of a simple bistable switch for performance or robustness to perturbations pushes any classical tool of nonlinear control to its limits. In this paper we begin to address these issues by proposing a robust, tractable approach to nonlinear analysis and feedback design for systems whose attractor landscape is characterized by the presence of multiple fixed points. The approach builds on differential positivity (Forni and Sepulchre (2016)), which studies systems by looking at the positivity of the linearized dynamics (along any trajectory of the system). Tractability follows from a two-step optimization iteration based on Linear Programming, which provides a novel numerical tool for analysis and feedback design for monostable and multistable closed-loop systems.

Differential positivity extends differential analysis (Forni and Sepulchre (2014b)). The approach is similar to contraction theory (Lohmiller and Slotine (1998); Pavlov et al. (2006); Russo et al. (2010); Forni and Sepulchre (2014a)), which characterizes the stability/contraction of a system from the stability of its linearizations along any possible system trajectory. This makes contraction a powerful approach for problems where the fixed point varies with parameters. Similarly to how contraction theory links

the convergence of system trajectories to the stability of the linearizations, differential positivity links the behavior of the nonlinear system (monotonicity, multistability, etc.) to the positivity of its linearizations.

A linear system is positive if its trajectories contract a cone. This is a form of projective contraction that leads to the Perron-Frobenius theorem, which entails the existence of a slow dominant mode in the system dynamics and is also related to the presence of a dominant eigenvector within the cone (Bushell (1973)). In a similar way, a system is differentially positive if its linearized dynamics contract a cone field (a cone that depends on the state of the system). When the cone field is constant and the system state belongs to a vector space, a differentially positive system is a monotone system (Hirsch and Smith (2006); Angeli and Sontag (2003)), therefore its trajectories preserve a partial order relation on the system state space. Furthermore, almost all bounded trajectories of the system converge to some fixed point, asymptotically. We call these systems  *$\mathcal{K}$ -cooperative*, to emphasize the role of the constant cone field  $\mathcal{K}$  and important connections with the literature (Hirsch and Smith (2006)).

Finding the contracting cone of a  $\mathcal{K}$ -cooperative system is provably difficult (Protasov (2010)). For this reason, most of the literature assumes that a suitable cone is readily available (Hirsch and Smith (2006); Angeli and Sontag (2003)). In contrast, in this paper we provide a heuristic algorithm to attempt to find such cones. The theory builds on a previous attempt (Kousoulidis and Forni (2019)). The key difference from our previous algorithm is that each iteration of the new algorithm modifies existing rays instead of adding new ones. This has computational advantages and allows us to find cones of lower complexity

---

<sup>\*</sup> D. Kousoulidis is supported by the Engineering and Physical Sciences Research Council (EPSRC) of the United Kingdom.

(fewer extreme rays). We also make use of this to attempt to synthesize  $\mathcal{K}$ -cooperativity, enabling a new approach to robust design for multistable systems.

Our algorithm is based on optimization and Linear Programming. To make use of optimization techniques, we introduce a quantification of how far a given system is from being  $\mathcal{K}$ -cooperative with respect to a given cone. This is then used to iteratively reshape the cone to improve this distance until we obtain a contracting cone. We illustrate our algorithm in practice in Section 6 by analyzing and designing a robust electrical switch. We also investigate a nonlinear consensus problem with repulsive interactions. Finally, through a suitable system augmentation, in our third example we show how to use our algorithm to build polyhedral Lyapunov functions for stability analysis.

The first two sections below introduce the language of the paper. The main cone finding algorithm is presented in Sections 4 and 5 for analysis and synthesis respectively. This is followed by the examples.

**Notation:** When used on matrices or vectors, inequalities are always meant in the element-wise sense. Combined with inequalities and used on a matrix,  $\text{off-diag}(\cdot)$  implies that the inequalities apply only to the off-diagonal entries of the matrix. The interior of a set  $\mathcal{S}$  is denoted by  $\text{int}(\mathcal{S})$ . A proper cone is a set  $\mathcal{K}$  such that: (i) if  $r_1, r_2 \in \mathcal{K}$  and  $0 \leq p_1, p_2 \in \mathbb{R}$ , then  $p_1 r_1 + p_2 r_2 \in \mathcal{K}$ ; (ii)  $\text{int}(\mathcal{K}) \neq \{\}$ ; and (iii) if  $r \in \mathcal{K} \setminus \{0\}$ , then  $-r \notin \mathcal{K}$ . The dual cone of a cone  $\mathcal{K}$  is denoted by  $\mathcal{K}^*$  and defined as  $\{h : h^T r \geq 0, \forall r \in \mathcal{K}\}$ . The dual cone can also be interpreted geometrically as the set of all half-spaces that contain the cone. Considering two cones  $\mathcal{K}_1$  and  $\mathcal{K}_2$ ,  $\mathcal{K}_1 \subseteq \mathcal{K}_2$  denotes the usual set inclusion. We use  $\mathcal{K}_1 \subset \mathcal{K}_2$  to denote  $r \in \mathcal{K}_1 \setminus \{0\} \implies r \in \text{int}(\mathcal{K}_2)$ . When  $\mathcal{A}$  denotes a finite set, we use  $|\mathcal{A}|$  to denote its index set.

## 2. $\mathcal{K}$ -COOPERATIVITY

### 2.1 Definitions and Properties

We say that a continuous time linear system  $\dot{x} = Ax$ ,  $x \in \mathbb{R}^n$ , leaves a cone  $\mathcal{K}$  invariant if

$$x(0) \in \mathcal{K} \implies x(t) \in \mathcal{K} \text{ for all } t \geq 0$$

This is equivalent to the *sub-tangentiality* condition (Berman et al., 1989, Theorem 3.11):

$$x \in \mathcal{K}, y \in \mathcal{K}^*, y^T x = 0 \implies y^T A x \geq 0 \quad (1)$$

Similarly, we say that the system contracts a cone  $\mathcal{K}$  if

$$x(0) \in \mathcal{K} \setminus \{0\} \implies x(t) \in \text{int}(\mathcal{K}) \text{ for all } t > 0$$

This is implied by the *strict sub-tangentiality* condition (Berman et al., 1989, Theorems 3.7 and 3.26):

$$x \in \mathcal{K} \setminus \{0\}, y \in \mathcal{K}^* \setminus \{0\}, y^T x = 0 \implies y^T A x > 0 \quad (2)$$

If we find a cone that the system contracts, we certify that the system is strictly positive. Strict positivity restricts the qualitative behavior of the system: the dominant eigenvector is an attractor of the system.

Differential positivity (Forni and Sepulchre (2016)) generalizes the notion of positivity to the nonlinear setting. For nonlinear systems,  $\dot{x} = f(x)$ , this generalization is based on system linearization and the notion of prolonged system dynamics:

$$\dot{x} = f(x) \quad \delta \dot{x} = \partial f(x) \delta x,$$

where  $\partial f(x)$  is the Jacobian of the vector field at  $x \in \mathbb{R}^n$ . Differential positivity is defined with respect to a cone field  $\mathcal{K}(x)$ . In this paper we will study the simpler case of a constant cone field  $\mathcal{K}$ . We say that a nonlinear system differentially contracts a cone  $\mathcal{K}$  if

$$\delta x(0) \in \mathcal{K} \setminus \{0\} \implies \delta x(t) \in \text{int}(\mathcal{K}) \text{ for all } t > 0. \quad (3)$$

In analogy with the linear case, we call these systems *strictly differentially positive*. Equation (3) is implied by the *strict differential sub-tangentiality* condition that requires that, for all  $x \in \mathbb{R}^n$

$$\delta x \in \mathcal{K} \setminus \{0\}, y \in \mathcal{K}^* \setminus \{0\}, \quad y^T \delta x = 0 \implies y^T \partial f(x) \delta x > 0 \quad (4)$$

Similarly to strict positivity in linear systems, strict differential positivity restricts the qualitative behavior of nonlinear systems: in our setting, it implies that almost all bounded trajectories converge to some fixed point (Forni and Sepulchre, 2016, Corollary 5).

The strict differential sub-tangentiality conditions provide a way to verify strict differential positivity. We call systems that satisfy (4) for some cone *strictly  $\mathcal{K}$ -cooperative*. Since (4)  $\implies$  (3), strictly  $\mathcal{K}$ -cooperative systems are strictly differentially positive systems and share the same convergence result. For the rest of the paper, we attempt to verify or synthesize strictly  $\mathcal{K}$ -cooperative systems. The main obstacle in doing so is finding a  $\mathcal{K}$  for which (4) holds.

For linear systems, strict positivity has a spectral characterization: a linear system  $\dot{x} = Ax$  is strictly positive if and only if the rightmost eigenvalue of  $A$  is simple and real (Vandergraft (1968)). Unfortunately, for a nonlinear system  $\dot{x} = f(x)$ , this does not generalize to a sufficient condition on the eigenvalues of the Jacobians  $\partial f(x)$ . The presence of a rightmost eigenvalue in the Jacobian at every  $x$  is only a *necessary* condition. This is analogous to how we cannot, generally, establish the convergence of all trajectories of a system to a unique fixed point from the analysis of the Jacobian eigenvalues (see related counterexamples in (Khalil, 2002, Example 4.22) and Leonov et al. (2010)).

## 3. GENERAL FORMULATION

### 3.1 Conical Relaxation and Robustness

To verify (4) with finite computations we relax the set of all Jacobians,  $\{\partial f(x)\}$ , to a *finitely* generated set of matrices,  $\mathcal{A}$ , such that for each  $x \in \mathbb{R}^n$ ,

$$\partial f(x) \in \text{conic-hull}(\mathcal{A}) \setminus \{0\}, \quad (5)$$

where  $\text{conic-hull}(\mathcal{A})$  is defined as

$$\text{conic-hull}(\mathcal{A}) = \left\{ A : A = \sum_{i=1}^k p_i A_i, p_i \geq 0 \right\}$$

We refer to a set  $\mathcal{A}$  that satisfies (5) as a *conical relaxation* of  $\{\partial f(x)\}$ . The use of a conical relaxation to verify (4) follows from the next Lemma.

*Lemma 1.* (Convexity of Sub-tangentiality). If  $\mathcal{A}$  is a conical relaxation of  $\{\partial f(x)\}$ , and each  $A_i \in \mathcal{A}$  satisfies (2) with respect to the same cone  $\mathcal{K}$ , then  $f(x)$  satisfies (4).

**Proof.** Assuming left-hand side of (4),

$$y^T \partial f(x) \delta x = y^T \left( \sum_{i=1}^k p_i^x A_i \right) \delta x = \sum_{i=1}^k p_i^x (y^T A_i \delta x) > 0,$$

where the strict inequality follows from (2), using the fact that at least one  $p_i^x > 0$  (since  $\partial f(x) \neq 0$ ). ■

We can also use conical relaxations to analyze robustness to uncertainty in the dynamics:

- We get robustness to some perturbations ‘for free’ since our certificate also holds for any  $\dot{x} = f(x) + g(x)$  where  $\partial g(x) \in \text{conic-hull}(\mathcal{A})$ .
- We can adapt a conical relaxation to incorporate specific perturbations. If our perturbed system can be represented in the linearizations by  $\partial f(x) + \mathcal{Q}$ , where  $\mathcal{Q}$  is a given family of perturbations, we can use a new conical relaxation  $\mathcal{A}_{\mathcal{Q}}$  that satisfies  $\partial f(x) + \mathcal{Q} \in \text{conic-hull}(\mathcal{A}_{\mathcal{Q}}) \setminus \{0\}$ .

We therefore treat the problem of finding cones for system analysis as the problem of finding a common cone that is contracted by (all elements of) a given  $\mathcal{A}$ .

*Remark 1.* Naively producing a tight conical relaxation for a general system can lead to a combinatorial explosion in the number of elements in  $\mathcal{A}$ . We do not presently focus on the general construction of conical relaxations and instead manually derive them for the applications considered.

### 3.2 Cone Representations

In our search for cones, we limit ourselves to the family of *polyhedral* cones. Polyhedral cones can be represented in two ways:

**R-representation** given matrix  $R$ ,

$$\mathcal{K}_{\mathcal{R}}(R) = \{r : r = Rp, p \geq 0\} \quad (6)$$

**H-representation** given matrix  $H$ ,

$$\mathcal{K}_{\mathcal{H}}(H) = \{r : Hr \geq 0\} \quad (7)$$

Assuming the cones are proper, strict inequalities characterize their interior. We focus on verification and synthesis of  $\mathcal{K}$ -cooperative systems with respect to R-representation cones, noting that the analysis and algorithms can be extended to H-representation cones through duality. Specifically, if we introduce  $\mathcal{A}^* = \{A^T : A \in \mathcal{A}\}$ , then it can be shown that using  $\mathcal{A}$  to verify  $\mathcal{K}$ -cooperativity with respect to  $\mathcal{K}_{\mathcal{H}}(H)$  yields the same conditions as using  $\mathcal{A}^*$  and  $\mathcal{K}_{\mathcal{R}}(H^T)$ .

### 3.3 Feasibility Formulation

We reformulate (1) and (2) for R-representation polyhedral cones:

*Lemma 2.* For a given matrix  $A$  and proper cone  $\mathcal{K}_{\mathcal{R}}(R)$ ,  $A$  satisfies (1) for  $\mathcal{K}_{\mathcal{R}}(R)$  if (and only if) there exists a matrix  $P$  with nonnegative off-diagonal entries (off-diag( $P$ )  $\geq 0$ ) such that:

$$AR = RP \quad (8)$$

Furthermore,  $A$  also satisfies (2) for  $\mathcal{K}_{\mathcal{R}}(R)$  if (and only if) there exists a  $P$  with off-diag( $P$ )  $> 0$ .

**Proof.** If: Let  $r \in \mathcal{K} \setminus \{0\}$ ,  $h \in \mathcal{K}^* \setminus \{0\}$ ,  $h^T r = 0$ , and  $\mathcal{K}_{\mathcal{R}}(R)$  is proper.

Then, for any  $\alpha \in \mathbb{R}$ ,  $h^T Ar = h^T (A + \alpha I)r = h^T (A + \alpha I)Rp = h^T R(P + \alpha I)p$ , where the second identity follows from the additional identity  $r = Rp$ , for some  $p \geq 0$ .

From our assumptions we have  $0 \neq p \geq 0$  and  $0 \neq (h^T R) \geq 0$ . As such we satisfy (1) for  $A$  if  $(P + \alpha I) \geq 0$ . Since we can choose  $\alpha$  freely, this is equivalent to off-diag( $P$ )  $\geq 0$ . Similarly, we satisfy (2) for  $A$  if off-diag( $P$ )  $> 0$ .

Only if: See (Berman et al., 1989, Theorems 3.3.9 and 3.3.41). ■

Putting everything together:

*Proposition 1.* (Feasibility Formulation). A nonlinear system  $\dot{x} = f(x)$  is strictly  $\mathcal{K}$ -cooperative if, given an  $\mathcal{A}$  that satisfies (5), there exists a matrix  $R$  satisfying:

- $\mathcal{K}_{\mathcal{R}}(R)$  is a proper cone
- For each  $A_i \in \mathcal{A}$ , there exists a matrix  $P_i$  such that  $A_i R = R P_i$  and off-diag( $P_i$ )  $> 0$

**Proof.** Implied by combining Lemmas 1 and 2. ■

If  $R$  is left as a free parameter this test is intractable. On the other hand, when  $R$  is fixed the test can be carried out using Linear Programming (LP). However, this LP test only gives a ‘Yes’ or ‘No’ answer. For our goal of *finding* a suitable  $R$  we are interested in introducing a measure of how far a given  $R$  is from satisfying Proposition 1.

### 3.4 Measuring Distance to $\mathcal{K}$ -cooperativity

We want to quantify how far a given set of positive matrices is from contracting a given candidate cone. This measure should be independent of the representation of the cone and compatible with LP. To this end, we introduce the set of widened operators  $\bar{\mathcal{A}}_w$ , where for each  $\bar{A}_i \in \bar{\mathcal{A}}_w$ :

$$\bar{A}_i = A_i + w_i (r_i h_i^T) \quad (A_i \in \mathcal{A}) \quad (9)$$

where  $w_i$  is a scalar ‘widening coefficient’,  $r_i$  is some fixed vector in  $\text{int}(\mathcal{K})$ , and  $h_i$  is some fixed vector in  $\text{int}(\mathcal{K}^*)$  (we explore some strategies for selecting these later). The rank one matrix  $(r_i h_i^T)$  projects all points  $x$  to  $(h_i^T x) r_i$ .

*Lemma 3.* For any given matrix  $A$ , cone  $\mathcal{K}$ ,  $r \in \text{int}(\mathcal{K})$ , and  $h \in \text{int}(\mathcal{K}^*)$ , there will exist some finite  $w$  such that  $[A + w(rh^T)]$  satisfies (1) with respect to  $\mathcal{K}$

**Proof.** We are only concerned with the sign of  $y^T [A + w(rh^T)]x$ , so we fix  $\|y\|_2 = \|x\|_2 = 1$  (the implication is trivial when  $y$  or  $x$  are 0). Because  $r \in \text{int}(\mathcal{K})$ , and  $h \in \text{int}(\mathcal{K}^*)$ ,  $y^T r > 0$  and  $h^T x > 0$ . We denote  $\inf_y (y^T r) = \alpha > 0$  and  $\inf_x (h^T x) = \beta > 0$ .

Then:  $y^T [A + w(rh^T)]x = y^T Ax + w(y^T r)(h^T x) \geq -\|A\|_2 + w\alpha\beta$

As such, we can guarantee that  $y^T [A + w(rh^T)]x \geq 0$  if  $w \geq \|A\|_2 / (\alpha\beta)$ . ■

*Lemma 4.* Given  $[A + w(rh^T)]$  and  $\mathcal{K}$  that satisfy (1) with  $r \in \text{int}(\mathcal{K})$  and  $h \in \text{int}(\mathcal{K}^*)$ ,  $[A + w^*(rh^T)]$  will satisfy (2) for all  $w^* > w$  with the same  $\mathcal{K}$ .

**Proof.** Since  $[A + w(rh^T)]$  satisfy (1),  $y^T [A + w(rh^T)]x \geq 0$ . Also, since  $r \in \text{int}(\mathcal{K})$ , and  $h \in \text{int}(\mathcal{K}^*)$ ,  $y^T r > 0$  and

$h^T x > 0$ . Letting  $\bar{w} = w^* - w > 0$ ,  $y^T[A + w^*(rh^T)]x = y^T[A + w(rh^T)]x + \bar{w}(y^T r)(h^T x) > 0$  ■

Lemmas 3 and 4 mean that we can use  $\{w_i\}$  to quantitatively measure how close we are to a contracting cone: if we leave  $\{w_i\}$  as free parameters, the tests in Lemma 2 will always be feasible for  $\bar{A}_i \in \bar{\mathcal{A}}_w$  and a fixed  $\mathcal{K}_{\mathcal{R}}(R)$ , and, if we find  $\{w_i\} < 0$ , we can conclude strict  $\mathcal{K}$ -cooperativity.

To summarize:

*Proposition 2.* (Optimization Formulation). Consider the following optimization problem:

$$\begin{aligned} \min_{\{w_i, P_i\}} \quad & w \\ \text{subject to, } \forall i \in |\mathcal{A}|: \quad & w \geq w_i, \\ & \text{off-diag}(P_i) \geq 0, \\ & [A_i + w_i(r_i h_i^T)] R = R P_i \end{aligned} \quad (10)$$

Where  $|\mathcal{A}|$  is the index set of  $\mathcal{A}$ ,  $r_i \in \text{int}(\mathcal{K}_{\mathcal{R}}(R))$ , and  $h_i \in \text{int}(\mathcal{K}_{\mathcal{H}}(R)^*)$ .

If we find a solution with  $w < 0$ , we can conclude strict  $\mathcal{K}$ -cooperativity with respect to the corresponding  $\mathcal{K}_{\mathcal{R}}(R)$  for all nonlinear systems  $\dot{x} = f(x)$  that satisfy (5) for the  $\mathcal{A}$  used.

**Proof.** Implied by combining Lemmas 1-4. ■

We can now make use of  $w$  to iteratively refine an initial candidate cone  $R$ , searching for a final contracting cone, as described in Section 4.2.

## 4. VERIFYING $\mathcal{K}$ -COOPERATIVITY

### 4.1 Necessary Conditions and Geometric Constraints

In Section 2 we mentioned that to verify strict  $\mathcal{K}$ -cooperativity we need to find a contracting cone but that the linearizations directly provide us with some necessary conditions. We will now use Propositions 4-6 from Kousoulidis and Forni (2019) to elaborate on the necessary conditions and the geometric constraints imposed by the linearizations.

Firstly, every  $A_i \in \mathcal{A}$  has to be strictly positive, i.e. have a simple, real right-most eigenvalue. We refer to such eigenvalues as dominant. We also apply the same terminology to their associated right and left eigenvectors, which we denote by  $\bar{r}_i$  and  $\bar{h}_i$ , and scale such that  $\|\bar{h}_i\|_1 = 1$  and  $\bar{h}_i^T \bar{r}_i = 1$ . The dominant eigenvectors tell us that all trajectories of  $\dot{x} = A_i x$  that start in the open half-space  $\bar{h}_i^T x(0) > 0$  will converge (as rays) to the ray  $p_0 \bar{r}_i$ , for some  $0 < p_0 \in \mathbb{R}$ .

It follows that for a cone  $\mathcal{K}$  to be contracted by each  $A_i$ , there must exist orientations of the dominant eigenvectors  $\bar{r}_i$  and  $\bar{h}_i$  such that  $\bar{r}_i \in \text{int}(\mathcal{K})$  and  $\bar{h}_i \in \text{int}(\mathcal{K}^*)$ . This must hold for each  $A_i \in \mathcal{A}$ . For this to be possible, there must exist an orientation of the dominant eigenvectors such that  $\bar{h}_i^T \bar{r}_j > 0$ . We can efficiently find such an orientation if it exists or conclude that it does not (see Algorithm 1 in Kousoulidis and Forni (2019)). If it does not exist, we can again exclude  $\mathcal{K}$ -cooperativity. If it does exist, it is unique (up to the orientation of one of the eigenvectors). We then combine our dominant eigenvectors to form two cones,  $\mathcal{K}_{\mathcal{R}}(R_{inner})$  and  $\mathcal{K}_{\mathcal{H}}(H_{outer})$ :

$\mathcal{K}_{\mathcal{R}}(R_{inner})$  is a R-representation polyhedral cone formed from right dominant eigenvectors  $\bar{r}_i$ , while  $\mathcal{K}_{\mathcal{H}}(H_{outer})$  is a H-representation polyhedral cone formed from left dominant eigenvectors  $\bar{h}_i$ . We enforce that  $\bar{r}_i \in \text{int}(\mathcal{K})$  and  $\bar{h}_i \in \text{int}(\mathcal{K}^*)$  by only considering cones  $\mathcal{K}$  that satisfy:

$$\mathcal{K}_{\mathcal{R}}(R_{inner}) \subset \mathcal{K} \subset \mathcal{K}_{\mathcal{H}}(H_{outer}) \quad (11)$$

Since  $\bar{r}_i \in \text{int}(\mathcal{K})$  and  $\bar{h}_i \in \text{int}(\mathcal{K}^*)$ , they serve as ideal candidates for  $\{r_i\}$  and  $\{h_i\}$  in Proposition 2. By using them to produce  $\bar{\mathcal{A}}_w$ , each new matrix  $\bar{A}_i$  will have the same eigenvectors as  $A_i$ , will have its dominant eigenvalue shifted by  $w_i$  to the right, and will have all other eigenvalues unaffected.

### 4.2 The Algorithm

The first step of the verification algorithm is to compute an initial candidate cone. We want this cone to satisfy (11). This is because (11) is a necessary condition for an invariant cone and, since we set  $r_i = \bar{r}_i$ ,  $h_i = \bar{h}_i$ , (11) also guarantees that we can solve (10) for a finite  $w$ . Additionally, because we will not be adding new extreme rays in other steps of the algorithm, we want to be able to specify the number of extreme rays  $m$  ( $\geq n$ ) as a parameter to our initialization algorithm (generally, more oscillatory dynamics require cones with a larger number of extreme rays - Benvenuti and Farina (2004)).

To achieve this, we begin with matrix  $R_{inner}$ . Then, until our matrix has  $m$  columns, we expand it by appending randomly perturbed columns of  $R_{inner}$ , rejecting new vectors that do not increase the number of extreme rays in our cone or are not inside  $\mathcal{K}_{\mathcal{H}}(H_{outer})$ . To ensure the strict inclusion  $\mathcal{K}_{\mathcal{R}}(R_{inner}) \subset \mathcal{K}_{\mathcal{R}}(R^{(0)})$ , we subtract some small enough positive multiple of the centroid ray of  $\mathcal{K}_{\mathcal{R}}(R_{inner})$  from each column of our matrix. In this way we now have a matrix  $R^{(0)}$  that provides a R-representation of a suitable initial cone.

We then proceed to solve (10) with  $R = R^{(0)}$ , obtaining values for  $\{w_i, P_i\}$  in the process. If  $w < 0$ , our system is  $\mathcal{K}$ -cooperative and our search is done. If it is not, we introduce a new optimization problem that approximates the effect of changes to  $R$  on the solution to (10). This is done by using the values we have for  $R$  and  $\{w_i, P_i\}$  to linearize the equality constraints of (10) (which are nonlinear when  $R$  is treated as a variable) about the previous solution by taking their derivatives. The variations of  $(R, \{w_i, P_i\})$  about their previous values are represented by the new variables  $(\delta R, \{\delta w_i, \delta P_i\})$ . The linearized equality constraints are given by:

$$[A_i + w_i(\bar{r}_i \bar{h}_i^T)](\delta R) + (\delta w_i)(\bar{r}_i \bar{h}_i^T)R = (\delta R)P_i + R(\delta P_i)$$

We also have to ensure that (11) holds. As such, when solving the approximate problem, we need to impose that:

$$\mathcal{K}_{\mathcal{R}}(R_{inner}) \subset \mathcal{K}_{\mathcal{R}}(R + \delta R) \subset \mathcal{K}_{\mathcal{H}}(H_{outer}) \quad (12)$$

The right hand side set inclusion in (12) ends up being a linear constraint on  $\delta R$  that we can include directly. It can be expressed as  $H_{outer}(R + \delta R) > 0$ . However, the left-hand side constraint isn't linear. Instead of also approximating this constraint via its derivative, we characterize a linear subset of  $\delta R$  for which it is *guaranteed* to hold. To guarantee  $\mathcal{K}_{\mathcal{R}}(R_{inner}) \subset \mathcal{K}_{\mathcal{R}}(R + \delta R)$ , we first

introduce a new matrix  $R'$  that satisfies  $\mathcal{K}_{\mathcal{R}}(R_{inner}) \subset \mathcal{K}_{\mathcal{R}}(R') \subset \mathcal{K}_{\mathcal{R}}(R)$  and has the same dimensions as  $R$ . We can obtain such a matrix by, for example, adding a small enough positive multiple of the centroid ray of  $\mathcal{K}_{\mathcal{R}}(R_{inner})$  to each column of  $R$ . We then add a free matrix variable  $M$  and the constraints that  $(R + \delta R) = R'M$ ,  $1^T M = 1^T$ , and  $\text{off-diag}(M) \leq 0$ , to our optimization problem. The constraints imply  $-M$  will be Metzler with  $-1$  as its Perron-Frobenius eigenvalue, and so, all eigenvalues of  $M$  will have real part  $\geq 1$ . This guarantees that  $M$  is an invertible M-matrix ( $M^{-1}$  exists and  $M^{-1} \geq 0$ ). Since  $R' = (R + \delta R)M^{-1}$ , we have that  $\mathcal{K}_{\mathcal{R}}(R') \subseteq \mathcal{K}_{\mathcal{R}}(R + \delta R)$ , which in-turn guarantees  $\mathcal{K}_{\mathcal{R}}(R_{inner}) \subset \mathcal{K}_{\mathcal{R}}(R + \delta R)$ .

To ensure that  $(\delta R, \{\delta w_i, \delta P_i\})$  are kept small (so that the linearized equality constraints offer good approximations) and that the algorithm converges, we add a penalty and/or a constraint on their norms. We combine everything to obtain the following LP problem:

$$\begin{aligned} \min_{\delta R, M, \{\delta w_i, \delta P_i\}} \quad & \bar{w} & (13) \\ \text{subject to:} \quad & H_{outer}(R + \delta R) > 0, (R + \delta R) = R'M, \\ & 1^T M = 1^T, \text{off-diag}(M) \geq 0, \\ & (\delta R, \{\delta w_i, \delta P_i\}) \text{ small,} \\ \text{and, } \forall i \in |\mathcal{A}|: \quad & \bar{w} \geq w_i + \delta w_i, \text{off-diag}(P_i + \delta P_i) \geq 0, \\ & [A_i + w_i(\bar{r}_i \bar{h}_i^T)](\delta R) + (\delta w_i)(\bar{r}_i \bar{h}_i^T)R \\ & = (\delta R)P_i + R(\delta P_i) \end{aligned}$$

We then update  $R$  to  $R + \delta R$  and solve the original (10) again, expecting the updated  $R$  to be able to solve (10) for a lower  $w$ . The process is repeated, stopping when  $w < 0$  or the algorithm has converged (which we check by looking at  $\|\delta R\|$ ). Algorithm 1 summarizes the overall process.

---

**Algorithm 1**  $\mathcal{K}$ -cooperativity Verification Algorithm

---

**Data:** The set of matrices  $\mathcal{A}$ ,

the maximum number of extreme rays in the cone  $m$

**Result:**  $R$  satisfying Proposition 1 if found, else *False*

**Procedure:**

$\mathcal{K}_{\mathcal{R}}(R_{inner}), \mathcal{K}_{\mathcal{H}}(H_{outer}), R^{(0)} = \text{Initialize}(\mathcal{A}, m)$

$k = 0$

**while** *True*:

$w, \{w_i, P_i\} = \text{Solution to (10) with } R = R^{(k)}$

**if**  $w < 0$ :

return  $R^{(k)}$

$\delta R = \text{Solution to (13) with } R = R^{(k)}$

**if**  $\|\delta R\| < \epsilon$ :

return *False*

$R^{(k+1)} = R^{(k)} + \delta R$

$k := k + 1$

---

Although this algorithm is not guaranteed to be able to verify  $\mathcal{K}$ -cooperativity in general, if it terminates successfully then the system is guaranteed to be  $\mathcal{K}$ -cooperative. It also appears to be effective in practice, as we demonstrate in the Examples section. By modifying existing rays, instead of continuously adding new ones, this algorithm allows us to reuse cones found as starting points for modified problems. This is crucial for the synthesis problem.

## 5. SYNTHESIZING $\mathcal{K}$ -COOPERATIVITY

### 5.1 Problem Formulation

We next consider the design problem: how do we find system parameters that make the dynamics  $\mathcal{K}$ -cooperative.

We use the following formulation:

$$\dot{x} = f_c(x) = f(x) + \left( \sum_{j=1}^l c_j U_j \right) x \quad (14)$$

Where  $U_j$  are fixed matrices, and  $c_j$  are scalars we are free to choose.

Many control design problems can be expressed in this formulation. For example, consider a linear system with state feedback  $\dot{x} = (A + BF)x$ . Assume it has  $n$  states and 1 input. Then, take  $l = n$  and  $U_j = (Be_j^T)$ , where  $e_j$  is a  $n$  dimensional vector with a 1 in position  $j$  and 0 in all other places. Each  $c_j$  corresponds to  $F_j$ , the  $j^{\text{th}}$  element of  $F$ .

To find parameters that make the dynamics  $\mathcal{K}$ -cooperative we need to search for a contracting cone while also adapting  $c$ . As discussed in Section 3.1, to be able to test  $\mathcal{K}$ -cooperativity with a finite number of computations, we require a conical relaxation  $\mathcal{A}(c)$ , which now depends on  $c$ . For our formulation,  $\mathcal{A}(c)$  has a simple parameterization. We start with a conical relaxation of  $f(x)$ ,  $\mathcal{A}$ . Then, for each matrix  $A_i \in \mathcal{A}$ , we define the parameter dependent matrix  $A_i(c) \in \mathcal{A}(c)$  as:

$$A_i(c) = A_i + \left( \sum_{j=1}^l c_j U_j \right)$$

It is easy to see that, since  $\mathcal{A}$  satisfies (5) for  $\partial f(x)$ ,  $\mathcal{A}(c)$  satisfies (5) for  $\partial f_c(x) = \partial f(x) + \left( \sum_{j=1}^l c_j U_j \right)$ . We can think of  $\mathcal{A}(\cdot)$  as a function that can be evaluated for a given  $c$  to return a valid conical relaxation for  $\dot{x} = f_c(x)$ . Propositions 1 and 2 still hold for a given  $c$  if we set  $\mathcal{A} = \mathcal{A}(c)$  and  $f(x) = f_c(x)$ , providing us with a way to test  $\mathcal{K}$ -cooperativity and measure distance to  $\mathcal{K}$ -cooperativity, respectively. We can also analyze robustness through  $\mathcal{A}(c)$  in the same ways as those described in Section 3.1.

*Remark 2.* The algorithm that follows can also be used for nonlinear controllers where  $f_c(x)$  cannot be decomposed in the same way as (14). To make our exposition cleaner we do not consider this here.

### 5.2 The Algorithm

The high-level structure of the algorithm remains unchanged from Algorithm 1. However, since the matrices in  $\mathcal{A}(c)$  now depend on  $c$ , we can no longer directly apply the necessary conditions of Section 4.1 and do not have obvious candidates for  $r_i$  and  $h_i$ . We instead set all  $r_i$  to a pre-selected fixed  $r$ , and all  $h_i$  to a pre-selected fixed  $h$ . This replaces (11) with  $\mathcal{K}_{\mathcal{R}}(r) \subset \mathcal{K} \subset \mathcal{K}_{\mathcal{H}}(h^T)$ , constraining our search to cones that have  $r \in \text{int}(\mathcal{K})$  and  $h \in \text{int}(\mathcal{K}^*)$ .

The cone is initialized in the same way as in Section 4.2 but with  $\mathcal{K}_{\mathcal{R}}(r)$  in place of  $\mathcal{K}_{\mathcal{R}}(R_{inner})$  and  $\mathcal{K}_{\mathcal{H}}(h^T)$  in place of  $\mathcal{K}_{\mathcal{H}}(H_{outer})$ .

We next consider the necessary modifications to the two optimization steps corresponding to (10) and (13). When incorporating  $c$  to our optimization problems we have two choices: we can either treat  $c$  as fixed when solving the first optimization step and update  $c$  only through solutions to the second optimization step (as we do with  $R$ ), or we can treat  $c$  as a variable directly from the first optimization step (as we do with  $\{w_i, P_i\}$ ). Here we chose to implement the former because it generalizes to nonlinear formulations of  $\mathcal{A}(c)$  and allows us to bias the values of  $c$ . When treating  $c$  in this way we also have to provide initial values  $c^{(0)}$  to the algorithm.

Optimization problem (10) is replaced by:

$$\begin{aligned} & \min_{\{w_i, P_i\}} w & (15) \\ & \text{subject to, } \forall i \in |\mathcal{A}(c)|: & w \geq w_i, \text{ off-diag}(P_i) \geq 0, \\ & & [A_i(c) + w_i(rh^T)] R = RP_i \end{aligned}$$

To obtain an optimization problem analogous to (13) we repeat the approximation procedure presented in Section 4.2, this time applied to (15). The only difference is in having to include the derivative with respect to  $c$  when computing the total derivative of the equality constraints. The following optimization problem is obtained:

$$\begin{aligned} & \min_{\delta R, \delta c, M, \{\delta w_i, \delta P_i\}} \bar{w} & (16) \\ & \text{subject to:} & H_{outer}(R + \delta R) > 0, (R + \delta R) = R'M, \\ & & 1^T M = 1^T, \text{ off-diag}(M) \geq 0, \\ & & (\delta R, \{\delta w_i, \delta P_i\}) \text{ small,} \\ & \text{and, } \forall i \in |\mathcal{A}(c)|: & \bar{w} \geq w_i + \delta w_i, \text{ off-diag}(P_i + \delta P_i) \geq 0, \\ & & [A_i(c) + w_i(rh^T)](\delta R) + (\delta w_i)(rh^T)R \\ & & + \left( \sum_{j=1}^l (\delta c_j) U_j \right) R \\ & & = (\delta R)P_i + R(\delta P_i) \end{aligned}$$

The overall process is summarized in Algorithm 2:

---

**Algorithm 2**  $\mathcal{K}$ -cooperativity Synthesis Algorithm

---

**Data:** The set of matrix functions  $\mathcal{A}(\cdot)$ ,

the initial set of parameters  $c^{(0)}$ ,

a vector that will lie inside the cone  $r$ ,

a vector that will lie inside the dual cone  $h$ ,

the number of rays in the cone  $m$

**Result:**  $R$  and  $c$  such that  $R, \mathcal{A}(c)$ , and  $f_c(x)$  satisfy Proposition 1 if found, else *False*

**Procedure:**

$R^{(0)} = \text{Initialize}(r, h^T, m)$

$k = 0$

**while** *True*:

$w, \{w_i, P_i\} = \text{Solution to (15) with } R = R^{(k)}, c = c^{(k)}$

**if**  $w < 0$ :

return  $R^{(k)}, c^{(k)}$

$\delta R, \delta c = \text{Solution to (16) with } R = R^{(k)}, c = c^{(k)}$

**if**  $\|\delta R\| < \epsilon_R$  and  $\|\delta c\| < \epsilon_c$ :

return *False*

$R^{(k+1)} = R^{(k)} + \delta R$

$c^{(k+1)} = c^{(k)} + \delta c$

$k := k + 1$

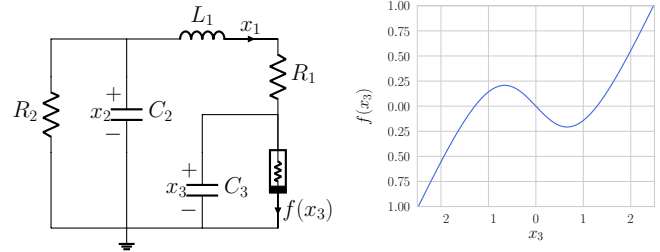


Fig. 1. **Left:** Circuit used in Section 6.1. **Right:** Plot of static nonlinearity  $f(x_3)$ .

## 6. EXAMPLES

### 6.1 Electrical Switch

Taking inspiration from the bistable and oscillatory circuits analyzed in Miranda-Villatoro et al. (2018, 2019), we consider the electrical circuit with three states shown in Fig. 1 (left). The dynamics of this circuit are given by:

$$\begin{aligned} L_1 \dot{x}_1 &= -R_1 x_1 + x_2 - x_3 \\ C_2 \dot{x}_2 &= -x_1 - x_2/R_2 \\ C_3 \dot{x}_3 &= x_1 - f(x_3) \end{aligned} \quad (17)$$

Here  $x_1$  represents the current across the inductor, and  $x_2$  and  $x_3$  represent the voltages across the capacitors. The scalar function  $f(\cdot)$  represents a nonlinear resistor. We take its derivative,  $f'$ , to be bounded by  $f'_l \leq f' \leq f'_u$ .

For different choices of parameters, this system exhibits a range of different qualitative behaviors. We fix  $L_1 = 30, C_2 = 10, R_2 = 5, C_3 = 1$  and let  $f(x_3) = 1.5 \tanh(x_3)$ , shown in Fig. 1 (right), such that  $f'_l = -0.5$  and  $f'_u = 1$ . Then, for  $R_1 = 1$ , simulations show the existence of a limit cycle, while for  $R_1 = 50$  the system appears to be bistable. This is illustrated in Fig. 2 (left).

Algorithm 1 finds a contracting cone for the case where  $R_1 = 50$ , proving that it is  $\mathcal{K}$ -cooperative. Since the chosen nonlinear resistor guarantees bounded trajectories and our system has two stable equilibrium points, this in turn proves that it is bistable. To further investigate this scenario, we incorporate robustness considerations: (i) We add matrix  $[0, 0, 1]^T [0, 0, 1]$  to our conical relaxation. This removes the lower bound on  $f'$  ( $f(\cdot)$  must still be Lipschitz). (ii) We allow all passive components (the resistances, capacitors, and inductor) other than  $R_1$  to vary by  $\pm 10\%$ , generating matrices for each possible scenario. The robust cone found is shown in Fig. 2 (right).

We now attempt to use Algorithm 2 to find an  $R_1$  for which we can prove that the system is  $\mathcal{K}$ -cooperative even if the passive components are allowed to vary by  $\pm 20\%$ . We initialize the nominal values at those of the oscillatory scenario from Fig. 1. We still include matrix  $[0, 0, 1]^T [0, 0, 1]$  in our conical relaxation, removing the lower bound on  $f'$ . As such we set  $r = h = [0, 0, 1]^T$ ,  $c = R_1, c^{(0)} = 1$ . We find the cone shown in Fig. 3 (left), which is contracted for our system provided  $R_1 = 61$  and all passive components (other than  $R_1$ ) are within  $\pm 20\%$  of their nominal values. We can therefore conclude that, for the above range of component values, the system will be bistable as long as trajectories are bounded and

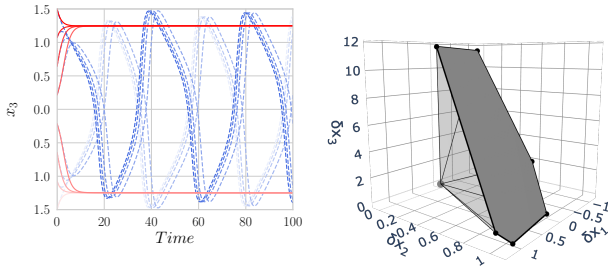


Fig. 2. **Left:** Example trajectories of (17) with  $R_1 = 1$  (blue and dashed) and  $R_1 = 50$  (red). **Right:** Cone found for (17) using Algorithm 1 ( $R_1 = 50$ , component tolerance  $\pm 10\%$ ).

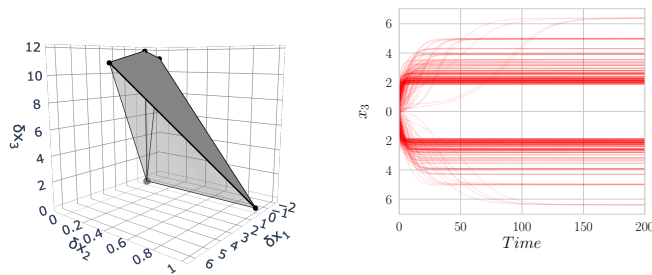


Fig. 3. **Left:** Cone found for (17) using Algorithm 2 ( $R_1^{(0)} = 1$ , final  $R_1 = 61$ , component tolerance  $\pm 20\%$ ). **Right:** Example trajectories of (17) with different component values for which the cone on the left guarantees bistability.

the equilibrium at 0 is unstable. Some possible example trajectories are shown in Fig. 3 (right).

### 6.2 First Order Consensus

We consider dynamics of the form:

$$\dot{x}_i = \sum_{j=1}^N f_{ij}(x_j - x_i) \quad 0 < i \leq N, \quad (18)$$

where each agent  $x_i$  represents a simple integrator driven by the weighted differences with its neighboring agents, characterized by functions  $f_{ij} : \mathbb{R} \rightarrow \mathbb{R}$  such that  $f_{ij}(0) = 0$ . We say that the agents reach consensus when  $x_1 = \dots = x_N$ .

We can use  $\mathcal{K}$ -cooperativity to study consensus of nonlinear or uncertain systems: the trajectories of a strictly  $\mathcal{K}$ -cooperative system with  $1_N \in \text{int}(\mathcal{K})$  converge to consensus.

Here, we revisit the consensus example from Section V.A of Kousoulidis and Forni (2019), where we considered a network of 5 agents with the topology shown in Fig. 4. The black edges represent linear connections with weights normalized to 1, the blue edges represent nonlinear connections with slopes restricted to  $-1 \leq f'_{15} \leq 1$  and  $-1 \leq f'_{42} \leq 1$ , and the red edge represents a linear connection with weight set to  $k$ .

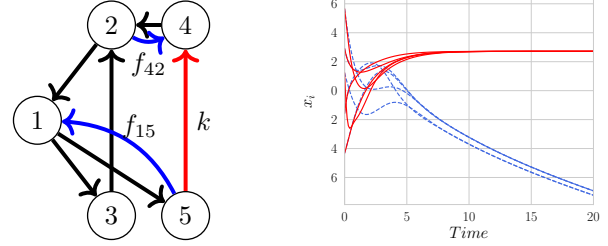


Fig. 4. **Left:** Consensus topology. **Right:** Comparison of a random trajectory of the consensus system with  $f_{15}(x), f_{42}(x) = -2 \sin(x)$  between when  $k = 1$  (blue) and when  $k = 4.8$  (red).

We initially fix  $k = 1$  and attempt to use Algorithm 1. We successfully find a contracting cone with 7 rays. For comparison, when we use the older Algorithm from Kousoulidis and Forni (2019) on this problem, we obtain a cone composed of 52 rays.

Next, we attempt to find a  $k$  such that the system reaches consensus for any  $-2 \leq f'_{15} \leq 2$  and  $-2 \leq f'_{42} \leq 2$ . For this we use Algorithm 2 with  $r, h = 1_N, c = k$  and  $c^{(0)} = 1$ . We find a contracting cone with  $k = 4.8$ . An example of a random trajectory when  $f_{15}(x), f_{42}(x) = -2 \sin(x)$  with  $k = 1$  and  $k = 4.8$  is shown in Fig. 4. Since  $-2 \leq f'_{15} \leq 2$  and  $-2 \leq f'_{42} \leq 2$ , we can only guarantee consensus for the case where  $k = 4.8$ , and plot the trajectory when  $k = 1$  for comparison.

### 6.3 Spring-Damper: Lyapunov Analysis

As a final example, we show some initial results on how our algorithm can also be adapted to find Lyapunov functions and design robust stabilizing controllers for linear time-varying systems described by polytopic Linear Differential Inclusions (LDIs),  $\dot{x} = A(t)x$  with  $A(t) \in \text{convex-hull}(\mathcal{A})$ ; where we define  $\text{convex-hull}(\mathcal{A})$  as:

$$\text{convex-hull}(\mathcal{A}) = \left\{ A : A = \sum_{i=1}^k p_i A_i, p_i \geq 0, \sum_{i=1}^k p_i = 1 \right\}$$

We can do this by finding a bounded polytope  $\mathcal{C}$  that is contracted by  $\mathcal{A}$ . That is a  $\mathcal{C}$  such that for every  $A_i \in \mathcal{A}$ , if  $\dot{x} = A_i x$  then

$$x(0) \in \mathcal{C} \implies x(t) \in \text{int}(\mathcal{C}) \text{ for all } t \geq 0. \quad (19)$$

The Minkowski functional of  $\mathcal{C}$  is then a Lyapunov function for the LDI (Blanchini (1999)).

To use our algorithm to find bounded polytopes instead of polyhedral cones, we begin by transforming our  $n$  dimensional matrices  $A_i \in \mathcal{A}$  into extended  $n + 1$  dimensional matrices  $\hat{A}_i \in \hat{\mathcal{A}}$ :

$$\hat{A}_i = \begin{bmatrix} 0 & 0_n^T \\ 0_n & A_i \end{bmatrix} \quad (20)$$

And we set all  $r_i, h_i = [1, 0_n]^T$ .

We then apply Algorithm 1 or 2 as normal, obtaining a  $n + 1$  dimensional  $\mathcal{K}_{\mathcal{R}}(R)$  that satisfies (2) for each  $\hat{A}_i \in \hat{\mathcal{A}}$  and, in the case of Algorithm 2, a corresponding vector of parameters  $c$ .

We map a  $\mathcal{K}_{\mathcal{R}}(R)$  that satisfies (2) to a  $\mathcal{C}$  that satisfies (19) by taking a slice of the cone about the hyperplane



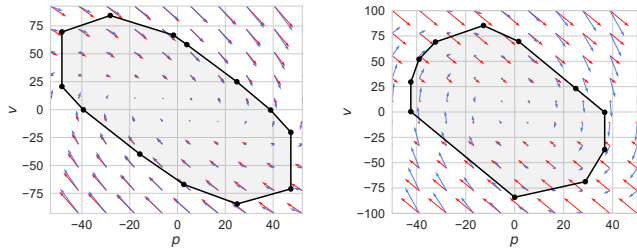


Fig. 5. Contracting Polytopes found for (21) and the vector fields generated by each of the two  $A_i \in \mathcal{A}$ .  
**Left:** Analysis problem with  $1 \leq \phi(t) \leq 3$  and  $k_p = 0$ .  
**Right:** Synthesis problem with  $-3 \leq \phi(t) \leq 3$  and  $k_p^{(0)} = 0$ . Final  $k_p = 3.02$ .

$[1, 0_n]^T x = 1$ . Numerically, if each column of  $R$  is scaled such that the first element is 1, this is equivalent to removing the first row of  $R$ , with each column of the matrix obtained representing a vertex of  $\mathcal{C}$ .

We apply this to the following time varying spring-damper system:

$$\begin{bmatrix} \dot{p} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -(\phi(t) + k_p) & -2 \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix} \quad (21)$$

Where  $\phi(t)$  represents an uncertain (and potentially negative) spring constant and  $k_p$  represents a proportional position feedback gain.

We initially consider the analysis problem by setting  $1 \leq \phi(t) \leq 3$  and  $k_p = 0$ . We use Algorithm 1 to obtain the contracting polytope shown in Fig. 5 (left). We then attempt the synthesis problem, setting  $-3 \leq \phi(t) \leq 3$  and letting  $c = k_p$  be a free parameter initialized to 0. Using Algorithm 2, we simultaneously obtain a stabilizing gain ( $k_p = 3.02$ ) and a contracting polytope, as shown in Fig. 5 (right). It is worth noting that attempting to find a Quadratic Lyapunov function for the LDI using linear matrix inequalities fails when  $-3 \leq \phi(t) \leq 3$  and  $k_p = 3.02$ .

In this example we have shown a new method for computing polyhedral Lyapunov functions (Blanchini and Miani (2015)). We leave a thorough investigation of this application area, including its differential version (Forni and Sepulchre (2014a)), as future work.

## 7. CONCLUSIONS

We presented a novel algorithm for constructing contracting polyhedral cones. The algorithm is based on Linear Programming and keeps the number of vectors in the representation of the cone fixed. This enabled us to verify and synthesize  $\mathcal{K}$ -cooperative systems without fixing a cone a priori. Building on differential positivity,  $\mathcal{K}$ -cooperative systems capture more behaviors than traditional differential analysis approaches, which we illustrated by providing examples of robust analysis and synthesis for bistable and multi-agent systems.

## ACKNOWLEDGEMENTS

The authors wish to thank I. Cirillo and F. Miranda for useful comments and suggestions to the manuscript.

## REFERENCES

- Angeli, D. and Sontag, E.D. (2003). Monotone control systems. *IEEE Transactions on Automatic Control*, 48(10), 1684–1698.
- Benvenuti, L. and Farina, L. (2004). Eigenvalue regions for positive systems. *Systems & Control Letters*, 51(3-4), 325–330.
- Berman, A., Neumann, M., and Stern, R.J. (1989). *Non-negative Matrices in Dynamic Systems*. Wiley.
- Blanchini, F. (1999). Set invariance in control. *Automatica*, 35(11), 1747–1767.
- Blanchini, F. and Miani, S. (2015). *Set-Theoretic Methods in Control*. Systems & Control: Foundations & Applications. Birkhäuser, Cham, second edition.
- Bushell, P.J. (1973). Hilbert’s metric and positive contraction mappings in a Banach space. *Archive for Rational Mechanics and Analysis*, 52, 330–338.
- Forni, F. and Sepulchre, R. (2014a). A Differential Lyapunov Framework for Contraction Analysis. *IEEE Transactions on Automatic Control*, 59(3), 614–628.
- Forni, F. and Sepulchre, R. (2016). Differentially Positive Systems. *IEEE Transactions on Automatic Control*, 61(2), 346–359.
- Forni, F. and Sepulchre, R. (2014b). Differential analysis of nonlinear systems: Revisiting the pendulum example. In *53rd IEEE Conference on Decision and Control*, 3848–3859.
- Hirsch, M.W. and Smith, H.L. (2006). Chapter 4 Monotone Dynamical Systems. *Handbook of Differential Equations: Ordinary Differential Equations*, 239–357.
- Khalil, H.K. (2002). *Nonlinear Systems*. Prentice Hall.
- Kousoulidis, D. and Forni, F. (2019). Finding cones for  $\mathcal{K}$ -cooperative systems. In *58th IEEE Conference on Decision and Control*.
- Leonov, G.A., Kuznetsov, N.V., and Bragin, V.O. (2010). On problems of Aizerman and Kalman. *Vestnik St. Petersburg University: Mathematics*, 43(3), 148–162.
- Lohmiller, W. and Slotine, J. (1998). On Contraction Analysis for Non-linear Systems. *Automatica*, 34(6), 683–696.
- Miranda-Villatoro, F.A., Forni, F., and Sepulchre, R. (2018). Differentially passive circuits that switch and oscillate. In *2nd IFAC Conference on Modelling, Identification and Control of Nonlinear Systems MICNON 2018*.
- Miranda-Villatoro, F.A., Forni, F., and Sepulchre, R. (2019). Dissipativity analysis of negative resistance circuits. *Submitted to Automatica (arxiv.org/abs/1908.11193)*.
- Pavlov, A.V., van de Wouw, N., and Nijmeijer, H. (2006). *Uniform Output Regulation of Nonlinear Systems: A Convergent Dynamics Approach*. Systems & Control: Foundations & Applications. Birkhäuser Basel.
- Protasov, V.Y. (2010). When do several linear operators share an invariant cone? *Linear Algebra and its Applications*, 433(4), 781–789.
- Russo, G., di Bernardo, M., and Sontag, E.D. (2010). Global Entrainment of Transcriptional Systems to Periodic Inputs. *PLOS Computational Biology*, 6(4), e1000739.
- Vandergraft, J.S. (1968). Spectral Properties of Matrices which Have Invariant Cones. *SIAM Journal on Applied Mathematics*, 16(6), 1208–1222.