Lyapunov Function Synthesis - Infeasibility and Farkas' Lemma

Leth, Tobias; Sloth, Christoffer; Wisniewski, Rafal; Sankaranarayanan, Sriram

Published in:
IFAC-PapersOnLine

DOI (link to publication from Publisher):
10.1016/j.ifacol.2017.08.339

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 50-1 (2017) 1667–1672

Lyapunov Function Synthesis - Infeasibility and Farkas' Lemma ⋆

Tobias Leth * **Christoffer Sloth** * **Rafał Wisniewski** *
Sriram Sankaranarayanan **

* *Department of Electrical Engineering, Section of Automation & Control, University of Aalborg, 9220 Aalborg East, Denmark (e-mail: {tol, ces, raf}@es.aau.dk)*
** *Department of Computer Science, University of Colorado, Boulder, CO, USA (e-mail: srirams@colorado.edu)*

Abstract: In this paper we prove the convergence of an algorithm synthesising continuous piecewise-polynomial Lyapunov functions for polynomial vector fields defined on simplices. We subsequently modify the algorithm to sub-divide locally by utilizing information from infeasible linear problems. We prove that this modification does not destroy the convergence of the algorithm. Both methods are accompanied by examples.

Keywords: Lyapunov methods, Stability of nonlinear systems, Algorithmic design

1. INTRODUCTION

Lyapunov stability has been the go to technique for asserting stability properties of dynamical systems for decades. The theory offers a way of proving the stability of equilibria of non-linear systems by means of the existence of a particular function. The theory is non-constructive in obtaining this function, and this non-constructiveness has been, and still is, the focus of significant research effort (Giesl and Hafstein, 2015). For general non-linearities some physical insight is often utilized to guide the search for Lyapunov functions, where they represent a generalized notion of energy (van der Schaft and Jeltsema, 2014). Sometimes no such insight is available and a more systematic approach is necessary.

A notable contribution to systematic Lyapunov function synthesis for polynomial systems came in 2000 when Pablo A. Parrilo published his PhD thesis on Semi-definite Programs (Parrilo, 2000). By formulating the Lyapunov stability criteria using sum-of-squares polynomials, the synthesis can be cast as linear matrix inequalities and solved by semi-definite programming. For more than a decade the toolbox SOSTOOLS (Prajna et al., 2005) has been the state of the art when it comes to Lyapunov function synthesis.

So far, no method has efficiently escaped the so-called *curse of dimensionality*. One technique to address this is to restrict the relaxation even further and require (scaled) diagonally dominant sum-of-squares polynomials (Ahmadi and Hall, 2015). This results in linear and second order cone programs, and approximates the solution to the semi-definite program. Linear and second order cone programs scale better than semi-definite programs, but at the same time the technique restricts the class of polynomials.

⋆ This work is supported by the Danish Council for Independent Research under grant number DFF - 4005-00452 in the project CodeMe.

One way to obtain linear programs without restricting the class of polynomials, is to formulate the synthesis using either the Handelman basis for general convex polytopes or the Bernstein basis for box or simplex polytopes. Using the Handelman basis, Kamyar et al. (2014) relaxes the Lyapunov criteria into inequalities on the coefficients resulting in a linear program. Whenever they encounter infeasibility they increase the degree of the Lyapunov function which they search for, while maintaining fixed polytopes. In Sassi and Sankaranarayanan (2015) properties of the Bernstein basis polynomials are utilized to obtain a hierarchy of linear program relaxations. They use box polytopes and a fixed sub-division method to handle infeasibility.

The work presented in this paper differs from Kamyar et al. (2014) and Sassi and Sankaranarayanan (2015) by using the positivity of coefficients together with an adaptive sub-division method and piecewise polynomials. We build on the work presented in Leth et al. (2016) and expand the techniques presented therein. Polynomials are defined in the simplicial Bernstein basis, and often the initial triangulation will result in an infeasible linear program. By the use of Farkas' lemma, information from an infeasible linear program is used to target the reason for the infeasibility and to sub-divide locally. In the examples this is shown to reduce the problem size and the linear programs are solved faster.

The paper is organised as follows: Section 2 offers a short introduction to the computations presented in Leth et al. (2016) and covers additional notation and definitions. In Section 3, the convergence of regular sub-division is proven. After deriving an irregular sub-division method in Section 4, Section 5 contains the main contribution which proves the convergence of the method. Section 6 outlines some differences between synthesizing continuous piecewise-polynomial and continuous differentiable polynomial Lyapunov functions, and a conclusion is given in Section 7.

2405-8963 © 2017, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2017.08.339

2. PRELIMINARIES

The paper Leth et al. (2016) covers how to derive a linear programming (LP) problem for synthesizing a continuous piecewise-polynomial Lyapunov function for polynomial vector fields. The vector field and Lyapunov function are described in the simplicial Bernstein basis, which, together with some additional notation, is introduced in the following.

A non-degenerate n-simplex (a simplex of dimension n) $\sigma \equiv [\sigma_0, \ldots, \sigma_n]$, with ordering $<$ such that $\sigma_i < \sigma_j$ if $i < j$, is the convex hull of $n+1$ affinely independent vertices $\sigma_i \in \mathbb{R}^n$, i.e.,

$$\sigma = \left\{ \sum_{i=0}^{n} \lambda_i \sigma_i \;\middle|\; \sum_{i=0}^{n} \lambda_i = 1 \wedge \lambda_i \geq 0, i = 1, \ldots, n \right\} \subset \mathbb{R}^n,$$

where λ_i are the barycentric coordinates for σ.

Denoting $\mathbb{N} \cup \{0\}$ by $\mathbb{N}_0$, we employ the following shorthand notation. For $\alpha \in \mathbb{N}_0^{n+1}$ we write

$$|\alpha| = \sum_{i=0}^{n} \alpha_i, \qquad \lambda^\alpha = \prod_{i=0}^{n} \lambda_i^{\alpha_i}, \tag{2}$$

and

$$\mathcal{B}_\alpha^D = \binom{D}{\alpha} \lambda^\alpha, \qquad \binom{D}{\alpha} = \frac{D!}{\alpha_0! \alpha_1! \cdots \alpha_n!}. \tag{3}$$

With this, any polynomial p of degree d in n variables can be described in the Bernstein basis of degree $D \geq d$ on a simplex σ as

$$p = \sum_{|\alpha|=D} b_\alpha(p, D, \sigma) \mathcal{B}_\alpha^D(\sigma) = b(p, D, \sigma) \mathcal{B}^D(\sigma), \tag{4}$$

where the vector $b(p, D, \sigma)$ contains the coefficients which uniquely describe p on σ, and $\mathcal{B}^D(\sigma)$ are the Bernstein basis polynomials of degree D on σ.

The notion of a simplex is extended to collections of simplices. Let $K = \{\sigma^1, \cdots, \sigma^m\}$ be a finite set of m non-overlapping (except for faces) n-simplices, and let them be ordered by $<$.

Notation 1. Let $U \subset \mathbb{R}^n$ be a closed polytope with the origin in its interior. Let the collection of simplices K be a triangulation of U such that

$$|K| = \bigcup_{\sigma \in K} \sigma = U, \tag{5}$$

where $|K|$ is the realisation of K (Basu et al., 2006). We say that K covers U.

Denote all coefficients in the collection by C (ordered by $<$), and let $\Delta = [1, \ldots, N_\Delta]$ be a list of the indices of the coefficients in the collection as seen in Fig. 1, where N_Δ is the number of coefficients. On each simplex σ, each $|\alpha| = D$ defines one coefficient as in (4). We reference to the coefficients on σ^i by $C^i = C(\Delta^i)$ where Δ^i denote the indices related to σ^i, i.e., $\Delta^1 = [1, 2, 3, 6, 7, 8]$. Comparing to (4) we have $b(p, D, \sigma^i) = C^i$. We employ this shorthand notation and assume the polynomial p and the degree D obvious from context. We define polynomials as:

Definition 2. (Continuous Piecewise-Polynomial (CPP) on Collection of Simplices). Let K be a collection of m simplices, let p be a CPP of degree d in n variables, and let C be the vector of all coefficients in the collection. Then

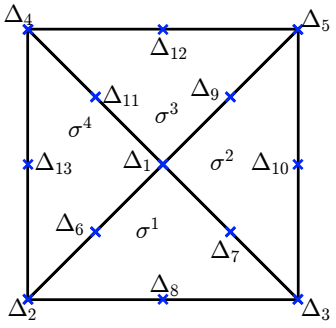

Fig. 1. Defining polynomials in the Bernstein basis on a collection of simplices. σ relates to the simplices and Δ relates to the coefficients.

$$p(x) = p_i(x) \; \forall x \in \sigma^i \text{ for } i = 1, \ldots, m, \tag{6}$$

where

$$p_i = \sum_{|\alpha|=d} C_\alpha^i \mathcal{B}_\alpha^d(\sigma^i) = C^i \mathcal{B}^d(\sigma^i). \tag{7}$$

Note that the continuity of p across faces is ensured since adjacent p_i's have the same coefficients on the shared faces, thus

$$p_i(x) = p_j(x) \; \forall x \in \sigma^i \cap \sigma^j. \tag{8}$$

The classical Lyapunov criteria can be translated to conditions on the coefficients as follows.

Lemma 3. (Sloth and Wisniewski, 2014). Let V be a CPP of degree d defined as in Definition 2, and denote its coefficients by CV. For a given polynomial vector field, let $\dot{V}$ be the Lie derivative of degree $\hat{d}$, and denote its coefficient vector by CL. Let V and $\dot{V}$ be defined on a collection of m simplices and assume, without loss of generality, that the simplices are numbered such that the origin is a vertex of the first $\bar{m}$ simplices. Denote by e_j the j^{th} canonical unit vector. If

$$CV \geq 0 \tag{9a}$$
$$CV_{de_0}^i = 0 \; \forall i \in \{1, \ldots, \bar{m}\} \tag{9b}$$
$$CV_{de_j}^i > 0 \; \forall i \in \{1, \ldots, \bar{m}\}, \forall j \in \{1, \ldots, n\} \tag{9c}$$
$$CV_{de_j}^i > 0 \; \forall i \in \{\bar{m}+1, \ldots, m\}, \forall j \in \{0, \ldots, n\} \tag{9d}$$
$$CL \leq 0 \tag{9e}$$
$$CL_{\hat{de}_0}^i = 0 \; \forall i \in \{1, \ldots, \bar{m}\} \tag{9f}$$
$$CL_{\hat{de}_j}^i < 0 \; \forall i \in \{1, \ldots, \bar{m}\}, \forall j \in \{1, \ldots, n\} \tag{9g}$$
$$CL_{\hat{de}_j}^i < 0 \; \forall i \in \{\bar{m}+1, \ldots, m\}, \forall j \in \{0, \ldots, n\}, \tag{9h}$$

then $x = 0$ is a local asymptotically stable equilibrium point.

The importance of this reformulation comes from the fact that the coefficient vector of the Lie derivative is linear in the unknown coefficients of the Lyapunov function, specifically

$$CL = A \; CV, \tag{10}$$

where A is a matrix as in (11). For the Lie derivative on the collection of simplices K, we stack the matrices like

$$A = \begin{bmatrix} A_1 \\ \cdots \\ A_m \end{bmatrix}, \tag{11}$$

where A_i relates to simplex σ^i. Thus (9), (10) constitutes an LP. See Leth et al. (2016) for a derivation of A.

Notation 4. Let $f : U \to \mathbb{R}^n$, where $U \subset \mathbb{R}^n$ is a closed polytope, be a polynomial vector field defining the dynamical system $\Sigma : \dot{x} = f(x)$. The LP defined by (9) and (10) for Σ on a collection of simplices K which covers U, is written $LP(\Sigma, K)$. To be read as "The linear synthesis problem for Σ, when V and f are defined on K".

In addition we define the following notions.

Notation 5. A vector of coefficients CV is called feasible if it fulfils (9), and $LP(\Sigma, K)$ is called solvable if a feasible CV exists. Otherwise it is called unsolvable.

Since Bernstein's Theorem (Leroy, 2011) is only valid for positive polynomials and since we strive to design positive definite polynomials we introduce the following.

Notation 6. Let $p : U \to \mathbb{R}$, with $U \subset \mathbb{R}^n$ a closed polytope, be a positive definite polynomial with one point $x^* \in U$ such that $p(x^*) = 0$. A collection K covering U is said to be a Bernstein basis certifying collection for p, if $C^i \geq 0$ for all $i \in \{1, \ldots, m\}$, when p is described on K.

As a natural consequence, the point x^* needs to be at a vertex in K. The notion is extended to Lyapunov functions.

Notation 7. Let V be a Lyapunov function and $\dot{V}$ its Lie derivative for a given polynomial vector field Σ. The collection K is said to be a Lyapunov Bernstein basis certifying collection for V, if it is a Bernstein basis certifying collection for V and $-\dot{V}$.

In addition, if the vertices in a collection K are such that

$$\sigma_i^j \in \mathbb{Q} \ \forall j \in \{1, \ldots, m\}, i \in \{0, \ldots, n\}, \tag{12}$$

with $\mathbb{Q}$ the rational numbers, then K is called a rational collection. Lastly we need the notion of diameter of simplices.

Notation 8. Let the diameter of a simplex be the maximal distance between any two points in the simplex, and denote the diameter of σ by $h(\sigma)$. For the collection $K = \{1, \ldots, m\}$ let

$$h = \max_{\sigma \in K} h(\sigma) \tag{13}$$

be the maximal diameter of any simplex in K.

This concludes the needed notation. The following sections introduce strategies for modifying K when the initial $LP(\Sigma, K)$ is unsolvable.

3. CONVERGENCE FOR REGULAR SUB-DIVISION

This section restates the algorithm first presented in Leth et al. (2016), proves its convergence, and gives an example of its use. First, define the shrinking factor $S \in \mathbb{R}$ of a sub-division routine as

$$Sh^{\{k\}} = h^{\{k+1\}}, \tag{14}$$

where $h^{\{k\}}$ is the maximal diameter in the collection at iteration k. Thus S is a measure of how much the diameters of the simplices in a collection shrinks by applying the routine once. We will refer to any sub-division routine with $0 < S < 1$ as a regular sub-division routine. Two such routines are the Standard Triangulation and the Binary Splitting (Leroy, 2011).

Algorithm 9. (BBAlgorithm)
Input: Dynamical system Σ, closed polytope U, and degree of the Lyapunov function $d_V \geq 1$.
Output: Triangulation of U and Lyapunov function V defined on the resulting simplices.
Procedure:

0) Set iteration counter $k = 1$ and get initial $K^{\{k\}}$.
1) If $LP(\Sigma, K^{\{k\}})$ is solvable, then return $K^{\{k\}}$ and CV. Otherwise, go to 2).
2) Use a regular sub-division routine on all simplices in $K^{\{k\}}$ to get $K^{\{k+1\}}$ and set $k = k + 1$. Go to 1).

We start with the following Lemma.

Lemma 10. Let a polynomial p of degree d be defined in the Bernstein basis on a simplex $\hat{\sigma}$, and let $b(p, d, \hat{\sigma}) \geq 0$. Then any sub-division of $\hat{\sigma}$ into a collection K with m simplices, such that

$$\hat{\sigma} = |K| = \bigcup_{\sigma \in K} \sigma, \tag{15}$$

will preserve $b(p, d, \sigma^i) \geq 0, \ \forall i \in \{1, \ldots, m\}$.

Proof. This follows from the fact that $b(p, d, \sigma^i)$ are calculated as convex combinations of $b(p, d, \hat{\sigma})$. See Leroy (2011) for details. ■

Proposition 11. Let V^* be a polynomial Lyapunov function of degree $d_{V^*} \leq d_V$ (d_V is the degree input to the BBAlgorithm) for the locally asymptotically stable system Σ. Assume the existence of a rational Lyapunov Bernstein basis certifying collection K^* for V^*. Then the BBAlgorithm converges to a collection K' making $LP(\Sigma, K')$ solvable in finitely many steps, and the solution V' is a Lyapunov function for Σ.

Proof. Let $K^* = \{\sigma^{*1}, \ldots, \sigma^{*m^*}\}$ be the rational Lyapunov Bernstein basis certifying collection for V^*. Because K^* is rational, finitely many applications of any regular sub-division routine will result in a collection $\bar{K}$ satisfying

$$\sigma^{*i} = |K^{*i}| = \bigcup_{\bar{\sigma} \in K^{*i}} \bar{\sigma} \ \forall i \in \{1, \ldots, m^*\}, \tag{16}$$

where $K^{*i} \subset \bar{K}$. That is, any simplex σ^{*i} in K^* can be expressed as a collection of simplices K^{*i} taken from the simplices in $\bar{K}$. This also results in all vertices of K^* being vertices of $\bar{K}$. Setting $K' = \bar{K}$, it follows from Lemma 10 that the coefficients of V' when described on K' are $CV' \geq 0$, since the coefficients of V^* when described on K^*, by assumption, are $CV^* \geq 0$.

Prior to obtaining the collection $\bar{K}$, it may happen at iteration k that the BBAlgorithm obtains a collection $K^{\{k\}}$ which makes $LP(\Sigma, K^{\{k\}})$ solvable without $K^{\{k\}}$ fulfilling (16). In this case $K' = K^{\{k\}}$ and the algorithm has still converged. ■

We have two remarks on the existence of Lyapunov functions and the BBAlgorithms ability to find them.

Remark 12. In the case where $d_{V^*} > d_V$, the convergence cannot be proven without additional assumptions on K'. Since V' is piecewise continuous it can approximate polynomials of higher degree by additional sub-divisions, but it cannot be guaranteed that a rational Lyapunov Bernstein basis certifying collection will exists for the approximation.

Assuming this existence makes the BBAlgorithm converge even for $d_{V^*} > d_V$.

Remark 13. For locally exponentially stable systems, Giesl and Hafstein (2014) proves that the existence of a continuous piecewise affine Lyapunov function is a necessary and sufficient condition. Their proof relies on a specific triangulation method which results in a rational collection, thus the assumption on the existence of a rational Lyapunov Bernstein basis certifying collection *is not* conservative when the system is locally exponentially stable.

Consider next an example on the use of the BBAlgorithm.

Example 14. In this example we will apply the BBAlgorithm on a vector field Σ from Ahmadi and Parrilo (2011) given as

$$\begin{aligned}\dot{x}_1 =& -x_1^3x_2^2 + 2x_1^3x_2 - x_1^3 + 4x_1^2x_2^2 - 8x_1^2x_2 + 4x_1^2 \\ & - x_1x_2^4 + 4x_1x_2^3 - 4x_1 + 10x_2^2 \\ \dot{x}_2 =& -9x_1^2x_2 + 10x_1^2 + 2x_1x_2^3 - 8x_1x_2^2 - 4x_1 - x_2^3 \\ & + 4x_2^2 - 4x_2.\end{aligned} \tag{17}$$

We wish to investigate the local stability of the origin by considering the domain $U = [-3.5, 3.5]^2$ with a Lyapunov function of degree $d_V = 4$. The initial collection, consisting of 4 simplices, renders $LP(\Sigma, K^{\{1\}})$ unsolvable. We employ the Binary Splitting to refine the collection in step 2. During the 7^{th} iteration Binary Splitting has been applied 6 times and the collection consists of 256 simplices. This leaves $LP(\Sigma, K^{\{7\}})$ solvable and the resulting Lyapunov function is show in Fig. 2. The solvable LP consists of 2,212 variables and 11,520 constraints. Note that an objective function minimizing the sum of the Bernstein coefficients was added to improve the visual presentation. This does not affect the feasibility of (9).

4. INFEASIBILITY INVESTIGATION

In the attempt of obtaining a solvable $LP(\Sigma, K^{\{k\}})$ with a low number of variables and constraints, the initial $K^{\{1\}}$ will triangulate the domain of interest using as few simplices as possible, while making the equilibrium point a vertex in the collection. By Algorithm 9 and Proposition 11 any regular sub-division routine will converge to a Lyapunov function. However, the algorithm requires the sub-division of every simplex in the collection at every iteration. This can potentially result in the sub-division of simplices which do not need further sub-division. It is reasonable to expect that the source of the unsolvability of $LP(\Sigma, K^{\{k\}})$ does not originate from every simplex in the collection, but rather from a (possibly small) subset of the simplices.

In the following, using Farkas' lemma, we show how to identify the constraints, and thus simplices, responsible for the infeasibility.

Lemma 15. (Farkas' Lemma). (Boyd and Vandenberghe, 2004), (Andersen, 2013). Let $A \in \mathbb{R}^{l\times k}$ and $b \in \mathbb{R}^l$ define an LP. Then exactly one of the two propositions is true:

1. $\exists x \;:\; Ax = b,\; x \geq 0,$
2. $\exists y \;:\; A^T y \leq 0,\; b^T y > 0.$

That is, if the LP 1. is infeasible, then there exists a $y \in \mathbb{R}^l$ certifying the infeasibility. And moreover, the non-zero elements of y correspond to (some of) the constraints

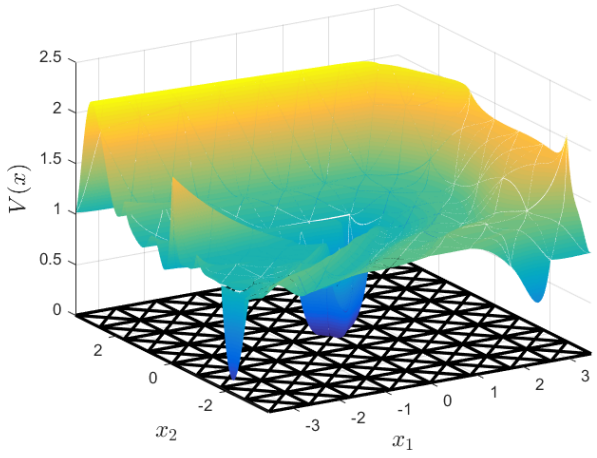

Fig. 2. Lyapunov function synthesised when Binary Splitting is applied on all simplices in the collection. The solid lines are the facets of the simplices.

responsible for the infeasibility (Andersen, 2013). Note that when solving using either simplex or dual-simplex methods, the certificate is a so-called basis certificate (Andersen, 2001). When solving using an interior-point algorithm the obtained certificate is a linear combination of all basis certificates, but the basis certificates can be recovered using the methods described in Andersen (2001). These methods are not detailed here, and henceforth when referring to a certificate of infeasibility we assume it to be a basis certificate.

Using this, we propose the following method for obtaining a sub-division where simplices are sub-divided by a regular sub-division routine, but where only some simplices are sub-divided at each iteration.

Notation 16. (Certificate of Infeasibility Identifying (CII) Method) When an $LP(\Sigma, K)$ is unsolvable, use Farkas' lemma to get a certificate of infeasibility y to identify, possibly a subset, of the constraints responsible for the infeasibility. Simplices containing any of these constraints are sub-divided. We will reference to such simplices as CII simplices.

5. CONVERGENCE FOR IRREGULAR SUB-DIVISION

This section covers how to utilize the CII method in the algorithm and expands the proof to encompass this. First we introduce the notion of a reduced collection.

Notation 17. Let $K = \{\sigma^1, \ldots, \sigma^{j-1}, \sigma^j, \sigma^{j+1} \ldots, \sigma^m\}$ be the entire collection and let

$$\hat{K} = \{\sigma^1, \ldots, \sigma^{j-1}, \sigma^{j+1}, \ldots, \sigma^m\} \tag{18}$$

denote the collection with the simplex σ^j removed. We will reference to $\hat{K}$ as the reduced collection. The $\hat{}$ notation can in general reference to more than one simplex being removed, but this will be obvious from context.

The BBAlgorithm is modified such that step 2 becomes:

2) *Use a regular sub-division on CII simplices to get* $K^{\{k+1\}}$ *and set* $k = k + 1$. *Go to step 1.*

Proposition 18. Let V^* be a polynomial Lyapunov function of degree $d_{V^*} \leq d_V$ (d_V is the degree input to the

modified BBAlgorithm) for the local asymptotically stable system Σ. Assume the existence of a rational Lyapunov Bernstein basis certifying collection K^* for V^*. Then the modified BBAlgorithm converges to a collection K' making $LP(\Sigma, K')$ solvable in finitely many steps, and the solution V' is a Lyapunov function for Σ.

Proof. It suffices to show that the CII method eventually will select all simplices with constraints which are responsible for the infeasibility of $LP(\Sigma, K^{\{k\}})$, then the rest follows from the proof of Proposition 11.

Let $I^{\{k\}}$ be an index set of all simplices with one or more constraints contributing to the infeasibility of $LP(\Sigma, K^{\{k\}})$, and let $I_y^{\{k\}}$ be an index set of the CII simplices at the k^{th} iteration. Then $LP(\Sigma, \hat{K}^{\{k\}})$ is solvable, where $\hat{K}^{\{k\}}$ is the reduced collection with the $I^{\{k\}}$ simplices removed. Then $I_y^{\{k\}}$ contains at least one element from $I^{\{k\}}$, and the $I_y^{\{k\}}$ simplices are sub-divided. Let $I_\sigma^{\{k\}}$ denote index set of the simplices created by the sub-division in the k^{th} iteration.

In the $(k+1)^{\text{th}}$ iteration, given that $LP(\Sigma, K^{\{k+1\}})$ is unsolvable, one of the following two scenarios will happen.

1) No element from $I_\sigma^{\{k\}}$ is in $I^{\{k+1\}}$ which makes $I^{\{k+1\}} = I^{\{k\}} \backslash I_y^{\{k\}}$. Thus $I_y^{\{k+1\}}$ contains at least one element from $I^{\{k+1\}}$. Continuing in this fashion, $I^{\{k+i\}}$ becomes empty after at most $i_{\max}$ iterations where $i_{\max}$ is the number of elements in $I^{\{k\}}$.
2) At least one element from $I_\sigma^{\{k\}}$ is in $I^{\{k+1\}}$. This makes $I_y^{\{k+1\}}$ contain at least one element from $I^{\{k\}} \backslash I_y^{\{k\}}$ or at least one element from $I_\sigma^{\{k\}} \cap I^{\{k+1\}}$. If $I_y^{\{k+1\}}$ contains elements from $I^{\{k\}} \backslash I_y^{\{k\}}$ the scenario is exactly what happens in 1), except, for some $i \in \{1, \ldots, i_{\max}\}$, $I_y^{\{k+i\}}$ will contain at least one element from the set

$$\bigcup_{j=1}^{i} I_\sigma^{\{k+j-1\}} \cap I^{\{k+j\}}. \tag{19}$$

For this reason, it is inevitable that simplices which are the result of sub-division will eventually become sub-divided even further. The above argument can be applied recursively, and (16) will be satisfied. The rest of the proof follows the proof of Proposition 11. ■

Consider next an example utilizing the CII method.

Example 19. This example revisits the vector field Σ from the previous example with the same domain and Lyapunov function degree. We employ Binary Splitting only on the CII simplices in step 2 of the BBAlgorithm. This makes the BBAlgorithm terminate after 6 iterations, and the resulting Lyapunov function is shown in Fig. 3. Again the sum minimizing objective function was added to improve visual presentation. It is evident that only sub-dividing the CII simplices results in a customized triangulation. The final collection K' consists of 21 simplices, and the program has 186 variables and 945 constraints. This is an improvement of 91 % in the number of variables compared to applying Binary Splitting on all simplices.

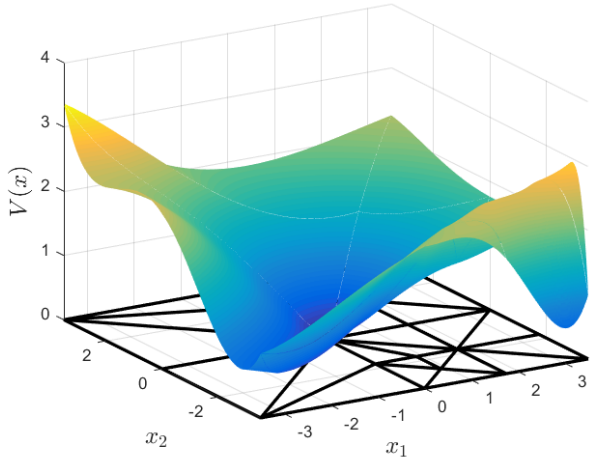

Fig. 3. Lyapunov function synthesised when Binary Splitting is applied on CII simplices in the collection. The solid lines are the facets of the simplices.

6. CONTINUOUS DIFFERENTIABLE VERSUS CONTINUOUS PIECEWISE

The analysis has solely focused on continuous piecewise-polynomials (CPP) as defined in Section 2. With a slight modification of the above, the class of Lyapunov functions can be restricted to continuous differentiable polynomials (CDP) on collections of simplices. In (4) p is defined on a simplex σ. By a linear change of base, p can be described on any simplex. Define ${}^jB_i$ as the matrix transforming base from σ^i to σ^j, with ${}^jB_i = I$ if $i = j$.

Definition 20. (Continuous Differentiable Polynomial (CDP) on Collection of Simplices). Let p be defined as in Def. 2, but also require that

$$C^j = {}^jB_i C^i. \tag{20}$$

This connection between coefficients on different simplices essentially makes p one polynomial defined on the collection. This makes the BBAlgorithm search a smaller class of functions, but the number of design variables is invariant to the number of simplices. As m grows $LP(\Sigma, K)$ only grows in terms of equality constraints (20) and inequality constraints (9). The vector field Σ from the previous examples is used to assess the differences between the two methods. First, note that in order for the BBAlgorithm to terminate in a reasonable manner when searching for CDP Lyapunov functions, the domain was shrunk to $U = [-2.5, 2.5]^2$. The vector field was analysed using CPP and CDP Lyapunov functions of degree $d_V = 4$, and using both Binary Splitting on all simplices and on CII simplices. The results are reported in Table 1.

For CDP, note that using the CII method results in more simplices than sub-dividing all simplices. The reason for this is that during the first two iterations the CII method fails to identify all simplices which needs to be sub-divided. The method only finds a sub-set of the simplices, and the remaining are then identified during the following iterations. At this point, however, some of the smaller simplices are simultaneously identified for further sub-division even though they do not contribute to the infeasibility. Seeing as the LPs have the same number of variables and are sill quite small, most of the computation

Table 1. Comparison between synthesising CDP and CPP Lyapunov functions.

		#Var	#Con	#σ	Runtime	Iterations
CDP	BS all	14	3425	64	5.7 [s]	5 (BS 4 times)
	BS CII	14	3480	65	8.1 [s]	8 (BS 7 times)
CPP	BS all	288	1440	32	3.6 [s]	4 (BS 3 times)
	BS CII	112	540	12	3.5 [s]	5 (BS 4 times)

time is spent on overhead setting up the programs. This shows in the longer runtime for the CII method.

For CPP, first of all notice the advantage by searching over a larger class of functions. For Binary Splitting on all simplices there exists a CPP Lyapunov function when V and f are defined on 32 simplices, whereas 64 simplices are needed before a CDP Lyapunov function exists. Contrary to CDP, applying the CII method when searching for a CPP Lyapunov function does speed up the runtime. The CII method requires an additional iteration before obtaining a solvable $LP(\Sigma, K')$, but since the program is keep small by minimizing the number of simplices the runtime is shorter than when sub-dividing all simplices.

The tendencies in Tab. 1 have not been investigated on a comprehensive case study. It is of yet unknown whether the CII method is advantageous in high dimension.

6.1 Sparsity

When searching for a CDP Lyapunov function the number of design variables is constant during the iterations of the BBAlgorithm. As such, the inequality matrix A in (10) will maintain its width and only become taller throughout the iterations. For this reason, the linear program for searching for a CDP Lyapunov function offers no sparsity.

When searching for a CPP Lyapunov function the number of design variables grows as the number of simplices grows during the iterations. The coefficients located on facets of the simplices in the collection will be common to two or more simplices, and appear multiple times in the inequality matrix A. The coefficients located in the interior of the simplices will be unique to a single simplex. Thus the inequality matrix A can be ordered into a structure known as dual block angular (Lubin et al., 2013). This structure offers a high degree of sparsity and can potentially be exploited to develop a dedicated solver.

7. CONCLUSION

In this paper the convergence of a Lyapunov function synthesizing algorithm was proven. Whenever infeasibility occurs, using Farkas' lemma we were able to target specific simplices which were responsible for the infeasibility. This enables a method where the problem size is kept at a minimum while obtaining a feasible linear program and synthesizing a Lyapunov function. A comparison between continuous piecewise-polynomials and continuous differentiable polynomials showed that the proposed method is best suited when searching for a continuous piecewise-polynomial Lyapunov function.

Future research will address the gap between positive definite polynomials and polynomials admitting a Bernstein basis certifying collection.

ACKNOWLEDGEMENT

The authors thanks Joachim Dahl and Erling D. Andersen from MOSEK for bringing Farkas' Lemma to our attention, fruitful discussions, and technical support.

REFERENCES

Ahmadi, A.A. and Hall, G. (2015). Sum of squares basis pursuit with linear and second order cone programming. *Contemporary Mathematics*, 1–26.

Ahmadi, A.A. and Parrilo, P.A. (2011). Converse results on existence of sum of squares Lyapunov functions. *Proceedings of the Conference on Decision and Control*, 6516–6521.

Andersen, E.D. (2001). Certificates of primal or dual infeasibility in linear programming. *Computational Optimizations and Applications*, 20(2), 171–183.

Andersen, E.D. (2013). How to use Farkas' lemma to say something important about infeasible linear problems. Technical report, MOSEK. URL `http://docs.mosek.com/whitepapers/infeas.pdf`.

Basu, S., Pollack, R., and Roy, M.F. (2006). *Algorithms in real algebraic grometry*, volume 10. Springer.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Giesl, P. and Hafstein, S. (2015). Review on computational methods for Lyapunov functions. *Discrete and Continuous Dynamical Systems*, 20(8), 2291–2331.

Giesl, P.A. and Hafstein, S.F. (2014). Revised CPA method to compute Lyapunov functions for nonlinear systems. *Journal of Mathematical Analysis and Applications*, 410(1), 292–306.

Kamyar, R., Murti, C., and Peet, M.M. (2014). Constructing piecewise-polynomial Lyapunov functions for local stability of nonlinear systems using Handelmans theorem. *CDC, Los Angeles, USA*, 5481–5487.

Leroy, R. (2011). Certificates of positivity in the simplicial Bernstein basis. *hal-00589945*.

Leth, T., Sloth, C., and Wisniewski, R. (2016). Lyapunov function synthesis - algorithm and software. *MSC, Buenos Aires, Argentina*, 641–647.

Lubin, M., Hall, J.A.J., Petra, C.G., and Anitescu, M. (2013). Parallel distributed-memory simplex for large-scale stochastic LP problems. *Computational Optimizations and Applications*, 55(3), 571–596.

Parrilo, P.A. (2000). *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. Ph.D. thesis, California Institute of Technology.

Prajna, S., Papachristodoulou, A., Seiler, P., and Parrilo, P.A. (2005). SOSTOOLS and its control applications. *Positive Polynomials in Control*, 312, 273–292.

Sassi, M.A.B. and Sankaranarayanan, S. (2015). Linear relaxations of polynomial positivity for polynomial Lyapunov function synthesis. *Journal of Mathematical Control and Information*, 1–34.

Sloth, C. and Wisniewski, R. (2014). Robust stability of switched systems. *CDC, Los Angeles, USA*, 4685–4690.

van der Schaft, A. and Jeltsema, D. (2014). *Port-hamiltonian systems theory: an introductiry overview*, volume 1, 173–378. Foundations and Trends® in Systems and Control.