

Linear relaxations of polynomial positivity for polynomial Lyapunov function synthesis

MOHAMED AMIN BEN SASSI AND SRIRAM SANKARANARAYANAN*

Department of Computer Science, University of Colorado, Boulder, CO, USA

*Corresponding author: srirams@colorado.edu

AND

XIN CHEN AND ERIKA ÁBRAHÁM

Department of Computer Science, RWTH Aachen University, Aachen, Germany

[Received on 5 April 2014; revised on 14 November 2014; accepted on 11 January 2015]

We examine linear programming (LP) based relaxations for synthesizing polynomial Lyapunov functions to prove the stability of polynomial ordinary differential equations (ODEs). Our approach starts from a desired parametric polynomial form of the polynomial Lyapunov function. Subsequently, we encode the positive definiteness of the function, and the negation of its derivative, over the domain of interest. We first compare two classes of relaxations for encoding polynomial positivity: relaxations by sum-of-squares (SOS) programmes, against relaxations based on Handelman representations and Bernstein polynomials, that produce linear programmes. Next, we present a series of increasingly powerful LP relaxations based on expressing the given polynomial in its Bernstein form, as a linear combination of Bernstein polynomials. Subsequently, we show how these LP relaxations can be used to search for Lyapunov functions for polynomial ODEs by formulating LP instances. We compare our techniques with approaches based on SOS on a suite of automatically synthesized benchmarks.

Keywords: positive polynomials; sum-of-squares; Bernstein polynomials; interval arithmetic; Handelman representations; stability; Lyapunov functions.

1. Introduction

The problem of discovering **stability proofs** for closed-loop systems in the form of Lyapunov functions is an important step in the **formal verification of closed-loop control systems** (Tabuada, 2009). Furthermore, extensions of Lyapunov functions such as *control Lyapunov functions* can be used to design controllers, and *input-to-state stability (ISS) Lyapunov functions* are used to verify the stability of interconnected systems in a component-wise fashion.

In this paper, we focus on the **synthesis of polynomial Lyapunov functions for proving the stability of autonomous systems with polynomial dynamics** using linear programming (LP) relaxations. At its core, this requires us to **find a positive-definite polynomial whose Lie derivatives are negative definite**. Therefore, the problem of finding a Lyapunov function depends intimately on techniques for finding positive-definite polynomials over the domain K of interest. By finding a Lyapunov function over K , we ensure the existence of a region (neighbourhood of the equilibrium) contained in K such that the system is stable. But **proving that a multivariate polynomial is positive definite over an interval is co-NP hard**, and therefore considered to be a hard problem (Garey & Johnson, 1979). Many relaxations to this problem have been studied, wherein a relaxed procedure can either conclude that

the polynomial is positive definite with certainty, or fail with no conclusions. We examine two main flavours of relaxation:

1. The first class of *linear representations* involve the expression of the target polynomial to be proved non-negative over the set K of interest as a linear combination of polynomials that are known to be non-negative over the set K . This approach reduces the polynomial positivity problem to a LP.
2. Alternatively, a different class of approaches uses ‘Sum-Of-Squares representations’ (Datta, 2002). This approach yields relaxations based on semi-definite programming (SDP) (Shor, 1987; Lasserre, 2001; Parillo, 2003).

As a first contribution of this paper, we extend the so-called Handelman representations, considered in our previous work (Sankaranarayanan *et al.*, 2013), using the idea of Bernstein polynomials from approximation theory (Bernstein, 1915; Farouki, 2012; Muñoz & Narkawicz, 2013). Bernstein polynomials are a special basis of polynomials that have many rich properties, especially over the unit interval $[0, 1]$. For instance, tight bounds on the values of these polynomials over the unit interval are known. We show three LP relaxations, each more precise than the previous, that exploit these bounds in the framework of a *reformulation linearization approach* (Sherali & Tuncbilek, 1991, 1997). Next, we compare Bernstein relaxations against SOS relaxations, demonstrating polynomials that can be shown to be positive using one, but not the other.

Finally, the main contribution of the paper consists of adapting Bernstein relaxations for finding Lyapunov functions over rectangular domain K . The key difference is that, to find a Lyapunov function, we search for a parametric polynomial $V(\vec{x}, \vec{c})$ for unknown coefficients \vec{c} , which is positive definite, and whose derivative is negative definite over the region of interest. A straightforward approach leads to a bilinear programme, that can be dualized as a multiparametric programme. We apply the basic requirements for a Lyapunov function, to cast the multiparametric programme back into an LP, *without any loss in precision*.

We have implemented the approach and describe our results on a suite of examples. We also compare our work with an SOS programming relaxation using the SOSTOOLS package (Papachristodoulou *et al.*, 2013). On the one hand, we find that LP-based relaxations presented in this paper can find Lyapunov functions for more benchmark instances while suffering from fewer numerical issues when compared to an SOS programming approach. Overall, the LP relaxations are shown to present a promising approach for synthesizing Lyapunov functions over a bounded rectangle K .

1.0.1. Organization Section 2 presents some preliminary notions of Lyapunov functions, representations of positive polynomials including Handelman, Schmüdgen and Putinar representations. We then present the basic framework for synthesizing Lyapunov functions by formulating a parametric polynomial that represents the desired function. Section 3 presents the basic properties of Bernstein polynomials and three LP relaxations for proving polynomial positivity. In Section 4, we compare first linear and SOS relaxations and then we compare the proposed Bernstein relaxations with existing linear ones. Next, we describe the synthesis of Lyapunov functions using Bernstein relaxations in Section 5. Section 6 presents the numerical results.

An extended version of this paper including the benchmark examples used in our evaluation along with the Lyapunov functions found for each is available through arXiv (Sassi *et al.*, 2014).

1.1 Related work

In this section, we restrict our discussion to those works that are closely related to the overall problem of finding Lyapunov functions for polynomial systems.

Much research has focused on the topic of stability analysis for polynomial systems, which continues to be a challenging problem. The sum-of-squares (SOS) relaxation approach is quite popular, and has been explored by many authors (Papachristodoulou & Prajna, 2002; Jarvis-Wloszek, 2003; Tan & Packard, 2007; Topcu *et al.*, 2007). Papachristodoulou & Prajna (2002) were among the first to use SOS relaxations for finding polynomial Lyapunov functions. The core idea is to express the polynomial and its negative Lie derivative as sum-of-square polynomials for global stability analysis, or use a suitable representation such as Putinar representation for finding Lyapunov functions over a bounded region. Their approach is implemented in the SOSTOOLS package (Papachristodoulou *et al.*, 2013). Extensions have addressed the problem of controller synthesis (Jarvis-Wloszek, 2003), finding the region of stability (Tan & Packard, 2007); and using a combination of numerical simulations with SOS programming to estimate the region of stability (Topcu *et al.*, 2007). A related set of approaches directly relax the positivity of the Lyapunov form and the negativity of its derivative using Linear Matrix Inequalities (LMIs) (Tibken, 2000; Chesi *et al.*, 2005; Chesi, 2005, 2010; Henrion and Lasserre, 2006). Algebraic methods based, for example, on Gröbner basis (Forsman, 1991), or on constructive semi-algebraic systems techniques have been explored (She *et al.*, 2009, 2013).

While the approach in this paper focuses on polynomial system stability using polynomial Lyapunov functions, the general problem of analysing non-linear systems with rational, trigonometric and other non-linear terms has received lesser attention. Significantly, Papachristodoulou *et al.* present SOS relaxations for the stability of non-polynomial systems through a process of *algebraization* that augments the original ordinary differential equation (ODE) with more state variables to create an equivalent system involving rational functions (Papachristodoulou & Prajna, 2005). Work by Chesi addresses the use of LMI relaxations for the stability analysis of a class of genetic regulatory networks involving ODEs with rational functions on the right-hand sides (Chesi, 2010).

Conversely, polynomial systems often require non-polynomial Lyapunov functions. Ahmadi *et al.* present an example of a polynomial system that is globally stable but does not admit a polynomial Lyapunov function (Ahmadi *et al.*, 2011). Some previous research, including Papachristodoulou & Prajna (2005), has focused on the generation of non-polynomial Lyapunov functions. Recent work by Goubault *et al.* presents techniques for finding rational, trigonometric and exponential Lyapunov functions for polynomial systems through ideas from formal integration (Goubault *et al.*, 2014). Their approach also reduces to polynomial optimization problems, providing a future avenue for the application of the linear relaxations developed here.

Recently, Ahmadi *et al.* have proposed a different set of linear relaxations for polynomial optimization problems called the DSOS approach. This approach further relaxes the positive-semi-definiteness conditions in the SDP formulation using the condition of diagonal dominance, which yields LP relaxations (Ahmadi & Majumdar, 2014). This idea has also been extended to synthesize polynomial Lyapunov functions (Majumdar *et al.*, 2014). A detailed comparison of the ideas of Ahmadi *et al.*'s with those in this paper will be carried out as part of our future work.

However, the use of LP relaxation has not received as much attention. Johansen presented an approach based on linear and quadratic programming (Johansen, 2000). This approach needs a so-called *linear parametrization form* to reduce the stability conditions to an infinite number of linear inequalities, which are reduced to a finite number by discretizing the state space. As a consequence, the number of linear inequalities characterizing the Lyapunov functions grows exponentially with both the dimension

of the state space and the required accuracy. Another approach using LP was presented by Hafstein (2002, 2014). This approach searches for a piecewise affine Lyapunov function, and requires a triangulation of the state space. Our approach derives polynomial (as opposed to affine) Lyapunov function but also benefits from a subdivision of the state space to increase accuracy. The use of Bernstein polynomial properties to formulate relaxations is a distinguishing feature of our approach. The recent work of Kamyar and Peet, which remains under submission at the time of writing, also examines linear relaxations for polynomial optimization problems using Handelman representations, Bernstein polynomial representations (which are closely related) and a linear relaxation based on the well-known Polya's theorem for characterizing positive polynomials on a simplex (Kamyar & Peet, 2014). As in this paper, they have used their approach to search for Lyapunov functions by decomposing the state space. A key difference between the two papers lies in our use of reformulation linearization that considers non-trivial linear relationships between Bernstein polynomials. As shown through examples in this paper, these relationships strictly increase the set of polynomials that can be proved non-negative through our linear relaxations. It must be mentioned that Kamyar *et al.* consider more applications including searching for piecewise polynomial Lyapunov functions and the robust H_∞ control of systems. Our future work will consider the application of the LP relaxations to those considered in Kamyar *et al.*, facilitating an experimental comparison. Ratschan and She use interval arithmetic relaxations with branch-and-bound to discover Lyapunov-like functions to prove a notion of region stability of polynomial systems (Ratschan & She, 2010). This is extended in our previous work to find LP relaxations using the notion of Handelman representations (Sankaranarayanan *et al.*, 2013). In practice, the interval arithmetic approach is known to be quite coarse for proving polynomial positivity, especially for intervals that contain 0. Therefore, Ratschan and She restrict themselves to region stability by excluding a small interval containing the equilibrium from their region of interest. Furthermore, the coarseness of interval relaxation is remedied by resorting to branch-and-bound over the domain. A detailed comparison between interval and Handelman approach is provided in our previous work (Sankaranarayanan *et al.*, 2013), wherein we conclude that both approaches have complementary strengths. A combined approach is thus formulated.

In this paper, we start from such a combined approach and generalize it further through Bernstein polynomials. We use non-trivial properties of Bernstein polynomials that cannot be proved through interval analysis or Handelman representations, to further improve the quality of these relaxations. Section 4 provides detailed comparisons between the various approaches presented in this paper with the approaches based on interval arithmetic, Handelman representations and SOS programming relaxations.

2. Preliminaries

In this section, we recall the definition of Lyapunov functions and discuss procedures for synthesizing them. Subsequently, we examine two techniques for proving the positivity of polynomials: so-called *Handelman representation* technique that produces LP relaxations and a *Putinar representation* technique that produces SDP relaxations. We extend these to recall algorithmic schemes for synthesizing Lyapunov functions, wherein we treat constraints that arise from the positivity of polynomials parametrized by unknown coefficients.

DEFINITION 2.1 (Positive semi-definite functions) A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is *positive semi-definite* over a domain $U \subseteq \mathbb{R}^n$ iff

$$(\forall \vec{x} \in U) f(\vec{x}) \geq 0.$$

Furthermore, f is *positive definite* iff f is positive semi-definite, and additionally, (a) $f(\vec{x}) > 0$ for all $\vec{x} \in U \setminus \{0\}$, and (b) $f(0) = 0$.

2.1 Lyapunov functions

We now recall the key concepts of stability and Lyapunov functions. Let \mathcal{S} be a continuous system over a state space $\mathcal{X} \subseteq \mathbb{R}^n$ specified by a system of **ODEs**

$$\frac{d\vec{x}}{dt} = f(\vec{x}), \quad \vec{x} \in \mathcal{X}.$$

We assume that the right-hand side function $f(\vec{x})$ is Lipschitz continuous over \vec{x} . An equilibrium of the system $\vec{x}^* \in \mathcal{X}$ satisfies $f(\vec{x}^*) = 0$.

DEFINITION 2.2 (Lyapunov and Asymptotic Stability) A system is *Lyapunov stable* over an open region U around the equilibrium \vec{x}^* if, for every neighbourhood $N \subseteq U$ of \vec{x}^* , there is a neighbourhood $M \subset N$ such that $(\forall \vec{x}(0) \in M) (\forall t \geq 0) \vec{x}(t) \in N$. A system is *asymptotically stable* if it is Lyapunov stable and all trajectories starting from U approach \vec{x}^* as $t \rightarrow \infty$.

Lyapunov functions are useful in proving that a system is stable in a region around the equilibrium.

Without loss of generality, we assume that $\vec{x}^* = \vec{0}$. The definitions below are based on the terminology used by [Meiss \(2007\)](#).

DEFINITION 2.3 A continuous and differentiable function $V(\vec{x})$ is a *weak Lyapunov function* over a region $U \subseteq \mathcal{X}$ iff the following conditions hold:

1. $V(\vec{x})$ is positive definite over U , i.e. $V(\vec{x}) > 0$ for all $\vec{x} \in U \setminus \{\vec{0}\}$ and $V(\vec{0}) = 0$.
2. $dV/dt = (\nabla V \cdot f) \leq 0$ for all $\vec{x} \in U$.

Additionally, V is a *strong Lyapunov function* if $(-dV/dt)$ is positive definite.

Weak Lyapunov functions are used to prove that a system is Lyapunov stable over a subset of region U , whereas a strong Lyapunov function proves asymptotic stability. The approaches presented in this paper can be used to search for weak as well as strong Lyapunov functions.

Stability is an important property of control systems. Techniques for discovering Lyapunov functions to certify the stability of a closed-loop model are quite useful in control systems design.

2.2 Proving polynomial positivity

At the heart of Lyapunov function synthesis, we face the challenge of establishing that a given function $V(\vec{x})$ is positive (negative) definite over U . The problem of deciding whether a given polynomial $V(\vec{x})$ is positive definite is NP hard ([Garey & Johnson, 1979](#)). A precise solution requires a decision procedure over the theory of reals ([Tarski, 1951](#); [Collins, 1975](#)). To wit, we check the validity of the formula: $(\forall \vec{x} \in U) V(\vec{x}) \geq 0$ using tools such as QEPcad ([Collins & Hong, 1991](#)) and REDLOG ([Dolzmann & Sturm, 1997](#)). This process is *exact*, but intractable for all but the smallest of systems and low-degree polynomials for V . Therefore, we seek stricter versions of positive semi-definiteness that yield a more tractable system of constraints.

We examine relaxations to the problem of establishing that a given polynomial is positive semi-definite over a region $K \subseteq \mathbb{R}^n$. In the literature, we can distinguish two kinds of techniques for establishing that a given polynomial is positive semi-definite ([Powers & Reznick, 2000](#)). Here, we call them *linear representations* and *SOS representations*.

2.2.1 Linear representations The first approach writes the given polynomial p as a conic combination of products of the constraints defining K . This idea was first examined by Bernstein for proving the positivity of univariate polynomials over the unit interval $[0, 1]$ (Bernstein, 1915). Furthermore, Hausdorff (1921) extended it to the interval $[-1, 1]$.

THEOREM 2.1 (Bernstein and Hausdorff) A polynomial $p(x)$ is strictly positive over $[-1, 1]$ iff there exist a degree $d > 0$ and non-negative constants $\lambda_0, \dots, \lambda_d \geq 0$, such that

$$p(x) \equiv \sum_{i=0}^d \lambda_i (1-x)^i (1+x)^{d-i}, \quad (2.1)$$

This approach is generalized to multivariate polynomials over $\vec{x} : (x_1, \dots, x_n)$ and general semi-algebraic sets $K \subseteq \mathbb{R}^n$ rather than the unit interval. Let K be defined as a semi-algebraic set:

$$K : (p_1(\vec{x}) \geq 0 \wedge \dots \wedge p_m(\vec{x}) \geq 0)$$

for multivariate polynomials p_1, \dots, p_m . A power product over the set of polynomials $P : \{p_1, \dots, p_m\}$ is a polynomial of the form $f : p_1^{n_1} p_2^{n_2} \dots p_m^{n_m}$. The degree of the power product is given by (n_1, \dots, n_m) . We say that $(n_1, \dots, n_m) \leq D$ iff $n_j \leq D$ for each $j \in [1, m]$. Let $\text{PP}(P, D)$ represent all power products from the set P bounded by degree D .

THEOREM 2.2 (Conic combination of power products) If a polynomial p can be written as a conic combination of power products of $P : \{p_1, \dots, p_m\}$, i.e.

$$p(\vec{x}) \equiv \sum_{f \in \text{PP}(P, D)} \lambda_f f, \text{ s.t. } (\forall f \in \text{PP}(P, D)) \lambda_f \geq 0, \quad (2.2)$$

then the polynomial p is non-negative over K :

$$(\forall \vec{x} \in \mathbb{R}^n) \vec{x} \in K \Rightarrow p(\vec{x}) \geq 0.$$

The proof is quite simple. The conic combination of power products in $\text{PP}(P, D)$ as shown in Equation (2.2), is said to be a *Handelman representation* for a polynomial p (Datta, 2002). However, the converse of Theorem 2.2 does not hold, in general. Therefore, polynomials that are positive semi-definite over K need not necessarily have a Handelman representation.

EXAMPLE 2.1 Consider the first orthant in \mathbb{R}^2 given by $K_1 : (x_1 \geq 0 \wedge x_2 \geq 0)$ and the polynomial $p : x_1^2 - 2x_1x_2 + x_2^2$. It is easily seen that p cannot be written as a conic combination of power products over x_1, x_2 , no matter what the degree limit D is chosen to be.

An important question is when the converse of Theorem 2.2 holds. One important case for a compact polyhedron K defined as $K : \bigwedge_{j=1}^m \underbrace{(\vec{a}_j \vec{x} - \vec{b}_j)}_{f_j} \geq 0$ is given by Handelman (1988). Let P denote the set

$\{f_1, \dots, f_m\}$, and $\text{PP}(P, D)$ denote the power products of degree up to D , as before.

THEOREM 2.3 (Handelman) If p is strictly positive over a compact polyhedron K , then there exists a degree bound $D > 0$ such that

$$p \equiv \sum_{f \in \text{PP}(P, D)} \lambda_f f \quad \text{for } \lambda_f \geq 0. \quad (2.3)$$

EXAMPLE 2.2 Consider a polynomial $p(x_1, x_2) = -2x_1^3 + 6x_1^2x_2 + 7x_1^2 - 6x_1x_2^2 - 14x_1x_2 + 2x_2^3 + 7x_2^2 - 9$ over the set $K : \underbrace{(x_1 - x_2 - 3)}_{f_1} \geq 0 \wedge \underbrace{(x_2 - x_1 - 1)}_{f_2} \geq 0$. We can establish the positivity of p over K through its Handelman representation:

$$p \equiv 2f_1^2f_2 + 3f_1f_2.$$

The problem of checking if a polynomial p is positive semi-definite over a set $K : \bigwedge_{j=1}^m p_j(\vec{x}) \geq 0$ is therefore tackled as follows:

1. Choose a degree limit D and construct all terms in $\text{PP}(P, D)$, where $P = \{p_1, \dots, p_m\}$ are the polynomials defining K .
2. Express $p \equiv \sum_{f \in \text{PP}(P, D)} \lambda_f f$ for *unknown multipliers* $\lambda_f \geq 0$.
3. Equate coefficients on both sides (the given polynomial and the Handelman representation) to obtain a set of linear inequality constraints involving λ_f .
4. Use a LP solver to solve these constraints. If feasible, the result yields a proof that p is positive semi-definite over K .

We note that the procedure fails if p is not positive definite over K , or p does not have a Handelman representation over K . Nevertheless, it provides a useful LP relaxation for polynomial positivity.

2.2.2 *SOS representations* Another important approach to proving positivity is through the well-known SOS decomposition.

DEFINITION 2.4 A polynomial $p(\vec{x})$ is a SOS iff there exist polynomials p_1, \dots, p_k over \vec{x} such that p can be written as

$$p \equiv p_1^2 + \dots + p_k^2.$$

It is easy to show that any SOS polynomial is positive semi-definite over \mathbb{R}^n . On the other hand, not every positive semi-definite polynomial is SOS (the so-called Motzkin polynomial provides a counter-example) (Motzkin, 1967).

Schmüdgen representation: Whereas SOS polynomials are positive semi-definite over \mathbb{R}^n , we often seek if p is positive semi-definite over a semi-algebraic set $K : (p_1 \geq 0 \wedge \dots \wedge p_m \geq 0)$.

We define the *pre-order* generated by a set $P = \{p_1, \dots, p_m\}$ of polynomials as the set

$$R(P) = \{p_1^{e_1} p_2^{e_2} \dots p_m^{e_m} \mid (e_1, \dots, e_m) \in \{0, 1\}^m\}.$$

It is easy to see that if, for some given \vec{x} , $p_i(\vec{x}) \geq 0$ for all $i \in [1, m]$, then, for each $r \in R(P)$, we have $r(\vec{x}) \geq 0$. In fact, the following result follows easily.

THEOREM 2.4 If a polynomial p can be expressed as an *SOS polynomial combination* of elements in $R(P)$,

$$p \equiv \sum_{r \in R(P)} q_r r \quad \text{for SOS polynomials } q_r, \quad (2.4)$$

then p is positive semi-definite over K .

Decomposing a polynomial p according to Equation (2.4) will be called the *Schmüdgen representation* of p . The terminology is inspired by the following result due to Schmüdgen (1991).

THEOREM 2.5 (Schmüdgen Positivstellensatz) If K is compact, then every polynomial $p(\vec{x})$ that is strictly positive over K has a Schmüdgen representation of the form given in Equation (2.4).

While Schmüdgen representations are powerful, and in fact, subsume the Handelman representation approach, or even the Bernstein polynomial relaxations to be presented in Section 3, the computational cost of using them is prohibitive. Using the form Equation (2.4) requires finding 2^m SOS polynomials. In our applications, K typically represents the unit rectangle $[-1, 1]^n$, which makes the size of a Schmüdgen representation exponential in the size of the variables. As a result, we will not consider this representation any further in this paper.

Putinar representation: The Putinar representation approach provides a less expensive alternative. Once again, let $K : (p_1 \geq 0 \wedge \dots \wedge p_m \geq 0)$ be a set of interest.

THEOREM 2.6 If a polynomial p can be expressed as

$$p \equiv q_0 + q_1 p_1 + \dots + q_m p_m \quad (2.5)$$

for SOS polynomials q_0, \dots, q_m , then p is positive semi-definite over K .

Decomposing a polynomial p according to Equation (2.5) is said to provide a *Putinar representation* for p . The converse of Theorem 2.6 was proved by Putinar (1993).

THEOREM 2.7 (Putinar) Let $K : (p_1 \geq 0 \wedge \dots \wedge p_m \geq 0)$ be a compact set, suppose that there exists a polynomial p_0 of the form $p_0 = r_0 + \sum_{j=1}^m r_j p_j$ where r_0, \dots, r_m are all SOS and let the set $\hat{K} : \{\vec{x} \mid p_0(\vec{x}) \geq 0\}$ be also compact.

It follows that every polynomial $p(\vec{x})$ that is strictly positive on K has a Putinar representation: $p \equiv q_0 + \sum_{j=1}^m q_j p_j$ for SOS polynomials q_0, \dots, q_m .

A Putinar representation of p for a set $P = \{p_1, \dots, p_m\}$ involves expressing $p \equiv q_0 + \sum_{j=1}^m q_j p_j$ for an SOS polynomial q_j . Searching whether a polynomial p is positive semi-definite over $K : \bigwedge_{p_j \in P} p_j \geq 0$ involves searching for a Putinar representation.

$$\text{find } q_0, \dots, q_m \text{ s.t. } p \equiv q_0 + \sum_{j=1}^m q_j p_j, \quad q_0, \dots, q_m \text{ are SOS.}$$

The key steps involve parametrizing q_0, \dots, q_m in terms of polynomials of bounded degree D over a set of unknown coefficients \vec{c} , and then solving the resulting problem through a relaxation to SDP, originally proposed by Shor and further developed by Parillo (Shor, 1987; Parillo, 2003). The resulting optimization problem is called a SOS programming problem.

2.3 Synthesis of Lyapunov functions

We now summarize the standard approach to synthesizing Lyapunov functions using Handelman or Putinar representations. The Handelman approach reduces the synthesis to solving a set of linear programmes, and was presented in our previous work (Sankaranarayanan et al., 2013). The Putinar

representation approach uses SOS programming, and was presented by Papachristodoulou & Prajna (2002). This approach is implemented in a package SOSTOOLS that provides a user-friendly interface for posing SOS programming problems and solving them by relaxing to a semi-definite programme (Papachristodoulou *et al.*, 2013).

Let $U \subseteq \mathbb{R}^n$ be a compact set and \mathcal{S} be a system defined by the ODE $d\vec{x}/dt = f(\vec{x})$. We assume that the origin is the equilibrium of \mathcal{S} , i.e. $f(\vec{0}) = 0$, $\vec{0} \in \text{interior}(U)$, and wish to prove the local asymptotic (or Lyapunov) stability of \mathcal{S} for a subset of the region U .

Therefore, we seek a Lyapunov function of the form $V(\vec{x}, \vec{c})$, wherein V is a polynomial form over \vec{x} whose coefficients are polynomials over \vec{c} . Let V' denote the Lie derivative of V , i.e. $V'(\vec{x}, \vec{c}) = (\nabla_{\vec{x}} V) \cdot f$. We define the set C as follows:

$$C = \{\vec{c} \mid V(\vec{x}, \vec{c}) \text{ is positive definite for } \vec{x} \in U\}. \quad (2.6)$$

Also, let \hat{C} be defined as follows:

$$\hat{C} = \{\vec{c} \mid V'(\vec{x}, \vec{c}) \text{ is negative definite for } \vec{x} \in U\}. \quad (2.7)$$

We replace negative definiteness for negative semi-definiteness if Lyapunov stability, rather than asymptotic stability is of interest. The overall procedure for synthesizing Lyapunov functions proceeds as follows:

1. Fix a template form $V(\vec{x}, \vec{c})$ with parameters \vec{c} .
2. Compute constraints $\psi[\vec{c}]$ whose solutions yield the set C in Equation (2.6).
3. Compute constraints $\hat{\psi}[\vec{c}]$ whose solutions yield the set \hat{C} from Equation (2.7).
4. Compute a value $\vec{c} \in C \cap \hat{C}$ by solving the constraints $\psi \wedge \hat{\psi}$. The resulting function $V_c(\vec{x})$ is a Lyapunov function.

The main problem, therefore, is to characterize a set C for the unknown parameters \vec{c} , so $V_c(\vec{x})$ is positive definite over U for all $\vec{c} \in C$. Thus, the process of searching for Lyapunov functions of a given form devolves into the problem of finding a system of constraints for the sets C, \hat{C} .

REMARK 2.1 It must be remarked that finding a (strong) Lyapunov function $V(\vec{x})$ inside a region U , as presented thus far, does not necessarily prove that the system is asymptotically stable for every initial state $\vec{x} \in U$. For instance, trajectories starting from $\vec{x} \in U$ may exit the set U .

However, let γ represent the largest value such that, for all $\vec{x} \in U$, $V(\vec{x}) \leq \gamma$:

$$\gamma : \max_{\vec{x} \in U} V(\vec{x}).$$

It can be shown that the system is asymptotically stable inside the set $V_\gamma : \{\vec{x} \mid V(\vec{x}) \leq \gamma\}$.

Handelman representations and Putinar representations provide us two approaches to encoding the positive definiteness of V and negative definiteness of V' to characterize the sets C, \hat{C} .

Handelman representations: We now briefly summarize our previous work that uses Handelman representations for Lyapunov function synthesis (Sankaranarayanan *et al.*, 2013).

Let us assume that the set U is written as a semi-algebraic set:

$$U : \bigwedge_{j=1}^m p_j(\vec{x}) \geq 0.$$

Let $P = \{p_1, \dots, p_m\}$ represent these constraints. Given a degree limit D , we construct the set $\text{pp}(P, D)$ of all power products of the form $\prod_{j=1}^m p_j^{n_j}$ wherein $0 \leq n_j \leq D$.

We encode positive semi-definiteness of a form $V(\vec{x}, \vec{c})$ by writing it as

$$V(\vec{x}, \vec{c}) \equiv \sum_{f \in \text{pp}(P, D)} \lambda_f f \text{ wherein } \lambda_f \geq 0. \quad (2.8)$$

Positive definiteness is encoded using a standard trick presented by [Papachristodoulou & Prajna \(2002\)](#). Briefly, the idea is to write $V = \hat{V} + \sum_{j=1}^n \epsilon x_j^{2p}$ for $\hat{V}(\vec{x}, \vec{c})$, an unknown positive semi-definite function and a fixed positive-definite contribution given by setting ϵ, p . This idea is used in all our examples wherein positive definiteness is to be encoded rather than positive semi-definiteness.

We eliminate \vec{x} by equating the coefficients of monomials on both sides of Equation (2.8), and obtain a set of linear constraints $\psi[\vec{c}, \vec{\lambda}]$ involving \vec{c} and $\vec{\lambda}$. The set C is characterized as a polyhedron obtained by the projection

$$C : \{\vec{c} \mid \exists \vec{\lambda} \geq 0 \psi(\vec{c}, \vec{\lambda})\}.$$

In practice, we do not project $\vec{\lambda}$, but instead retain ψ as a set of constraints involving both $\vec{c}, \vec{\lambda}$. Similarly, we consider the Lie derivative $V'(\vec{c}, \vec{x})$ and obtain constraints $\psi(\vec{c}, \vec{\mu})$ for a different set of multipliers $\vec{\mu}$.

The overall problem reduces to finding a value of \vec{c} that satisfies the constraints

$$\psi(\vec{c}, \vec{\lambda}) \wedge \hat{\psi}(\vec{c}, \vec{\mu})$$

for some $\vec{\lambda}, \vec{\mu} \geq 0$. This is achieved by solving a set of linear programmes.

EXAMPLE 2.3 Consider a parametric polynomial $p(\vec{c}, \vec{x}) : c_1 x_1^2 + c_2 x_2^2 + c_3 x_1 x_2 + c_4 x_1 + c_5 x_2 + c_6$ and the set K defined by the constraints: $x_2 - x_1 \geq -1 \wedge x_1 + x_2 \geq 2$. We will use a Handelman representation to characterize a set of parameters C s.t. $\vec{x} \in K \models p(\vec{c}, \vec{x}) \geq 0$. Using degree-2 Handelman representation, we obtain the following larger set of constraints:

$$\begin{aligned} x_2 - x_1 + 1 &\geq 0 \wedge x_1 + x_2 - 2 \geq 0 \wedge \\ x_2^2 + x_1^2 - 2x_1 x_2 + 2x_2 - 2x_1 + 1 &\geq 0 \wedge \leftarrow (x_2 - x_1 + 1)^2 \geq 0 \\ x_1^2 + x_2^2 + 2x_1 x_2 - 4x_1 - 4x_2 + 4 &\geq 0 \wedge \leftarrow (x_1 + x_2 - 2)^2 \geq 0 \\ -x_1^2 + x_2^2 + 3x_1 - x_2 - 2 &\geq 0 \leftarrow (x_2 - x_1 + 1)(x_1 + x_2 - 2) \geq 0. \end{aligned}$$

We express p as a linear combination of these constraints yielding the following equivalence:

$$c_1x_1^2 + c_2x_2^2 + c_3x_1x_2 + c_4x_1 + c_5x_2 + c_6 \equiv \begin{pmatrix} \lambda + \lambda_0(x_2 - x_1 + 1) + \lambda_1(x_1 + x_2 - 2) \\ +\lambda_2(x_2^2 + x_1^2 - 2x_1x_2 + 2x_2 - 2x_1 + 1) \\ +\lambda_3(x_1^2 + x_2^2 + 2x_1x_2 - 4x_1 - 4x_2 + 4) \\ +\lambda_4(-x_1^2 + x_2^2 + 3x_1 - x_2 - 2) \end{pmatrix},$$

where $\lambda, \lambda_0, \dots, \lambda_4 \geq 0$. Matching coefficients of monomials on both sides, we obtain linear inequality constraints involving variables c_1, \dots, c_6 and $\lambda_1, \dots, \lambda_4$:

$$\begin{aligned} c_1 &= \lambda_2 + \lambda_3 - \lambda_4 && \leftarrow \text{Matching } x_1^2 \\ c_2 &= \lambda_2 + \lambda_3 + \lambda_4 && \leftarrow \text{Matching } x_2^2 \\ c_3 &= -2\lambda_2 + 2\lambda_3 && \leftarrow \text{Matching } x_1x_2 \\ &\vdots && \leftarrow \text{Matching } x_1, x_2 \\ c_6 &\geq \lambda_0 - 2\lambda_1 + \lambda_2 + 4\lambda_3 - 2\lambda_4 && \leftarrow \text{Matching constant term} \\ &\lambda_0, \dots, \lambda_4 \geq 0. \end{aligned}$$

Any non-zero solution yields a set of values for \vec{c} and the corresponding Handelman representation for degree 2.

Putinar representations : Papachristadoulou and Prajna present the Putinar representations approach to synthesizing Lyapunov functions. Once again, we consider a semi-algebraic set U , as before. We fix a form $V(\vec{c}, \vec{x})$ for the Lyapunov and write

$$V \equiv q_0 + \sum_{j=1}^m q_j p_j$$

for SOS polynomials q_0, \dots, q_m . The approach fixes the degree of each q_j and uses SOS programming to encode the positivity. The result is a system of constraints over the parameters \vec{c} for V and the unknowns $\vec{\lambda}$ that characterize the SOS multipliers q_j . The same approach encodes the negative semi-definiteness of V' over U . The combined result is a semi-definite programme that jointly solves for the positive definiteness of V and the negative definiteness of V' . A solution is recovered by solving the feasibility problem for an SDP to yield the values for \vec{c} that yield a Lyapunov certificate for stability.

3. LP relaxations based on Bernstein polynomials

In this section, we recall the use of Bernstein polynomials for establishing bounds on polynomials in intervals. Given a multivariate polynomial p , proving that p is positive semi-definite in K is equivalent to showing that the optimal value of the following optimization problem is non-negative:

$$\begin{aligned} &\text{minimize } p(\vec{x}) \\ &\text{s.t } \vec{x} \in K. \end{aligned} \tag{3.1}$$

Whereas (3.1) is hard to solve, we will construct a LP relaxation, whose optimal value is guaranteed to be a lower bound on p^* . If the bound is tight enough, then we can prove the positivity of polynomial p on K .

In general, the Handelman representation approach presented in Section 2.2 can be used to construct a LP relaxation (Sankaranarayanan *et al.*, 2013). In this section, we will use Bernstein polynomials for the special unit box ($K = [0, 1]^n$). Bernstein polynomials extend the Handelman approach, and will be shown to be strictly more powerful when K is the unit box. In our application examples, K is often a hyper-rectangle but not necessarily the unit box. We use an affine transformation to transform p and K back to the unit box, so that the Bernstein polynomial approach can be used.

3.1 Overview of Bernstein polynomials

Bernstein polynomials were first proposed by Bernstein as a constructive proof of the Weierstrass approximation theorem (Bernstein, 1912). Bernstein polynomials are useful in many engineering design applications for approximating geometric shapes (Farouki, 2012). They form a basis for approximating polynomials over a compact interval, and have nice properties that will be exploited to relax the optimization Equation (3.1) to a linear programme. Here, we should mention that a relaxation using Bernstein polynomials was provided in the context of reachability analysis for polynomial dynamical systems (Dang and Salinas, 2009) and improved in Sassi *et al.* (2012). The novelty in this work is not only the adaptation of these relaxations in the context of polynomial Lyapunov function synthesis but also a new tighter relaxation will be introduced by exploiting the induction relation between Bernstein polynomials. More details on Bernstein polynomials are available elsewhere (Muñoz & Narkawicz, 2013).

We first examine Bernstein polynomials and their properties for the univariate case and then extend them to multivariate polynomials (see Bernstein (1952, 1954)).

DEFINITION 3.1 (Univariate Bernstein polynomials) Given an index $i \in \{0, \dots, m\}$, the i th univariate Bernstein polynomial of degree m over $[0, 1]$ is given by the following expression:

$$\beta_{i,m}(x) = \binom{m}{i} x^i (1-x)^{m-i}, \quad i \in \{0, \dots, m\}. \quad (3.2)$$

In the Bernstein polynomial basis, a univariate polynomial $p(x) : \sum_{j=0}^m p_j x^j$ of degree m , can be written as

$$p(x) = \sum_{i=0}^m b_{i,m} \beta_{i,m}(x),$$

where, for all $i = 0, \dots, m$,

$$b_{i,m} = \sum_{j=0}^i \frac{\binom{i}{j}}{\binom{m}{j}} p_j. \quad (3.3)$$

The coefficients $b_{i,m}$ are called the *Bernstein coefficients* of the polynomial p .

Bernstein polynomials have many interesting properties. We summarize the most relevant ones for our applications, below.

LEMMA 3.1 For all $x \in [0, 1]$, and for all $m \in \mathbb{N}$, the Bernstein polynomials $\{\beta_{0,m}, \dots, \beta_{m,m}\}$ have the following properties:

1. Unit partition: $\sum_{i=0}^m \beta_{i,m}(x) = 1$.
2. Bounds: $0 \leq \beta_{i,m}(x) \leq \beta_{i,m}(i/m)$, for all $i = 0, \dots, m$.
3. Induction property: $\beta_{i,m-1}(x) = ((m-i)/m)\beta_{i,m}(x) + ((i+1)/m)\beta_{i+1,m}(x)$ for all $i = 0, \dots, m-1$.

Using the unit partition and positivity of Bernstein polynomials, the following bounds' result holds:

COROLLARY 3.1 On the interval $[0, 1]$, a polynomial p with Bernstein coefficients $b_{0,m}, \dots, b_{m,m}$, the following inequality holds (Garloff, 1993):

$$\min_{i=0,\dots,m} b_{i,m} \leq p(x) \leq \max_{i=0,\dots,m} b_{i,m}. \quad (3.4)$$

We generalize the previous notions to the case of multivariate polynomials, i.e. $p(\vec{x}) = p(x_1, \dots, x_n)$ where $\vec{x} = (x_1, \dots, x_n) \in U = [0, 1]^n$. For multi-indices, $I = (i_1, \dots, i_n) \in \mathbb{N}^n$, $J = (j_1, \dots, j_n) \in \mathbb{N}^n$, we fix the following notation:

- $I \leq J \iff i_l \leq j_l$, for all $l = 1, \dots, n$.
- $I/J = (i_1/j_1, \dots, i_n/j_n)$ and $\binom{I}{J} = \binom{i_1}{j_1} \cdots \binom{i_n}{j_n}$.
- $I_{r,k} = (i_1, \dots, i_{r-1}, i_r + k, i_{r+1}, \dots, i_n)$, where $r \in \{1, \dots, n\}$ and $k \in \mathbb{Z}$.

Let us fix our maximal degree $\delta = (\delta_1, \dots, \delta_n) \in \mathbb{N}^n$ for a multivariate polynomial p (δ_l is the maximal degree of x_l for all $l = 1, \dots, n$). Then the multivariate polynomial p will have the following form:

$$p(\vec{x}) = \sum_{I \leq \delta} p_I \vec{x}^I, \quad \text{where } p_I \in \mathbb{R}, \forall I \leq \delta.$$

Multivariate Bernstein polynomials are given by products of the univariate polynomials:

$$B_{I,\delta}(\vec{x}) = \beta_{i_1,\delta_1}(x_1) \cdots \beta_{i_n,\delta_n}(x_n), \quad \text{where } \beta_{i_j,\delta_j}(x_j) = \binom{\delta_j}{i_j} x_j^{i_j} (1-x_j)^{\delta_j-i_j}. \quad (3.5)$$

Owing to the previous notations, these polynomials can also be written as follows:

$$B_{I,\delta}(\vec{x}) = \binom{\delta}{I} \vec{x}^I (1_n - \vec{x})^{\delta-I}. \quad (3.6)$$

Now, we can have the general expression of a multivariate polynomial in the Bernstein basis:

$$p(\vec{x}) = \sum_{I \leq \delta} b_{I,\delta} B_{I,\delta}(\vec{x}),$$

where Bernstein coefficients $(b_{I,\delta})_{I \leq \delta}$ are given as follows:

$$b_{I,\delta} = \sum_{J \leq I} \frac{\binom{i_1}{j_1} \cdots \binom{i_n}{j_n}}{\binom{\delta_1}{j_1} \cdots \binom{\delta_n}{j_n}} p_J = \sum_{J \leq I} \frac{\binom{I}{J}}{\binom{\delta}{J}} p_J. \quad (3.7)$$

Therefore, the generalization of Lemma 3.1 will lead to the following properties:

LEMMA 3.2 For all $\vec{x} = (x_1, \dots, x_n) \in U$, we have the following properties:

1. Unit partition: $\sum_{I \leq \delta} B_{I,\delta}(\vec{x}) = 1$.
2. Bounded polynomials: $0 \leq B_{I,\delta}(\vec{x}) \leq B_{I,\delta}(I/\delta)$, for all $I \leq \delta$.
3. Induction relation: $B_{I,\delta_{r-1}} = ((\delta_r - i_r)/\delta_r)B_{I,\delta} + ((i_r + 1)/\delta_r)B_{I_{r,1},\delta}$, for all $r = 1, \dots, n$, and all $I \leq \delta_{r-1}$.

Finally, in the case of general rectangle $K = [x_1, \bar{x}_1] \times \dots \times [x_n, \bar{x}_n]$ it suffices to make a change of variables $x_j = \underline{x}_j + z_j(\bar{x}_j - \underline{x}_j)$ for all $j = 1, \dots, n$ to obtain new variables $\vec{z} = (z_1, \dots, z_n) \in U$.

3.2 Bernstein relaxations

We assume that K is a bounded rectangle. Without loss of generality, we can assume that $K = [0, 1]^n$ since we can be reduced to the unit box by a simple affine transformation. Using the previous properties, we are going to construct three LP relaxations to problem (3.1).

Reformulation linearization technique (RLT): We first recall a simple approach to relaxing polynomial optimization problems to linear programmes (Sherali & Tuncbilek, 1991, 1997). We then carry out these relaxations for Bernstein polynomials, and show how the properties in Lemma 3.2 can be incorporated into the relaxation schemes. Recall, once again, the optimization problem Equation (3.1):

$$\begin{array}{ll} \text{minimize} & p(\vec{x}) \\ \text{s.t.} & \vec{x} \in K. \end{array}$$

For simplicity, let us assume $K : [0, 1]^n$ is the unit rectangle; K is represented by the constraints $K : \bigwedge_{j=1}^n (x_j \geq 0 \wedge (1 - x_j) \geq 0)$. The standard RLT approach consists of writing $p(\vec{x}) = \sum_I p_I \vec{x}^I$ as a linear form $p(\vec{x}) : \sum_I p_I y_I$ for *fresh variables* y_I that are place holders for the monomials \vec{x}^I . Next, we write down as many *facts* about \vec{x}^I over K as possible. The basic approach now considers all possible power products of the form $\pi_{J,\delta} : \vec{x}^J (1 - \vec{x})^{\delta-J}$ for all $J \leq \delta$, where δ is a given degree bound. Clearly, if $\vec{x} \in K$, then $\pi_{J,\delta}(\vec{x}) \geq 0$. Expanding $\pi_{J,\delta}$ in the monomial basis as $\pi_{J,\delta} : \sum_{I \leq \delta} a_{I,J} \vec{x}^I$, we write the linear inequality constraint:

$$\sum_{I \leq J} a_{I,J} y_I \geq 0.$$

The overall LP relaxation is obtained as follows:

$$\begin{array}{ll} \text{minimize} & \sum_I p_I y_I \\ \text{s.t.} & \sum_{I \leq J} a_{I,J} y_I \geq 0, \text{ for each } J \leq \delta. \end{array} \quad (3.8)$$

Additionally, it is possible to augment this LP by adding inequalities of the form $\ell_I \leq y_I \leq u_I$ through the interval evaluation of \vec{x}^I over the set K .

PROPOSITION 3.1 For any polynomial p , the optimal value computed by the LP Equation (3.8) is a lower bound on that of the polynomial programme Equation (3.1).

EXAMPLE 3.1 We suppose to compute a lower bound for the following POP:

$$\begin{aligned} & \text{minimize} && x_1^2 + x_2^2 \\ & \text{s.t.} && (x_1, x_2) \in [0, 1]^2. \end{aligned} \quad (3.9)$$

Using the RLT technique for a degree $\delta = 2$ we denote by y_{ij} the fresh variables replacing the non-linear terms $x^{(i,j)} = x_1^i x_2^j$ for all $(i, j) \in \mathbb{N}^2$ such that $i + j \leq 2$. Using these notations the objective function of the relaxation will be $y_{20} + y_{02}$. The constraints are given by the linearized form of the possible products (degree less than δ) of the following constraints : $x_1 \geq 0$, $x_2 \geq 0$, $1 - x_1 \geq 0$ and $1 - x_2 \leq 0$. For example the constraint $y_{11} \geq 0$ is obtained by multiplying $x_1 \geq 0$ and $x_2 \geq 0$.

RLT using Bernstein polynomials: The success of the RLT approach depends heavily on writing ‘facts’ involving the variables y_I that substitute for \vec{x}^I . We now present the core idea of using Bernstein polynomial expansions and the richer bounds that are known for these polynomials from Lemma 3.2 to improve upon the basic RLT approach.

First, we write $p(\vec{x})$ as a sum of Bernstein polynomials of degree δ :

$$p(\vec{x}) : \sum_{I \leq \delta} b_{I,\delta} B_{I,\delta},$$

wherein $b_{I,\delta}$ are calculated using the formula in Equation (3.7). Let us introduce a fresh variable $z_{I,\delta}$ as a place holder for $B_{I,\delta}(\vec{x})$. Lemma 3.2 now gives us a set of linear inequalities that hold between these variables $z_{I,\delta}$. Therefore, we obtain three LP relaxations of increasing precision using Bernstein polynomials. Once again, let $(b_{I,\delta})_{I \leq \delta}$ denote Bernstein coefficients of p with respect to a maximal degree δ . We formulate three LP relaxations, each providing a better approximation for the feasible region of the original problem Equation (3.1).

The first relaxation uses the fact that $B_{I,\delta}(\vec{x}) \geq 0$ for all $\vec{x} \in [0, 1]^n$ and that $\sum_{I \leq \delta} B_{I,\delta} \equiv 1$.

$$\begin{aligned} p_\delta^{(1)} = \text{minimize} & \quad \sum_{I \leq \delta} b_{I,\delta} z_{I,\delta} \\ \text{s.t} & \quad z_{I,\delta} \in \mathbb{R}, \quad I \leq \delta, \\ & \quad z_{I,\delta} \geq 0, \quad I \leq \delta, \\ & \quad \sum_{I \leq \delta} z_{I,\delta} = 1. \end{aligned} \quad (3.10)$$

From corollary 3.1, it is easy to see that $p_\delta^{(1)} = \min_{I \leq \delta} b_{I,\delta}$ (minimal of Bernstein coefficients for p). The optimization is superfluous here, but will be useful subsequently.

Next, we incorporate sharper bounds for $B_{I,\delta}(\vec{x})$ for each I for $\vec{x} \in K$.

$$\begin{aligned} p_\delta^{(2)} = \text{minimize} & \quad \sum_{I \leq \delta} b_{I,\delta} z_{I,\delta} \\ \text{s.t} & \quad z_{I,\delta} \in \mathbb{R}, \quad I \leq \delta, \\ & \quad 0 \leq z_{I,\delta} \leq B_{I,\delta} \left(\frac{I}{\delta} \right), \quad I \leq \delta, \\ & \quad \sum_{I \leq \delta} z_{I,\delta} = 1. \end{aligned} \quad (3.11)$$

Finally, the recurrence relation between the polynomials are expressed as equations in the LP relaxation to constrain the relaxation even further.

$$\begin{aligned}
 p_\delta^{(3)} &= \text{minimize} && \sum_{I \leq \delta} b_{I,\delta} z_{I,\delta} \\
 \text{s.t} &&& z_{I,\delta} \in \mathbb{R}, \quad I \leq \delta, \\
 &&& z_{J,\delta'} \in \mathbb{R}, \quad J \leq \delta', \quad \delta' < \delta, \\
 &&& 0 \leq z_{I,\delta} \leq B_{I,\delta} \left(\frac{I}{\delta} \right), \quad I \leq \delta, \\
 &&& 0 \leq z_{J,\delta'} \leq B_{J,\delta'} \left(\frac{J}{\delta'} \right), \quad J \leq \delta', \quad \delta' < \delta, \\
 &&& \sum_{I \leq \delta} z_{I,\delta} = 1, \\
 &&& \sum_{J \leq \delta'} z_{J,\delta'} = 1, \quad \delta' < \delta, \\
 &&& z_{J,\delta'} = \frac{\delta'_r - j_r}{\delta'_r} z_{J,\delta'_{r,1}} + \frac{j_r + 1}{\delta'_r} z_{J_{r,1},\delta'_{r,1}}, \quad J \leq \delta', \quad \delta' < \delta.
 \end{aligned} \tag{3.12}$$

Relaxation (3.10) is obtained once the unit partition and the positivity of Bernstein polynomials (Lemma 3.2) are injected. In relaxation (3.11), we add lower bounds on polynomials $B_{I,\delta}(y)$ (given by the second property of Lemma 3.2) which allow us to obtain a more precise result (greater) since we are adding more constraints for the previous minimization problem. The third one (3.12) is obtained by adding new variables for Bernstein polynomials of lower degree and exploiting the induction property of Lemma 3.2. It will be the more precise one but also the more costly.

We will show using the properties of Bernstein polynomials that each of these relaxations provides a lower bound on the original polynomial optimization problem.

PROPOSITION 3.2 $p_\delta^{(1)} \leq p_\delta^{(2)} \leq p_\delta^{(3)} \leq p^*$, where p^* is the optimal value of (3.1).

Proof. First, consider any feasible solution y to the problem Equation (3.1):

$$\begin{aligned}
 &\text{minimize} && \sum_{I \leq \delta} b_{I,\delta} B_{I,\delta}(y) \\
 \text{s.t} &&& y \in [0, 1]^n.
 \end{aligned}$$

We note that injecting $z_I = B_{I,\delta}(y)$, the vector of all z_I s form a feasible solution to each of the three relaxation Equation (3.10–3.12). Therefore, $p_\delta^{(j)} \leq p^*$ for $j \in \{1, 2, 3\}$.

Next, considering the formulation Equations (3.12), (3.11) and (3.10), we note that their objective functions are the same. Furthermore, the decision variables in Equations (3.11) and (3.10) are the same; while the decision variables in Equation (3.11) are a strict subset of those in Equation (3.12). Next, each constraint in Equation (3.10) is present in Equation (3.11), and likewise, each constraint in Equation (3.11) is also present in Equation (3.12).

As a consequence, any feasible solution for $p_\delta^{(3)}$ is, in turn, a feasible solution for $p_\delta^{(2)}$ with the extra variables in formulation Equation (3.12) removed. Therefore, since the objectives coincide and we seek to minimize, we have $p_\delta^{(2)} \leq p_\delta^{(3)}$. Similarly, any feasible solution for Equation (3.11) is, in turn, a feasible

solution for Equation (3.10). Here, no projection is needed, since the two LPs consider the same set of variables. Once again, we have $p_\delta^{(1)} \leq p_\delta^{(2)}$. Putting it all together, we have $p_\delta^{(1)} \leq p_\delta^{(2)} \leq p_\delta^{(3)} \leq p^*$. \square

REMARK 3.1 The choice of the appropriate relaxation is a trade-off between complexity and precision. In fact, the third relaxation (3.12) which gives the more precise result, can be very expensive, especially when the number of variables and/or their degrees increase.

Finally, Equation (3.12) can be used for a *fixed level* to alleviate the drastic increase in the number of decision variables; i.e. instead of exploiting all the constraints arising for the degrees $\delta' < \delta$, we may restrict ourselves to δ' such that $\delta'_r = \delta_r - 1$.

4. Comparison between Representations

In this section, we will first start by comparing linear and SOS representations. Next, we compare the new Bernstein relaxations with other existing linear relaxations including Handelman and interval representations.

4.1 Comparison between linear and SOS representations

Comparing the presentations of linear vs. SOS representations, the trade-offs look quite obvious. Whereas linear representations produce linear programmes, which can be solved using exact arithmetic, SOS representations produce SOS programmes, which are solved numerically by relaxation to semi-definite programmes. In fact, numerical issues sometimes arise, and have been noted in our previous work (Sankaranarayanan *et al.*, 2013). On the other hand, it also seems that Handelman representations may be weaker than Putinar representations. Consider the example below.

PROPOSITION 4.1 The polynomial $p(x) : x^2$ does not have a Handelman representation inside the interval $[-1, 1]$.

Proof. Suppose we were able to express x^2 as a (non-trivial) conic combination of power products of the form

$$x^2 \equiv \sum_{j=1}^m \lambda_j (1-x)^{n_j} (x+1)^{m_j}, \quad \lambda_j > 0.$$

We note that, at $x = 0$, the LHS is zero, whereas the RHS is strictly positive. This implies that $\lambda_j = 0$ for $j = 1, \dots, m$. Thus, the RHS is identically zero, yielding a contradiction. \square

On the other hand, the polynomial x^2 is SOS, and thus trivially shown to be positive over $[-1, 1]$ (if not over \mathbb{R}) by a Putinar representation.

However, in such a situation, we can show that Handelman representations can be useful in showing positivity where Putinar representations can fail. Consider the set $K : [0, 1] \times [0, 1]$ and the bivariate polynomial $p(x, y) = xy$.

PROPOSITION 4.2 There do not exist SOS polynomials q_0, q_1, q_2, q_3, q_4 such that

$$xy \equiv q_0 + q_1x + q_2y + q_3(1-x) + q_4(1-y).$$

In other words, the polynomial xy does not have a Putinar representation over the unit box $K : [0, 1] \times [0, 1]$.

Proof. Suppose, for a contradiction, there exist SOS polynomials q_0, q_1, q_2, q_3, q_4 such that

$$xy \equiv q_0 + q_1x + q_2y + q_3(1-x) + q_4(1-y). \quad (4.1)$$

We establish a contradiction by considering the lowest degree terms of the polynomials q_0, \dots, q_4 . We use the notation $\text{COEFF}(q, x^i y^j)$ to refer to the coefficient of the monomial $x^i y^j$ in the polynomial $q(x, y)$.

First, plugging in $x=0, y=0$, we observe that $\text{COEFF}(q_0, 1) = \text{COEFF}(q_3, 1) = \text{COEFF}(q_4, 1) = 0$. In other words, the constant coefficients of q_0, q_3, q_4 are zero.

Since q_0, q_3, q_4 are psd, if they have zero constant terms, then they do not have linear terms involving x or y .

$$\text{COEFF}(q_j, x) = \text{COEFF}(q_j, y) = \text{COEFF}(q_j, 1) = 0, \quad j \in \{0, 3, 4\}.$$

Therefore, the constant terms of q_1, q_2 are zero as well:

$$\text{COEFF}(q_1, 1) = \text{COEFF}(q_2, 1) = 0.$$

Otherwise, the RHS will have non-zero terms involving x, y but the LHS has no such terms. Once again, from the positivity of q_1, q_2 , we have

$$\text{COEFF}(q_j, x) = \text{COEFF}(q_j, y) = 0, \quad j \in \{1, 2\}.$$

Having established that no constant or linear terms can exist for q_0, \dots, q_4 , we turn our attention to the quadratic terms x^2, y^2, xy . Consider the coefficients of x^2 on both sides of Equation (4.1):

$$\text{COEFF}(q_0, x^2) + \text{COEFF}(q_3, x^2) + \text{COEFF}(q_4, x^2) = 0, \quad \text{COEFF}(q_0, y^2) + \text{COEFF}(q_3, y^2) + \text{COEFF}(q_4, y^2) = 0.$$

Since q_i are psd and lack constant/linear terms, we can show that

$$\text{COEFF}(q_j, x^2) \geq 0, \quad \text{COEFF}(q_j, y^2) \geq 0, \quad j \in \{0, \dots, 4\}.$$

Therefore, we conclude that

$$\text{COEFF}(q_j, x^2) = \text{COEFF}(q_j, y^2) = 0, \quad j \in \{0, 3, 4\}.$$

Finally, we compare xy terms on both sides of Equation (4.1) to obtain

$$\text{COEFF}(q_0, xy) + \text{COEFF}(q_3, xy) + \text{COEFF}(q_4, xy) = 1.$$

Therefore, we have $\text{COEFF}(q_j, xy) > 0$ for some $j \in \{0, 3, 4\}$, while $\text{COEFF}(q_j, x^2) = \text{COEFF}(q_j, y^2) = 0$. We now contradict the assumption that q_j is psd. Based on what we have shown thus far, we can write

$$q_j(x, y) = cxy + \text{third or higher-order terms}, \quad \text{where } c > 0.$$

Let us fix $x = \epsilon, y = -\epsilon$ for some $\epsilon > 0$.

$$q_j(\epsilon, -\epsilon) = -c_0\epsilon^2 + o(\epsilon^3).$$

Therefore, we conclude for ϵ small enough, $q_j(\epsilon, -\epsilon) < 0$, thus contradicting the positivity assumption for q_j for some $j \in \{0, 3, 4\}$. \square

Furthermore, the SOS relaxation to SDP relies on numerical interior point solvers to find a feasible point. From the point of view of a *guaranteed method*, such an approach can be problematic. On the other hand, LP solvers can use exact arithmetic in spite of the high cost of doing so, to obtain results that hold up to verification. Examples of numerical issues in SOS programming for Lyapunov function synthesis are noted in our previous work (Sankaranarayanan *et al.*, 2013), and will not be reproduced here. Consequently, much work has focused on the problem of finding rational feasible points for SOS to generate polynomial positivity proofs in exact arithmetic (Harrison, 2007; Platzer *et al.*, 2009; Monniaux & Corbineau, 2011). Recently, a self-validated SDP solver VSDP has been proposed by Härter *et al.* (2012). However, its application to SOS optimization has not been investigated.

4.2 Comparison of Bernstein relaxations with other Linear Representations

We are going to compare the Bernstein relaxation Equations (3.12), (3.11) and (3.10) with other existing linear relaxations including Handelman and interval LP relaxations. More precisely, we will show the benefit of using Bernstein relaxations instead of the LP relaxations given by Ratschan & She (2010) and our previous work (Sankaranarayanan *et al.*, 2013).

Our earlier work (Sankaranarayanan *et al.*, 2013) uses RLT with a combination of Handelman representation augmented by interval arithmetic constraints to prove polynomial positivity, as a primitive for Lyapunov function synthesis. As long as the domain K of interest is a hyper-rectangle, the relaxations provided by Bernstein polynomials will provide results that are guaranteed to be at least as good, if not strictly better. For simplicity, let us fix K as the unit box $[0, 1]^n$ and compare the two relaxations.

A first remark will be that the Handelman representation contains polynomials with degree less than or equal to the fixed degree δ , whereas our Bernstein polynomials are all of degree equal to δ . This does not affect the optimal value of the relaxation, as noted by Sherali & Tuncbilek (1991). Therefore, no gain of precision can be made using the Handelman relaxation owing to the additional polynomials of degree less than δ .

LEMMA 4.1 (Bernstein vs. Handelman representations) Let $K : [0, 1]^n$ represent the unit interval. Any polynomial that can be shown non-negative over K using a Handelman representation of degree δ can also be shown non-negative using the formulation in Equation (3.10) with the same degree.

Proof. We first note that Handelman representation for p seeks to express p as

$$p \equiv \sum_{I \leq \delta} \lambda_I \underbrace{\vec{x}^I (1 - \vec{x})^{\delta-I}}_{B_{I,\delta}}.$$

In fact, over the unit interval, the Handelman representation seeks to write p as a conic combination of Bernstein polynomials. Consider the relaxation to the POP:

$$\min_{\vec{x} \in K} p(\vec{x}).$$

Using a Handelman representation of $p(\vec{x})$, we obtain the following relaxation:

$$\begin{aligned} p_H = \text{minimize} \quad & \sum_{I \leq \delta} b_{I,\delta} z_{I,\delta} \\ \text{s.t} \quad & z_{I,\delta} \in \mathbb{R}, \quad I \leq \delta, \\ & z_{I,\delta} \geq 0, \quad I \leq \delta. \end{aligned}$$

In contrast, we compare this to the formulation Equation (3.10), recalled below:

$$\begin{aligned}
 p_\delta^{(1)} = \text{minimize} \quad & \sum_{I \leq \delta} b_{I,\delta} z_{I,\delta} \\
 \text{s.t} \quad & z_{I,\delta} \in \mathbb{R}, \quad I \leq \delta, \\
 & z_{I,\delta} \geq 0, \quad I \leq \delta, \\
 & \sum_{I \leq \delta} z_{I,\delta} = 1.
 \end{aligned}$$

Comparing the two LPs, it is easy to see that $p_H \leq p_\delta^{(1)}$. Also, if $p_H \geq 0$, then $p_\delta^{(1)} \geq 0$. Therefore, the result follows. \square

Comparison with interval representations: Now, we compare Bernstein relaxation to the interval relaxation over $[0, 1]^n$. Interval relaxations are presented by Ratschan & She (2010) and in our previous work (Sankaranarayanan et al., 2013). Notably, let K be a hyper-rectangular domain. The interval relaxation replaces each monomial \vec{x}^I with an interval over K . The interval for a polynomial p is obtained by summing up the interval for each term. While there exist many approaches to evaluate a polynomial over an interval, we will consider the scheme (implicitly) adopted by Ratschan & She (2010) and in our previous work (Sankaranarayanan et al., 2013) that uses an interval arithmetic-based LP relaxation for (parametric) polynomial optimization problems. Here, we will fix $K : [0, 1]^n$, mapping arbitrary, bounded hyper-rectangles to this domain through a linear change of variables (see Section 5.3).

LEMMA 4.2 A polynomial $p : \sum_{I \leq \delta} c_I \vec{x}^I$ can be shown to be non-negative over $[0, 1]^n$ using interval arithmetic if all its coefficients $c_I \geq 0$.

Following this, we note that if a polynomial $p : \sum_{I \leq \delta} c_I \vec{x}^I$ has $c_I \geq 0$, then all its Bernstein coefficients are non-negative following Equation (3.3).

LEMMA 4.3 Any polynomial p_I that can be shown non-negative over $K : [0, 1]^n$ using interval arithmetic can also be shown non-negative using the Bernstein polynomial-based formulation Equation (3.10).

Proof. It can be shown that the optimal value of Equation (3.10) is in fact the minimal Bernstein polynomial coefficient, which has to be non-negative for p_I . \square

Comparing Bernstein relaxations: Note that Proposition 3.2 has already demonstrated that any polynomial that can be shown non-negative over the unit interval by Equation (3.10) can be shown non-negative by Equation (3.11). Likewise, Equation (3.12) is at least as powerful as Equation (3.11) in this respect. Therefore, the major advantage of using Bernstein polynomials is that, in addition to positivity, the three non-trivial properties of Lemma 3.2 can be used to add linear relationships between the decision variables in the RLT. We first demonstrate that the second relaxation Equation (3.11) is strictly more powerful than the relaxation in Equation (3.10).

EXAMPLE 4.1 Consider the simple univariate polynomial below:

$$\text{Show that } p(x) : 4x^2 - 4x + 1 \geq 0 \quad \text{on } [0, 1]. \quad (4.2)$$

For this example, we find that the relaxation (3.11) with a degree 2 computes exact optimal value $p_\delta^{(2)} = p^* = 0$, proving the positivity of p over $[-1, 1]$. However, Equation (3.10) yields a minimal value of -1 and fails to prove positivity on $[0, 1]$.

Now, we demonstrate that the third relaxation Equation (3.12) is strictly more powerful through an example.

EXAMPLE 4.2 We will consider the following bivariate polynomial:

$$p(x) = x^2 + y^2 \quad \text{on } [-1, 1]^2. \quad (4.3)$$

For a degree $\delta = (2, 2)$, the optimal value of (3.10) is $p_\delta^{(1)} = -2$, which does not establish the positivity of p on $[-1, 1]$. If we use the second linear programme (3.11), the optimal value will be improved and we find $p_\delta^{(2)} = -0.5$, but still not sufficient to prove the positivity of p on $[-1, 1]$. Now, when we use the third linear programme (3.12), we obtain the exact optimal value $p_\delta^{(3)} = 0$ and ensure the positivity of p over $[-1, 1]^2$.

5. Synthesis of polynomial Lyapunov functions

Given an ODE in the form: $d\vec{x}/dt = f(\vec{x})$ with equilibrium $\vec{x}^* = 0$, we wish to find a Lyapunov function $V(\vec{x})$ over a given rectangular domain R_x containing 0.

NOTE 1 (Positive semi-definite vs. positive definite) As presented in Section 2.3, our approach fixes a polynomial template $V_{\vec{c}}(x) = V(\vec{x}, \vec{c})$ for the target Lyapunov function, and computes its Lie derivative form $V'(\vec{x}, \vec{c})$. It then searches for coefficients \vec{c} such that $V(\vec{x}, \vec{c})$ is positive definite over R_x and V' is negative definite. We recall the standard approach to encoding positive definiteness, following Papachristodoulou & Prajna (2002), by writing $V = U + \vec{x}' A \vec{x}$ for a positive semi-definite function U and a diagonal matrix A with small but fixed positive diagonal entries. Therefore, we will focus on encoding positive or negative semi-definiteness and use this approach to extend to positive/negative definiteness.

We will now demonstrate how the three LP relaxation Equations (3.10–3.12), described in Equation (3), extend to search for Lyapunov functions, wherein

- (a) the polynomial of interest is $V(\vec{x}, \vec{c})$ parametrized by unknowns \vec{c} ;
- (b) the interval of interest is a general box $\prod_{j=1}^n [\ell_j, u_j]$ rather than $[0, 1]^n$;
- (c) we wish to encode the positive semi-definiteness of $-V'(\vec{x}, \vec{c})$ rather than V itself (following the technique in Section 5.2).

5.1 Encoding positivity of parametric polynomial

We first consider the problem of extending the LP relaxation to find values of parameters \vec{c} , such that, a parametric polynomial $\mathcal{P}(\vec{x}, \vec{c})$ is positive semi-definite over the interval $[0, 1]^n$.

Recall that, given a known polynomial $p(\vec{x})$, our first step was to write down $p(\vec{x})$ using its Bernstein expansion as $p(\vec{x}) : \sum_{I \leq \delta} b_I B_{I, \delta}$. Thus, the overall form of Equations (3.10–3.12) can be written as

$$\min \sum_{I \leq \delta} b_I z_I \quad \text{s.t.} \quad A\vec{z} \leq \vec{b}.$$

However, Bernstein coefficients of a parametric polynomial $\mathcal{P}(\vec{x}, \vec{c})$ are not known in advance. Let \vec{m} denote a vector of monomials \vec{x}^I for $I \leq \delta$. The polynomial $\mathcal{P}(\vec{x}, \vec{c})$ can be written as $\vec{c}^t \cdot \vec{m}$. Furthermore, each monomial \vec{x}^I itself has a Bernstein expansion:

$$\vec{x}^I : \sum_{J \leq \delta} b_{J,I} B_{J,\delta}.$$

Consider a matrix \mathcal{B} , wherein, each row corresponds to a monomial \vec{x}^I , and each column to a Bernstein polynomial $B_{J,\delta}$. The coefficient corresponding to row I and column J is $b_{J,I}$, the Bernstein coefficient for \vec{x}^I corresponding to $B_{J,\delta}$. Therefore, we use \mathcal{B} to convert polynomials from monomial to the Bernstein basis:

$$\mathcal{P}(\vec{x}, \vec{c}) : \vec{c}^t \vec{m} = \vec{c}^t \mathcal{B} \vec{z}, \text{ wherein } \vec{z} \text{ represents the Bernstein polynomials.}$$

Therefore, the LP relaxation Equations (3.10–3.12) have the following form:

$$\begin{aligned} \min \quad & \vec{c}^t \mathcal{B} \vec{z} \\ \text{s.t.} \quad & A \vec{z} \leq \vec{b}. \end{aligned} \tag{5.1}$$

Equation (5.1) is, in fact, a bilinear programme which can be reformulated using its dual to a multiparametric linear optimization problem (Jones *et al.*, 2007; Kvasnica *et al.*, 2004). However, the direct resolution of a multiparametric programme is very expensive since it requires to find exponentially many *critical regions*, and for each region we have to find our optimal value which will be an affine function depending on the parameter vector \vec{c} . Therefore, rather than solve the optimization problem Equation (5.1), we simply seek values of \vec{c} such that

$$\text{find } \vec{c} \text{ s.t. } (\forall \vec{z}) A \vec{z} \leq \vec{b} \Rightarrow \vec{c}^t \mathcal{B} \vec{z} \geq 0. \tag{5.2}$$

We now use Farkas' lemma, a well-known result in LP, to dualize Equation (5.2).

LEMMA 5.1 \vec{c} is a solution to the problem in Equation (5.2) if and only if there exist multipliers $\vec{\lambda} \geq 0$ such that

$$A^t \vec{\lambda} = -\mathcal{B}^t \vec{c}, \quad \vec{b}^t \vec{\lambda} \leq 0, \quad \text{and } \vec{\lambda} \geq 0.$$

As a result, we now have a procedure to reduce the search for a parametric positive polynomial as the feasibility problem for a set of linear constraints.

REMARK 5.1 The trick of using Farkas' Lemma to handle multilinear constraint is well known from previous work on the synthesis of ranking functions (Colon & Sipma, 2001; Podelski & Rybalchenko, 2004).

5.2 Simplified encoding

Thus far, our approaches have encoded both the positive definiteness of $V(\vec{x}, \vec{c})$ and the negative definiteness of $V'(\vec{x}, \vec{c})$ to yield a combined linear or semi-definite programme that can be used to synthesize the Lyapunov function. In this section, we propose a simplified approach that simply focuses on encoding the negative definiteness of $V'(\vec{x}, \vec{c})$, extracting a solution \vec{c} and checking that the result $V_{\vec{c}}(\vec{x})$ is in fact positive definite.

1. Choose a form $V(\vec{x}, \vec{c})$.

2. Encode negative definiteness of $V'(\vec{x}, \vec{c})$ (the Lie derivative) over U . In particular, we do not encode the positive definiteness of $V(\vec{x}, \vec{c})$.
3. Compute a solution for \vec{c} and *check* that the solution is, in fact, positive definite over U .

The approach is motivated by the following result from [Vannelli & Vidyasagar \(1985, Page 72, Lemma 3\)](#). A proof of this theorem is also included for the sake of completeness.

THEOREM 5.1 If \mathcal{S} is an asymptotically stable system on U , $V(\vec{x})$ is a continuous function over U with $V(\vec{0}) = 0$ and V' is negative definite, then V is positive definite in some neighbourhood of $\vec{0}$.

Proof. Assume, for the sake of a contradiction, that every neighbourhood N of $\vec{0}$ has a point $\vec{x}_0 \neq 0$ such that $V(\vec{x}_0) \leq 0$. For each N, \vec{x}_0 , we will now show that the trajectory starting at \vec{x}_0 cannot converge asymptotically to $\vec{0}$. Let $t \in [0, T)$ represent a time interval for which $\vec{x}(t) \in N \setminus \{\vec{0}\}$. If $\vec{x}(t) \in N \setminus \{\vec{0}\}$ forever, then we set $T = \infty$. Consider any finite, or infinite sequence of time instances $t_0 = 0 < t_1 < t_2 \dots < T$. We observe that $0 \geq V(\vec{x}(t_0)) > V(\vec{x}(t_1)) > \dots$, since

$$V(\vec{x}(t_i)) = V(\vec{x}(t_{i-1})) + \underbrace{\int_{t_{i-1}}^{t_i} V'(\vec{x}(s)) ds}_{<0}$$

By the continuity of V , and the fact that $V(\vec{0}) = 0$, we conclude that the trajectory $\vec{x}(t)$ cannot converge asymptotically to $\vec{0}$. In other words, the system is not asymptotically stable. This directly contradicts our original claim. \square

We will focus on encoding the negative definiteness of $V'(\vec{x}, \vec{c})$ over the given domain R_x , without requiring that $V(\vec{x}, \vec{c})$ be positive definite. Once a suitable \vec{c} is found, we simply check if $V_{\vec{c}}(\vec{x})$ is positive definite over R_x . Failing this, we simply choose a point $\vec{y} \in R_x$ where V fails to be positive and simply repeat our procedure by adding an additional constraint that $V(\vec{y}, \vec{c}) > 0$.

REMARK 5.2 The advantage of this simplified encoding is that the synthesis part for $V_{\vec{c}}$ is replaced by a simple check of positivity. If the obtained $V_{\vec{c}}$ is non-positive, we can conclude using Theorem 5.1 that the system is already unstable.

As a result, the simplified encoding results in an LP relaxation with fewer constraints.

It now remains to address: (a) the transformation from a given domain R_x to the domain $[0, 1]^n$ for applying the Bernstein polynomial-based LP relaxations, and (b) encode negative definiteness of the parametric polynomial $V'(\vec{x}, \vec{c})$.

5.3 Transforming co-ordinates

Let $R_x : \prod_{j=1}^n [\ell_j, u_j]$ be the domain of interest. We consider the change-of-basis transformation from $\vec{x} \in R_x$ to a new set of variables $\vec{y} \in [0, 1]^n$

$$x_j \mapsto \ell_j + y_j(u_j - \ell_j).$$

Let \vec{m} denote the original monomial basis over \vec{x} consisting of monomials \vec{x}^I for $I \leq \delta$. Corresponding to this, we define $\hat{\vec{m}}$ as the monomial basis over \vec{y} , consisting of monomials \vec{y}^I for $I \leq \delta$. It is easy to

see that any monomial \vec{x}^J can be written as a polynomial involving monomials \vec{y}^J of degree J at most I . Therefore,

$$\vec{m} \equiv T \hat{\vec{m}} \text{ wherein}$$

each row of T corresponds to a monomial \vec{x}^J and each column to a monomial \vec{y}^J . Each row therefore lists the coefficients of the monomial \vec{x}^J as a function over \vec{y} .

Therefore, $\vec{c}^t \vec{m} \equiv \vec{c}^t T \hat{\vec{m}}$. Rather than encoding the positivity of the original polynomial $V(\vec{x}, \vec{c}) : \vec{c}^t \vec{m}$ over R_x , we encode that of $(T^t \vec{c})^t \hat{\vec{m}}$ over $[0, 1]^n$.

5.4 Lie derivatives

Finally, Lyapunov function synthesis requires us to encode the negative definiteness of $V'(\vec{x}, \vec{c})$ rather than V . Once again, this requires us to consider the coefficients of the form $V'(\vec{x}, \vec{c})$ as a linear transformation applied over \vec{c} .

Since the RHS of the ODE is polynomial, we consider the Lie derivative of each monomial \vec{x}^J as a polynomial p_I . Let \mathcal{D} represent the matrix wherein each row of \mathcal{D} represents the monomial \vec{x}^J and the contents of the row are the coefficients of the Lie derivative of \vec{x}^J .

Therefore, applying the Lie derivative to $V(\vec{x}, \vec{c}) : \vec{c}^t \vec{m}$, we obtain

$$V'(\vec{x}, \vec{c}) : \vec{c}^t \mathcal{D} \vec{m}'.$$

Here the vector \vec{m}' represents the set of monomials involved in the Lie derivative.

5.5 Overall encoding

To summarize, we are asked to find a value of \vec{c} such that the Lie derivative of the polynomial $V(\vec{x}, \vec{c})$ is non-negative over R_x . Let \mathcal{D} represent the matrix form of the Lie derivatives on the monomial basis \vec{m} , T represent the transformation of the monomials from R_x to $[0, 1]^n$ and, finally, \mathcal{B} represent the transformation to Bernstein form. The overall optimization involves finding \vec{c} such that

$$\text{find } \vec{c} \text{ s.t. } (\forall \vec{z}) A \vec{z} \leq \vec{b} \Rightarrow (\mathcal{B}^t \times T^t \times \mathcal{D}^t \vec{c})^t \vec{z} \leq 0. \quad (5.3)$$

As a result, applying Farkas lemma transforms this into solving the feasibility problem below:

$$\text{find } \vec{c} \text{ s.t. } (\exists \vec{\lambda}) \underbrace{A^t \vec{\lambda} = \mathcal{B}^t T^t \mathcal{D}^t \vec{c}, \vec{b}^t \vec{\lambda} \leq 0, \text{ and } \vec{\lambda} \geq 0}_{\text{LP feasibility}}. \quad (5.4)$$

We note that $\vec{c} = 0$ is seemingly a trivial solution to the feasibility problem in Equation (5.4). But, this does not yield a Lyapunov function. To address, this, we recall that our goal is to encode the *negative definiteness* and not the negative semi-definiteness of the derivative. On the other hand, Equation (5.4) encodes the negative semi-definiteness.

As mentioned earlier, we ensure that $U : V'(\vec{x}, \vec{c}) - \vec{x}' \Lambda \vec{x}$ is negative semi-definite using Equation (5.4) rather than V' itself. The matrix Λ is a diagonal matrix whose diagonal entries are all set to a small value $\epsilon > 0$, chosen by the user. We choose $\epsilon = 0.1$ for most of our experiments.

REMARK 5.3 The problem posed in Equation (5.4) can be simplified considerably for the LP relaxation Equation (3.10)). In the absence of further bounds about the Bernstein polynomials, the smallest Bernstein coefficient is a lower bound on the minimum value of a polynomial. Therefore, the constraints in

Equation (5.4) can be simplified as

$$\text{find } \vec{c} \text{ s.t. } \mathcal{B} \cdot \vec{c} \geq 0. \quad (5.5)$$

Effectively the form above imposes that all the Bernstein coefficients of $V(\vec{x}, \vec{c})$ are non-negative. This implicitly eliminates the multipliers $\vec{\lambda}$ from the LP relaxation.

REMARK 5.4 Infeasibility of Equation (5.4) means that our search failed to find a Lyapunov function. This can indicate many problems, including (a) the system is not stable, (b) the system is stable but no polynomial Lyapunov function exists (Ahmadi *et al.*, 2011), (c) the system is stable with a polynomial Lyapunov but it is not provable using the relaxation that we have chosen to arrive at our LP.

5.6 Higher relaxation degree

Our linear relaxations are based on a fixed degree for the Bernstein polynomial expansion. By default, this degree called δ is fixed to some chosen value at the beginning of the algorithm. However, if the technique fails to find a Lyapunov function, we may improve precision by increasing the degree δ . The following convergence result motivates the possible improvement in the lower bounds of the LP relaxation by increasing the degree bound δ (Lin & Rokne, 1996):

THEOREM 5.2 Let p be a multivariate polynomial of degree $\delta = (\delta_1, \dots, \delta_n)$ and let $b_{I,\delta} = b_I$ be its Bernstein coefficients with respect to the unit box $[0, 1]^n$:

$$\left| b_{I,\delta} - p\left(\frac{I}{\delta}\right) \right| = O\left(\frac{1}{\delta_1} + \dots + \frac{1}{\delta_n}\right) \quad \text{for all } I \leq \delta. \quad (5.6)$$

As a consequence, when the optimal value of our linear or bilinear programme is negative, we can just increase the degree of the relaxation allowing the relaxation to be more precise and then increasing the possibility to find our Lyapunov function.

5.7 Branch-and-bound decomposition

A second, more widely used approach to improving the relaxation, is to perform a branch-and-bound decomposition. The essential idea consists in verifying the so-called *vertex condition* (Garloff, 1993) for the given hyper-rectangle which guarantees that the LP relaxation coincides with the optimal value. Informally, this condition requires that no local minima for a polynomial p exist in the interior of the rectangle. If it does not hold, we will simply divide our rectangle and keep doing it until reaching the global minimum and getting exact bounds in each sub-box. In our case, we have two main differences:

1. We do not have a fixed polynomial, but a parametric polynomial $V(\vec{x}, \vec{c})$.
2. The global minimum for a Lyapunov function V is known in advance as the equilibrium $\vec{0}$. Likewise, the negation of its derivative also has $\vec{0}$ for a global minimum.

For these reasons, our branch-and-bound approach focuses on decomposing the given region R_x , so that the equilibrium $\vec{0}$ lies in the boundaries of our cells rather than the relative interior, in an attempt to satisfy the *vertex condition*. So, if a Lyapunov function is not found, we simply choose a variable x_j and consider two cells $R_x^{(1)} : R_x \cap \{x_j \leq 0\}$ and $R_x^{(2)} : R_x \cap \{x_j \geq 0\}$. The cells may be recursively subdivided if necessary. In the limit, this approach creates 2^n cells, and can be expensive for systems with more than 10s of variables. The computational complexity can be mitigated by examining a few cells in the

decomposition and trying to find a Lyapunov candidate based on the examined cells. We can then check if the Lyapunov candidates are indeed Lyapunov functions by considering the other cells. This approach can, in the worst case, examine every cell in the decomposition. However, if a good empirical strategy for selecting the cells can be found, the approach can save much effort involved in encoding the LP relaxations for an exponential number of cells.

6. Numerical results

In this section, we present an evaluation of various LP relaxations using Bernstein polynomial Equations (3.10–3.12), extended using the technique for encoding the positivity of a parametric form, presented in Section 5.

6.1 Implementation

Our approaches are implemented as a MATLAB(tm) toolbox for synthesizing Lyapunov function. Apart from a description of the system to be analysed, the inputs include the maximum degree $\delta : (\delta_1, \dots, \delta_n)$ for the Bernstein expansion in each variable, the region of interest (fixed to $[-1, 1]^n$ for all of our evaluation) and the number of subdivisions along each dimension. Furthermore, our toolbox implements three LP relaxations, each adding more constraints over the previous. The first relaxation, based on Equation (3.10), simply uses the non-negativity and the unit summation properties of Bernstein polynomials. The second LP relaxation, based on Equation (3.11), adds upper bounds to the Bernstein polynomials and, finally, the third approach Equation (3.12) adds the recurrence relations between the Bernstein polynomials. Each approach is used in the Lyapunov search by encoding the dual form Equation (5.4).

6.2 Numerical examples

We first compare and contrast the three LP relaxations here over some benchmark examples from our previous work (Sankaranarayanan *et al.*, 2013). Then, using a special problem generator, we compare the results we obtain for each benchmark with those obtained by using the `findlyap` function in SOS-TOOLS (Papachristodoulou *et al.*, 2013), and the Lyapunov functions obtained in our previous work. For all the examples, we wish to prove asymptotic stability over $R_x = [-1, 1]^n$. We will report for each programme the Lyapunov function, the number of boxes in our decomposition and two computational times.

SETUP is the needed time to compute the data for the linear programme. This includes:

1. the time needed to compute the matrix \mathcal{B} (for all three relaxations),
2. computing bounds on the Bernstein polynomials (for second and third relaxations) and
3. time needed to compute recurrences for each Bernstein polynomial (for the third relaxation).

In fact, much of the computation of \mathcal{B} and the bounds on it are independent of the actual problem instance. They can be performed once, and cached for a given number n of variables and given degree bounds δ , instead of recomputing them separately for each problem.

LPTIME is the computational time associated with solving the LP relaxation using the `linprog` function provided by MATLAB(tm). Also, we should mention that since all the LPs are feasibility problems, the objective function is set to be the maximization of the sum of the coefficients.

6.2.1 Benchmarks from Sankaranarayanan et al. (2013) and comparison with Handelman representations

EXAMPLE 6.1 Consider the system over (x, y) :

$$\frac{dx}{dt} = -x^3 + y, \quad \frac{dy}{dt} = -x - y. \quad (6.1)$$

The Handelman relaxation technique in our previous work (Sankaranarayanan et al., 2013) finds the Lyapunov function $x^2 + y^2$ taking < 0.1 s, whereas SOS discovers $1.2118x^2 + 1.6099 \times 10^{-5}xy + 1.212y^2$, requiring 0.4 s. The three relaxations each using a subdivision of $[-1, 1]^2$ discover the function $x^2 + xy + \frac{1}{2}y^2$ (with a multiplicative factor, and modulo small perturbations due to floating point error). Interestingly, the system is globally asymptotically stable, and the Lyapunov function discovered by our approach is valid globally.

Next, we consider the system

$$\frac{dx}{dt} = -x^3 - y^2, \quad \frac{dy}{dt} = xy - y^3. \quad (6.2)$$

The Handelman relaxation approach (Sankaranarayanan et al., 2013) finds a 4-degree Lyapunov function $x^4 + 2x^2y^2 + y^4$, requiring < 0.1 s, whereas the SOS approach produces $0.62788x^4 + 0.052373x^3 + 0.65378x^2y^2 + 1.1368x^2 - 0.18536xy^2 + 0.60694y^4 + 1.1368y^2$ after deleting terms with coefficients $< 10^{-7}$. The SOS approach requires roughly 0.4 s for this example. Our approach discovers degree 2 Lyapunov function $x^2 + y^2$ that is also globally stable.

Finally, we consider the system

$$\frac{dx}{dt} = -x - 1.5x^2y^3, \quad \frac{dy}{dt} = -y^3 + 0.5x^2y^2. \quad (6.3)$$

The approach in Sankaranarayanan et al. (2013) proves asymptotic stability over $[-1, 1]^2$ through the function $0.2x^2 + y^2$, requiring 0.4 seconds, whereas the SOS approach finds $2.4229x^2 + 4.4868y^2$ requiring a running time of 8.8 seconds.

The specific Lyapunov functions found for systems Equations (6.1–6.3), the running times and number of subdivisions needed are summarized in Table 1.

6.2.2 Synthetic Benchmarks and comparison with SOS We now consider a second class of *synthetic benchmarks* that were generated using a special problem generator, constructed for generating challenging examples of locally stable polynomial vector fields of varying degrees and number of variables to evaluate the various techniques presented here. Our overall idea is to fix two homogeneous polynomials $V_1(\vec{x})$ and $V_2(\vec{x})$ that are positive definite over a region of interest, chosen to be $K: [-1, 1]^n$ for our examples. The benchmarks described in this section along with the Lyapunov functions synthesized are available online through arXiv (Sassi et al., 2014).

Subsequently, for each choice of V_1, V_2 , we attempt to find a system $\frac{d\vec{x}}{dt} = F(\vec{x})$ such that the Lie derivative of V_1 is $-V_2$, and with an equilibrium at $\vec{0}$.

$$(\nabla V_1) \cdot F = -V_2, \quad \text{and} \quad F(\vec{0}) = \vec{0}. \quad (6.4)$$

Naturally, any such system using the vector field F is guaranteed to be asymptotically stable due to the existence of V_1, V_2 . To synthesize a benchmark that is guaranteed to have asymptotic stability, we need

TABLE 1 Table showing Lyapunov functions computed by each of the three LP relaxations on the three systems considered in Example 6.1. The column Relaxation indicates which of the three LP relaxations was used, the Lyapunov function for each approach, the number of Boxes in the subdivision and the computational times split into computing the matrices and LP data, and the actual time needed to solve the LP. All timings are in seconds

System	Relaxation	Lyapunov	# Boxes	SETUP	LPTIME
Equation (6.1)	LP1	$4.5807x^2 + 4.5807xy + 2.2906y^2$	2	0.06	0.36
	LP2	$5x^2 + 4.9995xy + 2.5002y^2$	2	0.09	0.34
	LP3	$5x^2 + 5xy + 2.5y^2$	2	0.15	0.37
Equation (6.2)	LP1	$4.3039x^2 + 4.3039y^2$	4	0.13	0.38
	LP2	$4.9998x^2 + 5y^2$	4	0.18	0.41
	LP3	$5x^2 + 5y^2$	4	0.37	0.77
Equation (6.3)	LP1	$4.6809x^2 + 4.9547y^2$	4	0.16	0.36
	LP2	$4.9998x^2 + 5y^2$	4	0.18	0.40
	LP3	$5x^2 + 5y^2$	4	0.33	0.43

to find a suitable F within a given degree bound. To this end, we parametrize our system F by a set of parametric polynomials and attempt to find parameters that satisfy Equation (6.4). It is easy to show that our approach leads to a set of linear equations on the parameters defining the entries in F and solving these equations yields a suitable system F . The difficulty here lies in choosing appropriate V_1, V_2 so that the system F can be found. In our experience, if V_1, V_2 are chosen arbitrarily, the likelihood of finding a function F that satisfies Equation (6.4) seems quite small. Furthermore, since F involves n polynomials, the technique yields prohibitively large equations for $n \geq 6$. Our approach to synthesize benchmarks is based on carefully controlling the choice of V_1, V_2 and repeated trial and error, until a feasible system of equations is discovered, to synthesize a benchmark. Having synthesized our benchmark, we *hide* the functions V_1, V_2 used to generate it and simply present the system F to our implementation, as well as for SOS programmes.

The key to finding benchmarks lies in the generation of the polynomials V_1, V_2 . We generated V_1 as one of two simple forms: (a) $V_1 : \vec{x}^t \Lambda_1 \vec{x}$, or (b) $V_1 : \vec{m}^t \Lambda_2 \vec{m}$, wherein \vec{m} is a vector of squares of the system variables of the form $[x_1^2, x_2^2, \dots, x_n^2]^t$, and Λ_1, Λ_2 are diagonal matrices with non-negative diagonal entries chosen at random.

The polynomial V_2 is chosen to be a positive-definite polynomial over $[-1, 1]^n$. The key idea here is to generate V_2 that is guaranteed to be positive definite over $[-1, 1]^n$ by writing

$$V_2(\vec{x}) : \vec{x}^t \Lambda \vec{x} + \sum_j q_j \prod_{i=1}^n (1 + x_i)^{p_{j,i}} (1 - x_i)^{q_{j,i}},$$

essentially as a Schmüdgen representation involving the polynomials $(1 - x_i), (1 + x_i)$ for $i \in [1, n]$ and SOS polynomials q_j obtained by squaring and adding randomly generated polynomials together.

REMARK 6.1 Even though our approach synthesizes an ODE $d\vec{x}/dt = F(\vec{x})$ that by design has a Lyapunov function $V(\vec{x})$, we note that the resulting system may (and often does) admit many other Lyapunov functions with a possibility of a larger domain of attraction towards the equilibrium $\vec{0}$.

TABLE 2 Performance of our approach on the synthesized benchmarks. The column n : number of variables, d_{\max} : maximum degree of the vector field, SUCC? indicates whether the approach succeeded in finding a Lyapunov function, 3: succeeded with Lyapunov, NP: numerical problem, MO: out-of-memory, d_L : degree of Lyapunov function, d_Q : degree of SOS multipliers, SETUP: setup time, T_{SDP} : SDP Solver time, Rel. Typ.: Relaxation Type, #Box: number of boxes in decomposition, T_{LP} : LP solver time. All times are reported in seconds

ID	n	d_{\max}	Putinar (SOS)					Bernstein (our approach)				
			SUCC?	d_L	d_Q	SETUP	T_{SDP}	Rel. Typ.	SUCC?	# Box	SETUP	T_{LP}
1	2	3	✓	2	2	0.35	0.9	LP1	✓	4	0.17	0.43
								LP2	✓	4	0.20	0.42
								LP3	✓	2	0.17	0.38
2	2	3	✓	2	2	0.3	0.67	LP1	✓	4	0.17	0.42
								LP2	✓	4	0.19	0.38
								LP3	✓	2	0.17	0.35
3	2	3	✓	2	2	0.33	0.61	LP1	✓	4	0.17	0.37
								LP2	✓	4	0.18	0.35
								LP3	✓	2	0.16	0.35
4	2	3	✓	2	2	0.3	0.97	LP1	✓	4	0.17	0.37
								LP2	✓	4	0.21	0.39
								LP3	✓	2	0.17	0.35
5	3	3	✓	2	2	0.86	1.12	LP1	✓	8	0.81	0.47
								LP2	✓	8	0.97	0.61
								LP3	✓	4	1.24	0.70
6	3	5	X(NP)	2	2	0.81	2.3	LP1	✓	8	7.15	6.4
			✓	2	4	39.5	4.2	LP2	✓	8	7.83	17.17
								LP3	X(NP)	8	17.4	102.3
7	3	5	X(NP)	2	2	0.8	2.2	LP1	✓	8	6.50	5.2
			X(NP)	2	4	40	4.6	LP2	✓	8	7.42	5.7
			X(NP)	4	4	40.5	7	LP3	✓	8	13.2	26.8

In many cases, the process of trial and error is required to find pairs V_1, V_2 that yield a feasible vector field. Using this process, 15 different benchmarks were synthesized with 5 each of degrees 2, 3 and 4, respectively. The benchmark descriptions are available from the extended report (Sassi *et al.*, 2014).

6.2.3 Results Tables 2 and 3 compare the performance of the three LP relaxations implemented in our prototype with an implementation Putinar (SOS), based on Putinar representation of the Lyapunov function and the negation of its derivative, built using SOSTOOLS. Here we should mention that, in order to reduce the complexity of the ‘LP3’ relaxation, we reduce ourselves to a first level of lower degrees (see Remark 3.1). For each of the 15 benchmarks, we run both tools under different setups. The Putinar (SOS) approach is run with varying degrees of the Lyapunov function d_L , and degrees of the SOS multipliers d_Q . We attempted three sets $(d_L, d_Q) = (2, 2), (2, 4), (4, 4)$ in succession, stopping as soon as a Lyapunov function is found without a failure. To experiment with our approach and enable a full comparison, we attempt all the three relaxations for all the benchmarks.

TABLE 3 Performance of our approach on the synthesized benchmarks (continued)

ID	n	d_{\max}	Putinar (SOS)					Bernstein (our approach)				
			SUCC?	d_L	d_Q	SETUP	T_{SDP}	Rel. Typ.	SUCC?	# Box	SETUP	T_{LP}
8	3	5	X(NP)	2	2	0.8	1.7	LP1	✓	8	10.63	10.9
			X(NP)	2	4	40.9	7.9	LP2	✓	8	11.91	30.97
			X(NP)	4	4	40.1	5.5	LP3	X(NP)	8	22.38	130.77
9	3	2	X(NP)	2	2	0.9	4.1	LP1	X(NP)	8	1.99	0.61
			X(NP)	2	4	42.2	3.7	LP2	✓	8	2.06	0.92
			✓	4	4	41.9	3.1	LP3	✓	8	3.04	3.81
10	3	5	X(NP)	2	2	0.9	2.9	LP1	X(NP)	8	3.48	3.19
			X(NP)	2	4	38.3	5.3	LP2	✓	8	1.23	1.88
			✓	4	4	38.7	5.54	LP3	✓	8	1.56	0.60
11	4	3	✓	2	2	3.7	3.1	LP1	✓	16	3.58	3.25
								LP2	✓	16	4.34	17.27
								LP3	✓	16	9.05	53.5
12	4	3	X(NP)	2	2	3.7	2.1	LP1	✓	16	5.16	16.85
			X(MO)	2	4	> 600		LP2	✓	16	6.38	12.86
								LP3	X(NP)	16	22.23	224.23
13	4	6	X(NP)	2	2	4	3.1	LP1	X(NP)	16	41.36	627.25
			X(MO)	2	4	> 600		LP2	X(NP)	16	43.38	988.31
								LP3	X(TO)	16	> 1200	
14	4	6	X(NP)	2	2	3.8	3.6	LP1	X(NP)	16	37.45	339.86
			X(MO)	2	4	> 600		LP2	X(NP)	16	41.93	1049.53
								LP3	X(TO)	16	> 1200	
15	4	6	X(NP)	2	2	3.8	3.9	LP1	X(NP)	16	38.55	368.48
			X(MO)	2	4	> 600		LP2	X(NP)	16	45.32	888.33
								LP3	X(TO)	16	> 1200	

MO: out-of-memory termination; TO: time-out. All times are reported in seconds.

We note that the LP relaxation approach is generally successful in discovering Lyapunov functions. In 7 out of 15 cases, all three LP relaxations succeed, while at least one LP relaxation succeeds in 12 out of 15 cases. On the other hand, the Putinar (SOS) approach succeeds in 9 out of the 15 attempts, with *numerical problems* (NP) being the most common failure mode. These may arise due to many reasons, but commonly due to the Hessian matrix becoming ill-conditioned during the calculation of a Newton step. For benchmarks 12-15, the polynomials involved become so large, that the Putinar (SOS) approach runs out of memory during the problem set up, causing MATLAB(tm) to crash. Our approach also suffers from the same set of problems, but to a noticeably lesser extent. For instance, 11 out of the 45 linear programmes failed due to numerical problems, and 3 more due to timeouts. On the other hand, 15 out of the 29 SDPs terminate with a numerical problem with an additional 4 out-of-memory issues.

On most of the smaller benchmarks, all approaches have comparable timings. In general, the third relaxation (LP3) is the most expensive, often more expensive than the other two LP relaxations or the Putinar (SOS) approach. Likewise, when the degree of the SOS multipliers d_Q is increased from 2 to 4, we witness a corresponding 40× factor increase in the time taken to setup the SDP, with a smaller

increase in the time taken to solve the SDP. For the larger examples, the LP relaxation requires more time, but is generally successful in finding an answer.

Finally, all approaches fail on benchmarks 13–15. A key issue is the blowup in the number of monomial terms to be considered in the parametric polynomial forms for the Lyapunov function and its derivatives. This blowup seems to overwhelm both our approach and the SOS programming approach. We conclude that handling large parametric polynomials efficiently remains a challenging problem for our approach as well as the Putinar (SOS) approach.

7. Conclusion

To conclude, we have examined three different LP relaxations for synthesizing polynomial Lyapunov functions for polynomial systems. We compare these approaches to the standard approaches using Schmüdgen and Putinar representations that are used in SOS programming relaxations of the problem. In theory, the Schmüdgen representation approach subsumes the three LP relaxations. In practice, however, we are forced to use the Putinar representation. We show that the Putinar representation can prove some polynomials positive semi-definite that our approaches fail to. On the other hand, the reverse is also true: we demonstrate a polynomial that is easily shown to be positive semi-definite on the interval $[-1, 1]^n$ through LP relaxations. However, the same fact cannot be demonstrated by a Putinar representation approach. We then compare both approaches over a set of numerical benchmarks. We find that LP relaxations succeed in finding Lyapunov functions for all cases, while the Putinar representation fails in many benchmarks due to numerical (conditioning) issues while solving the SDP. As future work, we wish to extend our approach to a larger class of Lyapunov functions. We also are looking into the problem of analysing systems with non-polynomial dynamics and the synthesis of non-polynomial Lyapunov functions.

Finally, Lyapunov functions have, thus far, remained important theoretical tools for analysing the stability of control systems. However, these tools are seldom used, in practice, for industrial scale systems. This is chiefly due to the burden of manually specifying Lyapunov functions. Therefore, the stability of complex industrial designs is often ‘verified’ by extensive simulations. Recent work by Kapinski *et al.* argues that Lyapunov functions can be of practical value for automotive designs, provided they can be discovered easily, and certified using formal verification tools (Kapinski *et al.*, 2014). We hope that the use of linear relaxations can provide us with more approaches to synthesize Lyapunov functions that can serve as certificates for stability.

Funding

M.A.B.S. and S.S. were supported by the US National Science Foundation (NSF) under CAREER Award # 0953941. All opinions expressed are those of the authors and not necessarily of the NSF.

REFERENCES

- AHMADI, A. A., KRSTIC, M. & PARRILO, P. A. (2011) A globally asymptotically stable polynomial vector field with no polynomial Lyapunov function. *Proceedings of the 50th IEEE Conference on Decision and Control, 2011*, pp. 7579–7580.
- AHMADI, A. & MAJUMDAR, A. (2014) DSOS and SDSOS optimization: LP and SOCP-based alternatives to sum of squares optimization. *International Conference on Information Sciences and Systems (CISS)*, Princeton University. IEEE Press, pp. 1–5.
- BERNSTEIN, S. N. (1912) Démonstration du théorème de Weierstrass fondée sur le calcul des probabilités. *Comm. Kharkov Math. Soc.*, **13**, 1–2.

- BERNSTEIN, S. N. (1915) On the representation of positive polynomials. *Soobshch Kharkov maren ob-va*, **2**, 227–228.
- BERNSTEIN, S. (1952) Collected Works, vol. 1. USSR Academy of Sciences. Russia.
- BERNSTEIN, S. (1954) Collected Works, vol. 2. USSR Academy of Sciences. Russia.
- CHESI, G. (2005) Estimating the domain of attraction via union of continuous families of Lyapunov estimates. *Syst. Control Lett.*, **56**, 326–333.
- CHESI, G. (2010) Polynomial relaxation-based conditions for global asymptotic stability of equilibrium points of genetic regulatory networks. *Int. J. Syst. Sci.*, **41**, 65–72.
- CHESI, G., GARULLI, A., TESI, A. & VICINO, A. (2005) LMI-based computation of optimal quadratic Lyapunov functions for odd polynomial systems. *Int. J. Robust Nonlinear Control*, **15**, 35–49.
- COLLINS, G. (1975) Quantifier elimination for real closed fields by cylindrical algebraic decomposition. *Automata Theory and Formal Languages* (H.Brakhage, ed.), Lecture Notes in Computer Science 33. Berlin: Springer, pp. 134–183.
- COLLINS, G. E. & HONG, H. (1991) Partial cylindrical algebraic decomposition for quantifier elimination. *J. Symb. Comput.*, **12**, 299–328.
- COLON, M. & SIPMA, H. (2001) Synthesis of linear ranking functions. *Tools and Algorithms for Construction and Analysis of Systems* (T. Margaria and W. Yi, eds), vol. 2031. Berlin: Springer, pp. 67–81.
- DANG, T. & SALINAS, D. (2009) Image computation for polynomial dynamical systems using the Bernstein expansion. *CAV'09*, LNCS 5643. Berlin: Springer, pp. 219–232.
- DATTA, R. (2002) Computing Handelman representations. *Mathematical Theory of Networks and Systems*, 2002. Cf. math.berkeley.edu/~datta/MTNSHandelman.ps.
- DOLZMANN, A. & STURM, T. (1997) REDLOG: computer algebra meets computer logic. *ACM SIGSAM Bull.*, **31**, 2–9.
- FAROUKI, R. T. (2012) The Bernstein polynomial basis: a centennial retrospective. *Comput. Aided Geom. Des.*, **29**, 379–419.
- FORSMAN, K. (1991) Construction of Lyapunov functions using Gröbner bases. *Proceedings of the 30th IEEE Conference on Decision and Control*. IEEE, Brighton, December, pp. 798–799.
- GAREY, M. R. & JOHNSON, D. S. (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Oxford: W.H.Freeman.
- GARLOFF, J. (1993) The Bernstein algorithm. *Reliab. Comput.*, **2**, 154–168.
- GOUBAULT, E., JOURDAN, J.-H., PUTOT, S. & SANKARANARAYANAN, S. (2014) Finding non-polynomial positive invariants and Lyapunov functions for polynomial systems through Darboux polynomials. *Proc. American Control Conference (ACC)*. Portland: IEEE Press, pp. 3571–3578.
- HAFSTEIN, S. (2002) Stability analysis of nonlinear systems with linear programming. *Ph.D. Thesis*, Gerhard-Mercator-University Duisburg.
- HAFSTEIN, S. (2014) Revised CPA method to compute Lyapunov functions for nonlinear systems. *J. Math. Anal. Appl.*, **4**, 610–640.
- HANDELMAN, D. (1988) Representing polynomials by positive linear functions on compact convex polyhedra. *Pacific J. Math*, **132**, 35–62.
- HARRISON, J. (2007) Verifying nonlinear real formulas via sums of squares. *Proceeding of the International Conference on Theorem Proving in Higher Order Logics* (K. Schneider and J. Brandt, eds), Lecture Notes in Computer Science 4732. Berlin: Springer, 2007, pp. 102–118.
- HÄRTER, V., JANSSON, C. & LANGE, M. (2012) VSDP: A matlab toolbox for verified semidefinite-quadratic-linear programming. Cf. <http://www.ti3.tuhh.de/jansson/vsdp/>.
- HAUSDORFF, F. (1921) Summationsmethoden und Momentfolgen i. *Math. Zeit*, **9**, 74–109.
- HENRION, D. & LASSERRE, J. (2006) Convergent relaxations of polynomial matrix inequalities and static output feedback. *TransIEEE Trans. Autom. Control*, **51**, 192–202.
- JARVIS-WLOSZEK, Z. W. (2003) Lyapunov based analysis and controller synthesis for polynomial systems using sum-of-squares optimization. *Ph.D. Thesis*, UC Berkeley.

- JOHANSEN, A. (2000) Computation of Lyapunov functions for smooth nonlinear systems using convex optimization. *Automatica*, **36**, 1617–1626.
- JONES, C., BARIC, M. & MORARI, M. (2007) Multiparametric linear programming with applications to control. *Eur. J. Control*, **13**, 152–170.
- KAMYAR, R. & PEET, M. M. (2014) Polynomial optimization with applications to stability analysis and control—alternatives to sum of squares. arXiv, vol. abs/1408.5119, Available online: <http://arxiv.org/abs/1408.5119>.
- KAPINSKI, J., DESHMUKH, J. V., SANKARANARAYANAN, S. & ARECHIGA, N. (2014) Simulation-guided Lyapunov analysis for hybrid dynamical systems. *Hybrid Systems: Computation and Control (HSCC)*. Berlin, Germany: ACM Press, pp. 133–142.
- KVASNICA, M., GRIEDER, P., BAOTIC, M. & MORARI, M. (2004) Multi-parametric toolbox (MPT). *HSCC (Hybrid Systems: Computation and Control)*. Lecture Notes in Computer Science, vol. 2993, pp. 448–462.
- LESSERRE, J. B. (2001) Global optimization with polynomials and the problem of moments. *SIAM J. Optim.*, **11**, 796–817.
- LIN, Q. & ROKNE, J. (1996) Interval approximation of higher order to the ranges of functions. *Comput. Math. Appl.*, **31**, 101–109.
- MAJUMDAR, A., AHMADI, A. A. & TEDRAKE, R. (2014) Control and verification of high-dimensional systems via DSOS and SDSOS optimization. *IEEE Conference on Decision and Control (CDC)*, Los Angeles, December 2014.
- MEISS, J. D. (2007) *Differential Dynamical Systems*. SIAM, Philadelphia.
- MONNIAUX, D. & CORBINEAU, P. (2011) On the generation of Positivstellensatz witnesses in degenerate cases. *ITP*, Lecture Notes in Computer Science 6898. Berlin: Springer, pp. 249–264.
- MOTZKIN, T. (1967) The arithmetic-geometric inequality. *Proceedings of a Symposium on Inequalities*. Wright-Patterson Air Force Base, Ohio: Academic Press, pp. 205–224.
- MUÑOZ, C. & NARKAWICZ, A. (2013) Formalization of a representation of Bernstein polynomials and applications to global optimization. *J. Autom. Reasoning*, **51**, 151–196.
- PAPACHRISTODOULOU, A., ANDERSON, J., VALMORBIDA, G., PRAJNA, S., SEILER, P. & PARRILO, P. A. (2013) SOS-TOOLS: Sum of squares optimization toolbox for MATLAB Version 3.00.
- PAPACHRISTODOULOU, A. & PRAJNA, S. (2002) On the construction of Lyapunov functions using the sum of squares decomposition. *IEEE CDC*. Las Vegas, Nevada: IEEE Press, 2002, pp. 3482–3487.
- PAPACHRISTODOULOU, A. & PRAJNA, S. (2005) Analysis of non-polynomial systems using the sum of squares decomposition. *Positive Polynomials in Control* (D. Henrion and A. Garulli, eds), Lecture Notes in Control and Information Science 312. Berlin, Heidelberg: Springer, pp. 23–43.
- PARILLO, P. A. (2003) Semidefinite programming relaxation for semialgebraic problems. *Math. Program. Ser. B*, **96**, 293–320.
- PLATZER, A., QUESEL, J.-D. & RÜMMER, P. (2009) Real world verification. *Proceedings of International Conference on Automated Deduction*. Berlin: Springer, pp. 485–501.
- PODELSKI, A. & RYBALCHENKO, A. (2004) *A Complete Method for the Synthesis of Linear Ranking Functions*, Lecture Notes in Computer Science 2937. Berlin: Springer, pp. 239–251.
- POWERS, V. & REZNICK, B. (2000) Polynomials that are positive on an interval. *Trans. Amer. Maths. Soc.*, **352**, 4677–4692.
- PUTINAR, M. (1993) Positive polynomials on compact semi-algebraic sets. *Indiana Univ. Math.*, **42**, 969–984.
- RATSHAN, S. & SHE, Z. (2010) Providing a basin of attraction to a target region of polynomial systems by computation of Lyapunov-like functions. *SIAM J. Control Optim.*, **48**, 4377–4394.
- SANKARANARAYANAN, S., CHEN, X. & ÁBRAHAM, E. (2013) Lyapunov function synthesis using Handelman representations. *IFAC Conference on Nonlinear Control Systems (NOLCOS)*, pp. 576–581.
- SASSI, M. A. B., SANKARANARAYANAN, S., CHEN, X. & ÁBRAHAM, E. (2014) Linear relaxations of polynomial positivity for polynomial Lyapunov function synthesis. arXiv, vol. arXiv:1407.2952 [math.DS].

- SASSI, M. B., TESTYLIER, R., DANG, T. & GIRARD, A. (2012) Reachability analysis of polynomial systems using linear programming relaxations. *Automated Technology for Verification and Analysis*, Thiruvananthapuram, India, pp. 137–151.
- SCHMÜDGEN, K. (1991) The k-moment problem for compact semi-algebraic sets. *Math. Ann.*, **289**, 203–206.
- SHE, Z., LI, H., XUE, B., ZHENG, Z. & XIAB, B. (2013) Discovering polynomial Lyapunov functions for continuous dynamical systems. *J. Symbolic Comput.*, **58**, 41–63.
- SHE, Z., XIAB, B., XIAOB, R. & ZHENG, Z. (2009) A semi-algebraic approach for asymptotic stability analysis. *Nonlinear Anal.: Hybrid Syst.*, **3**, 588–596.
- SHERALI, H. & TUNCBILEK, C. (1991) A global optimization algorithm for polynomial programming using a reformulation linearization technique. *J. Global Optim.*, **2**, 101–112.
- SHERALI, H. & TUNCBILEK, C. (1997) New reformulation linearization/convexification relaxations for univariate and multivariate polynomial programming problems. *Oper. Res. Lett.*, **21**, 1–9.
- SHOR, N. (1987) Class of global minimum bounds on polynomial functions. *Cybernetics*, **23**, 731–734. Originally in Russian: Kibernetika (6), 1987, 9–11.
- TABUADA, P. (2009) *Verification and Control of Hybrid Systems: A Symbolic Approach*. Berlin: Springer.
- TAN, W. & PACKARD, A. (2007) Stability region analysis using sum of squares programming. *Proceedings of the 2006 American Control Conference Minneapolis*, Minnesota, USA, 14–16 June 2006.
- TARSKI, A. (1951) A decision method for elementary algebra and geometry. Tech. rep., Univ. of California Press, Berkeley.
- TIBKEN, B. (2000) Estimation of the domain of attraction for polynomial systems via LMIs. *IEEE CDC*, vol. 4. Sydney, Australia: IEEE Press, pp. 3860–3864.
- TOPCU, U., PACKARD, A., SEILER, P. & WHEELER, T. (2007) Stability region analysis using simulations and sum-of-squares programming. *Proceeding of the ACC*. New York: IEEE Press, pp. 6009–6014.
- VANNELLI, A. & VIDYASAGAR, M. (1985) Maximal Lyapunov functions and domains of attraction for autonomous nonlinear systems. *Automatica*, **21**, 69–80.