Sampling Quotient-Ring Sum-of-Squares Programs for Scalable Verification of Nonlinear Systems

Sampling Quotient-Ring Sum-of-Squares Programs for Scalable Verification of Nonlinear Systems

Shen Shen and Russ Tedrake

***Abstract*— This paper presents a novel method, combining new formulations and sampling, to improve the scalability of sum-of-squares (SOS) programs-based system verification. Region-of-attraction approximation problems are considered for polynomial, polynomial with generalized Lur'e uncertainty, and rational trigonometric multi-rigid-body systems. Our method starts by identifying that Lagrange multipliers, traditionally heavily used for S-procedures, are a major culprit of creating bloated SOS programs. In light of this, we exploit inherent system properties such as continuity, convexity, and implicit algebraic structure, and reformulate the problems as quotient-ring SOS programs, thereby eliminating all the multipliers. These new programs are smaller, sparser, less constrained, yet less conservative. Their computation is further improved by leveraging a recent result on sampling algebraic varieties. Remarkably, solution correctness is guaranteed with just a finite (in practice, very small) number of samples. Altogether, the proposed method can verify systems well beyond the reach of existing SOS-based approaches (29 states); on smaller problems where a baseline is available, it computes tighter solution 2-3 orders faster. Source code is included.**

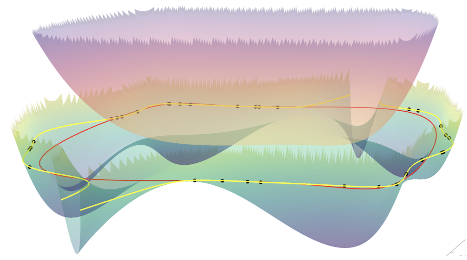


Fig. 1: The proposed method significantly reduces *both* formulation and computation overhead. One resulting improvement is visualized above on the ROA approx. of the van der pol. Traditional methods typically involve conditions on, e.g., the set of all states enclosed within the yellow line, and solving an optimization globally. Our method, *provably* correct and less conservative, only needs to examine few random samples, shown as blue dots, on the yellow line.

I. Introduction

We consider the fundamental verification problem of region-of-attraction (ROA) approximations for polynomial, polynomial with generalized Lur'e uncertainty, and multi-rigid-body systems. Sum-of-squares programs are widely accepted as a standard approach to this problem. Powered by semidefinite programs (SDP)s, SOS provides a systematic way to optimize over polynomial Lyapunov functions' sub-levelsets for these approximation tasks [1], [2].

Despite the popularity and rich theories, the problems solved by these approaches are still of only modest dimension (10-15 states) [3]. This is limiting, as many interesting real-world applications, e.g., mechanical systems consisting of many rigid bodies, are well beyond that scale.

What, then, could be causing the scalability challenge?

Typical scale-improving techniques, rightly so, identifies the low-level SDPs as a computational bottleneck. However, the SDPs are far from the only issue; in fact, we argue that they are, to a large extent, a scapegoat for the inefficient high-level problem formulations.

Specifically, traditional formulations heavily rely on the recipe of (in)equality implication, S-procedure, and auxiliary high-degree Lagrange multipliers. These multipliers not only introduce a large number of auxiliary decision variables and possibly extra expensive constraints, they inflate the problem dimension or degree as well, all of which responsible for creating bloated SDPs. If the dynamics are not exactly polynomial, like the Lur'e-uncertain or rigid-body dynamics that we consider, auxiliary indeterminates are additionally necessary, aggravating the complexity even further [4].

Motivated to eliminate all these multipliers (and most of the auxiliary indeterminates), we exploit inherent system properties – continuity in polynomial, convexity in Lur'e uncertain, and implicit algebraic structure in rigid-body systems – and reformulate the ROA approximation problems as quotient-ring SOS programs. These are programs that directly reason on algebraic varieties (objects defined by polynomial equations; for example, the yellow line in Figure 1 is a variety) without relying on multipliers. Basic algebraic geometry properties imply these reformulated programs are smaller, sparser, less constrained, yet less conservative.

The computation of the new quotient-ring SOS programs is further improved, significantly, by leveraging a sampling algebraic variety approach. The method, recently introduced in [5], reduces a quotient-ring SOS program to sampled instances on the defining variety, resulting in small SDPs with low-rank data and better numerical conditions. Remarkably, solution correctness is guaranteed with just a finite (in practice, very small) number of such samples.

Combining the new formulations and sampling, the proposed method can verify systems well beyond the reach of existing SOS-based approaches (29 states). On smaller problems where a baseline is available, it computes tighter solution 2-3 orders faster.

Finally, while this paper focuses on ROA verification, extensions to the closely related problems such as reachability analysis or barrier certification [2] are immediate.

The authors are with the Computer Science and Artificial Intelligence Lab at MIT. Email: {shenshen, russt}@mit.edu.

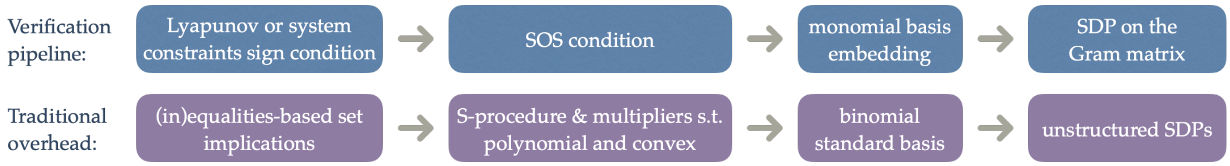


Fig. 2: Standard SOS-based verification pipeline and the traditional overhead. We follow the same pipeline but use different ingredients *throughout*. Thus, unlike most scale-improving methods that are SDP-oriented, we reduce all these overhead.

Our general contributions are:

(i) We present three new quotient-ring SOS programs, one for each ROA approximation problems considered. Different inherent system structures are exploited, all leading to *smaller* yet *stronger* formulations.

(ii) We apply the efficient sampling variety approach from polynomial optimization to the context of general nonlinear system verification.

To our knowledge, all of these are proposed for the first time.

A. Related Work

SOS programs have had success in verification for a wide variety of systems and tasks, from polynomial to hybrid, deterministic to stochastic, and stability to robustness to safety [1], [2], [6], [7], [4], and not only theoretically but demonstrated on hardware [3] as well.

All these work essentially follow a standard pipeline, as illustrated in Fig. 2. While we follow the same pipeline, our ingredients differ from the beginning, e.g. our Lyapunov or Lur'e conditions are not the usual inequality implications. These new conditions lead us to smaller yet stronger quotient-ring SOS programs; multipliers, traditionally needed to segue to conformed convex and polynomial but bloated programs, are thereby eliminated entirely.

These new quotient-ring SOS programs also allow us to take advantage of [5] and improve the downstream computation differently. Existing methods commonly assume special structures such as compositional [8], low-rank solution [9], or chordal sparse [10] in SDPs, or symmetry or sparsity in polynomials [11], [12]. Other methods, while general, either approximate the semidefinite cone with linear or second-order cones [3], or rely on first-order methods such as the augmented Lagrangian [13]; scalability are therefore achieved at the cost of conservatism or accuracy.

In contrast, via sampling, [5] exploits the inherent geometric structure in our quotient-ring SOS programs. It constructs orthogonal and low-dimensional (implicit) Gröbner basis, and produces SDPs that are small, better conditioned, and of low-rank (data, not solution). Remarkably, efficiency is significantly increased, without sacrificing the program's generality, correctness, and less-conservatism.

II. Problem Statement and Approach

Given a continuous-time closed-loop nonlinear system with dynamics $\dot{x} = f(x)$ and a fixed positive definite polynomial Lyapunov candidate $V(x)$, we consider the task of quantitatively verifying if the system is locally asymptotically stable around the fixed point (assumed to be the origin). Concretely, we are interested in finding a sub-levelset $\mathcal{E}(V,\rho) := \{x \mid V(x) < \rho\}$ whose volume grows with ρ. The connected component of $\mathcal{E}$ that includes the origin is an inner approximation of the ROA if the constraint in

$$\begin{aligned} \max \quad & \rho \\ \text{s.t.} \quad & \dot{V}(x) = \frac{\partial V}{\partial x} f(x) < 0, \ \forall x \in \mathcal{E}(V,\rho) \setminus \{0\}, \end{aligned} \tag{1}$$

is satisfied. The cost on ρ encourages enlarging the sub-levelset, thus providing a tighter approximation.

We consider solving the ROA approximation on three sub-problems; they differ in the dynamics characterization.

Polynomial problem: the "vanilla" case where $f(x)$ is polynomial in x.

Lur'e problem: The dynamics is a nominally polynomial $f_0(x)$ subject to additive uncertainty

$$f(x) = f_0(x) + \delta(x),$$

where $\delta(x)$ satisfies a generalized Lur'e type condition $(\alpha(x) - \delta(x))(\beta(x) - \delta(x)) \leq 0$ with $\alpha(x)$ and $\beta(x)$ both polynomial in x.

Rigid-body problem: The dynamics of rigid-body mechanical system, which come from the equations of motion, are given as

$$f(x) = M^{-1}(x)F(x),$$

where both $M(x)$ and $F(x)$ include terms like $\sin(x)$, thus $f(x)$ is rational trigonometric.

The overall approach in this paper is two-pronged: reformulate the three ROA verification problems as simpler yet stronger quotient-ring SOS programs, and apply the efficient sampling algebraic variety method to solve them.

We begin by describing the complete solution to the polynomial problem. For Lur'e and rigid-body problems, we focus on illustrating their tailor-made formulations only, as the sampling subroutine is identically applied.

III. Formulation - Polynomial Problem

A. Existing formulations

There are two known SOS programs formulations for the polynomial problem. The more popular one, which we call program (IE), is based on a straightforward inequality implication $V \leq \rho \Rightarrow \dot{V} \leq 0$, S-procedure and multipliers:

$$\text{(IE)} \quad \begin{aligned} \text{find} \quad & \lambda(x) \\ \text{s.t.} \quad & \lambda(x)(V(x) - \rho) - \dot{V}(x) \text{ is SOS} \\ & \lambda(x) \text{ is SOS} \end{aligned}$$

Note that, this is a feasibility program and requires a line-search of a fixed ρ since otherwise the program would be bilinear (non-convex) in ρ and λ.

An alternative equality constrained formulation can be found in [1], [3]. In particular, under the assumption that the Hessian of $\dot{V}$ is negative definite at the origin, the following is also sufficient for problem Eq. (1):

$$\text{(E)} \quad \begin{aligned} \max_{\rho, Q, \lambda(x)} \quad & \rho \\ \text{s.t.} \quad & (x'x)^d (V(x) - \rho) - \lambda(x)\dot{V}(x) = \dots \\ & m'(x)Qm(x)\ \forall x \end{aligned}$$

Here we explicitly write out the SOS factorization constraint on the right-hand side (for easy reference later); $m(x)$ denotes the standard monomial basis of appropriate degree.

Both formulations need to optimize over auxiliary multipliers $\lambda(x)$. When the λ are of the same degree choices, the SOS programs translate to SDPs of similar dimension and lead to similar optimal ρ. However, the equality constrained (E) is much simpler to solve due to the elimination of the SOS condition on the multipliers and the line-search.

B. Proposed formulation

The proposed formulation is closely related to (E). However, since it was given in the references without proof, some important and subtle questions were left un-addressed. For example, what is the formulation based on? and what is the purpose of the $(x'x)^d$ term? To answer these, we first reverse-engineer the formulation to discover its underlying implication, described below.

Theorem 1: Under the assumption that the Hessian of $\dot{V}$ is negative definite at the origin, the implication condition

$$\dot{V}(x) = 0 \Rightarrow V(x) \geq \rho \text{ or } x = 0 \tag{2}$$

is a sufficient condition for Eq. (1).

Proof: The Hessian condition ensures that $\dot{V}(0) = 0$ is a local maximum. Therefore, locally around the origin, we must have $\dot{V} < 0$. If $\dot{V}$ is negative definite, the system is globally asymptotically stable, and Eq. (2) gives $\rho = \infty$ which in turn correctly implies Eq. (1). The more interesting case is when the system is locally stable, implying that at some states $\dot{V} > 0$. Since V and f are both polynomial by assumption, so is $\dot{V}(x)$ and it is thus continuous. Given this continuity, and that $\dot{V}$ changes sign eventually, zero-crossing event(s) must have occurred at some states.

If at all such states where $\dot{V}(x) = 0$, the evaluation of $V \geq 0$ or it is precisely the origin, as encoded by Eq. (2), then by contraposition, it is equivalent to

$$x \in \{x \mid V(x) < \rho, x \neq 0\} \Rightarrow \dot{V}(x) \neq 0$$

Given the local behavior of $\dot{V}$ around the origin, the connected component of the ρ sub-levelset that includes the origin, must have $\dot{V} < 0$ (except for the origin itself). ∎

An interactive visualization of the proof idea is available online[1]. Figure 1 shows a snapshot of it, where the yellow line precisely defines those important non-origin zero-crossings $\dot{V}(x) = 0$.

[1] http://web.mit.edu/shenshen/www/VDP-animation.html

With Theorem1 in place, it should be clear that Formulation (E) is a multiplier-based sufficient condition for Eq. (2), therefore sufficient for Eq. (1) as well. Note the importance of the negative definite Hessian condition[2], it sufficiently implies the local maximum condition needed in the proof. Note also the importance of the $(x'x)^d$ term, where d is a strictly positive integer user chooses. Without this term, the optimization (E) is meaningless because it would always return the trivial solution $\rho = 0$. To see this, plug in $x = 0$. The left-hand side become $0 - \rho - 0$ which has to match a non-negative right-hand side; the maximal value of ρ must be zero.

Our formulation is a direct application of algebraic geometry on Eq. (2), using basic objects such as affine variety, quotient ring, and Gröbner basis (due to space limitation, we prioritize making the high-level idea clear, and refer to [14] Chapter 1 for the background and definitions).

In particular, simply by defining an algebraic variety $\mathcal{V} := \{x | \dot{V}(x) = 0\}$, a sufficient condition to Eq. (2) is given by the following quotient-ring SOS program (Q):

$$\text{(Q)} \quad \begin{aligned} \max_{\rho, Q} \quad & \rho \\ \text{s.t.} \quad & (x'x)^d (V - \rho) = n'(x)Qn(x),\ \forall x \in \mathcal{V} \end{aligned}$$

where $n(x)$ is a Gröbner basis.

(Q) and (E) may seem trivially equivalent and only differ in terminology; after all, they stem from the same high-level polynomial equality constraint Eq. (2). However, there are four facts that make the reformulation (Q) more appealing.

(i) The decision variable $\lambda(x)$ is eliminated
(ii) The basis $n(x)$ in (Q) is of lower dimension than $m(x)$ in (E), due to Gröbner basis (see [14], Chapter 2);
(iii) The fixed degree d can be lower in (Q), due to the elimination of the $\lambda(x)\dot{V}(x)$ term;
(iv) (Q) is intrinsically stronger than (E), i.e., optimal solution of (E) is in general only suboptimal to (Q).

The last fact is important but subtle. It is due to that (E) relies on degree-bounded multiplier whereas (Q) relies on geometric description of the variety. An explicit example to make this distinction clear:

Degree-bounded multipliers are "bounded": Suppose we need to check if this implication $x+1 = 0 \Rightarrow x^2 - 1 \leq 0$ is true. Multiplier-based formulation would search for a $\lambda(x)$ such that $(x^2 - 1) + \lambda(x)(x + 1) \leq 0, \forall x$. This optimization can not be feasible if λ is limited to be a constant, even though the implication is true. It takes at least an affine multiplier, for example $\lambda(x) = -(x-1)$ to make the problem feasible. In contrast, quotient-ring formulation interprets the left-hand side of the implication as $x \equiv -1$, so the right-hand side becomes $1^2 - 1 = 0 \leq 0$ which is trivially true.

To recap, facts (i)-(iii) mean that the quotient-ring SOS program (Q) leads to a much smaller SDP; yet it is also stronger (unless the multipliers can be of infinite degree) due to fact (iv). Therefore, (Q) is a strictly better formulation, in theory. The only downside, in practice, is that Gröbner

[2] even though one reference does not make this assumption explicit and the other has a sign flip

basis themselves may be challenging to find, especially when the defining equations for the variety get complicated or high-dimensional. To overcome this potential difficulty, we leverage an efficient sampling-based method.

IV. Sampling on Algebraic Varieties

We apply the sampling algebraic varieties method introduced in [5] to solve the quotient-ring problem (Q). The high-level idea is rather straightforward: instead of solving the optimization for all real-valued x with Gröbner basis, solve it at only a set of sampled numerical instances $\{x_i\}$:

$$\text{(S)} \quad \begin{aligned} \max_{\rho, Q} \quad & \rho \\ \text{s.t.} \quad & \dot{V}(x_i) = 0, \ \forall x_i \\ & (x_i' x_i)^d \left(V(x_i) - \rho\right) = \tilde{n}'(x_i) Q \tilde{n}(x_i), \ \forall x_i \end{aligned}$$

using $\tilde{n}(\cdot)$, which *can be* a standard basis or an implicit Gröbner basis; this is to be described later.

As hinted, there are certain numerical benefits of solving the sampled version (S). But given that the ultimate goal is to produce stability certificate, we should be immediately asking: a solution to (S) is necessarily a solution to (Q), does there exist guarantee regarding sufficiency (as required to claim correctness)? Also, what is the sample complexity and sampling procedure? The detailed answers and rigorous treatments can be found in [5], we include a brief high-level summary for completeness.

A. Correctness and sample bound

The sampled program (S) is equivalent to the original program (Q), with probability one, if the samples x_i are generic. The genericity condition can be interpreted as checking if *enough* samples are drawn *randomly*. In theory, there exists a *finite* sample bound. This bound depends on many factors including the problem size, variety structure, etc. Concretely, genericity is checked by a simple rank test of a matrix whose elements are simple monomial evaluations at the samples.

Through this practical case-by-case numerical rank check, we accumulated enough empirical evidence that the samples needed are in fact, very small. Usually, this number is far less than the number of elements in the Gram matrix. In Section VII, we document the number of samples used for each program, which could serve as an empirical reference.

An intuitive explanation might be helpful; after all, "with probability one guarantee" is usually stated in the asymptotic regime. Intuitively, the combination of "being exactly on the variety" and "degree-bounded polynomial parameterization" imposes a constraint so strong that finite samples are capable of capturing. To some extend, it is similar to polynomial interpolation, where a finite number of samples can faithfully determine the coefficients of a degree-bounded polynomial.

We finally point out that, the sampling procedure itself involves finding roots to polynomial equation(s). In the simple case where dynamics itself is polynomial, sampling means finding roots of a single multi-variate polynomial $\dot{V}$, which can be easily done via open-source packages (in our case, we use shooting and numpy). As the variety gets more complicated (usually when having more defining equations), so will the sampling process. Fortunately, sampling is a trivially parallelizable process, where each thread only comes with very low processing and memory requirement.

B. Computational benefits.

The computational gains come from the paradigm shift: whereas traditional methods match polynomial coefficients, sampled approach matches polynomial evaluations.

One direct consequence is the low-dimensional numerical basis $\tilde{n}(x_i)$. First, this basis *can* be chosen as *standard* monomials evaluations, because the generic samples numerically capture the underlying variety. In comparison, problem (Q) must rely on explicit Gröbner basis to symbolically encode the variety. This in and of itself is a huge improvement.

$\tilde{n}(x)$ can be further simplified (from e.g. the standard basis) to an even lower-dimensional implicit Gröbner basis by leveraging the underlying variety and a simple SVD procedure. These implicit basis can be thought of as the orthogonal basis with respect to a natural inner product supported on the samples. Orthogonalization, as a byproduct of this size-reduction procedure, has been shown to improve SDP numerical condition as well [15].

Finally, (S) results in an SDP with low-rank data structures, which can be readily exploited by solvers. Note that the right-hand side of (S) is a scalar evaluation. Via the trace cyclic property, $\tilde{n}'(x_i)Q\tilde{n}(x_i) = \operatorname{tr}(Q, \tilde{n}(x_i)\tilde{n}'(x_i))$, where $\tilde{n}(x_i)\tilde{n}'(x_i)$, the problem data in the SDP, is of rank at most one by construction (because recall that $\tilde{n}(x_i)$ is a numerical vector). Such low-rank data does not appear in traditional SOS programs, since $n(x)$ there are symbolic monomials. Note that it is the problem data (rather than the decision variable Q) that is of low-rank.

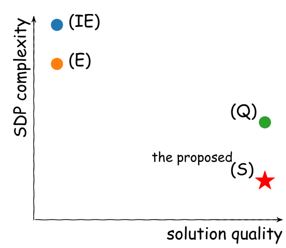


Fig. 3: Qualitative comparison of the four programs.

Comparison of the four SOS programs.: We have presented four different SOS programs for the polynomial ROA problem. Figure 3 summarizes a qualitative comparison of the solution quality and underlying SDP complexity. Note that the relative scale of the gap varies case-by-case. For example, on very simple problems, all programs might be overkill and arrive at the same solution. On the other hand, the proposed method achieves more significant computational gain for more complex systems; Section VII includes these quantitative comparisons.

V. Formulation - Lur'e Problem

Consider dynamics with generalized Lur'e uncertainties:

$$f(x) = f_0(x) + \delta(x) \tag{3}$$

where $f_0(x)$ is the nominal polynomial dynamics; the uncertainty $\delta(x)$ satisfies $(\alpha(x) - \delta(x))(\beta(x) - \delta(x)) \leq 0$, where $\alpha(x)$ and $\beta(x)$ are both polynomial (generalized from the standard linear). A one-dimensional example of $\delta(x)$ is visualized in Figure 4.

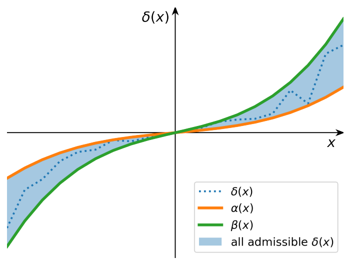


Fig. 4: Generalized Lur'e type sector uncertainty. $\alpha(x)$ and $\beta(x)$, both polynomial, define the "boundaries" of the sector; the uncertainty $\delta(x)$ can take any function "in between".

A. *Existing formulation*

For Lur'e problem, we would like to eliminate a standard S-procedure dedicated to encoding the uncertainty δ. This is a separate issue / overhead from those arising from encoding sub-levelset (discussed in Section III). To isolate the two and highlight the new improvements here, we first present the standard and proposed formulations for global analysis. Local extension is discussed later.

The standard way to verify global asymptotically stable (g.a.s.) is via an IQC-type treatment [16]:

$$\begin{aligned} \text{find} \quad & \xi(x,\delta) \\ \text{s.t.} \quad & \xi(x,\delta) > 0 \\ & \underbrace{\frac{\partial V}{\partial x}(f_0(x) + \delta)}_{\dot{V}(x,\delta)} - \underbrace{\xi(x,\delta)(\alpha(x) - \delta)(\beta(x) - \delta)}_{\text{S-procedure encoding } (x,\,\delta) \text{ dependency}} < 0, \end{aligned} \tag{4}$$

where δ is an auxiliary indeterminate, independent from x (thus the notation $\dot{V}(x,\delta)$); its true dependency on x is incorporated using multiplier $\xi(x,\delta)$ and the S-procedure.

B. *Proposed formulation*

The proposed formulation is *simpler* yet *stronger* (in aspect different from quotient-ring structure): it eliminates the auxiliary multiplier $\xi(x,\delta)$ and indeterminate δ, and allow us to analyze all admissible dynamics f (defined in Eq. (3)) by examining the boundaries $f_\alpha := f_0(x) + \alpha(x)$ and $f_\beta := f_0(x) + \beta(x)$ with a less conservative condition.

Lemma 1: For a given positive definite $V(x)$, define

$$\begin{cases} \dot{V}_\alpha(x) := \frac{\partial V(x)}{\partial x} f_\alpha < 0 \\ \dot{V}_\beta(x) := \frac{\partial V(x)}{\partial x} f_\beta < 0, \end{cases} \tag{5}$$

then (4) $\Longrightarrow$ (5) $\Longrightarrow$ g.a.s. $\not\Longleftarrow$ (5) $\not\Longleftarrow$ (4) (slight abuse of notation here, (4) denotes that the optimization is feasible).

Proof: (4) $\Longrightarrow$ (5): If (4) holds, it holds for all admissible δ. It must hold when $\delta = \alpha(x)$, plug this in and (4) reduces to exactly $\dot{V}_\alpha < 0$. Similarly $\dot{V}_\beta < 0$ is implied.

(5) $\Longrightarrow$ g.a.s.: First, note that for any *fixed* x, $\delta(x)$ can be written as a *convex* combination of $\alpha(x)$ and $\beta(x)$. Second, $\dot{V}(x) = \frac{\partial V}{\partial x}(f_0(x) + \delta(x))$ is linear with respect to $\delta(x)$. Combining the two observations, $\dot{V}$ can also be written as a convex combination of $\dot{V}_\beta$ and $\dot{V}_\alpha$ (with generally state-dependent combination coefficients). Therefore, (5) implies $\dot{V}(x) < 0$ for all admissible f.

g.a.s $\not\Longleftarrow$ (5): Well-known fact. Finally, (5) $\not\Longleftarrow$ (4) is a consequence of the multiplier limitation (Section III). ■

Lemma 1 shows that (5) is *stronger* than (4). Also, in terms of computation, not only is the multiplier $\xi(x,\delta)$ and S-procedure eliminated, so is the auxiliary indeterminate δ. These simplifications are preserved, and also combined with those in Section III (e.g. elimination of multipliers λ) when extended to local analysis via the following.

Theorem 2: Given a positive definite $V(x)$, if program (S) is feasible for dynamics f_α and f_β, with optimal solutions ρ_α and ρ_β, then $\dot{V}(x) < 0$, $\forall x \in \mathcal{E}\left(V, \underline{\rho}\right) := \{x \mid V(x) \leq \min(\rho_\alpha, \rho_\beta)\}$, and for all admissible f defined by Eq. (3).

Proof: Given Theorem 1, the optimal solutions of (S) imply that $\dot{V}_\alpha(x) < 0$, $\forall x \in \mathcal{E}\left(V, \underline{\rho}\right)$, and similarly for $\dot{V}_\beta$. For all admissible f, $\dot{V}(x) < 0$ within this set can be almost identically proved as "(5) $\Longrightarrow$ g.a.s." part in Lemma 1. ■

Finally, the new formulation affords additional high-level insights into application too; details in Section VII-B.

VI. Formulation - Rigid-body Problem

Multi-body rigid-body systems are challenging to analyze, not only because of their often large scale, but more importantly because the Equations of Motions (EoM) and rotation and potential energy make their dynamics complicated rational trigonometric. Shown in Figure 5 is an N-link pendulum on a cart system, whose dynamics exhibit such characteristic. We use it to illustrate more explicitly the challenge and the existing and proposed approach.

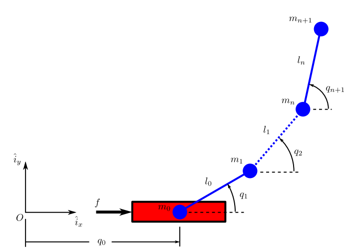


Fig. 5: N-link pendulum on a cart. [3]

Without loss of generality, assume $N = 1$ (the classical cart-pole). Also, suppose the cart and pole both have unit mass and the pole has unit length.

Let the states be $y := [q_0, q_1, \dot{q}_0, \dot{q}_1]$, where q_0 is the cart position, q_1 the pole angle, and $\dot{q}_0, \dot{q}_1$ the corresponding velocities. Let the force on the cart be u and the gravitational

[3] figure taken from the python PyDy package.

constant be g. The EoM is:

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & -\sin q_1 \\ 0 & 0 & -\sin q_1 & 1 \end{bmatrix}}_{M(y)} \underbrace{\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \ddot{q}_0 \\ \ddot{q}_1 \end{bmatrix}}_{\dot{y}} = \underbrace{\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_1^2 \cos q_1 + u \\ -g \cos q_1 \end{bmatrix}}_{F(y,u)} \tag{6}$$

where $M(y)$ is the (positive definite) mass matrix, $F(y, u)$ the force matrix, and $\dot{y}$ the dynamics. Since $\dot{y} = M^{-1}F$, the dynamics is explicitly rational trigonometric.

A. *Existing formulations*

Taylor expansion is commonly used to handle the rational trigonometric dynamics. This approach has two major limitations. One is the error; or the complication introduced by error-bounding. The other limitation is Taylor expansion's own scalability; expansion can be challenging when the dimension or degree gets non-trivial. For example, to expand the dynamics of the N-linked cart example to third order, Sympy fails at link 4, whereas Matlab fails at link 5. If higher-orders are needed to reduce the error, the scalability is even worse.

A "less lossy" transformation-type technique based on [17] has also been applied [4], [7]. This technique deals with the dynamics in two steps. First, a change-of-variables recasting technique can turn the trigonometric components into polynomials and the dynamics are simplified as rationals. The second step is to clear the rationals' denominator, perhaps not surprising by now, via multipliers, which carry all the complications discussed so far.

B. *Proposed formulation*

The proposed method to deal with the rational trigonometric dynamics is a combination of change of variable, differential algebraic equations (DAE), and implicit algebraic variety.

We start with the standard change of variable. Let $x := [q_0, s_1, c_1, \dot{q}_0, \dot{q}_1]$ where $s_1 \equiv \sin q_1$ and $c_1 \equiv \cos q_1$; suppose a feedback controller $u(x)$ is given to close the loop. The new coordinates first must satisfy the unit circle condition $s_1^2 + c_1^2 = 1$, which is equivalent to $x'Sx = 1$ where $S = \text{diag}(0, 1, 1, 0, 0)$.

Secondly, simple variable substitution of the recast states x into the EoM Eq. (6) gives $M(x)\dot{y} = F(x)$. This is the key step, where we do not explicitly write out the $\dot{y}$ but rather leave it *implicit*. In other words, instead of dealing with ordinary differential equations, we describe the dynamic via DAEs, which are in fact more general as well.

Thirdly, due to the coordinate transformation, the dynamics of the recast $\dot{x}$ follows a transformation from $\dot{y}$ as:

$$\dot{x} = \begin{bmatrix} \dot{q}_0 \\ \dot{s}_1 \\ \dot{c}_1 \\ \ddot{q}_0 \\ \ddot{q}_1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_1 & 0 & 0 \\ 0 & -s_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{T(x)} \underbrace{\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \ddot{q}_0 \\ \ddot{q}_1 \end{bmatrix}}_{\dot{y}} = T(x)\dot{y}$$

where $T(x)$ is the recasting transformation matrix (purely dependent on x). Consequentially, the derivative of the Lyapunov function is then

$$\dot{V}(x) = \frac{\partial V}{\partial x}\dot{x} = \frac{\partial V}{\partial x} T(x)\dot{y} \tag{7}$$

With these preparations, sampling quotient-ring can be now readily applied. In the 'vanilla' case where the dynamics is polynomial, the variety has only one component that is $\dot{V} = 0$. Here, due to the recasting, the variety $\mathcal{V}$ is more involved $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2 \cup \mathcal{V}_3$ where the defining equations are:

$$x'Sx = 1 \tag{8a}$$

$$M(x)\dot{y} = F(x) \tag{8b}$$

$$\frac{\partial V(x)}{\partial x} T(x)\dot{y} = 0 \tag{8c}$$

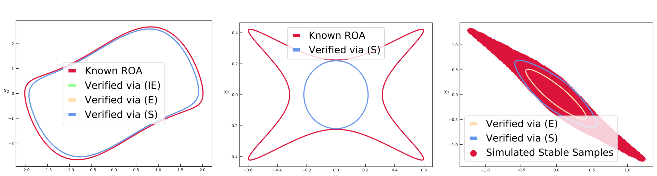
Note that $\dot{y} = [\dot{q}_0, \dot{q}_1, \ddot{q}_0, \ddot{q}_1]$ may seem necessary to be included in the SOS program basis. However, because the first half of the elements (first-order derivatives of q) is included in x, whereas the second half (second-order derivatives of q) is otherwise independent of x, the numerical samples via the variety capture all dependencies between $\dot{y}$ and x. In other words, it is not necessary for $\dot{y}$ to appear explicitly in the basis, which is another major advantage over multiplier-based formulation.

VII. Examples

A. *Polynomial problems*

We first consider three polynomial systems: Van der Pol oscillator, Ninja star, and Pendubot. Programs (IE), (E), and (S) are compared. Across all three examples, the proposed method (S) demonstrates speed improvement of up to 2-3 orders (Table I). On Pendubot and Ninja star example, the proposed method also produces better solutions.

Van der Pol: (Time-reversed) Van der Pol is a 2 state, degree 3 polynomial systems. It has a known ROA, and has thus been a staple benchmark. Using a degree 6 candidate V, all programs produce almost tight approximation (Fig. 6a), though the proposed method is the fastest.



(a) Van der Pol (b) Ninja Star (c) Pendubot

Fig. 6: ROA approximations of polynomial systems. Qualitatively, for Van der pol, all three programs return identical result; for Ninja star, only the proposed method (S) succeeds; for Pendubot, the proposed method is tighter.

	Van der Pol			Ninja Star			Pendubot		
	(IE)	(E)	(S)	(IE)	(E)	(S)	(IE)	(E)	(S)
PSD variable dim	45	45	15	220	220	55	495	495	70
num. scalar var.	46	46	1	221	221	1	496	496	1
num. constraints[a]	165	152	29	540	632	78	4844	4840	118
time (sec)	0.09	0.07	0.01	err[b]	err	0.13	err	217.96	0.33

[a]equals the number of samples on the variety for method (S)
[b]"err" indicates the solver encounters numerical error

TABLE I: Numerical comparison of three methods for ROA verification.

Ninja star: To showcase the efficacy of our method in numerically challenging situations, we purposely create a system with "badly conditioned" dynamics[4]. The system, with known true ROA that resembles a Ninja star shown in Figure 6, is low dimensional (2 states) but of high degree (7 degree). Further, the coefficients are very unbalanced (relative scale difference is 10^3); and the dynamics linearization A matrix at the origin are precisely zero. We supply a unit quadratic function $V = x'x$, therefore coefficients of V and $\dot{V}$ are even more unbalanced than the those in the dynamics. Among the three programs, only the proposed succeeds at producing a result (and the result is tight).

Pendubot: We take from [18] an LQR-controlled four dimensional Pendubot system, Taylor expanded to degree three at the fixed point. A degree six Lyapunov function is provided. Figure 6 shows the produced approximation at slice (x_1, x_3). Our method not only produces better approximation, it does so about 10^3 times faster (Table I).

B. Lur'e problem - Path-tracking Dubins Vehicle

Consider a Dubins car defined in the error frame relative to the virtual vehicle along a path to be tracked, illustrated in Figure 7. The model is: $\dot{\psi}_E = u_1 - k_v\ell, \dot{X}_E = u_1 Y_E + u_2 - \ell\cos\psi_E, \dot{Y}_E = -u_1 X_E + \ell\sin\psi_E$ where $\psi_E, \dot{X}_E, \dot{Y}_E$ are the angle error and linear displacements, l and k_v are the target speed and path curvature, and u_1 and u_2 are the angular and linear torques. Stabilization at zero error means the car achieves perfect tracking.

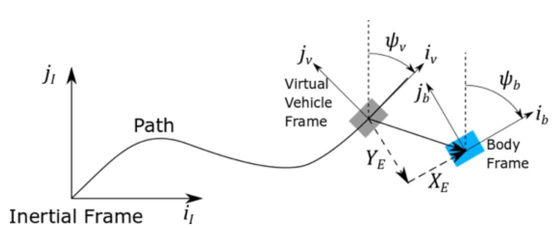


Fig. 7: Path-tracking Dubins vehicle in the virtual error frame

An LQR controller is designed for a constant nominal tracking curvature $k_v = 1$. The true curvature can be between $[0.8, 1.2]$, and potentially time-varying. The task is to find an ROA approximation robust to this run-time parameter variation. Formulation presented in Section V is applied to this problem, which allows us to verify the robust ROA by checking the two extreme cases. The result is shown in Figure 8.

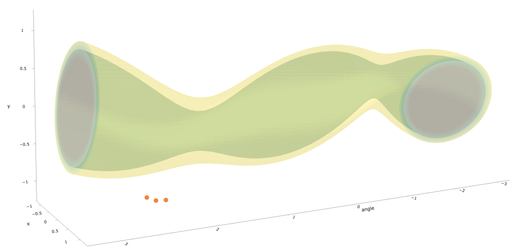


Fig. 8: Robust ROA analysis for Dubins vehicle tracking a path of varying curvature. The yellow outer tube corresponds to $k_v = 0.8$ (straighter path). The green inner-tube corresponds to $k_v = 1.2$. The inner-tube is also the robust ROA for any k_v varying within the given range. The red dots are counter-examples that do not converge to the origin; they show the tightness of the approximation.

We would like to point out an extra bit of insight the proposed method affords (which the existing formulation does not). Comparing the two ROAs produced for the extreme cases, the one corresponding to a straighter $k_v = 0.8$ path has larger volume than $k_v = 1.2$. In fact, we approximated the ROA for the nominal constant curvature $k_v = 1$ (note that this is not required for the robust ROA per se), and the result is sandwiched in between the two shown in Fig. 8 (so smaller than straighter-path one), even though the controller is designed based on this parameter. These findings agree with our intuition that tracking a "curvier" path feels dynamically more demanding, and it is interesting to have it, as a byproduct, emerging from the result.

C. Rigid-body problem - Cart with N-link Pole

Consider the problem of N-link pendulum on a cart, illustrated in Figure 5. The system states are the position and (angular) velocity of the cart and the N links. There are N inputs, one is a force applied on the cart, and the rest are torques applied on the $N-1$ links, starting from the attached on the cart (farthest one from the cart is not controlled directly). The task is to balance all the links upright.

We first produce an LQR controller and a quadratic Lyapunov function in the original coordinate. Then with

[4]Ninja Star dynamics: $\dot{x}_1/16 = -25x_1^3 + 2500x_1^7 + 48x_2^2x_1 - 14400x_2^2x_1^5 + 28432x_2^4x_1^3 - 19200x_2^6x_1; \dot{x}_2/16 = -100x_2^3 + 40000x_2^7 + 48x_2x_1^2 - 4800x_2x_1^6 + 28432x_2^3x_1^4 - 57600x_2^5x_1^2$

Link	1	2	3	4	5	6	7	8	9
$\dim(x)$	5	8	11	14	17	20	23	26	29
$\dim Q^a$	21	45	78	120	171	231	300	378	465
Constraints^b	45	89	179	298	390	531	690	890	1390
time (sec)	0.03	0.48	23.67	91.87	178.78	338.56	478.37	598.20	723.03

^{a}The matrix variable in the SDP is $\dim Q$-by-$\dim Q$.
^{b}The same as the number of samples.

TABLE II: Numerical results of the ROA problem with different number of links on the cart.

small-angle approximations, and techniques introduced Section VI, transform them all into the recast x coordinate. Figure 9 shows the ever-growing complexity disparity for the proposed method and its multiplier-based counterpart (E). Applying techniques detailed in, we are able to verify system of dimension 29, well beyond the scale of current SOS-based method. The numerical comparisons for different N are documented in Table II.

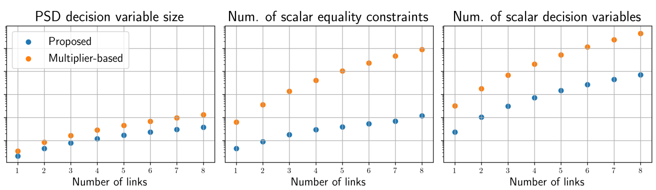


Fig. 9: SDP complexity as the number of links in the N-link on cart system Figure 5 grows. Note the log-scale.

VIII. Conclusion

We presented a novel framework, combining smaller yet stronger problem reformulations and sampling, to address the scalability issues of SOS-based verification. The three new ROA formulations each relies on (different) intrinsic system structures, and are thus general. The subsequent quotient-ring SOS programs leverage geometric problem description rather than algebraic and are thus smaller, sparser, less constrained, yet less conservative. Their computation is further improved via the application of sampling variety method. Altogether, scale and speed are significantly improved.

Future work includes extending the techniques to more applications. A direct example is the multi-contact example described in [4]. Where the mass matrix can be handled via DAE technique as described in Section VI, contact condition can be simplified as described in SectionV.

An important related question we did not elaborate on here (due to space limitation) is how to find a high-quality Lyapunov candidate to feed into this new verification pipeline. This is the motivation of a sister project, which we will discuss in an upcoming paper.

The source codes is at https://github.com/shensqured/S4VC; it includes all the algorithms and examples presented.

Acknowledgment

We thank Diego Cifuentes for tutorials and many helpful discussions of the sampling variety approach, and Pablo Parrilo for introducing to us this method. We thank Micah Fry for providing the Dubins vehicle model and feedback on the paper drafts. This work was supported by ONR award N00014-18-1-2210 and ONR-MURI N00014-17-1-2699.

References

[1] P. A. Parrilo, *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, Caltech, 2000.
[2] S. Prajna, *Optimization-based methods for nonlinear and hybrid systems verification*. PhD thesis, Caltech, 2005.
[3] A. Majumdar, *Funnel Libraries for Real-Time Robust Feedback Motion Planning*. PhD thesis, MIT, 2016.
[4] M. Posa, *Optimization for Control and Planning of Multi-contact Dynamic Motion*. PhD thesis, MIT, 2017.
[5] D. Cifuentes and P. A. Parrilo, "Sampling Algebraic Varieties for Sum of Squares Programs," *SIAM Journal on Optimization*, 2017.
[6] J. Steinhardt and R. Tedrake, "Finite-time regional verification of stochastic non-linear systems," *The International Journal of Robotics Research*, 2012.
[7] A. J. Barry, A. Majumdar, and R. Tedrake, "Safety verification of reactive controllers for UAV flight in cluttered environments using barrier certificates," *ICRA*, 2012.
[8] S. Shen and R. Tedrake, "Compositional Verification of Large-Scale Nonlinear Systems via Sums-of-Squares Optimization," in *ACC*, 2017.
[9] S. Burer and R. D. C. Monteiro, "A Nonlinear Programming Algorithm for Solving Semidefinite Programs via Low-Rank Factorization," vol. 357, pp. 329–357, 2003.
[10] R. P. Mason and A. Papachristodoulou, "Chordal Sparsity, Decomposing SDPs and the Lyapunov Equation,"
[11] P. A. Parrilo, "Exploiting Algebraic Structure in Sum of Squares Programs," *LNCIS*, vol. 312, pp. 181–194, 2005.
[12] F. N. Permenter, *Reduction Methods in Semidefinite and Conic Optimization*. PhD thesis, MIT, 2017.
[13] R. T. Rockafellar, "Augmented lagrangians and applications of the proximal point algorithm in convex programming," *Mathematics of Operations Research*, vol. 1, no. 2, pp. 97–116, 1976.
[14] D. Cox, J. Little, and S. Donal, *Ideals, Varieties, and Algorithms*. Springer, third ed., 2000.
[15] J. Löfberg and P. A. Parrilo, "From coefficients to samples: A new approach to SOS optimization," in *IEEE Conference on Decision and Control*, vol. 3, pp. 3154–3159, 2004.
[16] A. Megretski and A. Rantzer, "System analysis via integral quadratic constraints," *IEEE Transactions on Automatic Control*, 1997.
[17] A. Papachristodoulou and S. Prajna, "Analysis of Non-polynomial Systems Using the Sum of Squares Decomposition," 2005.
[18] U. Topcu, A. Packard, and P. J. Seiler, "Local stability analysis using simulations and sum-of-squares programming," *Automatica*, 2008.