Certification of fixed computation time first-order optimization-based controllers for a class of nonlinear dynamical systems

Milan Korda[1], Colin N. Jones[1]

***Abstract*— This paper proposes a stability verification method for systems controlled by an early terminated first-order method (e.g., an MPC problem approximately solved by a fixed number of iterations of the fast gradient method).**

The method is based on the observation that each step of the vast majority of first-order methods is characterized by a Karush-Kuhn-Tucker (KKT) system which (provided that all data are polynomial) is a basic semialgebraic set; M steps of a first-order method is then characterized by a basic semialgebraic set given by the intersection of M coupled KKT systems. Using sum-of-squares techniques, one can then search for a polynomial Lyapunov function that decreases between two consecutive time instances for all control inputs belonging to this coupled KKT system.

The proposed method applies to nonlinear dynamical systems described by polynomial (or trigonometric) data affected by a (possibly state-dependent) disturbance; in particular the method is not restricted to linear systems and/or convex cost functions. To the best of the authors' knowledge, this is the first verification approach for early terminated optimization schemes with this level of generality.

Keywords: First-order methods, model predictive control, early termination, certification.

I. Introduction

Verification of the stability of optimization-based controllers (e.g., MPC) under tight constraints on the computation time is a topic of active research and is crucial for practical deployment of these controllers, especially on embedded platforms.

One way of verifying stability is to first compute the number of iterations of a given optimization method to achieve a given precision ε and then enforce the stability of the closed-loop system under this imprecise control law, typically treating the solution inaccuracy ε as a disturbance. This approach typically suffers from two problems: First, the number of iterations needed to achieve a given precision is difficult to analyze and existing iteration bounds are often conservative and/or computationally difficult to obtain (e.g., requiring the solution of a mixed-integer optimization problem, whose size grows quickly with the input data dimension) [2], [18]. Second, treating the solution inaccuracy ε as a disturbance often leads to conservative robust MPC formulations where stability is enforced *by design*, often at the cost of a significant deterioration of control performance.

Another way of certifying stability of early terminated optimization schemes is to use the fact that the optimization problem underlying a traditional stabilizing MPC setup can be warm-started using a shifted sequence of the previous control inputs and that this warm-starting ensures a Lyapunov decrease. Augmenting the optimization problem by a constraint enforcing the Lyapunov decrease in conjunction with an optimization method ensuring primal feasibility after each iteration warm-started by the shifted sequence then guarantees stability; see [20] for details. This approach, however, assumes that the MPC problem is *designed* in such a way that the closed-loop system is stable when solved to the full accuracy and that the shifted sequence is feasible and ensures a Lyapunov decrease. This premise, although theoretically convenient, is rarely satisfied in practice.

The approach proposed in this paper tackles the problem of stability of early terminated optimization-based controllers from a very different angle – we propose a *verification* procedure that provides computational certificates for the stability of a dynamical system controlled by an optimization-based controller where the underlying optimization problem is solved by a first-order method. In particular we do not a priori assume that the optimization-based controller is stabilizing when solved to full accuracy, let alone possesses any robustness properties.

The proposed approach builds on the idea of Primbs in [14] who noticed that the Karush-Kuhn-Tucker (KKT) system associated to an optimization problem with convex quadratic cost function subject to linear dynamics and polytopic constraints is a basic semialgebraic set[1] and used the S-procedure to seek a quadratic Lyapunov function decreasing under the control law implicitly characterized by the KKT system. This work brings Primbs' approach one step further by characterizing each step of a given first-oder method by a KKT system and searching for a Lyapunov function decreasing on the conjunction of M coupled KKT systems corresponding to M steps of the first-order method. In addition our approach applies not only to linear systems with convex quadratic cost but to all problems with polynomial data; this is possible by replacing the S-procedure by powerful sum-of-squares (SOS) programming techniques. Moreover, the approach readily extends to the case of trigonometric data using sum-of-squares representation of non-negative trigonometric polynomials [6]. To the best of the authors' knowledge, this is the first verification approach for early terminated optimization schemes that applies to nonlinear dynamical systems described by polynomial or trigonometric data with no assumptions of convexity.

The approach is based purely on convex optimization, in fact on semidefinite programming (SDP). The size of

[1]Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne, Station 9, CH-1015, Lausanne, Switzerland. {milan.korda,colin.jones}@epfl.ch

[1]Basic semialgebraic set is the intersection of (sub/super)-level sets of finitely many polynomials.

the SDP problem to be solved depends on the problem dimensions and on the richness of the candidate Lyapunov functions we search over, which is a tuning parameter. The approach is easy to implement using high-level SOS modeling tools (e.g., Yalmip [9], SOSTOOLS [13] or more recently SOSOPT [15]).

II. Problem statement

We consider the nonlinear discrete-time dynamical system

$$x^+ = f(x, u), \tag{1}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ the control input, x^+ the successor state and $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ a *polynomial* transition mapping.

The goal of the paper is to provide computational certificates of stability of the system (1) when controlled by a controller which at each time instant *approximately* solves the optimization problem

$$\begin{array}{ll} \underset{y \in \mathbb{R}^{n_y}}{\text{minimize}} & J(x, y) \\ \text{s.t.} & q(x, y) \leq 0 \\ & q_{\text{eq}}(x, y) = 0, \end{array} \tag{2}$$

where the scalar function $J : \mathbb{R}^n \times \mathbb{R}^{n_y} \to \mathbb{R}$ and the vector-valued functions $q : \mathbb{R}^n \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_q}$ and $h : \mathbb{R}^n \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_{q_{\text{eq}}}}$ are assumed *polynomial*. The optimization problem (2) is purposely stated in a fairly general form so that particularities do not obstruct the main idea. However, in most cases the cost function $J(x, y)$ will correspond to the cost of an MPC problem, the decision variables y to the control inputs along a prediction horizon and the constraints $q(x, y) \geq 0$ and $q_{\text{eq}}(x, y) = 0$ to state and/or input constraints.

Given an (approximate) solution y to problem (2), the control input u applied to the system is given by $u = \kappa(y)$, where $\kappa : \mathbb{R}^{n_y} \to \mathbb{R}^m$ is a polynomial. For instance, if y is the control sequence along the prediction horizon of an MPC problem, then $\kappa(\cdot)$ is simply the selection of the first m components of y.

We focus our attention on the case when the problem (2) is solved *approximately* using (typically a small number of) iterations of a first-order method. Our approach applies to most of the existing first-order methods – the key property a method must satisfy is that each step of such a method is characterized by a system of polynomial equations. To be more specific we detail the approach on the probably simplest method – (primal) projected gradient – and later on comment on the particularities of other common approaches.

At this point we would like to stress that we make no assumptions on the problem data ensuring that a given first-order method converges (to local/global optimum or otherwise), nor do we assume that the underlying optimization problem results in closed-loop stability when solved to the full accuracy. In fact, the optimization problem (2) may have nothing to do with the system to be controlled (although then it is, of course, unlikely to obtain a stable closed-loop behavior).

III. Main idea

When applied to problem (2), the projected gradient method produces a sequence of points given by

$$y_{k+1} = P_{\mathcal{C}}(y_k - \tau \nabla_y J(x, y_k)), \tag{3}$$

where $P_{\mathcal{C}}(\cdot)$ denotes the Euclidean projection operator on the constraint set

$$\mathcal{C} = \{y \in \mathbb{R}^{n_y} \mid q(x, y) \leq 0,\ q_{\text{eq}}(x, y) = 0\}$$

and $\tau > 0$ is a step size.

Equation (3) can be equivalently written as

$$\begin{array}{rcll} y_{k+1} & = & \underset{y \in \mathbb{R}^{n_y}}{\arg\min} & \frac{1}{2}\|y - (y_k - \tau \nabla_y J(x, y_k))\|_2^2 \\ & & \text{s.t.} & q(x, y) \leq 0 \\ & & & q_{\text{eq}}(x, y) = 0. \end{array} \tag{4}$$

We assume that the projected gradient iterations (4) are initialized with $y_0 := \kappa_0(x)$, where $\kappa_0(\cdot)$ is a polynomial (e.g., zero, an LQ controller $K_{\text{lq}}x$, etc.) and that a fixed number $M > 0$ of iterations is performed. After that the control input $\kappa(y_M)$ is applied to the system.

We make the following standing assumption:

Assumption 1. 1) *(Feasibility) The problem (4) is feasible for all $x \in \mathbb{R}^n$ and $y_k \in \mathbb{R}^{n_y}$.*
2) *(Solvability) The minimum in (4) is attained for all $x \in \mathbb{R}^n$ and $y_k \in \mathbb{R}^{n_y}$.*
3) *(Constraint qualification) There exists a set of Lagrange multipliers characterising each global minimum of the problem (4).*

Note that at this level of generality the problem (4) may be a difficult nonconvex optimization problem; however, in most practical cases when the projected gradient method is computationally viable, the constraint set of (4) has a simple form allowing for a fast, often analytical, solution of (4). Note also that Assumption 1 implies that the problem (2) is feasible. This is a consequence of using the simple projected gradient method to explain the approach – more sophisticated first-order methods (e.g., ADMM treated in Section VI-B) allow for problem (2) to be infeasible while still having iterates characterized by an optimization problem satisfying Assumption 1.

Under this assumption, using the Karush-Kuhn-Tucker (KKT) necessary conditions of optimality, the last iterate y_M and therefore the control input $\kappa(y_M)$ are characterized by the following system of polynomial equalities and inequalities

$$\begin{aligned} & y_{k+1} - (y_k - \tau \nabla_y J(x, y_k)) + (\text{Jac}_y\, q(x, y_{k+1}))^T \lambda_{k+1} \\ & \quad + (\text{Jac}_y\, q_{\text{eq}}(x, y_{k+1}))^T \nu_{k+1} = 0 \end{aligned} \tag{5}$$

$$q_{\text{eq}}(x, y_{k+1}) = 0 \tag{6}$$

$$\lambda_{k+1}^T q(x, y_{k+1}) = 0 \tag{7}$$

$$q(x, y_{k+1}) \leq 0 \tag{8}$$

$$\lambda_{k+1} \geq 0, \tag{9}$$

$k = 0, \ldots, M-1$, $y_0 = \kappa_0(x)$. Here $\text{Jac}(\cdot)$ is the Jacobian matrix, and $\lambda_{k+1} \in \mathbb{R}^{n_q}$ and $\nu_{k+1} \in \mathbb{R}^{n_{q_{\text{eq}}}}$ are respectively

the Lagrange multipliers associated to the inequality and equality constraints at step $k+1$. Grouping all optimization variables and Lagrange multipliers together as

$$\boldsymbol{y} = \left[y_1^T, \ldots, y_M^T\right]^T, \quad \boldsymbol{\lambda} = \left[\lambda_1^T, \ldots, \lambda_M^T, \nu_1^T, \ldots, \nu_M^T\right]^T,$$

this system of equations can be written compactly as

$$\boldsymbol{h}(x, \boldsymbol{y}, \boldsymbol{\lambda}) = 0 \tag{10}$$

$$\boldsymbol{g}(x, \boldsymbol{y}, \boldsymbol{\lambda}) \leq 0, \tag{11}$$

for some polynomials $\boldsymbol{h}$ and $\boldsymbol{g}$.

Thus, the closed-loop evolution under the control given by M iterations of the projected gradient method is

$$x^+ = f(x, \kappa(y_M)) \tag{12}$$

subject to the polynomial equalities and inequalities (10) and (11).

A sufficient condition for the closed-loop system (12) to be globally asymptotically stable is the existence of a Lyapunov function $V(x, \boldsymbol{y}, \boldsymbol{\lambda})$ such that

$$V(x^+, \boldsymbol{y}^+, \boldsymbol{\lambda}^+) - V(x, \boldsymbol{y}, \boldsymbol{\lambda}) \leq -\|x\|_2^2 \tag{13}$$

$$V(x, \boldsymbol{y}, \boldsymbol{\lambda}) \geq \|x\|_2^2 \tag{14}$$

$$\forall\ (x, \boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{y}^+, \boldsymbol{\lambda}^+) \in \mathbf{K},$$

where

$$\mathbf{K} = \big\{(x, \boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{y}^+, \boldsymbol{\lambda}^+) \mid h(x, \boldsymbol{y}, \boldsymbol{\lambda}) = 0,\ \boldsymbol{g}(x, \boldsymbol{y}, \boldsymbol{\lambda}) \leq 0, \\ \boldsymbol{h}(x^+, \boldsymbol{y}^+, \boldsymbol{\lambda}^+) = 0,\ \boldsymbol{g}(x^+, \boldsymbol{y}^+, \boldsymbol{\lambda}^+) \leq 0\big\}, \tag{15}$$

and $x^+ = f(x, \kappa(y_M))$.

These equations require that the function V decreases on the basic semialgebraic set $\mathbf{K}$ implicitly characterizing the closed-loop control law $x \mapsto \kappa(y_M(x))$. Therefore we can tractably seek a Lyapunov function for system (12) by restricting V to be a polynomial of a pre-defined degree and replacing the inequalities (13) and (14) by sufficient sum-of-squares[2] (SOS) conditions. Setting

$$\boldsymbol{z} := (x, \boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{y}^+, \boldsymbol{\lambda}^+),$$

these SOS conditions read

$$\begin{aligned} V(x, \boldsymbol{y}, \boldsymbol{\lambda}) - V(x^+, \boldsymbol{y}^+, \boldsymbol{\lambda}^+) - \|x\|_2^2 = \\ = \sigma_0(\boldsymbol{z}) + \sigma_1(\boldsymbol{z})^T \boldsymbol{g}(x, \boldsymbol{y}, \boldsymbol{\lambda}) + \sigma_2(\boldsymbol{z})^T \boldsymbol{g}(x^+, \boldsymbol{y}^+, \boldsymbol{\lambda}^+) \\ + p_1(\boldsymbol{z})^T \boldsymbol{h}(x, \boldsymbol{y}, \boldsymbol{\lambda}) + p_2(\boldsymbol{z})^T \boldsymbol{h}(x^+, \boldsymbol{y}^+, \boldsymbol{\lambda}^+), \end{aligned} \tag{16}$$

$$\begin{aligned} \|x\|_2^2 - V(x, \boldsymbol{y}, \boldsymbol{\lambda}) = \\ = \bar{\sigma}_0(\boldsymbol{z}) + \bar{\sigma}_1(\boldsymbol{z})^T \boldsymbol{g}(x, \boldsymbol{y}, \boldsymbol{\lambda}) + \bar{\sigma}_2(\boldsymbol{z})^T \boldsymbol{g}(x^+, \boldsymbol{y}^+, \boldsymbol{\lambda}^+) \\ + \bar{p}_1(\boldsymbol{z})^T \boldsymbol{h}(x, \boldsymbol{y}, \boldsymbol{\lambda}) + \bar{p}_2(\boldsymbol{z})^T \boldsymbol{h}(x^+, \boldsymbol{y}^+, \boldsymbol{\lambda}^+), \end{aligned} \tag{17}$$

where $\sigma_i(\boldsymbol{z})$ and $\bar{\sigma}_i(\boldsymbol{z})$ are SOS multipliers and $p(\boldsymbol{z})$ and $\bar{p}(\boldsymbol{z})$ polynomial multipliers of compatible dimensions and pre-specified degrees (selection of the degrees is discussed in Section IV). The satisfaction of (16) implies the satisfaction of (17) and the satisfaction of (14) implies the satisfaction of (13) for all $z \in \mathbf{K}$; this follows readily since σ_i and $\bar{\sigma}_i$ are globally nonnegative, $\boldsymbol{g}$ is nonnegative on $\mathbf{K}$ and h is zero on $\mathbf{K}$ (see also footnote 2).

Therefore, the problem of verifying the stability of the closed-loop system (12) boils down (via sufficient conditions) to the following SOS feasibility problem

$$\begin{array}{lll} \text{find} & V, \sigma_0, \sigma_1, \sigma_2, p_1, p_2, \bar{\sigma}_0, \bar{\sigma}_1, \bar{\sigma}_2, \bar{p}_1, \bar{p}_2 & \\ \text{s.t.} & (16), (17) & \\ & \sigma_0, \sigma_1, \sigma_2, \bar{\sigma}_0, \bar{\sigma}_1, \bar{\sigma}_2 & \text{SOS polynomials} \\ & V, p_1, p_2, \bar{p}_1, \bar{p}_2 & \text{polynomials,} \end{array} \tag{18}$$

where the optimization is over the coefficients of the polynomials sought.

The following proposition summarizes the results of this section.

Proposition 1. *If optimization problem (18) is feasible, then the closed-loop system (12) is globally asymptotically stable.*

IV. Computational aspects

After specifying the degrees of the polynomials in (18), the problem (18) immediately translates to a semidefinite program (SDP). Indeed, the constraints (16) and (17) are affine in the coefficients of the polynomials sought and the constraint that a polynomial $\sigma(x)$ of degree $2d$ is a sum-of-squares is equivalent to the existence of a positive-semidefinite matrix $Q \succeq 0$ such that $\sigma(x) = v(x)^T Q v(x)$, where $v(x)$ is a vector of monomials of total degree up to d; see [11] or [8] for details. The transformation from the abstract form (18) can be carried out automatically using freely available modeling tools such as Yalmip [9], SOSTOOLS [13] or SOSOPT [15] and solved using SDP solvers such as SeDuMi [12] or MOSEK.

The computational tractability of the resulting SDP crucially depends on the degrees of the polynomials in (18). There a tradeoff has to be made between the richness of the class of Lyapunov functions we search over and tractability. The most important decision to make is on the degree of the Lyapunov function V. It is impossible to give general guidelines on how to select this degree for the general problem (2); however, for the standard setup with a linear system and quadratic cost minimized over a finite prediction horizon N, we found it sufficient to take V quadratic jointly in $(x, \boldsymbol{y})$, thereby encompassing the optimal value function of the problem (2) as already pointed out in [14]. For a general nonlinear MPC problem we recommend selecting V such that the cost $J(x, y)$ of (2) is among the class of Lyapunov functions V searched over.

Once the degree of V is selected, we need to select the degree of the polynomial multipliers σ_i and p_i in (16) and $\bar{\sigma}_i$ and $\bar{p}_i$ in (17). The smallest possible degree we can select is the one that makes the degree of the products between the multipliers and the functions $\boldsymbol{g}$ and $\boldsymbol{h}$ in (16) and (17) equal to the degree of the Lyapunov candidate V; this corresponds to the so-called Lasserre's relaxation of order

[2] A polynomial $\sigma(x)$ is sum-of-squares (SOS) if it can be expressed as $\sigma(x) = \sum_i s_i^2(x)$ for some polynomials $s_i(x)$. A sufficient condition for a polynomial $p(x)$ to be nonnegative on a given set $\{x \mid g_i(x) \leq 0\}$ is $p(x) = \sigma_0 - \sum_i \sigma_i(x) g_i(x)$ where σ_i are sum-of-squares.

one [8]. Increasing the degree of the polynomial multipliers then generates a hierarchy of semidefinite programs covering an increasingly richer class of functions V. In this case, this hierarchy is *not* guaranteed to be convergent in the sense of asymptotically covering all polynomial Lyapunov functions V of the prescribed degree; this is due to the discrepancy between SOS and nonnegative polynomials (see, e.g., [3]) since, in general, the set $\mathbf{K}$ defined in (24) need not be compact. Indeed, it is possible to manufacture an example of a dynamical system which admits a unique (up to a scaling) quadratic Lyapunov function whose derivative along the vector field is globally non-positive but not (minus) sum-of-squares [1, Section III].

In practical terms we often find the Lasserre's relaxations of order one sufficient but recommend to try out higher orders if the first one is infeasible. Crucially, even if a higher degree of the polynomial multipliers is selected, it is possible to discard some (typically, for this problem class, a large portion) of the monomials comprising these multipliers based on monomial reduction techniques (such as the Newton polytope [17]); this reduction is carried out automatically by SOS modeling tools such as Yalmip [9] or SOSOPT [15]. In particular SOSOPT carries out more efficient reduction techniques described in [16] and typically arrives at a more tractable SDP problem to be solved. However, for larger problems, the monomial reduction phase itself may become the bottleneck of the whole procedure and therefore care must be taken when selecting the degrees of the polynomial multipliers. One can, for instance, solve the problem for a small number of iterations of the gradient M, look at what monomials were discarded by SOSOPT and extrapolate from there the selection of monomials for a larger M; this heuristic procedure often significantly reduces the reduction phase and requires only a marginal additional effort on the side of the user.

V. Local version – state constraints

The method presented in Section 2 addresses the verification of global asymptotic stability. In order to get a local version and/or incorporate state constraints we follow the approach of Primbs [14] who proposes to simply enforce the Lyapunov conditions (13) and (14) on the intersection of the set $\mathbf{K}$ defined in (24) with the constraint set (or a given region of interest)

$$\mathbf{X} := \{x \in \mathbb{R}^n \mid \psi_i(x) \leq 0,\ i = 1, \ldots, n_\psi\}.$$

Provided that the functions $\psi_i(\cdot)$ defining $\mathbf{X}$ are polynomials, the set

$$\bar{\mathbf{K}} := \{(x, \boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{y}^+, \boldsymbol{\lambda}^+) \mid (x, \boldsymbol{y}, \boldsymbol{\lambda}, \boldsymbol{y}^+, \boldsymbol{\lambda}^+) \in \mathbf{K}, x \in \mathbf{X}\}$$

is basic semialgebraic, leading to a verification SOS problem completely analogous to (18). However, the pitfall here is that the satisfaction of Lyapunov conditions (13) and (14) does not ensure invariance of the closed-loop evolution in the set $\mathbf{X}$. Asymptotic stability is guaranteed only on the largest sub-level set of V contained in $\mathbf{X}$. To be more precise asymptotic stability is guaranteed in the largest sub-level set contained in $\mathbf{X}$ of the function

$$\bar{V}(x) = \sup_{\boldsymbol{y},\boldsymbol{\lambda}} \{V(x, \boldsymbol{y}, \boldsymbol{\lambda}) \mid (x, \boldsymbol{y}, \boldsymbol{\lambda}) \in \hat{\mathbf{K}}\},$$

where

$$\hat{\mathbf{K}} := \{(x, \boldsymbol{y}, \boldsymbol{\lambda}) \mid \boldsymbol{g}(x, \boldsymbol{y}, \boldsymbol{\lambda}) \leq 0, \boldsymbol{h}(x, \boldsymbol{y}, \boldsymbol{\lambda}) = 0, x \in \mathbf{X}\}.$$

Note that if we choose V as a function of x only, then trivially $\bar{V}(x) = V(x)$. In general, however, the function $\bar{V}(x)$ is a solution to a complicated parametric polynomial optimization problem and therefore not known analytically. Therefore, we resort to computing an inner approximation to the largest super level set of $\bar{V}(x)$ included in $\mathbf{X}$ by first approximating $\bar{V}(x)$ from above by a polynomial function $p(x)$ of a prescribed degree and computing the largest sub-level set of $p(x)$ included in $\mathbf{X}$. Approximating $\bar{V}(x)$ from above can be done using the following SOS optimization problem

$$\begin{array}{ll} \underset{p(x)}{\text{minimize}} & \int_{\mathbf{X}} p(x)\, dx \\ \text{s.t.} & p(x) \geq V(x, \boldsymbol{y}, \boldsymbol{\lambda}) \quad \forall\, (x, \boldsymbol{y}, \boldsymbol{\lambda}) \in \hat{\mathbf{K}}. \end{array} \tag{19}$$

Since the set $\hat{\mathbf{K}}$ is basic semialgebraic, the constraint immediately translates to an SOS constraint (with conservatism inversely proportional to the degree of the SOS multipliers), and the objective is a linear combination of the coefficients of the polynomial $p(x)$, where the coefficients of the linear combination are the moments of the Lebesgue measure over $\mathbf{X}$. Problem (19) therefore immediately translates to an SDP.

Once $p(\cdot)$ is found an inner approximation to the maximum level set of $\bar{V}(x)$ contained in $\mathbf{X}$ can be found by solving

$$\begin{array}{ll} \underset{\gamma \in \mathbb{R}_+, \{\sigma_{0,i}\}, \{\sigma_{0,i}\}}{\text{maximize}} & \gamma \\ \text{s.t.} & -\psi_i(x) = \sigma_{0,i}(x) + \sigma_{1,i}(x)(\gamma - p(x)),\ i \in \{1, \ldots, n_\psi\} \\ \{\sigma_{0,i}\},\ \{\sigma_{1,i}\} & \text{SOS polynomials} \end{array} \tag{20}$$

The satisfaction of the first constraint implies that $\psi_i(x) \leq 0$ for all x such that $p(x) \leq \gamma$ and all $i \in \{1, \ldots, n_\psi\}$; therefore $\{x \mid p(x) \leq \gamma\} \subset \mathbf{X}$ for any γ feasible in (20). Maximizing γ then maximizes the size of the inner approximation.

Problem (20) is only quasi-convex because of the bilinearity between σ_1 and γ but can be efficiently solved using a bi-section on γ. Indeed, for a fixed value of γ problem (20) is an SDP which can be efficiently solved.

The following Lemma summarizes the results of this section.

Lemma 1. *If a polynomial $p(\cdot)$ is feasible in (19) and $\gamma \in \mathbb{R}_+$ feasible in (20), then the set $\{x \mid p(x) \leq \gamma\}$ provides an inner approximation to the maximum sub-level set of $\bar{V}(X)$ included in $\mathbf{X}$.*

This immediately leads to the following proposition.

Proposition 2. *If a polynomial $V(\cdot, \cdot, \cdot)$ is feasible in (18), polynomial $p(\cdot)$ feasible in (19) and $\gamma \in \mathbb{R}_+$ feasible in (20),*

then all trajectories of the closed-loop system (12) starting from the set $\{x \mid p(x) \leq \gamma\}$ are asymptotically stable and lie in the constraint set $\mathbf{X}$.

VI. Other first-order methods

The main idea of the verification procedure described in Section III extends to many other first-order methods. This section presents a by no means exhaustive list of some of the more commonly encountered methods amenable to the presented approach.

A. Fast gradient

The approach immediately extends, with identical computation cost, to the fast gradient method whose iterates proceed as

$$\begin{aligned} y_{k+1} &= P_{\mathcal{C}}(z_k - \tau \nabla_y J(x, z_k)) \\ z_{k+1} &= y_{k+1} + \beta_k (y_{k+1} - y_k) \end{aligned}$$

for a suitably chosen momentum sequence β_k [10]. Indeed, as for the projected gradient method, each step involves one projection step that can be characterized via a KKT system in exactly the same fashion as before. The auxiliary sequence z_k is a linear combination of y_k and y_{k-1} and therefore does not enter the KKT system (5)-(9) as a new variable. Consequently, the dimension of the verification problem remains the same.

B. ADMM

The alternating direction method of multipliers (ADMM) (see, e.g., the survey [4]) can also be handled. In order to apply ADMM we assume that the general problem (2) can be written in the special form

$$\begin{array}{ll} \underset{y \in \mathbb{R}^{n_y}, z \in \mathbb{R}^{n_z}}{\text{minimize}} & J_1(x, y) + J_2(x, z) \\ \text{s.t.} & A(x)y + B(x)z = c(x), \end{array} \tag{21}$$

where $J_1(x, y)$ are polynomial and the matrices $A(x)$, $B(x)$ and the vector $c(x)$ depend polynomially on x. The function $J_2(x, z)$ (which take values in the extended real line) is assumed to have a proximal operator [5] that can be characterized by a KKT system described by polynomial equalities and inequalities; typically, $J_2(x, z)$ is the indicator function[3] of a basic semialgebraic constraint set.

Casting the general problem (2) in the form (21) can be done in various ways, each leading to different computational properties of the resulting problem. In problem formulations arising from control, the function $J_1(x, y)$ often corresponds to the cost $J(x, y)$ in (2), $J_2(z, x)$ is typically the indicator function of the constraint set and the equality constraint of (21) a consensus constraint $y - z = 0$.

[3]By indicator function of a set we mean a function which is equal to zero on the set and $+\infty$ otherwise.

The ADMM iterations are

$$\begin{aligned} y_{k+1} &= \underset{y \in \mathbb{R}^{n_y}}{\operatorname{argmin}} \big\{ J_1(x, y) + \\ &\qquad + (\rho/2) \|A(x)y + B(x)z_k - c(x) + \omega_k\|_2^2 \big\} \\ z_{k+1} &= \underset{z \in \mathbb{R}^{n_z}}{\operatorname{argmin}} \big\{ J_2(x, z) + \\ &\qquad + (\rho/2) \|A(x)y_{k+1} + B(x)z - c(x) + \omega_k\|_2^2 \big\} \\ \omega_{k+1} &= \omega_k + \big(A(x)y_{k+1} + B(x)z_{k+1} - c(x)\big), \end{aligned}$$

where $\rho > 0$ is a fixed parameter. We assume that the minima in both steps are attained and that a constraint qualification assumption analogous to Assumption (1) holds.

Then, the first two steps can be characterized by two KKT systems and therefore ADMM is also amenable to the approach of Section 2, leading to a verification problem of roughly twice the size of that of the (fast) gradient method. However, in most cases where it is viable to deploy ADMM, the first step admits an analytic solution (e.g., it is an unconstrained minimization of a strictly convex quadratic) and therefore only one KKT system for the second step is needed, leading to the same computation complexity as for the gradient methods.

An accelerated version of ADMM using an over-relaxation step applied to the sequence of ω_k (see, [7, Algorithm 8]) can also be handled in an analogous fashion to the fast gradient.

The alternating minimization algorithm (AMA) proposed in [19] can also be handled analogously to ADMM.

VII. Robust version

This section describes how the proposed approach extends to the case where the controlled system is affected by a disturbance at each time taking values in a given basic semi-algebraic set (which can possibly depend on x). Consider the dynamical system

$$x^+ = f(x, u, w), \quad w \in \mathbf{W}(x), \tag{22}$$

where the disturbance set $\mathbf{W}(x)$ is given by

$$\mathbf{W}(x) = \{w \in \mathbb{R}^{n_w} \mid \psi_w(x, w) \leq 0\}, \tag{23}$$

where the components of the vector function $\psi_w : \mathbb{R}^{n+n_w} \to \mathbb{R}^{n_{\psi_w}}$ are polynomial. Note that we do not a priori assume that the set $\mathbf{W}(x)$ is compact.

The extension is straightforward. Defining

$$\begin{aligned} \mathbf{K}_w = \big\{ &(x, \boldsymbol{y}, \boldsymbol{\lambda}, w, \boldsymbol{y}^+, \boldsymbol{\lambda}^+, w^+) \mid h(x, \boldsymbol{y}, \boldsymbol{\lambda}) = 0, \\ &\boldsymbol{g}(x, \boldsymbol{y}, \boldsymbol{\lambda}) \leq 0, \boldsymbol{h}(x^+, \boldsymbol{y}^+, \boldsymbol{\lambda}^+) = 0, \ \boldsymbol{g}(x^+, \boldsymbol{y}^+, \boldsymbol{\lambda}^+) \leq 0 \\ &\psi_w(x, w) \leq 0, \psi_w(x^+, w^+) \leq 0 \big\}, \end{aligned} \tag{24}$$

we can seek an input-to-state (ISS) Lyapunov function $V(x, \boldsymbol{y}, \boldsymbol{\lambda}, w)$ such that

$$V(x^+, \boldsymbol{y}^+, \boldsymbol{\lambda}^+, w^+) - V(x, \boldsymbol{y}, \boldsymbol{\lambda}, w) \leq -\|x\|_2^2 + \alpha \|w\|_2^2 \tag{25}$$

$$V(x, \boldsymbol{y}, \boldsymbol{\lambda}, w) \geq \|x\|_2^2 \tag{26}$$

$$\forall \ (x, \boldsymbol{y}, w, \boldsymbol{\lambda}, \boldsymbol{y}^+, \boldsymbol{\lambda}^+, w^+) \in \mathbf{K}_w,$$

where $x^+ = f(x, \kappa(y_M), w)$ with $\kappa(\cdot)$ being the polynomial extracting the control input applied to the system from the Mth iterate of a first-order method, and $\alpha \in \mathbb{R}_+$.

Since the set $\mathbf{K}_w$ is basic semialgebraic, the conditions (25) and (26) translate immediately to sum-of-squares conditions analogous to (16) and (17), leading to an SOS problem analogous to (18). Instead of solving a feasibility problem as in (18) we can minimize the ISS gain α which still leads to a convex SOS problem. Feasibility of this optimization problem implies that the closed-loop system is globally input-to-state stable provided that Assumption 1 holds for the optimization problem characterizing the iterates of the first-order method considered and provided that $\mathbf{W}(x)$ is non-empty for all $x \in \mathbb{R}^n$.

VIII. Numerical example

This section illustrates the approach on two numerical examples. The SOS problems were modeled using SOSOPT [15] and solved using MOSEK. For both examples we report the parsing time of SOSOPT, the time to carry out monomial reduction by SOSOPT and solve time of MOSEK. The bottleneck of the approach is the monomial reduction phase (which, however, is very effective in the sense of reducing the size of the problem significantly). The authors expect that a more efficient implementation of the reduction phase and polynomial handling in general would allow the approach to scale much beyond what is presented here.

A. Global asymptotic stability of an uncertain system

Consider the Quanser active suspension model in continuous-time $\dot{x} = A_c x + B_c u$ with

$$A_c = \begin{bmatrix} 0 & 1 & 0 & -1 \\ -K_s/M_s & -B_s/M_s & 0 & B_s/M_s \\ 0 & 0 & 0 & 1 \\ K_s/M_{us} & B_s/M_{us} & -K_{us}/M_{us} & -(B_s+B_{us})/M_{us} \end{bmatrix},$$

$$B_c = \begin{bmatrix} 0 & 1/M_s & 0 & -1/M_{us} \end{bmatrix}^T,$$

where $K_s = 1205$, $K_{us} = 2737$, $M_{us} = 1.5$, $B_s = 20$, $B_{us} = 20$ and the mass M_s is unknown and possibly time-varying in the interval $[2.85, 4]$. After discretization[4] with sampling period 0.01, this model can be written as $x^+ = (A_0 + A_1 w)x + (B_0 + B_1 w)x$, where $w := 1/M_s \in [1/4, 1/2.85]$.

The optimization problem considered (2) is the standard constrained LQ problem with cost function given by matrices $Q = I$ and $R = 20$ minimized over prediction horizon N subject to input constraint $|u| \leq 250$ and nominal dynamics $x^+ = A_0 x + B_0 u$. This problem is expressed in a dense form (i.e., the state is eliminated using the dynamics equation) to which we apply M steps of the projected gradient method (3) initialized with the LQ solution and seek a quadratic ISS Lyapunov function V while minimizing the ISS gain α in (26). The robust version of the SOS problem (18) described in Section VII is feasible (for all combinations of M and N tested) when we take the SOS multipliers σ_1, σ_2 in equation (16) of degree two in $(x, \boldsymbol{y})$ and the polynomial multipliers p_1, p_2 of degree one in $(x, \boldsymbol{y}, \boldsymbol{\lambda})$. The list of monomials $v(x, \boldsymbol{y}, \boldsymbol{\lambda}, w)$ constituting the multiplier σ_0 in the form $v(x, \boldsymbol{y}, \boldsymbol{\lambda}, w)^T \mathcal{Q} v(x, \boldsymbol{y}, \boldsymbol{\lambda}, w)$, $\mathcal{Q} \succeq 0$, is determined automatically by SOSOPT and contains monomials linear in $\boldsymbol{\lambda}$, x, $\boldsymbol{y}$ and w, and products $x \cdot w$ and $\boldsymbol{y} \cdot w$. In Eq. (17) we set all multipliers to zero except for $\bar{\sigma}_0$, monomials of which are again determined automatically by SOSOPT. Determining the smallest list of monomials $v(x, \boldsymbol{y}, \boldsymbol{\lambda}, w)$ takes the most time of the whole procedure; this is documented by Table I reporting the time breakdown for different values of N and M. The optimal ISS gain α is equal to zero, showing closed-loop global robust asymptotic stability (i.e., convergence $\|x_k\| \to 0$ for any sequence $\{w_k \in [1/4, 1/2.85]\}_{k=0}^{\infty}$). Figure 1 shows a sample trajectory of $\|x_k\|$, $V(x_k)$ (the Lyapunov function is function of x only in this case) and u_k and w_k for $N = M = 4$.

TABLE I

Global asymptotic stability of an uncertain system – timing breakdown as a function of the number of iteration of the projected gradient method M and the horizon length N used in the cost function. The parsing and monomial reduction was carried out by SOSOPT; the SDP solve by MOSEK.

	parsing	monomial reduction	SDP solve
$M, N = 1, 1$	0.93 s	1.2 s	0.11 s
$M, N = 2, 2$	2.68 s	14.5 s	0.35 s
$M, N = 3, 3$	45 s	350 s	1.4 s
$M, N = 4, 4$	1053 s	9450 s	9 s

B. Local stability of a quadcopter

This example investigates stability of a linearized attitude and vertical velocity model of a quadcopter. The system has seven states (Roll, Pitch and Yaw angles and angular velocities, and velocity in the vertical direction) and four control inputs (the thrusts of the four rotors). The system is controlled by a one-step MPC controller which at time k approximately minimizes the cost $x_k^T Q x_k + u_k^T R u_k + x_{k+1}^T P x_{k+1}$, where $Q = I$, $R = 10I$ and P is the infinite-time LQ matrix associated to the cost matrices Q and R, using one step of the projected gradient method (3) subject to the input constraints $\|u\|_\infty \leq 1$. This model is open-loop unstable and therefore we investigate closed-loop stability in the region $\mathbf{X} = [-1, 1]^7$ as described in Section V. The SOS problem (18) is feasible when seeking a quadratic Lyapunov function using SOS multipliers σ_1, σ_2 in equation (16) of degree two in x and the polynomial multipliers p_1, p_2 of degree one in $(x, \boldsymbol{y}, \boldsymbol{\lambda})$. The smallest set of monomials constituting σ_0 is chosen automatically by SOSOPT. In (17), we chose all multipliers zero except for $\bar{\sigma}_0$ whose monomials are determined automatically by SOSOPT. Computing the largest γ such that $\{x \mid V(x) \leq \gamma\}$ is included in $\mathbf{X}$ yields $\gamma = 6.37$; this proves that all trajectories starting in $\{x \mid V(x) \leq \gamma\}$ stay there and converge to the origin. One closed-loop trajectory of $\|x\|_2$, $V(x)$ and u are depicted in Figure 2; note that this trajectory does not start in $\{x \mid V(x) \leq \gamma\}$ but still converges to the origin and the Lyapunov function decreases. The parsing time and

[4]The matrices A_0, A_1, B_0, B_1 were found as a least-squares fit of the continuous-time dynamics discretized on a grid of values of $w \in [1/4, 1/2.85]$.

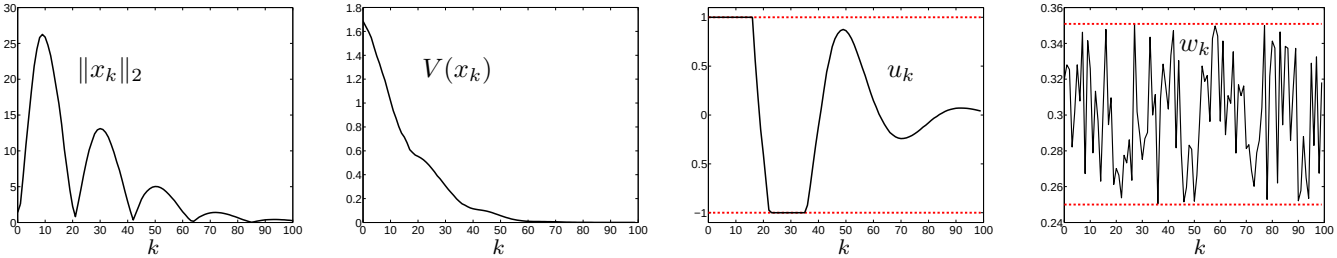

Fig. 1. Global asymptotic stability of an uncertain system – one trajectory starting from the initial condition $x_0 = [1\ 1\ 1\ 1]^T$ of the norm of the state $\|x_k\|$, the Lyapunov function $V(x_k)$, the control input u_k and the disturbance w_k.

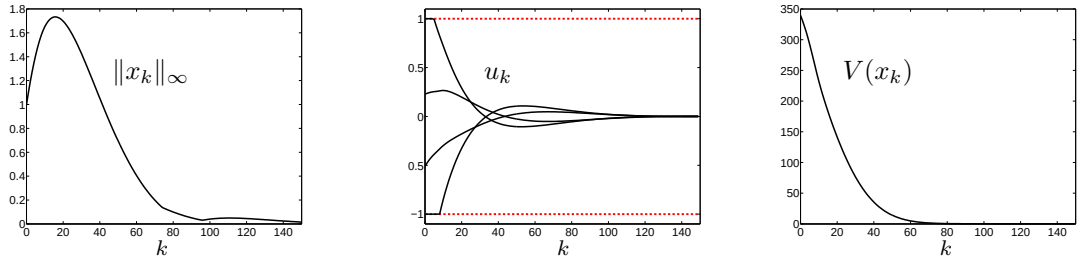

Fig. 2. Local stability of a quadcopter – trajectories of the norm of the state $\|x_k\|$, the Lyapunov function $V(x_k)$, the control input u_k for initial condition $x_0 = [1\ 1\ 1\ 1\ 1\ 1\ 1]^T$.

monomial reduction carried out by SOSOPT took 2.7 s and 16.2 s, respectively; the MOSEK solve time was 0.55 s.

IX. Conclusion

This paper presents a verification method for early-terminated first-order methods. The method is based on the observation that any control input generated by the vast majority of first-order methods terminated after a finite number of iterations belongs to a basic semialgebraic set related to the KKT systems characterizing each step of the first-order method. One can then seek a polynomial Lyapunov function decreasing on this semialgebraic set using sum-of-squares programming which reduces to a semidefinite program.

The sum-of-squares problem can be easily modeled using freely available high-level tools (e.g., SOSOPT [15] or Yalmip [9]). At present, the parsing and pre-processing phase carried out by these modeling tools seem to be the computational bottleneck of the method, not the size of the resulting semidefinite program. The authors are currently investigating alternative polynomial representations and more efficient monomial reduction techniques that could significantly improve scalability of the approach.

Acknowledgments

The first author would like to thank Ye Pu for a discussion that initiated the work, and Zlatko Emedji and Mahdieh Sadat Sadabad for providing the uncertain system for the numerical example.

References

[1] A. A. Ahmadi, P. A. Parrilo. Converse Results on Existence of Sum of Squares Lyapunov Functions. 50th IEEE Conference on Decision and Control, Orlando, 2011.

[2] A. Bemporad, P. Patrinos. Simple and Certiable Quadratic Programming Algorithms for Embedded Linear Model Predictive Control. 4th IFAC Nonlinear Model Predictive Control Conference, Noordwijkerhout, 2012.

[3] G. Blekherman. There are Significantly More Nonnegative Polynomials than Sums of Squares. Israel Journal of Mathematics, 153:355-380, 2006.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. Foundations and Trends in Machine Learning, 3(1):1-122, 2011.

[5] P. L. Combettes, J. C. Pesquet. Proximal splitting methods in signal processing. Fixed-Point Algorithms for Inverse Problems in Science and Engineering. Springer Optimization and Its Applications, pp 185-212, 2011.

[6] B. Dumitrescu. Positivstellensatz for trigonometric polynomials and multidimensional stability tests. IEEE Transactions on Circuits and Systems, 54: 353-356, 2007.

[7] T. Goldstein, B. O'Donoghue, S. Setzer. Fast alternating direction optimization methods. 2013

[8] J. B. Lasserre. Moments, positive polynomials and their applications. Imperial College Press, London, UK, 2009.

[9] J. Löfberg. YALMIP : A toolbox for modeling and optimization in MATLAB. In Proc. IEEE CCA/ISIC/CACSD Conference, Taipei, Taiwan, 2004.

[10] Y. Nesterov. Introductory Lectures on Convex Optimization: A Basic Course. Springer, 2003.

[11] P. A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. Mathematical Programming, 96:293-320, 2003.

[12] I. Pólik, T. Terlaky, and Y. Zinchenko. SeDuMi: a package for conic optimization. IMA workshop on Optimization and Control, Univ. Minnesota, Minneapolis, Jan. 2007.

[13] S. Prajna, A. Papachristodoulou, P. Seiler, P. A. Parrilo. SOSTOOLS: sum of squares optimization toolbox for MATLAB, 2004.

[14] J. A. Primbs. The analysis of optimization based controllers. Automatica, 37:933-938, 2001.

[15] P. Seiler. SOSOPT: A toolbox for polynomial optimization. University of Minnesota, 2010.

[16] P. Seiler, Q. Zheng, and G. J. Balas. Simplification Methods for Sum-of-Squares Programs. arXiv:1303.0714, 2013.

[17] B. Sturmfels. Polynomial equations and convex polytopes. American Mathematical Monthly, 105(10):907-922, 1998.

[18] S. Richter, C. N. Jones, M. Morari. Computational complexity certification for real-time MPC with input constraints based on the fast gradient method. IEEE Transactions on Automatic Control, 57:1391-1401, 2012.

[19] P. Tseng. Applications of splitting algorithm to decomposition in convex programming and variational inequalities. SIAM J. Control Optim., 29:119-138, 1991.

[20] M. N. Zeilinger, C. N. Jones, D. M. Raimondo, M. Morari. Real-time MPC - Stability through robust MPC design. 48th IEEE Conference on Decision and Control, Shanghai, 2009.