



HAL
open science

Fast and Robust Stability Region Estimation for Nonlinear Dynamical Systems

Eloïse Berthier, Justin Carpentier, Francis Bach

► **To cite this version:**

Eloïse Berthier, Justin Carpentier, Francis Bach. Fast and Robust Stability Region Estimation for Nonlinear Dynamical Systems. 2020. hal-02984348v1

HAL Id: hal-02984348

<https://hal.science/hal-02984348v1>

Preprint submitted on 30 Oct 2020 (v1), last revised 2 Sep 2021 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

FAST AND ROBUST STABILITY REGION ESTIMATION FOR NONLINEAR DYNAMICAL SYSTEMS

ELOÏSE BERTHIER, JUSTIN CARPENTIER AND FRANCIS BACH

*Inria - Ecole Normale Supérieure
PSL Research University, Paris, France*

ABSTRACT. A linear quadratic regulator can stabilize a nonlinear dynamical system with a local feedback controller around a linearization point, while minimizing a given performance criteria. An important practical problem is to estimate the region of attraction of such a controller, that is, the region around this point where the controller is certified to be valid. This is especially important in the context of highly nonlinear dynamical systems. In this paper, we propose two stability certificates that are fast to compute and robust when the first, or second derivatives of the system dynamics are bounded. Associated with an efficient oracle to compute these bounds, this provides a simple stability region estimation algorithm compared to classic approaches of the state of the art. We experimentally validate that it can be applied to both polynomial and non-polynomial systems of various dimensions, including standard robotic systems, for estimating region of attractions around equilibrium points, as well as for trajectory tracking.

1. INTRODUCTION

Controlling a robot typically involves a global motion planning to steer the system from one initial position to a target goal, as well as some local feedback corrections to accurately track the planned trajectory. For instance, the combination of rapidly-exploring random trees [LK01] and local trajectory stabilization led to a fruitful feedback motion planning algorithm named LQR-trees [TMTR10]. In this algorithm, a locally optimal trajectory is computed between sampled points of the state space. Each trajectory is then locally stabilized with a linear quadratic regulator (LQR). The aim is to design a global controller by covering the whole state space with overlapping funnels, *i.e.*, regions of attraction (ROA) around trajectories. An important subproblem is to estimate such an ROA: a set of starting points that the controlled dynamics brings back to an equilibrium. Crucially, it must be performed efficiently, as it will be called repeatedly to cover a potentially large dimensional state space. Ideally, the estimation must be fast to compute, but not overly conservative.

A controlled dynamical system can be stabilized around an equilibrium point with an adequate closed-loop controller. It is possible to synthesize an optimal feedback controller for some stability criterion [GSM90], but a simply available candidate is the LQR. The stability of a region is commonly assessed with a Lyapunov function, which again can be optimized [GH15, Joh00], or not. This paper focuses on finding the largest estimate of the ROA for a given controller and a given Lyapunov function, both obtained from LQR. The gold standard technique for this problem is based on sum of squares (SOS) programming and provides high quality estimates. Yet it is limited to polynomial dynamics, and grows computationally heavy in large dimensions, hence limiting its applicability in practice, especially in the context of robotics where fast methods are needed to accurately control and stabilize the motions of the robot such as for legged locomotion [CM18b].

Another stake in robotics is robustness with respect to model misspecification or uncertainties. In particular, there can be a shift between the behavior of a simulated robotic system and its physical counterpart [SCH⁺18]. Robust ROA estimation methods [Che04] must account for the uncertainty on the parameters of the dynamics. In particular, we focus on the case where the Jacobian or the Hessian of the dynamics is known to be bounded. This applies to robust control, but also to perfectly known dynamics that are computationally hard to handle. Bounding the Jacobian or Hessian is possible analytically for simple polynomial systems, or by sampling, taking advantage of automatic differentiation for complicated robotic systems [GNS⁺17]. Interestingly, the bounds can be computed offline, in parallel, or experimentally with

E-mail address: `firstname.lastname@inria.fr`.

a real physical system. With such information on the dynamics, our goal is to design fast, robust ROA estimation methods, practical in large state dimensions.

Our main contribution is to propose a general ROA estimation framework for non-polynomial systems, which is faster and simpler than SOS-based methods. The paper is organized as follows. After introducing the principle of LQR stabilization in Section 2, we adapt in Section 3 an existing robust stability certificate to systems with entrywise uncertainty bounds on their Jacobians. In Section 4, we present stability certificates for systems with entrywise bounds on their Hessians, and in Section 5, we propose an algorithm adapting robust certificates to systems with varying derivatives. In Section 6, we extend the methods to the trajectory tracking problem. Finally in Section 7, we compare the robust certificates, as well as those provided by SOS programming, on numerical examples of various dimensions. An implementation is available online.

2. PRELIMINARIES

We consider a nonlinear time-invariant control system:

$$\dot{x} = f(x, u),$$

where $x \in \mathbb{R}^d$, $u \in \mathbb{R}^m$, with $d, m \geq 1$. Assume there exists an equilibrium, without loss of generality at $(x_0, u_0) = (0, 0)$, that is $f(0, 0) = 0$, and that f is also smoothly differentiable:

$$f(x, u) = \underbrace{\frac{\partial f}{\partial x}(0, 0)x}_{Ax} + \underbrace{\frac{\partial f}{\partial u}(0, 0)u}_{Bu} + o(x) + o(u).$$

We assume that the pair (A, B) is controllable. For $Q \succeq 0$, $R \succ 0$ symmetric matrices respectively of size $d \times d$ and $m \times m$, we define the infinite-horizon LQR cost [Lib11]:

$$J(x) := \int_0^{+\infty} (x^\top(t)Qx(t) + u^\top(t)Ru(t))dt, \text{ with } x(0) = x.$$

The cost-minimizing controller is known to be:

$$u(x) = -R^{-1}B^\top Sx =: -Kx, \quad (1)$$

where S is the symmetric positive definite solution of the algebraic Riccati equation (ARE), which exists because (A, B) is controllable:

$$A^\top S + SA - SBR^{-1}B^\top S = -Q. \quad (2)$$

Under the closed-loop controller $u = -Kx$, the system is autonomous with

$$\dot{x} = f(x, -Kx) =: g(x). \quad (3)$$

In addition, the optimal cost-to-go $V(x) := x^\top Sx$ is used as a Lyapunov function of the nonlinear system. V is a Lyapunov function over a region $\mathcal{R} \subset \mathbb{R}^d$ around 0, if $V(0) = 0$, $V(x) > 0$ in $\mathcal{R} \setminus \{0\}$, and $\dot{V}(x) < 0$ in \mathcal{R} [SL91]. This certifies that the sublevel sets of V that are included in \mathcal{R} belong to the ROA of the equilibrium point: every trajectory beginning in this set will asymptotically stabilize to 0. In practice, it is convenient to choose \mathcal{R} as a sublevel set of V .

If the dynamics were exactly linear, then one would have:

$$\begin{aligned} \dot{V}(x) &= \nabla V(x) \cdot g(x) = 2x^\top Sg(x) = x^\top (SA + A^\top S - 2SBK)x \\ &= x^\top (-Q - SBR^{-1}B^\top S)x < 0, \quad \forall x \neq 0. \end{aligned}$$

Hence in the linear case, the ROA is the whole state space. In this work we will consider variations of this situation, and see how defects of linearity will affect this statement.

3. FIRST ORDER ROBUSTNESS

A linear differential inclusion (LDI) [AC84] is the following set-valued control problem:

$$\dot{x} \in \Omega x, \quad x(0) = x_0,$$

where Ω is a convex subset of $\mathbb{R}^{d \times d}$, and $\Omega x := \{Ax, A \in \Omega\}$. The constraint can be expressed as a linear matrix inequality (LMI) for some Ω [BEGFB94]. In particular, for $\Omega = \{A_0\}$ we get a linear system, for $\Omega = \text{Conv}(A_1, \dots, A_L)$ a polytopic LDI (PLDI), for $\Omega = \{A_0 + C\Delta E \mid \|\Delta\| \leq 1\}$ a norm-bound LDI (NLDI) where $\|\cdot\|$ is a matrix norm; if in addition Δ must be diagonal, we get a diagonal NLDI (DNLDI).

The asymptotic stability of an LDI around 0, *i.e.*, all initial conditions converge to 0, can be certified by a Lyapunov function of the form $V(x) = x^\top Px$. This amounts to finding:

$$P \succ 0 \text{ such that } A^\top P + PA \prec 0, \forall A \in \Omega.$$

This problem reduces to an LMI and is solved by interior-point methods [NN94] for general choices of Ω .

LDIs are used to model uncertain linear systems. Any differentiable dynamical system with an equilibrium at the origin and with bounded Jacobian (including the closed loop system in equation (3)) belongs to a suitable LDI, written as an uncertain linear system:

$$\dot{x} = A(x)x, \quad A(x) \in \Omega.$$

Ω is a convex set of matrices that accounts for nonlinearities, uncertainties or time-variations of the dynamics. In particular, Ω can bound the deviation of a nonlinear system $\dot{x} = g(x)$ from its linearization $\dot{x} = J_g(0)x$. This is similar to the problem considered in [TP07], except that the perturbations lie in a closed convex set instead of a semialgebraic set.

Suppose we are given individual bounds on the entrywise deviations of the Jacobian from a given matrix A_0 :

$$v_{ij} := \sup_{x \in \mathbb{R}^d} |\delta_{ij}(x)| = \sup_{x \in \mathbb{R}^d} |A(x)_{ij} - (A_0)_{ij}|. \quad (4)$$

Such bounds are easy to estimate from sampling or are computed in closed form. Stability is readily studied [BEGFB94] if Ω is a convex hull (PLDI) or a matrix ball (NLDI), which we now specify for our problem. Entrywise bounds can be fitted in both settings. Yet the description of the corresponding PLDI is intractable in large dimension: the number of vertices required to describe Ω scales as 2^d .

The following description of Ω with a DNLDI has polynomial length. Let $\mathbf{1}_d := (1 \dots 1)$, $\mathbf{0}_d := (0 \dots 0)$,

$$\Delta := \text{Diag} \left(\frac{\delta_{11}}{v_{11}}, \dots, \frac{\delta_{1d}}{v_{1d}}, \dots, \frac{\delta_{d1}}{v_{d1}}, \dots, \frac{\delta_{dd}}{v_{dd}} \right) \in \mathbb{R}^{d^2 \times d^2},$$

$$C := \begin{bmatrix} \mathbf{1}_d & \mathbf{0}_d \\ \mathbf{0}_d & \ddots \\ & & \mathbf{1}_d \end{bmatrix} = I_d \otimes \mathbf{1}_d \in \mathbb{R}^{d^2 \times d^2},$$

$E := [E_1 \dots E_d]^\top \in \mathbb{R}^{d^2 \times d}$, with $E_i = \text{Diag}(v_{i1}, \dots, v_{id})$. Hence $\|\Delta\|_2 = \sqrt{\lambda_{\max}(\Delta^\top \Delta)} = \sigma_{\max}(\Delta) \leq 1$, and the system belongs to the DNLDI defined by

$$\Omega = \{A_0 + C\Delta E \mid \|\Delta\| \leq 1, \Delta \text{ diagonal}\}.$$

Checking the asymptotic stability of a DNLDI is an LMI feasibility problem derived by applying the S-procedure [BEGFB94]:

Proposition 1. *Let $\dot{x} = A(x)x$ an uncertain linear system with entrywise bounded Jacobian. A sufficient condition for its global asymptotic stability at 0 is the feasibility of the following LMI, for A_0, C, E defined as above:*

Find $P \succ 0 \in \mathbb{R}^{d \times d}$, $\Lambda \succeq 0 \in \mathbb{R}^{d^2 \times d^2}$ diagonal such that:

$$\begin{bmatrix} A_0^\top P + PA_0 + E^\top \Lambda E & PC \\ C^\top P & -\Lambda \end{bmatrix} \prec 0. \quad (5)$$

One may optimize both P and Λ to obtain a Lyapunov function, or use a fixed predefined value of P , *e.g.*, S from the LQR, to check if $V(x) = x^\top Px$ is a valid Lyapunov function.

4. SECOND ORDER ROBUSTNESS

4.1. Condition on the Sublevel Sets. Let $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}^d$, such that each φ_k is twice continuously differentiable with bounded Hessian on a closed ball \mathcal{B} centered around 0. Then, using Taylor's formula, for any $x \in \mathbb{R}^d$, there exists a symmetric matrix $H^k(x)$ such that for all $k \in \{1, \dots, d\}$:

$$\varphi_k(x) = \varphi_k(0) + \nabla \varphi_k^\top x + \frac{1}{2} x^\top H^k(x)x,$$

$$\text{with } H_{ij}^k(x) = 2 \int_0^1 (1-t) \frac{\partial^2 \varphi_k}{\partial x_i \partial x_j}(tx) dt.$$

Hence $\forall i, j, k, \forall x \in \mathcal{B}$, $|H_{ij}^k(x)| \leq \max_{y \in \mathcal{B}} \left| \frac{\partial^2 \phi_k}{\partial x_i \partial x_j}(y) \right|$. Note that this is also true if \mathcal{B} is an ellipsoid around 0. This applies to the function g defined in Section 2, equation (3):

$$g_k(x) = f_k(x, -Kx) = (A - BK)_{k \cdot} x + \frac{1}{2} x^\top H^k(x) x,$$

where $M_k \cdot$ denotes the k -th line-vector of a matrix M . The derivative of the candidate Lyapunov function is:

$$\begin{aligned} \dot{V}(x) &= 2x^\top S \left((A - BK)x + \frac{1}{2} (x^\top H^k(x) x)_{k \in \{1, \dots, d\}} \right) \\ &= x^\top (-Q - SBR^{-1}B^\top S + \sum_{k=1}^d (S_k \cdot x) H^k(x)) x. \end{aligned}$$

Let $\mathcal{B}_\rho := \{x \mid x^\top S x \leq \rho\}$ for $\rho > 0$, a sublevel set of V . A sufficient condition for \mathcal{B}_ρ to be an ROA around 0 is that $-Q - SBR^{-1}B^\top S + \sum_k (S_k \cdot x) H^k(x) \prec 0$ for all $x \in \mathcal{B}_\rho$. Let $M := Q + SBR^{-1}B^\top S \succ 0$, the condition is equivalent to:

$$\forall x \in \mathcal{B}_\rho, \quad \sum_{k=1}^d (S_k \cdot x) \tilde{H}^k(x) \prec I_d,$$

where $\tilde{H}^k(x) := M^{-1/2} H^k(x) M^{-1/2}$.

To simplify the problem, we will decouple the two dependencies in x . The contribution of $S_k \cdot x$ will be bounded by two different bounds below. The tensor $\tilde{H}(x)$ is bounded globally, independently from ρ . A tighter analysis of the Hessian will be discussed in Section 5. For now, assume that we have an oracle on the magnitude $e_i^\top H^k e_j$ (e_i being the i -th unit vector) of H along d^2 directions for each H^k :

$$\forall x, \tilde{H}(x) \in \Xi := \{T \in \mathbb{R}^{d^3} \mid \forall i, j, k, |T_{ij}^k| \leq u_{ij}^k, T^{k^\top} = T^k\}. \quad (6)$$

A relaxation of the problem is then:

$$\sup_{x^\top S x \leq \rho} \sup_{T \in \Xi} \lambda_{\max} \left(\sum_{k=1}^d (S_k \cdot x) T^k \right) < 1.$$

With a simple change of variable and rescaling, the largest ρ fulfilling the condition is given by:

$$\rho = \frac{1}{\lambda^2}, \text{ where } \lambda := \sup_{\|y\|_2 \leq 1} \sup_{T \in \Xi} \lambda_{\max} \left(\sum_{k=1}^d (S_k^{1/2} y) T^k \right). \quad (7)$$

4.2. Two Upper Bounds on λ . The first bound is based on the following fact:

$$\sup_{\|y\|_2 \leq 1} \sup_{T \in \Xi} \left\| \sum_{k=1}^d (S_k^{1/2} y) T^k \right\|_2 \leq \sup_{\|y\|_2 \leq 1} \sum_{k=1}^d |S_k^{1/2} y| \sup_{T^k \in \Xi^k} \|T^k\|_2,$$

where Ξ^k is the projection of Ξ onto its k -th coordinate subspace. Let Z a matrix with lines $Z_k := (\sup_{T \in \Xi^k} \|T^k\|_2) S_k^{1/2}$,

$$\lambda \leq \sup_{\|y\|_2 \leq 1} \|Zy\|_1 \leq \sqrt{d} \sup_{\|y\|_2 \leq 1} \|Zy\|_2 = \sqrt{d} \|Z\|_2.$$

With equation (7), this guarantees that \mathcal{B}_{ρ_b} is an ROA for

$$\rho_b := \frac{1}{d \|DS^{1/2}\|_2^2}, \text{ with } D = \text{Diag} \left(\left(\sup_{T^k \in \Xi^k} \|T^k\|_2 \right)_k \right). \quad (8)$$

The following proposition explains how to compute D .

Proposition 2. Let ξ be the set of symmetric $d \times d$ matrices A such that for all $i, j \in \{1, \dots, d\}$, $|A_{ij}| \leq u_{ij}$. Let U be the nonnegative symmetric matrix with entries (u_{ij}) . Then:

$$\max_{A \in \xi} \|A\|_2 = \|U\|_2.$$

Proof. Since ξ is centered around 0, we only look for the largest eigenvalue:

$$\sup_{\|x\|_2 \leq 1, A \in \xi} x^\top A x = \sup_{\|x\|_2 \leq 1} \sup_{A \in \xi} \sum_i a_{ii} x_i^2 + 2 \sum_{i < j} a_{ij} x_i x_j.$$

Maximizing with respect to A , we get $a_{ii} = u_{ii}$, and

$$\forall i < j, \quad a_{ij} = \begin{cases} u_{ij} & \text{if } x_i x_j \geq 0 \\ -u_{ij} & \text{else.} \end{cases}$$

And then, $x^\top A x = \sum_i u_{ii} x_i^2 + 2 \sum_{i < j} u_{ij} |x_i x_j| = |x|^\top U |x|$. The full problem becomes:

$$\sup_{\|x\|_2 \leq 1, x \geq 0} x^\top U x.$$

The Perron-Frobenius theorem ensures that for any nonnegative square matrix, there exists a nonnegative real eigenvalue with at least one nonnegative eigenvector. Any other eigenvalue's modulus is smaller than this eigenvalue. U being symmetric, all its eigenvalues are real, hence the result. \square

Remark: If the bounds on the entries of A are not centered around 0, it is possible to write $A = A' + \bar{A}$, where the entries of \bar{A} are symmetrically bounded, and $\|A\|_2 \leq \|A'\|_2 + \|\bar{A}\|_2$.

The proposition leads to another bound on λ . If $T \in \Xi$:

$$\forall i, j, k, \quad \left| \sum_k (S_k^{1/2} y) T_{ij}^k \right| \leq \sum_k |S_k^{1/2} y| u_{ij}^k \leq \sum_k \|S_k^{1/2} y\| \cdot \|y\| u_{ij}^k.$$

Applying the previous result to the matrix whose entries are the middle term in the inequality above:

$$\lambda \leq \lambda_{\max} \left(\sum_k \sqrt{S_k \cdot S_k^{-1} S_k^\top U^k} \right) =: \lambda_a, \quad (9)$$

which guarantees that \mathcal{B}_{ρ_a} is an ROA for $\rho_a := 1/\lambda_a^2$.

5. ITERATIVE ALGORITHM

5.1. Stability Certificates. The bounds presented in Sections 3 and 4 readily give robust stability certificates that hold for a whole class of dynamics with suitably bounded derivatives. It is also possible to apply the same methods to a single known dynamics, if its derivatives can be bounded efficiently. In general the bounds on the derivatives depend on where they are computed: the larger the region, the larger the bounds. But such bounds must be computed on a sufficiently large region containing the sublevel set where stability is asserted.

With the notations of the previous two sections, and given equations (5,8,9), we define three stability certificates:

$$\begin{aligned} \mathcal{C}_1 &: (S, \rho_{up}, \Omega) \rightarrow \rho_{up} \mathbf{1}_{\text{LMI (5) is feasible}} \\ \mathcal{C}_2^{a,b} &: (S, \rho_{up}, \Xi) \rightarrow \min(\rho_{a,b}, \rho_{up}), \end{aligned}$$

meaning that $\{x^\top S x \leq \mathcal{C}_1(S, \rho_{up}, \Omega)\}$ is an ROA if the derivatives of the dynamics are bounded by Ω (resp. Ξ) and if ρ_{up} is an upper bound on ρ used to compute Ω (resp. Ξ).

5.2. Oracle on the Derivatives. Suppose we have an oracle \mathcal{O} computing, on a domain \mathcal{D} , a bound $\mathcal{O}(\mathcal{D})$ on the derivatives, corresponding to sets Ω or Ξ above. Our methods compute $\rho = \mathcal{C}(S, \rho_{up}, \mathcal{O}(\mathcal{D}))$. Then \mathcal{B}_ρ is an ROA if the whole trajectory to 0 stays inside \mathcal{D} (else the assumptions on the derivatives would be violated). A simple way to ensure that is to choose \mathcal{D} as a sublevel set of V containing \mathcal{B}_ρ , i.e., $\mathcal{B}_{\rho_{up}}$ for $\rho_{up} \geq \rho$.

For a quadratic dynamical system, each entry of the Jacobian is an affine function and each entry of the Hessian is constant. Exact entrywise bounds on an ellipsoid are:

$$\sup_{x^\top S x \leq \rho_{up}} c^\top x = \sqrt{\rho_{up}} \|S^{-1/2} c\|_2.$$

For third-order polynomial systems, the entries of the Hessian are affine and those of the Jacobian are polynomials of degree two. For a quadratic monomial, this formula can be used:

$$\sup_{x^\top S x \leq \rho_{up}} x^\top J x = \rho_{up} \lambda_{\max}(S^{-1/2} J S^{-1/2}).$$

In large dimensions, manually identifying each coefficient of the derivatives of a second or third order polynomial dynamics might be tedious: the Hessian tensor has d^3 entries. One solution is to work with symbolic expressions. Another one is to sample derivatives at a few different points with automatic differentiation [PGM⁺ 19], to fit a low order polynomial model, and then to maximize it in closed form.

For generic dynamics, one can sample derivatives, *e.g.*, by automatic differentiation, using analytical derivatives [CM18a] for rigid body dynamics, or from direct physical measurements on the system. The sampling process can be done in parallel. Bounding the samples provides estimates of the oracle, and efficiency improvements can be expected with Bayesian optimization [Moc12] or other global optimization tools.

5.3. Algorithm. A simple ROA estimation algorithm (see Algorithm 1) consists in iteratively bounding the derivatives and producing stability certificates, *i.e.*, alternating calls of \mathcal{O} and \mathcal{C} . ρ_0 is an initial upper bound on the size of the ROA. Each step of the loop provides a certificate that \mathcal{B}_ρ is an ROA, and this region grows at each iteration, the sequence of ρ s being nondecreasing. The number of iterations before the algorithm stops depends on the initial guess ρ_0 and the step size η . In our experiments, typically we use 10 to 20 iterations.

Algorithm 1 Adaptive stability certificates

Input: $S, \mathcal{C}(), \mathcal{O}(), \rho_0 > 0, \eta \in (0, 1)$
Output: An ROA certificate on $\{x \mid x^\top Sx \leq \rho\}$

- 1: $\rho_{up} \leftarrow \rho_0$
- 2: **repeat**
- 3: $U \leftarrow \mathcal{O}(\mathcal{B}_{\rho_{up}})$
- 4: $\rho \leftarrow \mathcal{C}(S, \rho_{up}, U)$
- 5: $\rho_{up} \leftarrow \eta \rho_{up}$
- 6: **until** $\rho \geq \rho_{up}$
- 7: **return** ρ

6. TRAJECTORY TRACKING

Let $(x_0(t), u_0(t))$, for $t \in [0, t_f]$ be a reference trajectory, with final state $x_f := x_0(t_f)$. For a nearby trajectory $(x(t), u(t))$, let $\bar{x}(t) := x(t) - x_0(t)$, $\bar{u}(t) := u(t) - u_0(t)$. The linearized dynamics is:

$$\dot{\bar{x}}(t) = A(t)\bar{x}(t) + B(t)\bar{u}(t) + o(\bar{x}(t)) + o(\bar{u}(t)).$$

Let \mathcal{B}_f a target region $\{x \mid (x - x_f)^\top S_f(x - x_f) \leq 1\}$, for some $S_f \succeq 0$. We define the finite-horizon LQR problem [Lib11] with the following tracking cost:

$$\int_0^{t_f} (\bar{x}(t)^\top Q \bar{x}(t) + \bar{u}(t)^\top R \bar{u}(t)) dt + \bar{x}^\top(t_f) S_f \bar{x}(t_f).$$

For $t \in [0, t_f]$, the optimal cost-to-go is $V(x, t) = \bar{x}^\top S(t) \bar{x}$, $S(t)$ being the solution of the Riccati differential equation (RDE):

$$\dot{S} = -Q + SBR^{-1}B^\top S - SA - A^\top S, \quad S(t_f) = S_f, \quad (10)$$

with controller $\bar{u}(t) = -K(t)\bar{x}(t) := -R^{-1}B^\top(t)S(t)\bar{x}(t)$.

We want to estimate the time-varying region (called “funnel” [TMTR10]) $\mathcal{B}(t) := \{x \mid F(x, t) \in \mathcal{B}_f\}$, where $F(x, t)$ is the integrated closed-loop dynamics with control $u(\cdot)$ from t to t_f . In particular, $\mathcal{B}(t_f) = \mathcal{B}_f$. $\mathcal{B}(t)$ is a region where applying $u(t) = u_0(t) + \bar{u}(t)$ will make the trajectory reach $\mathcal{B}(t_f)$ after time t_f . If in addition $\mathcal{B}(t_f)$ is included in an ROA around 0, the trajectory will then finally reach 0 in finite time.

We consider regions $\mathcal{B}(t) := \{x \mid 0 \leq V(x, t) \leq \rho(t)\}$. A sufficient condition for $\mathcal{B}(t)$ to be a funnel is [TMT11]:

$$V(x, t) \geq 0, \quad \forall x \in \mathcal{B}(t) \text{ and } \dot{V}(x, t) \leq \dot{\rho}(t), \quad \forall x \in \partial \mathcal{B}(t).$$

We drop some occurrences of the time variable to simplify the notations. $S(t)$ being a positive definite matrix [Lib11] for any $t \in [0, t_f]$, the first condition holds, the second one is:

$$\dot{V}(x, t) = 2\bar{x}^\top S \dot{\bar{x}} + \bar{x}^\top \dot{S} \bar{x} \leq \dot{\rho}, \quad \forall x \in \left\{x \mid \bar{x}^\top S \bar{x} = \rho\right\}.$$

If the closed-loop system is an LDI $\dot{\bar{x}} = \tilde{A}(t, x)\bar{x}$ in $\{x \mid \bar{x}^\top S \bar{x} = \rho\}$, with $\tilde{A} \in \Omega(\rho)$, a sufficient condition is:

$$\forall \tilde{A} \in \Omega(\rho), \quad \tilde{A}^\top S + S \tilde{A} + \dot{S} - \frac{\dot{\rho}}{\rho} S \preceq 0. \quad (11)$$

This can be fit into the LDI framework presented in Section 3, just by shifting the set $\Omega(\rho)$ to the set

$$\tilde{\Omega}(\rho, \dot{\rho}) := \left\{ \tilde{A} + \frac{1}{2}S^{-1}\dot{S} - \frac{1}{2}\frac{\dot{\rho}}{\rho}I_d \mid \tilde{A} \in \Omega(\rho) \right\}.$$

Now if the closed-loop system is known up to order two in $\{x \mid \bar{x}^\top S \bar{x} = \rho\}$, say

$$\dot{\bar{x}} = (A - BK)\bar{x} + \frac{1}{2}\bar{x}^\top H(t, x)\bar{x},$$

with $H(t, x) \in \Xi(\rho)$. Using that $S(\cdot)$ is a solution of equation (10), we obtain the following sufficient condition: $\forall y$ such that $\|y\|_2^2 = 1$, $\forall H \in \Xi(\rho)$,

$$-Q - SBR^{-1}B^\top S - \frac{\dot{\rho}}{\rho}S + \sqrt{\rho} \sum_{k=1}^d (S_k^{1/2}y)H^k \preceq 0.$$

If $N(\rho, \dot{\rho}) = Q + SBR^{-1}B^\top S + \frac{\dot{\rho}}{\rho}S \succ 0$, let $\tilde{\Xi}(\rho, \dot{\rho})$ the shifted set $\{N^{-1/2}HN^{-1/2} \mid H \in \Xi(\rho)\}$, we must check that:

$$\forall \|y\|_2^2 = 1, \forall \tilde{H} \in \tilde{\Xi}(\rho, \dot{\rho}), \sqrt{\rho} \sum_{k=1}^d (S_k^{1/2}y)\tilde{H}^k \preceq I_d. \quad (12)$$

Under such conditions, $\rho(\cdot)$ is built backwards in time by integration, beginning with $\rho(t_f) = 1$. At each time step, given ρ , a greedy strategy is to choose $\dot{\rho}$ as the smallest possible value such that the sufficient condition is enforced. Since $\rho(\cdot)$ is computed backwards, this maximizes $\rho(t - dt)$, hence locally the funnel's volume. Also, in both first and second order cases, the sufficient condition is monotonically more restrictive as $\dot{\rho}$ decreases. A simple algorithm is to start with a large positive $\dot{\rho}$, compute the set $\tilde{\Omega}(\rho, \dot{\rho})$ or $\tilde{\Xi}(\rho, \dot{\rho})$, check that the sufficient condition holds, and progressively decrease $\dot{\rho}$ until it no longer does (possibly with $\dot{\rho} < 0$ if $N \succ 0$ is still enforced, when applicable).

7. NUMERICAL EXPERIMENTS

7.1. Definition of the Systems and Implementation Details. The code to reproduce the experiments is available online¹. The first two systems, an electrical oscillator and a floating satellite with commanded torques, are taken from the Matlab material of [TMT11]. The third one is a double pendulum, with an actuated joint between the two arms (also called ‘‘acrobot’’ in [Sut96]). The last one corresponds to the UR5 robotic arm from *Universal Robots*², with 6 actuated joints.

Vanderpol. $d = 2, m = 0$ (unactuated), $x_0 = \mathbf{0}_2^\top, Q = I_2$. The dynamics is a polynomial of degree 3:

$$\forall x = (x_1, x_2) \in \mathbb{R}^2, \quad f(x) = (-x_2, x_1 + x_2(x_1^2 - 1))^\top.$$

Satellite. $d = 6, m = 3, (x_0, u_0) = (\mathbf{0}_6^\top, \mathbf{0}_3^\top), Q = I_6, R = 10 \times I_3$, the dynamics is a polynomial of degree 3. Let $J = \text{Diag}(5, 3, 2)$. For $x = (\omega^\top, \sigma^\top)^\top \in \mathbb{R}^6$, with $\omega, \sigma \in \mathbb{R}^3, f(x, u) = (\dot{\omega}^\top, \dot{\sigma}^\top)^\top$,

$$\begin{aligned} \dot{\omega} &= J^{-1}(u - \omega \times J\omega) \\ \dot{\sigma} &= \frac{1}{4} \left((1 - \|\sigma\|^2)I_3 + 2\sigma\sigma^\top - 2 \begin{bmatrix} 0 & \sigma_3 & \sigma_2 \\ \sigma_3 & 0 & \sigma_1 \\ \sigma_2 & \sigma_1 & 0 \end{bmatrix} \right) \omega. \end{aligned}$$

Pendulum. $d = 4, m = 1, u_0 = 0, x_0 = \mathbf{0}_4^\top$ (bottom) or $x_0 = (\pi, \pi, 0, 0)^\top$ (top), $Q = I_4, R = 1$. Let $g = 9.8, \ell = 0.5$ and $\mu = 1$. For $x = (\theta_1, \theta_2, p_1, p_2)^\top, f(x, u)$ is defined by:

$$\begin{aligned} \dot{\theta}_1 &= \frac{6}{\mu\ell^2} \frac{2p_1 - 3\cos(\theta_1 - \theta_2)p_2}{16 - 9\cos^2(\theta_1 - \theta_2)} \\ \dot{\theta}_2 &= \frac{6}{\mu\ell^2} \frac{8p_2 - 3\cos(\theta_1 - \theta_2)p_1}{16 - 9\cos^2(\theta_1 - \theta_2)} \\ \dot{p}_1 &= -\frac{\mu\ell^2}{2} \left(\dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + \frac{3g}{\ell} \sin \theta_1 \right) \\ \dot{p}_2 &= -\frac{\mu\ell^2}{2} \left(-\dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + \frac{g}{\ell} \sin \theta_1 \right) + u. \end{aligned}$$

¹www.github.com/eloiseberthier/Fast-Robust-ROA

²www.universal-robots.com/products/ur5-robot

TABLE 1. Radius and volume of the certified ROA for the different methods, relative to the values obtained by sampling for reference.

Dynamics	\mathcal{C}_1		\mathcal{C}_2^a		\mathcal{C}_2^b		SOS		sampling	
	ρ/ρ_s	v/v_s	ρ/ρ_s	v/v_s	ρ/ρ_s	v/v_s	ρ/ρ_s	v/v_s	ρ/ρ_s	v/v_s
Vanderpol	0.20	0.20	0.14	0.14	0.10	0.10	1	1	1	1
Satellite	2.9×10^{-2}	2.6×10^{-5}	9.3×10^{-2}	9.4×10^{-4}	7.9×10^{-2}	5.7×10^{-4}	0.93	0.82	1	1
Pend. (bot.)	3.2×10^{-2}	1.1×10^{-3}	3.5×10^{-2}	1.2×10^{-3}	4.2×10^{-2}	1.9×10^{-3}	1.4×10^{-2}	2.0×10^{-4}	1	1
Pend. (top)	5.1×10^{-3}	2.6×10^{-5}	4.5×10^{-2}	2.0×10^{-3}	4.7×10^{-2}	2.2×10^{-3}	N.A.	N.A.	1	1
Robot	2.4×10^{-3}	1.8×10^{-16}	7.1×10^{-3}	1.5×10^{-13}	1.5×10^{-2}	1.2×10^{-11}	N.A.	N.A.	1	1

TABLE 2. CPU time (s) per iteration, except for SOS (total time).

Dynamics	$\mathcal{O} + \mathcal{C}_1$	$\mathcal{O} + \mathcal{C}_2^a$	$\mathcal{O} + \mathcal{C}_2^b$	SOS
Vanderpol	1.8×10^{-3}	1.1×10^{-4}	1.6×10^{-4}	0.05
Satellite	1.2	0.17	0.17	32
Pend. (bot.)	2.3	15	15	132
Robot	2.3	32	33	N.A.

Robot. $d = 12$, $m = 6$, $x_0 = (q_0^\top, \mathbf{0}_6)^\top$, q_0 is the configuration $q_0 = (0, -\pi/5, -3\pi/5, 0, 0, 0)$, $Q = I_{12}$, $R = I_6$. u_0 is such that $f(x_0, u_0) = 0$ and is computed by the recursive Newton-Euler algorithm (RNEA) implemented in the C++ library Pinocchio [CVM⁺19] coming with Python bindings. The forward dynamics $f(x, u)$ is computed as a black box with the so-called articulated body algorithm (ABA).

The software used for the SOS based certificates is adapted from the Matlab material of [TMT11]. The oracles on the derivatives are computed either in closed form, for *Vanderpol* and the Hessian of *Satellite*, or by sampling p derivatives. Using automatic differentiation in PyTorch [PGM⁺19], we sample $p = 10^4$ Jacobians for *Satellite*, $p = 10^3$ Jacobians and Hessians for *Pendulum*. For *Robot*, $p = 5 \times 10^4$ and the Jacobians of the dynamics are computed analytically [CM18a], and we use finite differences on the first partial derivatives for the Hessians. It is important to notice at this stage that more advanced methods to efficiently compute these Hessians could improve the whole computation time of our methods, for instance by code-generating the second-order derivatives computed by automatic differentiation. Yet, the proposed solution already provides competitive timings.

7.2. Results. The performances of the certificates are compared in Table 1, in terms of radius of \mathcal{B}_ρ , and volume $v \propto \rho^{d/2} / \sqrt{|S|}$, the latter exacerbating differences in large dimensions. The volume, divided by the volume of the state space, is roughly the inverse of the number of ROAs that would have covered it. All the values in the table are divided by the ground truth ρ_s , the maximal ρ such that $\forall x^\top Sx \leq \rho$, $\dot{V}(x) < 0$, estimated by sampling a very large number of points. Apart from SOS on the first two problems, all methods are very far from estimating the true maximal ROA.

For the SOS method on *Pendulum*, because the dynamics is non polynomial, we substitute the odd function f by its Taylor expansion around the equilibrium, truncated at order $n = 7$. The result is sensitive to the order: for $n = 2$, f is linear hence $\rho = +\infty$, whereas ρ decreases for higher orders. It is unclear which one to choose, and the results are no longer certified. At the top position, an utterly unstable position, SOS fails to provide a positive ρ , regardless of $n \geq 3$.

Table 2 reports the corresponding CPU running times on a standard laptop. The code, in Python, is not optimized, except the SOS method and the LMI solver for \mathcal{C}_1 which are in Matlab. Our methods are much lighter than SOS, yet one must keep in mind that Algorithm 1 typically calls the oracle and the certificate 10 times. Nonetheless, this allows to tackle systems of larger dimensions, like *Robot*. If the oracle uses sampling, this dominates the running time. Figure 1 compares the running times of bounding the derivatives for \mathcal{C}_1 and \mathcal{C}_2^a , depending on the number of samples p , on *Satellite*. At fixed p , it is of course longer to sample Hessians than Jacobians. The sampling oracle overestimates ρ , but this tends to stabilize for reasonable values of p , as seen for ρ_2^a which can also be computed using a closed form oracle.

We also experiment trajectory tracking on a given trajectory of *Vanderpol*, with $x_f = (-1, -1)^\top$, $t_f = 2$. The target region $\mathcal{B}_f = \{x \mid \bar{x}^\top S_f \bar{x} \leq 1\}$ is the largest ellipsoid included in \mathcal{R} , an ROA around 0 computed

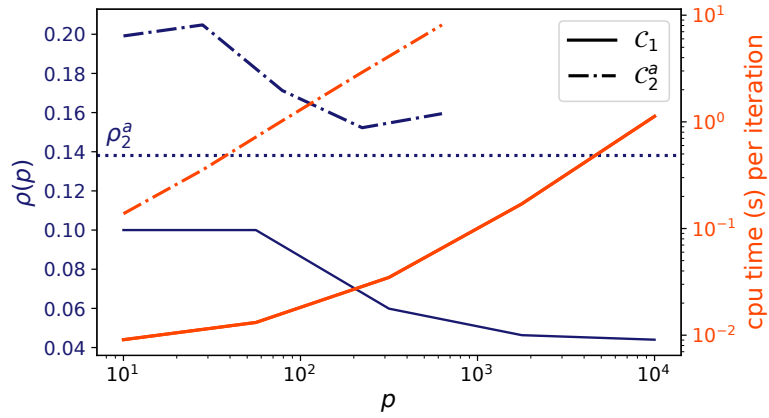


FIGURE 1. Results of \mathcal{C}_1 , \mathcal{C}_2^a , and cpu time on *Satellite*, depending on p .

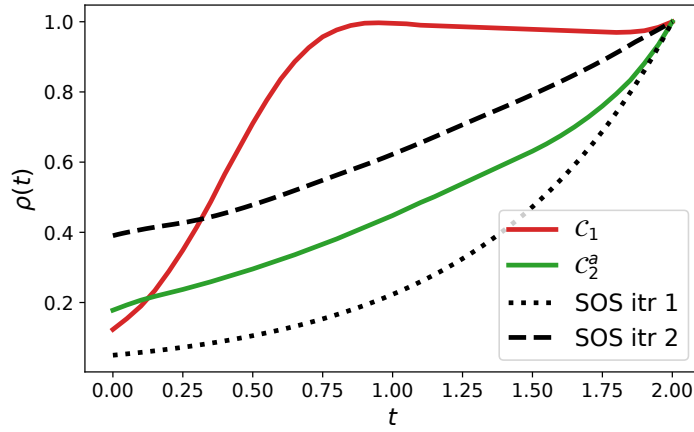


FIGURE 2. $\rho(t)$ with different certificates, around a trajectory of *Vanderpol*. The total CPU time is 7s for two iterations of SOS, roughly 1s for \mathcal{C}_1 , \mathcal{C}_2^a .

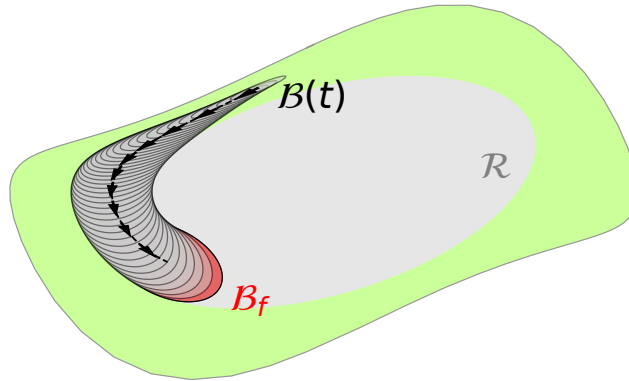


FIGURE 3. A funnel $\mathcal{B}(t)$ around a trajectory of *Vanderpol*, obtained with \mathcal{C}_1 . The state-space is in green, \mathcal{R} in light gray is an ROA around 0, and \mathcal{B}_f in red is the target region. The reference trajectory is displayed with arrows.

by SOS. In Figure 3, the state-space is in green, \mathcal{R} in light gray and \mathcal{B}_f in red. The funnel $\mathcal{B}(t)$, in gray, is computed backwards, with one or two iterations of the SOS-based algorithm of [TMT11], and with the methods of Section 6. Figure 2 shows our certificates lead to competitive values of $\rho(t)$, with faster computations.

8. CONCLUSION

The stability certificates presented in this paper are both fast to compute, and robust over a class of bounded-derivatives dynamics. They readily extend to the trajectory tracking problem, with a linear complexity in the number of time steps. Such certificates are easily implemented and enable handling non-polynomial, large dimensional control systems that were previously out of reach. The complexity is transferred from the certificate to a derivative-bounding oracle, which can be estimated efficiently in some cases, including rigid body dynamics in robotics. The certificates for trajectory tracking can in turn be integrated into the LQR-trees framework for global motion planning. They are more conservative than competing methods, yet faster, hence repeating calls to these certificates could be more efficient overall. Providing empirical evidence or counter-evidence for this trade-off phenomenon in real-world control systems would be an interesting avenue for future research.

ACKNOWLEDGEMENTS

This work was supported by the Direction Générale de l'Armement, and by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

REFERENCES

- [AC84] Jean-Pierre Aubin and Arrigo Cellina, *Differential inclusions: Set-valued maps and viability theory*, Springer, 1984.
- [BEGFB94] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan, *Linear matrix inequalities in system and control theory*, vol. 15, Siam, 1994.
- [Che04] Graziano Chesi, *Estimating the domain of attraction for uncertain polynomial systems*, *Automatica* **40** (2004), no. 11, 1981–1986.
- [CM18a] Justin Carpentier and Nicolas Mansard, *Analytical derivatives of rigid body dynamics algorithms*, Robotics: Science and Systems, 2018.
- [CM18b] ———, *Multicontact locomotion of legged robots*, *IEEE Transactions on Robotics* **34** (2018), no. 6, 1441–1460.
- [CVM⁺19] Justin Carpentier, Florian Valenza, Nicolas Mansard, et al., *Pinocchio: fast forward and inverse dynamics for poly-articulated systems*, <https://stack-of-tasks.github.io/pinocchio>, 2015–2019.
- [GH15] Peter Giesl and Sigurdur Hafstein, *Review on computational methods for Lyapunov functions*, *Discrete and Continuous Dynamical Systems-Series B* **20** (2015), no. 8, 2291–2331.
- [GNS⁺17] Markus Gifithaler, Michael Neunert, Markus Stäuble, Marco Frigerio, Claudio Semini, and Jonas Buchli, *Automatic differentiation of rigid body dynamics for optimal control and estimation*, *Advanced Robotics* **31** (2017), no. 22, 1225–1237.
- [GSM90] K Glover, J Sefton, and DC McFarlane, *A tutorial on loop shaping using H-infinity robust stabilization*, *IFAC Proceedings Volumes* **23** (1990), no. 8, 117–126.
- [Joh00] Tor A Johansen, *Computation of Lyapunov functions for smooth nonlinear systems using convex optimization*, *Automatica* **36** (2000), no. 11, 1617–1626.
- [Lib11] Daniel Liberzon, *Calculus of variations and optimal control theory: a concise introduction*, Princeton University Press, 2011.
- [LK01] Steven M LaValle and James J Kuffner, *Rapidly-exploring random trees: Progress and prospects*, *Algorithmic and Computational Robotics: New Directions* (2001), no. 5, 293–308.
- [Moc12] Jonas Mockus, *Bayesian approach to global optimization: Theory and applications*, vol. 37, Springer Science & Business Media, 2012.
- [NN94] Yurii Nesterov and Arkadii Nemirovskii, *Interior-point polynomial algorithms in convex programming*, SIAM, 1994.
- [PGM⁺19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala, *PyTorch: An imperative style, high-performance deep learning library*, *Advances in Neural Information Processing Systems* 32, 2019, pp. 8024–8035.
- [SCH⁺18] Sumeet Singh, Mo Chen, Sylvia L Herbert, Claire J Tomlin, and Marco Pavone, *Robust tracking with model mismatch for fast and safe planning: an SOS optimization approach*, *International Workshop on the Algorithmic Foundations of Robotics*, Springer, 2018, pp. 545–564.
- [SL91] Jean-Jacques E Slotine and Weiping Li, *Applied nonlinear control*, vol. 199, Prentice hall Englewood Cliffs, NJ, 1991.
- [Sut96] Richard S Sutton, *Generalization in reinforcement learning: Successful examples using sparse coarse coding*, *Advances in Neural Information Processing Systems*, 1996, pp. 1038–1044.
- [TMT11] Mark M Tobenkin, Ian R Manchester, and Russ Tedrake, *Invariant funnels around trajectories using sum-of-squares programming*, *IFAC Proceedings Volumes* **44** (2011), no. 1, 9218–9223.
- [TMTR10] Russ Tedrake, Ian R Manchester, Mark Tobenkin, and John W Roberts, *LQR-trees: Feedback motion planning via sums-of-squares verification*, *The International Journal of Robotics Research* **29** (2010), no. 8, 1038–1052.
- [TP07] Ufuk Topcu and Andrew Packard, *Stability region analysis for uncertain nonlinear systems*, *IEEE Conference on Decision and Control*, 2007, pp. 1693–1698.

APPENDIX A. IMPLEMENTATION SUMMARY

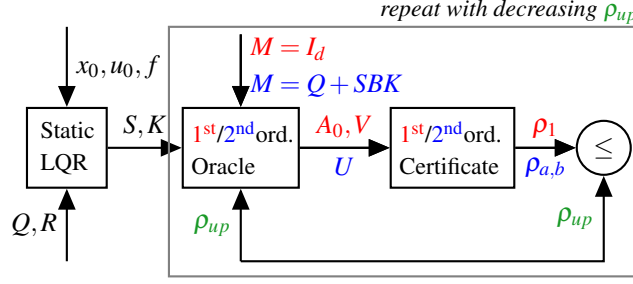


FIGURE 4. ROA estimation algorithm. Elements specific to the 1st order method are in red, to the 2nd order in blue. Framed steps are repeated until $\rho \geq \rho_{up}$.

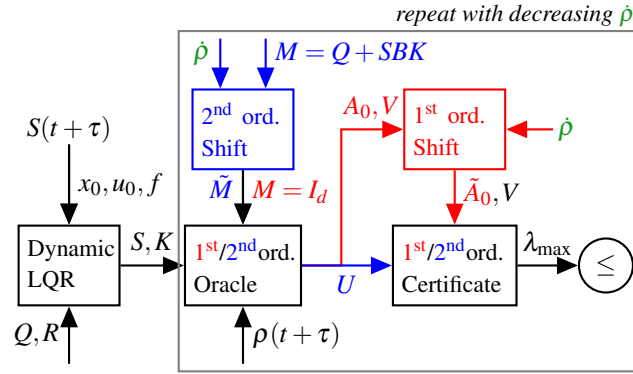


FIGURE 5. Trajectory tracking algorithm for one time-step. Framed steps are repeated while equations (11) for the 1st order, or (12) for the 2nd order hold.

The complete ROA estimation and trajectory tracking frameworks are summarized respectively in Figure 4 and 5. Each building block used in the diagrams is detailed below. The first one computes the LQR as in Section 2.

- **Static LQR:** $(Q, R, x_0, u_0, f) \rightarrow (S, K)$.

$A = \frac{\partial f}{\partial x}(x_0, u_0)$, $B = \frac{\partial f}{\partial u}(x_0, u_0)$, S is the positive definite solution of $A^\top S + SA - SBR^{-1}B^\top S = -Q$, and $K = R^{-1}B^\top S$.

The following block computes the LQR for one time step of trajectory tracking and is detailed in Section 6.

- **Dynamic LQR:** $(S(t+\tau), Q, R, x_0(t), u_0(t), f) \rightarrow (S(t), K(t))$.

Let $\bar{S} = S(t+\tau)$, then $S(t) = \bar{S} - \tau \dot{S}$, with $A = \frac{\partial f}{\partial x}(x_0(t), u_0(t))$, $B = \frac{\partial f}{\partial u}(x_0(t), u_0(t))$,

$$\dot{S} = -Q - \bar{S}A - A^\top \bar{S} + \bar{S}BR^{-1}B^\top \bar{S}, \text{ and } K(t) = R^{-1}B^\top S(t).$$

The next two blocks compute bounds on the derivatives of the dynamics. They can be implemented arbitrarily.

- **First-order oracle:** $(\rho_{up}, S, K, M, x_0, u_0, f) \rightarrow (A_0, V)$.

$$V_{ij} := \sup_{x^\top Sx \leq \rho_{up}} |J_{ij}(x) - (A_0)_{ij}|,$$

where J is the Jacobian of $x \mapsto f(x_0 + x, u_0 - Kx)$. A default choice for A_0 is $J(0) = A - BK$.

- **Second-order oracle:** $(\rho_{up}, S, K, M, x_0, u_0, f) \rightarrow U$.

$$U_{ij}^k := \sup_{x^\top Sx \leq \rho_{up}} \left[M^{-1/2} H^k(x) M^{-1/2} \right]_{ij},$$

where H is the Hessian of $x \mapsto f(x_0 + x, u_0 - Kx)$.

The next two blocks compute stability certificates, as detailed in Sections 3 and 4.

- **First-order certificate:** $(\rho_{up}, S, A_0, V) \rightarrow \rho_1 = \rho_{up} \mathbf{1}_{\text{LMI}}$ is feasible.

Let C, E defined as in section 3. The LMI feasibility problem is to find $\Lambda \succeq 0 \in \mathbb{R}^{d^2 \times d^2}$ diagonal such that:

$$\begin{bmatrix} A_0^\top S + SA_0 + E^\top \Lambda E & SC \\ C^\top S & -\Lambda \end{bmatrix} \prec 0.$$

- **Second-order certificate:** $(\rho_{up}, S, U) \rightarrow \rho_{a,b} = \frac{1}{\lambda_{a,b}^2}$.

$$\lambda_a = \lambda_{\max} \left(\sum_k \sqrt{S_k \cdot S^{-1} S_k^\top} U^k \right),$$

$$\lambda_b = \sqrt{d} \|DS^{-1/2}\|_2, \text{ with } D = \text{Diag}(\|U_k\|_2)_k.$$

The last two blocks are used in Section 6.

- **First-order shift:** $(A_0, \dot{\rho}, \rho(t+\tau), S(t), \dot{S}(t)) \rightarrow \tilde{A}_0$.

$\tilde{A}_0 = A_0 + \frac{1}{2} S^{-1} \dot{S} - \frac{1}{2} \frac{\dot{\rho}}{\rho} I_d$, where \dot{S} is given by the RDE (equation (10)).

- **Second-order shift:** $(M, \dot{\rho}, \rho(t+\tau), S(t)) \rightarrow \tilde{M} = M + \frac{\dot{\rho}}{\rho} S$.