Fast and Robust Stability Region Estimation for Nonlinear Dynamical Systems

European Control Conference 2021

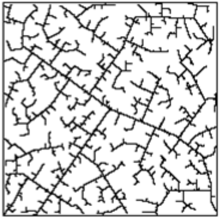
Eloïse Berthier, Justin Carpentier, Francis Bach

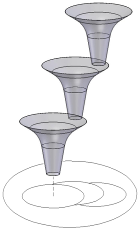
July 1, 2021

Motivation: Feedback Motion Planning

Three steps in *LQR-trees* [TMTR10]:

1. create a tree of trajectories [LK01];
2. **find a stability region around each trajectory**;
3. deduce a global controller [BRK99, TMT11].





1. Stability of the LQR Feedback
2. First-Order Perturbation
3. Second-Order Perturbation
4. Numerical Experiments

LQR Feedback Controller

Let a **nonlinear** controlled dynamical system:

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

with equilibrium point: $f(0,0) = 0$. Then:

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

Consider the LQR problem for the **linearized system** $\dot{x} = Ax + Bu$:

$$\min_{u(\cdot)} \int_0^{+\infty} (x^\top(t)Qx(t) + u^\top(t)Ru(t))\mathrm{d}t, \text{ with } x(0) = x.$$

LQR Feedback Controller

The cost-minimizing controller is:

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

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

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

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

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

Stability Region Estimation (fixed controller)

Find a maximal region $\mathcal{R}$ containing states x_0 such that

$$x(0) = x_0, \quad \dot{x} = f(x, -Kx) \implies \lim_{t \to +\infty} x(t) = 0.$$

One approach is to find a **Lyapunov function** V and a **region** $\mathcal{R}$ such that:

- $V(0) = 0$,
- $\forall x \in \mathcal{R} \setminus \{0\}, \; V(x) > 0$,
- $\forall x \in \mathcal{R} \setminus \{0\}, \; \dot{V}(x) < 0$.

Stability Region Estimation

A natural candidate Lyapunov function is the **LQR cost-to-go**:

$$V(x) = x^\top S x \geq 0.$$

The candidate $\mathcal{R}$ are the **sublevel sets of** V:

$$\mathcal{B}_\rho := \{x \mid x^\top S x \leq \rho\}.$$

Stability Region Estimation (fixed controller, fixed Lyapunov)

Find the maximal ρ such that

$$V(x) \leq \rho, \; x \neq 0 \implies \dot{V}(x) < 0.$$

Feedback Stability for Polynomial Systems

If the dynamics is **polynomial**, a sum-of-squares relaxation of the condition is that there exists a SOS polynomial $\sigma(x)$ such that:

$$\dot{V}(x) + \sigma(x)(\rho - V(x)) < 0.$$

In practice, this is solved with a **hierarchy of SDPs**, with a matrix of size $C_{n+d}^{d} \times C_{n+d}^{d}$, for $n \geq 1$.

$\rightarrow$ **intractable** in (not so) large dimensions $d \approx 10$.

Robustness for Nonlinear Systems

Back to the linear system:

$$g(x) = f(x, -Kx) = (A - BK)x.$$

Can we say something if the closed-loop system is *almost* linear, locally around the equilibrium?

$$g(x) = (A - BK)x + \delta(x).$$

This could account for **uncertainties** or **model misspecifications**, and hence the method is **robust**. We study two cases:

- First-order perturbation: $\delta(x) = \bar{A}x,\ \bar{A} \in \Omega,$
- Second-order perturbation: $\delta(x) = x^\top \bar{H} x,\ \bar{H} \in \Xi.$

Linear Differential Inclusion

Consider the **uncertain linear system** [AC84]:

$$g(x) = Ax, \ A \in \Omega,$$

where we know bounds on each entry of A:

$$\Omega = \{A_0 + \Gamma \mid \forall i, j, |\Gamma_{ij}| \leq U_{ij}\}.$$

This can be recast as $\Omega = \{A_0 + C\Delta E \mid \|\Delta\| \leq 1, \ \Delta \text{ diagonal}\}$, for suitable C and E. Δ has size $d^2 \times d^2$.

Linear Differential Inclusion

Suppose we have an **uncertain linear system**:

$$g(x) = Ax, \ A \in \Omega,$$

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

Stability of an LDI

The asymptotic stability of this system with a fixed Lyapunov function is equivalent to the feasibility of the following linear matrix inequality (LMI) [BEGFB94]:

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

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

Bounded-Hessians Systems

Suppose that we have:

$$g_k(x) = (A - BK)_{k.}x + \frac{1}{2}x^\top \bar{H}^k(x)x.$$

Then:

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

A **sufficient condition** for $\dot{V}(x) < 0$ is:

$$-Q - SBR^{-1}B^\top S + \sum_{k=1}^{d}(S_{k.}x)\bar{H}^k(x) \prec 0,\ \forall x \neq 0.$$

Bounded-Hessians Systems

Let $M := Q + SBR^{-1}B^\top S$, and
$T^k(x) := M^{-1/2}H^k(x)M^{-1/2}$.

If we know **entrywise bounds** on the rescaled Hessian T:

$$\Xi := \prod_{k=1}^{d} \Xi^k, \quad \Xi^k = \{T \in \mathbb{R}^{d\times d} \mid \forall i,j, |T_{ij}| \le U_{ij}^k\}.$$

Then we can extract the maximal sublevel set ρ:

Stability Region Estimation (bounded Hessian)

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

Stability Certificates

Theorem (A Second-Order Stability Certificate)

$\mathcal{B}_{\rho_a}$ is a stability region for $\rho_a := 1/\lambda_a^2$ and

$$\lambda_a := \lambda_{\max}\left(\sum_{k=1}^{d} \sqrt{S_{k.} S^{-1} S_{k.}^{\top}}\, U^k\right).$$

Theorem (Another Second-Order Stability Certificate)

$\mathcal{B}_{\rho_b}$ is a stability region for

$$\rho_b := \frac{1}{d\|DS^{1/2}\|_2^2}, \text{ with } D := \mathit{Diag}\Big(\|U^k\|_2\Big).$$

Iterative Algorithm

Two ingredients:

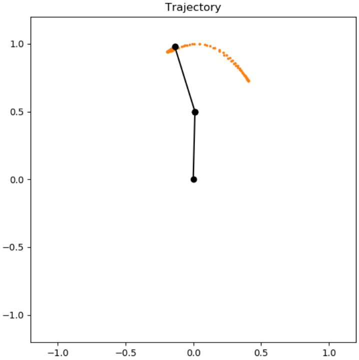
- $\mathcal{O}$ is an **oracle** bounding the derivatives of g on some region;
- $\mathcal{C}$ returns a **stability certificate**, given a candidate Lyapunov function S and bounds on the derivatives U.

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

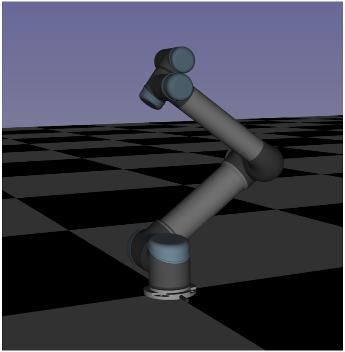
$\rho_{up} \leftarrow \rho_0$
repeat
  $U \leftarrow \mathcal{O}(\mathcal{B}_{\rho_{up}})$
  $\rho \leftarrow \mathcal{C}(S, \rho_{up}, U)$
  $\rho_{up} \leftarrow \eta \rho_{up}$
until $\rho \geq \rho_{up}$
return ρ

Examples of Dynamical Systems

We consider controlled dynamical systems of various dimensions around an equilibrium:



Pendulum
$d = 4$



UR-5 Robotic Arm
6 joints $\implies$ dimension $d = 12$

Results

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

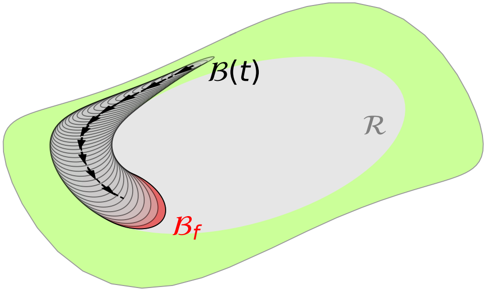
Dynamics	$\mathcal{C}_1$		$\mathcal{C}_2^a$		$\mathcal{C}_2^b$		**SOS**		**sampling**	
	ρ/ρ_s	v/v_s	ρ/ρ_s	v/v_s	ρ/ρ_s	v/v_s	ρ/ρ_s	v/v_s	ρ/ρ_s	v/v_s
Vanderpol	0.20	0.20	0.14	0.14	0.10	0.10	**1**	**1**	1	1
Satellite	2.9×10^{-2}	2.6×10^{-5}	9.3×10^{-2}	9.4×10^{-4}	7.9×10^{-2}	5.7×10^{-4}	**0.93**	**0.82**	1	1
Pend. (bot.)	3.2×10^{-2}	1.1×10^{-3}	3.5×10^{-2}	1.2×10^{-3}	$\mathbf{4.2\times10^{-2}}$	$\mathbf{1.9\times10^{-3}}$	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}	$\mathbf{4.7\times10^{-2}}$	$\mathbf{2.2\times10^{-3}}$	N.A.	N.A.	1	1
Robot	2.4×10^{-3}	1.8×10^{-16}	7.1×10^{-3}	1.5×10^{-13}	$\mathbf{1.5\times10^{-2}}$	$\mathbf{1.2\times10^{-11}}$	N.A.	N.A.	1	1

Running times

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

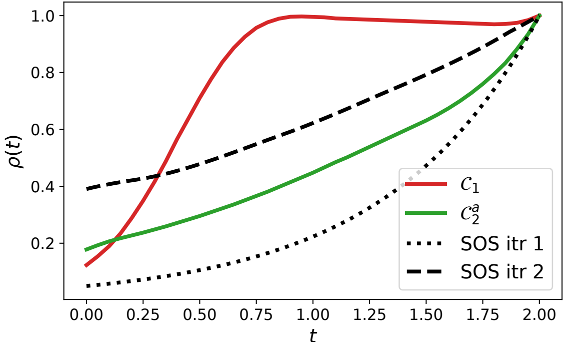
Dynamics	$\mathcal{O}+\mathcal{C}_{\mathbf{1}}$	$\mathcal{O}+\mathcal{C}_{\mathbf{2}}^{\mathbf{a}}$	$\mathcal{O}+\mathcal{C}_{\mathbf{2}}^{\mathbf{b}}$	**SOS**
Vanderpol	1.8×10^{-3}	1.1×10^{-4}	1.6×10^{-4}	0.05
Satellite	1.2	0.17	0.17	32
Pend. (bot.)	2.3	15	15	132
Robot	2.3	32	33	N.A.

Generalization to Trajectory Tracking



A funnel $\mathcal{B}(t)$ around a trajectory, obtained with $\mathcal{C}_1$.
$\mathcal{R}$ is a region of attraction around 0, and $\mathcal{B}_f$ is the target region.
The reference trajectory is displayed with arrows.

Extension to Trajectory Tracking



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

Conclusion

A **general method** to compute
stability certificates, that is:

- simple to implement;
- fast to compute;
- tractable in large dimensions;
- applicable to non-polynomial systems;
- BUT less precise than SOS based certificates.

When integrated into the **LQR-trees framework**, is it better to compute:

- a lot of low-quality certificates with this method,
- or a few tight certificates with SOS?

References I

- Jean-Pierre Aubin and Arrigo Cellina, *Differential inclusions: Set-valued maps and viability theory*, Springer, 1984.
- Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan, *Linear matrix inequalities in system and control theory*, vol. 15, Siam, 1994.
- Robert R Burridge, Alfred A Rizzi, and Daniel E Koditschek, *Sequential composition of dynamically dexterous robot behaviors*, The International Journal of Robotics Research **18** (1999), no. 6, 534–555.
- 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.
- 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.
- 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.