



HAL
open science

Computing regions of stabilizability for nonlinear control systems with input constraints

Philipp Schlehuber-Caissier, Nicolas Perrin

► **To cite this version:**

Philipp Schlehuber-Caissier, Nicolas Perrin. Computing regions of stabilizability for nonlinear control systems with input constraints. American Control Conference (ACC'18), 2018, Milwaukee, United States. pp.2869-2876, 10.23919/ACC.2018.8430798 . hal-03135362

HAL Id: hal-03135362

<https://hal.science/hal-03135362v1>

Submitted on 8 Feb 2021

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.

Computing regions of stabilizability for nonlinear control systems with input constraints

Philipp Schlehuber-Caissier¹ and Nicolas Perrin¹

Abstract—The growing interest in robust motion planning under safety constraints for robotic systems, such as autonomous vehicles or collaborative robots, lead to an increased interest to prove the stability or stabilizability of dynamic models of such systems. The proofs are usually based on Lyapunov theory and often rely on “funnels”, tubes or similar invariants that bound variations along reference trajectories. Recent methods have shown how to compute such funnels via sum-of-square (SoS) optimization. In this work we propose a new approach to solve this problem relying on optimal control based state-space partitioning and relaxation techniques for polynomial programming. In contrast to SoS techniques, our approach is inherently suitable to supply certificates of local stability and deal with bounded control inputs without increasing the complexity of the arising optimization problem. We present how to use our approach to obtain a large inner approximation of the true region of attraction around a given trajectory for which it is ensured that all states inside the funnel will reach goal states in a given time period. This property makes it interesting to integrate our approach in a general path or task planning algorithm. We compare our results in simulation with the ones obtained using a state-of-art method based on SoS techniques for a torque controlled simple pendulum and an underactuated double pendulum (Acrobot).

I. INTRODUCTION

Executing robotic tasks in the presence of safety or timing constraints in a robust fashion requires not only that the reference trajectory satisfies these constraints but also that there exists a region around this trajectory which will converge towards it and also respects the constraints. With a dynamic model of the robotic system, guaranteeing convergence on regions of the configuration space is usually done with stability certificates, which are computationally generated formal proofs based on Lyapunov or contraction theory. However obtaining stability certificates for nonlinear systems, such as walking or flying robots or even simpler systems such as the Acrobot [1], that induce such regions is notoriously difficult and remains a challenging problem despite the enormous progress made in recent years using a variety of different approaches. These approaches include, but are not limited to, outer approximation via occupation measures [2], counter-example guided synthesis [3] and sum-of-squares (SoS) approaches [4], which are based on approximating the system as polynomial using a truncated Taylor expansion and prove the convergence of the approximated system with respect to a quadratic Lyapunov function using SoS optimization. When the system has control inputs, what is required is not certificates of stability but certificates of stabilizability, proving that for bounded control inputs there always exists inputs that bring the system back to its reference. This is necessary to properly model

robots, since its actuators can only provide a limited amount of effort (e.g. joint actuators are usually limited in torque), which makes obtaining certificates more difficult, especially for SoS techniques. Moreover these certificates must be constructive and yield ways to compute such control inputs.

We therefore propose an approach to prove exponential stabilizability of a controlled polynomial system with input constraints with respect to polynomial Lyapunov function candidates based on two principles: state-space partitioning and convexification. This leads to a formulation that inherently yields local certificates of stabilizability and takes into account the boundedness of the input in a very natural way.

In the remainder of this article we show how to partition the state-space into subsets defined by the optimal control input (Sec. III-B) and how this relates to proving stabilizability for a given region and system (Sec. III-A). In section V we present our approach to prove nonpositiveness of a polynomial based on an extension of Reformulation and Linearisation Techniques (RLT) and finally in section VI we present numerical results for two common test cases, the Acrobot and a torque controlled simple pendulum.

II. PROBLEM STATEMENT AND NOTATION

We propose a new method to find an inner approximation of the true region of attraction (RoA) for polynomial control affine systems¹ by scaling a given quadratic Lyapunov function candidate $V(\mathbf{x}) = \mathbf{x}^T \cdot P \cdot \mathbf{x}$, $P \in \mathcal{S}_{++}^n$, where $\mathcal{S}_{+/++}^n$ denotes the cone of symmetric semi-definite/definite positive matrices of size $n \times n$ over \mathbb{R} . More precisely, we consider systems of the form

$$\dot{\mathbf{x}} = g(\mathbf{x}) + B \cdot \mathbf{u}, \mathbf{u} \in \mathcal{U} \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^n$ denotes a point in the state-space, $g(\cdot)$ represents the polynomial (system) dynamics $\mathbb{R}^n \mapsto \mathbb{R}^n$, $B \in \mathbb{R}^{n \times m}$ is a constant input matrix, $\mathbf{u} \in \mathbb{R}^m$ denotes the control input and \mathcal{U} is the set of admissible control inputs. We suppose that B has full column rank and that each control input $\mathbf{u}[i]$, where $[i]$ denotes the i -th element of the vector, is bounded, and that the constraints are independent of the other control inputs, $\mathbf{u}^- [i] \leq \mathbf{u}[i] \leq \mathbf{u}^+ [i]$ so that $\mathcal{U} = \{\mathbf{u} | \mathbf{u}^- \leq \mathbf{u} \leq \mathbf{u}^+\}$. This type of input constraints is typical for torque controlled articulated robots, which is our primary target as far as applications are concerned. We adopt the notation that $\leq \leq \geq \geq$ represent elementwise relations when applied to vectors or matrices, and $A \succeq \mathbf{0}$ means that the matrix A is positive semi-definite. The problem treated in this article is to find an as large as possible sublevel-set of the quadratic Lyapunov function candidate

¹Sorbonne Universités, UPMC Univ Paris 06, CNRS UMR 7222, Institut des Systèmes Intelligents et de Robotique, F-75005, Paris, France schlehuber@isir.upmc.fr

¹In Sec. VI-B the class of treated dynamic systems will be enlarged to polynomial systems with polynomial input dynamics. However, to introduce the basic idea we will restrain ourselves to input affine systems for the moment.

$V(\mathbf{x})$, denoted $\Omega = \{\mathbf{x} | V(\mathbf{x}) \leq \alpha\} \subset \mathbb{R}^n$ for which exists an admissible control input that makes this set exponentially stable, or more formally

$$\begin{aligned} & \text{prove } \forall \mathbf{x} \in \Omega : \exists \mathbf{u} \\ & \text{subject to } \mathbf{u}^- \leq \mathbf{u} \leq \mathbf{u}^+ \\ & \langle \nabla_{\mathbf{x}} V, g(\mathbf{x}) + B.\mathbf{u} \rangle \leq -\beta.V(\mathbf{x}) \end{aligned}$$

where $\langle \cdot, \cdot \rangle$ denotes the usual scalar product, $\nabla_{\mathbf{x}}$ denotes the gradient with respect to \mathbf{x} and $\beta \geq 0$ is called the convergence rate. The limit case $\beta = 0$ is equivalent to the set Ω being invariant. To avoid confusion with other approaches, note that our approach does not modify the given Lyapunov function candidate $V(\cdot)$ but seeks to enlarge the subset Ω by enlarging α . This approach is reasonable for dynamical systems for which a "good" Lyapunov function candidate can be found by other means as we show in section V-C.

To avoid ambiguity with other definitions, we call a function $V(\mathbf{x})$ a Lyapunov function candidate if it is differentiable, radially unbounded and everywhere strictly positive, except at the origin where it evaluates to zero. A function $V(\mathbf{x})$ is called a Lyapunov function for the dynamical system $\dot{\mathbf{x}} = f(\mathbf{x})$ if it is a Lyapunov function candidate and its derivative along any trajectory of the system is everywhere strictly negative except at the origin where it is zero. Moreover, a Lyapunov function proves exponential stability if its derivative along any trajectory is everywhere smaller than its current value multiplied with a negative factor except at the origin where it is zero. Due to their outstanding practical importance stemming from the ease of inclusion and intersection testing as well as their inherent suitability for second order systems, we restrain ourselves to quadratic Lyapunov functions of the form $V(\mathbf{x}) = \mathbf{x}^T.P.\mathbf{x}$. The conditions for $V(\mathbf{x})$ being a Lyapunov function candidate are met if $P \in S_{++}^n$. Finally we write $\mathbf{x} \models \mathcal{C}$ to indicate that the variable/vector of variables \mathbf{x} has to satisfy all constraints in the constraint set \mathcal{C} .

III. DERIVING POLYNOMIAL FUNCTIONS TO PROVE CONVERGENCE FOR A SUBSET OF THE STATE-SPACE

In this section we detail how to prove stabilizability for a region of the state-space based on Lyapunov theory for controlled systems. First we show how SoS techniques deal with the arising problems, and in the second part we detail our state-space partitioning approach.

A. Local stability of polynomial systems

Given a Lyapunov function candidate $V(\mathbf{x})$ and a subset of the state-space Ω , proving exponential convergence for a dynamical system $\dot{\mathbf{x}} = f(\mathbf{x})$ amounts to finding a certificate of negativeness for $\dot{V}(\mathbf{x})$ plus a convergence term valid within Ω . So one has to prove that

$$\forall \mathbf{x} \in \Omega \setminus \mathbf{0}, \dot{V}(\mathbf{x}) = \langle \nabla V(\mathbf{x}), f(\mathbf{x}) \rangle < -\beta.V(\mathbf{x}) \quad (2)$$

or equivalently

$$\max_{\mathbf{x} \in \Omega \setminus \mathbf{0}} \langle \nabla_{\mathbf{x}} V(\mathbf{x}), f(\mathbf{x}) \rangle + \beta.V(\mathbf{x}) < 0. \quad (3)$$

However there is no generic way to find such certificates for general nonlinear systems and Lyapunov function candidates. Therefore one has to restrain the class of functions considered.

In recent years enormous progress has been made² by restraining both, the dynamics and the Lyapunov function candidate to be polynomial in \mathbf{x} . By doing so $\dot{V}(\mathbf{x})$ is polynomial too and one can use SoS-decomposition techniques to prove nonpositiveness. However SoS-decomposition naturally implies global stability of the system. If one wants/can only prove local stability, additional multipliers need to be added, so (2) becomes $\forall \mathbf{x} \in \Omega, \dot{V}(\mathbf{x}) = \langle \nabla_{\mathbf{x}} V(\mathbf{x}), g(\mathbf{x}) \rangle - L(\mathbf{x})(V(\mathbf{x}) - \alpha) \leq -\beta.V(\mathbf{x})$ where $L(\mathbf{x})$ is a SoS-polynomial³. This, due to the multiplication of decision variables, causes the problem to be nonconvex and an alternating solution scheme has been proposed in [4] to solve this problem.

For controlled dynamical systems, the arising question is even more difficult: Given a controlled dynamical system of the form $\dot{\mathbf{x}} = g(\mathbf{x}) + b(\mathbf{x}, \mathbf{u})$, does a control input \mathbf{u} exist which satisfies simultaneously some input constraints so that $\mathbf{u} \in \mathcal{U}$ and ensures convergence $\langle \nabla_{\mathbf{x}} V(\mathbf{x}), g(\mathbf{x}) + b(\mathbf{x}, \mathbf{u}) \rangle \leq -\beta.V(\mathbf{x})$. So the arising problem asks whether for each point in the considered subset there exists an admissible control input that causes the derivative to be smaller than the demanded convergence rate,

$$\max_{\mathbf{x} \in \Omega \setminus \mathbf{0}} \min_{\mathbf{u} \in \mathcal{U}} \langle \nabla_{\mathbf{x}} V(\mathbf{x}), g(\mathbf{x}) + b(\mathbf{x}, \mathbf{u}) \rangle \leq -\beta.V(\mathbf{x}). \quad (4)$$

This is an even more difficult problem due to the interweaving of optimizations and cannot be efficiently solved even if V , g and b are polynomials. The approach taken by [4] is to search for a polynomial control law, denoted $c(\mathbf{x})$, that replaces \mathbf{u} in the previous optimization problem, simplifying its structure. However this solution comes at the price of introducing new variables to represent the polynomial control law. Moreover, for each element of the control vector new multipliers need to be added in order to ensure that the output of the polynomial control law locally satisfies the constraints on \mathbf{u} and the highest order of the polynomial system grows since b and c are composed, which is a new source of complexity and nonconvexity. This results in a necessary additional step in the alternating solution scheme proposed in [4]. In this article we propose a different approach to this problem. Instead of searching for a polynomial control law we derive conditions that partition the state-space into subsets for which the optimal control input satisfying the input constraints can be determined. This optimal control law is then viewed as proof that an exponentially stabilising control input exists and we will derive more practical control laws based on this proof in Sec. IV-B

B. State-space partitioning based on optimal input

In this section we explain how to partition the state-space such that in each region the optimal control input, with respect to convergence, is independent of the actual point considered. We moreover show that this optimal control input inevitably consists only of elements of \mathbf{u}^+ and \mathbf{u}^- .

Lemma 3.1: Optimal input partition

For polynomial control affine systems and quadratic Lyapunov candidate functions, the state-space can be partitioned into 2^m subsets $\mathcal{H}_{i \in [1, 2^m]}$ and an associated optimal control input with respect to convergence \mathbf{u}_i^* can be defined.

²See [4] and references therein.

³For details see [4]

In order to prove this claim, reconsider the quadratic Lyapunov function candidate $V(\mathbf{x}) = \mathbf{x}^\top.P.\mathbf{x}$ and the polynomial control affine system $\dot{\mathbf{x}} = g(\mathbf{x}) + B.\mathbf{u}$. Then the time derivative of V is given as

$$\begin{aligned}\dot{V}(\mathbf{x}) &= \langle \nabla_{\mathbf{x}}V(\mathbf{x}), \dot{\mathbf{x}} \rangle \\ &= 2\mathbf{x}^\top.P.(g(\mathbf{x}) + B.\mathbf{u}).\end{aligned}\quad (5)$$

This indicates that the derivative can be separated into two parts: An uncontrollable part resulting from the system dynamics $2\mathbf{x}^\top.P.g(\mathbf{x})$ denoted \dot{V}_g and an input dependent part $2\mathbf{x}^\top.P.B.\mathbf{u}$ denoted \dot{V}_u . In order to obtain the desired partitioning of the state-space, \dot{V}_u is explicitly written as sum

$$\dot{V}_u = \sum_j \mathbf{x}^\top.P.B[:,j].\mathbf{u}[j] \quad (6)$$

where we denote $B[:,j]$ the j -th column of the matrix B treated as a column vector.

So the input dependent part of the derivative can be written as the sum of each control input element $\mathbf{u}[j]$ multiplied by the scalar $\mathbf{x}^\top.P.B[:,j]$. By denoting $\mathbf{n}_j = P.B[:,j]$ one gets $\mathbf{x}^\top.\mathbf{n}_j$ which is simply the minimal directed distance⁴ between the point considered and a separating hyperplane passing through the origin with normal vector \mathbf{n}_j denoted \mathcal{P}_j .

Demanding the system to converge as fast as possible with respect to the given Lyapunov function is therefore equal to minimizing \dot{V}_u which in turn is equal to minimizing each term in the sum in (6) since $\mathbf{u}[i]$ is independent of $\mathbf{u}[j]$ if $i \neq j$. To achieve this, the j -th control input has to be chosen as small as possible if $\mathbf{x}^\top.\mathbf{n}_j > 0$ (\mathbf{x} lies in the upper half-space of \mathcal{P}_j) and as large as possible if $\mathbf{x}^\top.\mathbf{n}_j < 0$ (\mathbf{x} lies in the lower half-space of \mathcal{P}_j) to obtain the minimal value of \dot{V}_u . The corresponding optimal control input in the sense of instantaneous convergence for the system is

$$\mathbf{u}^*(\mathbf{x})[j] = \begin{cases} \mathbf{u}^+[j] & \text{if } \mathbf{x}^\top.\mathbf{n}_j < 0 \\ \mathbf{u}^-[j] & \text{if } \mathbf{x}^\top.\mathbf{n}_j > 0 \\ a & \text{else} \end{cases} \quad (7)$$

where, in order to remove the ambiguity, any input a , satisfying $\mathbf{u}^-[j] \leq a \leq \mathbf{u}^+[j]$, can be chosen if the current state belongs to the hyperplane. The undefined character of a does not pose a problem for proving stability since its contribution to \dot{V}_u is 0 independently of the value of a . This control law partitions the state-space into two closed half-spaces for each of the m control inputs. Denoting \mathcal{H}_i the unbounded convex polytope defined as the intersection of m upper or lower half-spaces generated by the hyperplanes $\mathcal{P}_{j \in [1,m]}$ we get

$$\mathcal{H}_i = \left\{ \mathbf{x} \mid \forall j \in [1, m] \ c_j^i \mathbf{x}^\top.\mathbf{n}_j \leq 0 \right\} \quad (8)$$

where $c_j^i \in \{-1, 1\}$ is a switch to determine whether the upper or lower half-space of the j -th hyperplane is used. One can easily see that for each such polytope \mathcal{H}_i the optimal control input \mathbf{u} , denoted \mathbf{u}_i^* , is independent of \mathbf{x} . Since there exist m such hyperplanes, the state-space is partitioned into 2^m polytopes with different optimal inputs. Note that each of these polytopes has nonempty interior if the matrix P is

positive definite and B has full column rank (i.e. rank m). The definiteness of P is always given because $V(\mathbf{x}) = \mathbf{x}^\top.P.\mathbf{x}$ is a Lyapunov function candidate and in general, for dynamic systems, B is of rank⁵ m . This approach is based on the ideas developed in [5] for bilinear systems.

This approach allows us to obtain 2^m subsets of the state-space \mathcal{H}_i , the dynamics induced by the optimal control input $\dot{\mathbf{x}} = g(\mathbf{x}) + B.\mathbf{u}_i^*$ and the best obtainable derivative of the Lyapunov function candidate $\dot{V}_i^*(\mathbf{x}) = \langle \nabla_{\mathbf{x}}V(\mathbf{x}), \dot{\mathbf{x}}^* \rangle = 2.\mathbf{x}^\top.P.(g(\mathbf{x}) + B.\mathbf{u}_i^*)$.

Now the min-max problem (4) for proving stability can be reformulated. Since we determined the optimal input with respect to instantaneous convergence for each \mathcal{H}_i we can drop the inner minimization by checking each intersection of the partition with the sublevel-set of V considered. So the inner minimization

$$\min_{\mathbf{u} \in \mathcal{U}} \langle \nabla_{\mathbf{x}}V(\mathbf{x}), g(\mathbf{x}) + b(\mathbf{x}).\mathbf{u} \rangle \quad (9)$$

becomes

$$\mathbf{x} \in \mathcal{H}_i : 2.\mathbf{x}^\top.P.(g(\mathbf{x}) + B.\mathbf{u}_i^*) \quad (10)$$

and therefore

$$\forall i : \max_{\mathbf{x} \in (\Omega \cap \mathcal{H}_i) \setminus \emptyset} \dot{V}_i^*(\mathbf{x}) = 2.\mathbf{x}^\top.P.(g(\mathbf{x}) + B.\mathbf{u}_i^*) + \beta.\mathbf{x}^\top.P.\mathbf{x} \leq 0 \quad (11)$$

is equivalent to the initial problem (4) and therefore represents a proof of exponential convergence. It also proves, constructively, that Ω is a region of exponential stabilizability. What we want is to maximize the size of Ω , which depends only on one parameter, α . In section VI, we use a dichotomic search to quickly find a large value for α . This requires us to efficiently check the validity of 11. In section V we show how this can be done by using relaxations to deal with the nonconvex polynomial expressions in 11.

IV. RESULTING CLOSED-LOOP DYNAMICS AND LINKS TO SLIDING-MODE AND QP CONTROL

In this section we point out links between the optimal control law and first order sliding mode control. Since this control mode can induce chattering and premature wear out due to the high, possibly infinite, switching frequency on the sliding surface it is not suitable in real applications. We therefore introduce a quadratic programming (QP) control law that results in continuous control trajectories and provides the same certificates. To illustrate the approach and the resulting dynamics, a torque controlled simple pendulum is used as a showcase,

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{\theta} \\ \ddot{\theta} \end{pmatrix} = g(\mathbf{x}) + B.u = \begin{pmatrix} \dot{\theta} \\ \omega^2 \sin(\theta) \end{pmatrix} + \begin{bmatrix} 0 \\ k \end{bmatrix}.u \quad (12)$$

The reference $\theta = 0$ corresponds to the upright position.

A. From sliding mode control to a continuous control law

To showcase the resulting partition we impose the Lyapunov function candidate $V(\mathbf{x}) = \mathbf{x}^\top.I_d.\mathbf{x}$. Since the pendulum is a single input system, there exists only one separating hyperplane defined by the normal vector $\mathbf{n} = I_d.B = k\mathbf{e}_{\dot{\theta}}$ as shown in Fig. 1. We therefore obtain optimal control inputs associated to

⁵If not, this means that there exist at least two inputs which are linearly dependent. One can replace these two or more inputs by fewer and linearly independent inputs and return to the general case.

⁴Up to a scaling factor

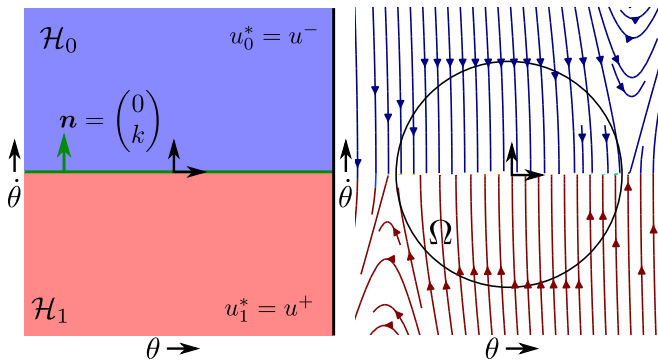


Fig. 1. On the left the partitioning of the state-space based on V is shown. u^-/u^+ is the optimal input for the upper/lower half-space. On the right, the resulting dynamics for the original system dynamics imposing u^* is shown. The black circle marks the largest exponentially converging subset Ω .

the partitioning of the state-space formed by two subsets \mathcal{H}_0 and \mathcal{H}_1 . From both half-spaces the system states converge to the hyperplane (or diverge) for the optimal control input. Once this surface obtained, switching between the u^- and u^+ occurs, at possibly infinite frequency, in order to maintain the state on the hyperplane. This corresponds to first order sliding mode control and the separating hyperplane is the sliding surface for this system.

In [6] a generic way to obtain stable sliding-mode control laws for a class of underactuated second-order systems, including the Acrobot, is proposed. Even though this approach also yields a hyperplane (not equivalent to ours) used to switch the sign of (a part of) the input, the resulting behaviour and properties are very distinct. The proof of convergence of the sliding mode control is performed in two steps: I) Prove that the system converges to the sliding surface (the hyperplane) II) Prove that all points on the hyperplane converge to the origin. Using such an approach, it is not obvious how to deal with bounded inputs or specify an invariant region. Since trajectories are allowed to stray very far from the origin while converging towards the sliding surface, it increases the risk of leaving the true RoA that takes into account the boundedness of the input.

B. QP-control for a continuous control law

As pointed out above, first order sliding control raises concerns from both theoretical and practical point of view: the high switching frequency once the hyperplane reached causes problems for real robotic applications, and from a theoretical point of view, the solution of the dynamical system is actually undefined. To remedy these problems we propose a QP-based control and prove, using Berge's Maximum Theorem [7], that it guarantees exponential convergence on $\Omega \setminus \mathbf{0}$ and continuous control trajectories if (11) holds.

The proposed QP-control is

$$\begin{aligned} & \underset{\mathbf{u}}{\text{minimize}} \quad h(\mathbf{x}, \mathbf{u}) = \mathbf{u}^\top \cdot Q \cdot \mathbf{u} + 2\mathbf{x}^\top \cdot P \cdot B \cdot \mathbf{u} \\ & \text{subject to} \quad \mathbf{u}^- \leq \mathbf{u} \leq \mathbf{u}^+ \\ & \quad \quad \quad 2\mathbf{x}^\top \cdot P \cdot B \cdot \mathbf{u} \leq -2\mathbf{x}^\top \cdot P \cdot g(\mathbf{x}) - \beta \cdot \mathbf{x}^\top \cdot P \cdot \mathbf{x} \end{aligned}$$

where $2\mathbf{x}^\top \cdot P \cdot B \cdot \mathbf{u}$ is the input dependent part of the derivative of the Lyapunov function, $Q \in \mathbb{S}_{++}$ is user chosen, typically diagonal, and reflects the cost of the control effort. The first

constraint represents the boundedness of the control input and the second constraint ensures the demanded exponential convergence.

Note that this optimization problem is guaranteed to have a solution for all $\mathbf{x} \in \Omega \setminus \mathbf{0}$ if (11) holds. Moreover, the form of the objective function (with $Q \in \mathbb{S}_{++}$), together with the convexity of the constraints, ensures uniqueness of solutions and makes the problem amenable to convex quadratic programming.

In order to prove continuity of the resulting control trajectory, we employ Berge's Maximum Theorem, which, combined with the uniqueness of solutions, guarantees continuous control trajectories. Berge's Maximum Theorem provides conditions to ensure the continuity (properly speaking the upper hemicontinuity) of the set of control inputs respecting the constraints and minimizing the objective function. Since this set collapses to a singleton \mathbf{u}^* (the unique solution) in our case, it ensures continuity. The fulfilled conditions are that the objective function $h(\mathbf{x}, \mathbf{u})$ is jointly continuous in \mathbf{x} and \mathbf{u} and that the set of inputs respecting the constraints is compact and hemicontinuous (upper and lower) with respect to \mathbf{x} .

The influence of the regularization matrix Q on the control effort and dynamics is shown in Fig. 2.

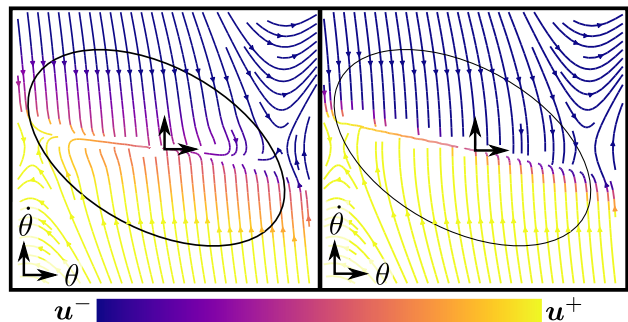


Fig. 2. In these figures the resulting dynamics for the inverse pendulum using the proposed QP-control are shown. The convergence rate $\beta = 0.5$ is the same for both figures. The regularization term Q is equal to 2 for the left and 0.5 for the right figure. As one can see, a high regularization induces a smoother control law (illustrated by a smoother transition in the colours). As the regularization term diminishes, the resulting control law approaches more and more the optimal control law \mathbf{u}^* , which maximizes instantaneous convergence. For points outside the ellipsoid (Ω) the QP-problem is not always feasible in which case we use \mathbf{u}^* .

Remark 1: We will see in sec. V-C that $V(\mathbf{x})$ (thus P) can be chosen such that the control inputs are not only continuous but also the slopes have reasonable absolute values.

Remark 2: The optimization objective is arbitrary (as long as Berge's Maximum Theorem hypotheses hold) and the decisive constraint assuring convergence is a simple linear constraint on the control input, which can be easily added to existing optimization based controllers.

V. RLT BASED CONVEXIFICATION

Up to now it was detailed how to partition the state-space into input optimal subsets and derive an associated nonconvex polynomial expression (11), whose negativity on the respective subset is a partial proof of exponential stabilizability. In this section we show how to efficiently treat this problem using a modified version of the well-known Reformulation and Linearisation Technique (RLT) first introduced in [8] for zero-one programming problems, extended to continuous variables in [9] and further enhanced with semi-definite constraints in [10].

A. RLT based convex underapproximation

In order to informally introduce the basic idea behind RLT consider the general quadratic objective subjected to a set of linear constraints \mathcal{C}_1

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} \quad \mathbf{x}^\top \cdot Q \cdot \mathbf{x} + L \cdot \mathbf{x} \\ & \text{subject to } \mathbf{x} \models \mathcal{C}_1 \end{aligned} \quad (13)$$

This problem is nonconvex and NP-hard if $Q \notin \mathcal{S}_+$ and therefore can not be efficiently solved. The Relaxation and Linearisation Technique copes with this problem by replacing the i -th nonlinear term in the optimization problem with a corresponding new variable X_i . This gives rise to a set of new variables X and the objective function becomes linear in the set $\mathbf{x} \cup X$, but is unbounded due to the lack of constraints on X . In the running example, one can replace $\mathbf{x}^\top \cdot Q \cdot \mathbf{x}$ with $\text{tr}(X \cdot Q)$, where the matrix variable X is introduced to linearise all quadratic terms $\mathbf{x}_i \mathbf{x}_j$ by the single variable X_{ij} .

We denote by \mapsto_{Lin} the operator replacing all nonlinear terms by the corresponding variables in the set of new variables, so for instance $a\mathbf{x}_i \mathbf{x}_j + b\mathbf{x}_k \mathbf{x}_l \mapsto_{\text{Lin}} aX_{ij} + bX_{kl}$.

In order to ensure that the gap between the solution of the original and the linearised problem is small, one has to construct valid constraints on X . This can be done by taking products of constraints on \mathbf{x} . For instance, reconsider the set of linear constraints $\mathcal{C}_1 = \cup_{k \in [1, K]} C_1^k$. By multiplying two constraints C_1^m, C_1^n one obtains a valid constraint with terms up to order 2 in \mathbf{x} , then again one can replace the nonlinear (quadratic in this case) terms by the corresponding variable in X and thereby a linear constraint in $\mathbf{x} \cup X$ is obtained. For short, $C_1^m \otimes C_1^n = C_2^{m,n} \mapsto_{\text{Lin}} \tilde{C}_X^{m,n}$, where \otimes denotes the product of constraints. E.g. if $C_1^m = a\mathbf{x}_i + b\mathbf{x}_j \geq 0$ and $C_1^n = \mathbf{x}_i \geq c$ then product becomes $C_1^m \otimes C_1^n = C_2^{m,n} = a\mathbf{x}_i^2 + b\mathbf{x}_i \mathbf{x}_j - ac\mathbf{x}_i - b\mathbf{x}_j \geq 0$ and can be linearised to $C_2^{m,n} \mapsto_{\text{Lin}} \tilde{C}_X^{m,n} = aX_{ii} + bX_{ij} - ac\mathbf{x}_i - b\mathbf{x}_j \geq 0$ which is linear in $\mathbf{x} \cup X$. By an abuse of notation, we will denote by \otimes also the product of constraint sets, enumerating all possible combinations, so $\mathcal{C}_1 \otimes \mathcal{C}_1 = \cup_{i \in [1, K]} \cup_{j \in [1, K]} C_1^i \otimes C_1^j$. We denote by \mathcal{C}_i a constraint set were all appearing monomials are of degree i or less and $\tilde{\mathcal{C}}_X$ denotes the corresponding linearised version, a set of linear constraints with variables in $\mathbf{x} \cup X$. Forming such product constraints ensures that the new variables are bounded while guaranteeing that the linearised objective function remains an underestimator of the original objective function, since the set of admissible values in $\mathbf{x} \cup X$ is strictly larger than the set of admissible values of the original problem⁶. This corresponds to the standard RLT and was originally proposed in [9].

In [10] an improvement to the standard approach is given by adding semidefinite constraints, reducing significantly the gap between the solution of the relaxed and original problem. To introduce these constraints reconsider the nonconvex part of the objective function $\mathbf{x}^\top \cdot Q \cdot \mathbf{x}$ and its linearisation $\text{tr}(X \cdot Q)$. It is clear that the minimum of the original objective is obtained if $X = \mathbf{x} \cdot \mathbf{x}^\top$ holds. However this condition cannot be imposed as a constraint to the optimization since it is nonconvex. But

$X \succeq \mathbf{x} \cdot \mathbf{x}^\top$ can be imposed⁷ as linear matrix inequality (LMI) constraint via the Schur-complement to the optimization by adding

$$\begin{bmatrix} 1 & \mathbf{x}^\top \\ \mathbf{x} & X \end{bmatrix} \succeq \mathbf{0}$$

which makes the problem amenable to semidefinite programming. The optimisation resulting from applying the enhanced RLT on (13) is given as

$$\begin{aligned} & \underset{\mathbf{x}, X}{\text{minimize}} \quad \text{tr}(X \cdot Q) + L \cdot \mathbf{x} \\ & \text{subject to } \mathbf{x} \models \mathcal{C}_1 \\ & \mathbf{x}, X \models \tilde{\mathcal{C}}_X \\ & \begin{bmatrix} 1 & \mathbf{x}^\top \\ \mathbf{x} & X \end{bmatrix} \succeq \mathbf{0} \end{aligned}$$

where $\tilde{\mathcal{C}}_X$ denotes the linearisation of the set of constraints $\mathcal{C}_1 \otimes \mathcal{C}_1 = \mathcal{C}_2 \mapsto_{\text{Lin}} \tilde{\mathcal{C}}_X$. This approach was successfully applied to nonconvex quadratic optimisation problems [11] or within branch and bound algorithms for general nonlinear programming [12].

B. Enhanced RLT for polynomials

Below, we apply our method to polynomial dynamics of degree 3, which gives a degree 4 to the polynomial expressions in (11): $\dot{V}_i^*(\mathbf{x}) = 2 \cdot \mathbf{x}^\top \cdot P \cdot (g(\mathbf{x}) + B \cdot \mathbf{u}_i^*) \leq -\beta \cdot \mathbf{x}^\top \cdot P \cdot \mathbf{x}$. So, the above presented enhanced RLT has to be modified in order to treat such polynomials. As seen above, a new set of variables, denoted X , is created to linearise quadratic terms in \mathbf{x} which we write as $\mathbf{x} \cdot \mathbf{x}^\top \mapsto_{\text{Lin}} X$. We denote \mathbf{z} the column vector composed of all quadratic terms in \mathbf{x} and $\text{vec}(X)$ the column vector where the upper triangle of X is stored. So we have $\mathbf{z} \mapsto_{\text{Lin}} \text{vec}(X)$ and one can construct the following matrix

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{z} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x} \\ \mathbf{z} \end{pmatrix}^\top \mapsto_{\text{Lin}} \begin{pmatrix} \mathbf{x} \\ \text{vec}(X) \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x} \\ \text{vec}(X) \end{pmatrix}^\top \mapsto_{\text{Lin}} \begin{bmatrix} X & Y \\ Y^\top & Z \end{bmatrix}$$

where the symmetric matrix X linearises all quadratic terms, $\mathbf{x} \cdot \text{vec}(X)^\top \mapsto_{\text{Lin}} Y$ linearises all cubic terms⁸ and $\text{vec}(X) \cdot \text{vec}(X)^\top \mapsto_{\text{Lin}} Z$ linearises all quadratic terms in X and therefore corresponds to quartic terms in \mathbf{x} ⁸. By applying the Schur-complement a valid LMI-constraint is generated for the linearisation of all monomials of degree up to 4

$$\begin{bmatrix} 1 & \mathbf{x}^\top & \text{vec}(X)^\top \\ \mathbf{x} & X & Y \\ \text{vec}(X) & Y^\top & Z \end{bmatrix} \succeq \mathbf{0}. \quad (14)$$

In order to ensure that the linearised objective gives tight bounds, valid constraints on X, Y and Z have to be constructed based on the original constraint sets $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ and \mathcal{C}_4 . Valid constraints can in this case be obtained if the degree of the resulting constraint is less or equal to 4. Some examples of valid constraint sets are $\mathcal{C}_1 \otimes \mathcal{C}_1 \otimes \mathcal{C}_1 = \mathcal{C}_3 \mapsto_{\text{Lin}} \tilde{\mathcal{C}}_Y$, $\mathcal{C}_1 \otimes \mathcal{C}_1 \otimes \mathcal{C}_2 = \mathcal{C}_4 \mapsto_{\text{Lin}} \tilde{\mathcal{C}}_Z$, where $\tilde{\mathcal{C}}_Y/\tilde{\mathcal{C}}_Z$ is a set of linear constraints in $\mathbf{x} \cup X \cup Y/\mathbf{x} \cup X \cup Y \cup Z$. Using this approach, which is inspired by the ideas developed in [13], we can

⁷This relaxation also conserves the property that the linearised objective function is an underestimator for the original problem as the set of admissible values is strictly enlarged.

⁸Note that not all elements in Y or Z are unique.

⁶For more details see [9].

underestimate the original objective function of degree 4

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} \quad (\mathbf{x}^\top \quad \mathbf{z}^\top) \cdot Q \cdot \begin{pmatrix} \mathbf{x} \\ \mathbf{z} \end{pmatrix} + L \cdot \mathbf{x} \\ & \text{subject to } \mathbf{x} \models \mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4 \end{aligned} \quad (15)$$

with

$$\begin{aligned} & \underset{\mathbf{x}, X, Y, Z}{\text{minimize}} \quad \text{tr} \left(\begin{bmatrix} X & Y \\ Y^\top & Z \end{bmatrix} \cdot Q \right) + L \cdot \mathbf{x} \\ & \text{subject to } \mathbf{x} \models \mathcal{C}_1 \\ & \mathbf{x}, X, Y, Z \models \tilde{\mathcal{C}}_X, \tilde{\mathcal{C}}_Y, \tilde{\mathcal{C}}_Z \\ & \begin{bmatrix} 1 & \mathbf{x}^\top & \text{vec}(X)^\top \\ \mathbf{x} & X & Y \\ \text{vec}(X) & Y^\top & Z \end{bmatrix} \succeq \mathbf{0} \end{aligned}$$

where $\tilde{\mathcal{C}}_X$, $\tilde{\mathcal{C}}_Y$ and $\tilde{\mathcal{C}}_Z$ are the sets of all obtainable constraints constructed via multiplication of \mathcal{C}_1 , \mathcal{C}_2 and \mathcal{C}_3 plus the linearisation of $\mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4$, the naturally arising constraints on the new variables and their respective products.

Remark 3: The number of variables in this optimisation problem is equal to the number of monomials of degree up to 4 in \mathbf{x} . The approach presented in [4] needs a similar number of variables to represent just one component of the polynomial control law (depending on the degree of the control law) plus the variables necessary to represent the multiplier functions which have to be of superior degree in general.

Remark 4: The number of linear constraints in the sets $\mathcal{C}_{1/2/3/4}$ and (even more so) $\tilde{\mathcal{C}}_{X/Y/Z}$ grows rapidly with the size of \mathbf{x} . However linear constraints are computationally cheap and additionally there exist methods to limit number of constraints. These methods are based on the redundancy often occurring, when taking all products possible between the constraint sets.

Remark 5: The above presented approach to construct enhanced RLT representation of nonconvex polynomial problems is by no means confined to polynomials of degree 4. By reapplying the method above k times one can linearise optimisation problems containing monomials of degree up to 2^k .

Remark 6: The decision variables in the resulting semidefinite program are associated with remarkably sparse matrices in the LMI constraints. This structure can probably be exploited to solve the problem even faster with a tailored solver.

C. Application

Above it was shown how to construct a linear underestimator in a higher dimensional space ($\mathbf{x} \cup X \cup Y \cup Z$) for a (nonconvex) polynomial function of order 4 in \mathbf{x} and how to construct constraint sets for the new variables ($X \cup Y \cup Z$) such that the underestimation usually results in reasonably tight bounds. Now we will shortly discuss how to apply this approach to prove stabilizability of a controlled system.

According to (11) we can prove the stabilizability of a polynomial control affine system on a sublevel-set Ω of a quadratic Lyapunov function candidate $V(\mathbf{x}) = \mathbf{x}^\top \cdot P \cdot \mathbf{x}$ by assuring the negativity of each of the 2^m (nonconvex) terms $2\mathbf{x}^\top \cdot P \cdot (g(\mathbf{x}) + B \cdot \mathbf{u}_i^*)$ on $\Omega \cap \mathcal{H}_i \setminus \mathbf{0}$. For each $i \in [1, 2^m]$,

this can be written as

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} \quad \begin{pmatrix} \mathbf{x} \\ \mathbf{z} \end{pmatrix}^\top \cdot Q \cdot \begin{pmatrix} \mathbf{x} \\ \mathbf{z} \end{pmatrix} + L_i \cdot \begin{pmatrix} \mathbf{x} \\ \mathbf{z} \end{pmatrix} \\ & \text{subject to } \mathcal{C}_1 = \left\{ \bigcup_{j \in [1, m]} c_j^i \mathbf{x}^\top \cdot \mathbf{n}_j \leq 0 \right\} \\ & \quad \mathcal{C}_2 = \left\{ \mathbf{x}^\top \cdot P \cdot \mathbf{x} \leq \alpha \right\} \end{aligned} \quad (16)$$

with

$$\begin{aligned} Q &= - \begin{bmatrix} P & \\ \mathbf{0} & \end{bmatrix} \cdot \tilde{A} - \tilde{A}^\top \cdot \begin{bmatrix} P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} - \beta \begin{bmatrix} P & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \text{ and} \\ L_i &= [-\mathbf{u}_i^* \cdot B^\top \cdot P \quad \mathbf{0}] , \end{aligned}$$

with $g(\mathbf{x}) = \tilde{A} \cdot \mathbf{z}$ being the truncated Taylor expansion⁹ of the true nonlinear system dynamics. \mathcal{C}_2 represents the quadratic constraint confining \mathbf{x} to the sublevel-set Ω and \mathcal{C}_1 is the set of linear constraints restricting \mathbf{x} to the i -th polytope of the optimal input partitioning. This shows how to bring the resulting optimal convergence for each of the 2^m input optimal subsets into the form of (15). If the minimal objective value of the optimization problem (16) is larger than 0 for each subset $\mathcal{H}_i \cap \Omega$, then Ω is exponentially stabilisable.

VI. NUMERICAL RESULTS

The above presented approach was implemented in python using `cvxopt` [14]¹⁰ and compared to the results obtained using the drake toolbox [15] for matlabTM relying on a very similar approach than the one presented in [4]. The largest region of attraction for the unstable position of a torque controlled simple pendulum and the Acrobot, a 2 DoF underactuated robotic arm, are presented and compared. Additionally, a time varying region of attraction (“Funnel”) is presented for a swing-up trajectory of the Acrobot using a reference trajectory generated with OMPL [16] and KPIECE1 [17].

Up to now it was shown how our approach can be used to prove that the optimal control law according to (7) ensures that the system converges exponentially with respect to a quadratic Lyapunov function candidate $V(\mathbf{x})$ on a given sublevel-set $\Omega = \{\mathbf{x} | V(\mathbf{x}) \leq \alpha\}$ and why this is a certificate that Ω is exponentially stabilisable with continuous control inputs. However neither how to obtain the candidate function nor how to maximize the certified inner approximation of the real region of attraction has been discussed. In contrast to the approach taken in [4] we do not seek to modify the Lyapunov function candidate, but rely on the similarity between the behaviour of the original system and the behaviour of its linearisation in the neighbourhood of an equilibrium point. Using the linearised system, suitable Lyapunov function candidates can be found using a Linear Quadratic Regulator (LQR) in its finite or infinite horizon version. It provides a positive definite matrix P and a cost optimal linear control such that $V(\mathbf{x}) = \mathbf{x}^\top \cdot P \cdot \mathbf{x}$ is a Lyapunov function for the closed-loop system in its linearised form. Then, in order to find a large inner approximation of the RoA for sublevel-sets of V , we perform a dichotomic search to quickly find a large α for which the exponential stabilizability can be proven for every point in $\Omega = \{\mathbf{x} | V(\mathbf{x}) \leq \alpha\}$.

⁹Without loss of generality it is assumed that the function is linearised at the origin.

¹⁰The code used for the examples in this article can be downloaded at <https://github.com/schlepil/RoA>

A. Simple Pendulum

The first example provided is a torque controlled simple pendulum for which we want to approximate the region of attraction of the upright position. The numerical values of the system are taken from the drake toolbox and we present the results obtained using the drake toolbox and our approach in Fig. 3.

Our approach provides in this example significantly larger

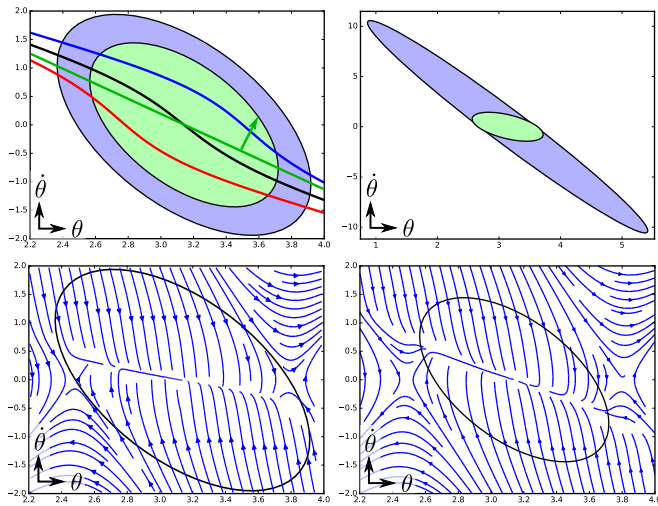


Fig. 3. Figure top left: Green ellipsoid: Region of attraction obtained with the drake toolbox for a cubic controller denoted RoA_{SoS} ; Purple ellipsoid obtained with our approach (denoted $\text{RoA}_{\text{relax}}$) initialised with the Lyapunov function of RoA_{SoS} . Green line: Separating hyperplane and defining normal vector; red, black and blue line: Set of points where the cubic control law (from drake toolbox) attains u^+ , 0 and u^- . Figure top right: Region of attraction obtained with our approach when initialised with the result of LQR with $Q = Id_n$ and $R = 5 \cdot Id_m$. Bottom row: Streamlines obtained when applying the generated control law. Left: Regularized QP-control with $\beta = 0.001$ and $Q = [0.5]$ based on our approach. Right: Cubic feedback control law saturated to meet input constraints. For both plots the Taylor expansion up to degree 3 of the simple pendulum is used as system dynamics.

regions of attractions, even-though it does not modify the shape of the region. It is worth noting that in [4] the ellipsoids are normalized by the condition $V(e) = 1$, where e is the vector of all ones. Even though this normalization does not introduce any conservativeness in the sense that a class of functions is excluded from the optimization, it introduces a bias since it is not equivalent to normalizing the ellipsoids by their volume. The ellipsoids on the top right of Fig. 3 have almost identical cost values for e , but the blue ellipsoid has a seven times larger surface. One can see that the resulting closed-loop dynamics are relatively similar for the chosen regularization value Q . As one can see, the generated RoA_{SoS} could be scaled without changing the control law and still be an invariant set, however the conservativeness introduced by the multiplier terms seems to inhibit this. A last remark on a qualitative difference between the two generated sets: while the set RoA_{SoS} is an *invariant* set according to [4], $\text{RoA}_{\text{relax}}$ is an *exponentially converging* set. This is worth noting since this change does not impact the runtime of our approach but has a significant impact on SoS approaches due to the additional SoS constraint necessary on the multiplier terms.

B. Acrobot

The underactuated two link robot called Acrobot does not belong to the class of polynomial control affine systems since its mass matrix is state-dependent and we therefore obtain nonlinear system and input dynamics $\dot{x} = \tilde{g}(x) + \tilde{b}(x) \cdot u$. We can nevertheless use the above detailed approach by first constructing the state-space partition based on the value of \tilde{b} at the origin, so $B = \tilde{b}(0)$. Then, we use the truncated Taylor expansion up to degree 3 to approximate the dynamics, $g(x) \approx \tilde{g}(x)$, $b(x) \approx \tilde{b}(x)$ and the expression for the best obtainable convergence (11) becomes $\dot{V}_i^*(x) = 2 \cdot x^T \cdot P \cdot (g(x) + b(x) \cdot u_i^*)$. Note that due to the nonlinearity of $b(x)$ it is no longer guaranteed that u_i^* is indeed optimal for each point in the subset \mathcal{H}_i . However, since the nonlinear part of the input dynamics is usually small compared to the linear part and the points for which u^* is not optimal are usually close to the separating hyperplane and have therefore little influence on \dot{V}_i , u_i^* is a good guess as it can be seen in the top left in Fig. 4.

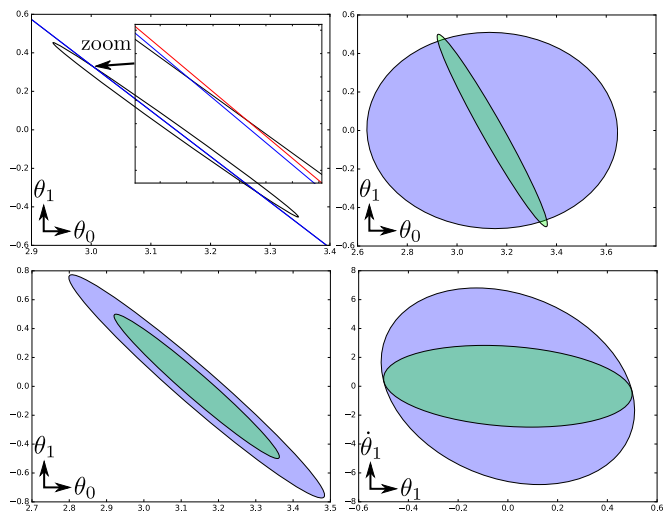


Fig. 4. Figure top left: Difference between separating hyperplane \mathcal{P} (red line) and separating hypersurface for polynomial input dynamics (blue line). Bottom left: Projection of RoA_{SoS} (green) and $\text{RoA}_{\text{relax}}$ (blue) when imposing RoA_{SoS} as initial condition for our approach. Right: Two projections of RoA_{SoS} (green) and $\text{RoA}_{\text{relax}}$ (blue) when our approach is initialised with the result of LQR with $Q = \text{diag}([10, 10, 1, 1])$ and $R = 0.01 \cdot Id_m$.

1) *Stabilising the upright position:* Again we compare the region of attraction found for the upright position under input constraints. Due to numerical issues solving the resulting semi-definite optimization problem when searching for a cubic controller using the drake toolbox, only a linear feedback controller is used. In Fig. 4 the projections of the two regions of attraction on different planes (the θ_0/θ_1 -plane and the θ_1/θ_1 -plane) are shown, except for the top right figure, where the intersection between the ellipsoid and the θ_0/θ_1 -plane is drawn. Here again, the volume of $\text{RoA}_{\text{relax}}$ is significantly larger (about 27 times) than the volume of RoA_{SoS} , but the value of the Lyapunov functions for e are very similar ($20 \cdot 10^{-3}$ to $22 \cdot 10^{-3}$).

2) *Funnel around a swing-up trajectory:* As final example we seek to construct “the largest” time-varying exponentially converging region, called a funnel, around a given reference

trajectory. The reference trajectory describes a swing up motion, i.e. a motion taking the Acrobot from the stable "hanging" position to the unstable upright position. We want to compute a funnel that takes as many states as possible, measured by the volume of the ellipsoids defining the funnel, to a small ellipsoid centred on the upright position. To achieve this, we first distribute N sample points along the trajectory, for which we will actually prove convergence as done in [4]. Then our algorithm retro-propagates the goal set Ω_N along the reference trajectory, seeking at each step to maximize the size of Ω_{i-1} . A suitable initial guess for the Lyapunov function candidate $V_i(\mathbf{x}) = \mathbf{x}^T \cdot P_i \cdot \mathbf{x}$ ¹¹ can be computed using a slightly modified time-dependent LQR controller. Attention has to be paid to properly account for the time-dependence of the funnel since the derivative of V becomes $\dot{V}(\mathbf{x}, t) = 2\mathbf{x}^T \cdot P \cdot (g(\mathbf{x}) + b(\mathbf{x}) \cdot \mathbf{u}) + \mathbf{x}^T \cdot \dot{P} \cdot \mathbf{x}$, where $\mathbf{x}^T \cdot \dot{P} \cdot \mathbf{x}$ is the additional term required by the time-dependence. The projection of the resulting funnel onto different planes is shown in Fig. 4. For this example we prove convergence on 95 points and each retro-propagation step takes about 12 seconds, so the entire funnel is computed in about 20 minutes on a desktop pc with i7-3770 processor.

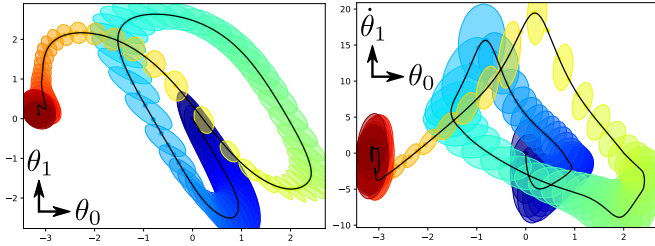


Fig. 5. Figure left: Projection of the time-dependent invariant set onto the θ_0/θ_1 -plane. Figure right: Projection of the time-dependent invariant set onto the $\theta_0/\dot{\theta}_1$ -plane. Due to the under-actuation of the system, a small neighbourhood around the origin has to be excluded to prove exponential convergence, so we have $\dot{V}(\mathbf{x}, t) \leq -\beta V(\mathbf{x}, t)$ on $\Omega = \{\mathbf{x} | \gamma \alpha(t) \leq V(\mathbf{x}, t) \leq \alpha(t)\}$ with $\beta, \gamma > 0$. The reference input is confined between -10 and 10 and we set the minimal/maximal input to $-20/20$, other numerical values defining the system are taken from the drake toolbox.

VII. CONCLUSION AND FUTURE WORK

We have presented an alternative approach to prove local exponential stabilizability of nonlinear control systems with input constraints based on ideas of sliding-mode control and relaxation methods for polynomial programming. The ability of our approach to find large inner approximations of the true region of attraction is shown for two common examples, a simple pendulum and the Acrobot. We obtain promising results in comparison to state-of-art methods relying on SoS-techniques. In order to make our approach more computationally efficient and allow for systems with more degrees of freedom ($n > 8$) we seek to replace the computationally heavy LMI-constraints with second-order cone constraints based on ideas in [18] or using semi-definite cuts based on [13]. Another line of work is to exchange interior point based optimization with ones relying on first order optimization methods, which are more suitable in case

¹¹Note that i here indicates the current time-point considered $i \in [1, N]$ and is not to be confused with indexing the different input optimal polytopes \mathcal{H}_i .

of large optimization problems [19]. Also there exists a huge potential speed-up using (warm-up) heuristics when computing funnels around reference trajectories since the polytope \mathcal{H}_i having the worst convergence rarely changes from one time-point to its predecessor.

VIII. ACKNOWLEDGEMENTS

We would like to thank Vincent Padois for fruitful discussions and his helpful comments on the manuscript.

REFERENCES

- [1] M. W. Spong, "The swing up control problem for the acrobot," *IEEE control systems*, vol. 15, no. 1, pp. 49–55, 1995.
- [2] D. Henrion and M. Korda, "Convex computation of the region of attraction of polynomial control systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 297–312, 2014.
- [3] H. Ravanbakhsh and S. Sankaranarayanan, "Counter-example guided synthesis of control lyapunov functions for switched systems," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, 2015, pp. 4232–4239.
- [4] A. Majumdar, A. A. Ahmadi, and R. Tedrake, "Control design along trajectories with sums of squares programming," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, 2013, pp. 4054–4061.
- [5] R. Longchamp, "Stable feedback control of bilinear systems," *IEEE Transactions on Automatic Control*, vol. 25, no. 2, pp. 302–306, 1980.
- [6] W. Wang, J. Yi, D. Zhao, and D. Liu, "Design of a stable sliding-mode controller for a class of second-order underactuated systems," *IEEE Proceedings-Control Theory and Applications*, vol. 151, no. 6, pp. 683–690, 2004.
- [7] C. Berge, "Espaces topologiques: Fonctions multivoques," 1959.
- [8] H. D. Serali and W. P. Adams, "A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems," *SIAM Journal on Discrete Mathematics*, vol. 3, no. 3, pp. 411–430, 1990.
- [9] H. D. Serali and C. H. Tuncbilek, "A reformulation-convexification approach for solving nonconvex quadratic programming problems," *Journal of Global Optimization*, vol. 7, no. 1, pp. 1–31, 1995.
- [10] L. Lovász and A. Schrijver, "Cones of matrices and set-functions and 0–1 optimization," *SIAM Journal on Optimization*, vol. 1, no. 2, pp. 166–190, 1991.
- [11] I. Kezurer, S. Z. Kovalsky, R. Basri, and Y. Lipman, "Tight relaxation of quadratic matching," in *Computer Graphics Forum*, vol. 34, no. 5, 2015, pp. 115–128.
- [12] H. D. Serali and C. H. Tuncbilek, "A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique," *Journal of Global Optimization*, vol. 2, no. 1, pp. 101–112, 1992.
- [13] H. D. Serali, E. Dalkiran, and J. Desai, "Enhancing rlt-based relaxations for polynomial programming problems via a new class of v-semidefinite cuts," *Computational Optimization and Applications*, vol. 52, no. 2, pp. 483–506, 2012.
- [14] M. S. Andersen, J. Dahl, and L. Vandenberghe, "Cvxopt: A python package for convex optimization, version 1.1. 6 (2013)."
- [15] R. Tedrake and t. D. D. Team, "Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems," 2016. [Online]. Available: <http://drake.mit.edu>
- [16] I. A. Şucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, December 2012, <http://ompl.kavrakilab.org>.
- [17] I. A. Şucan and L. E. Kavraki, "A sampling-based tree planner for systems with complex dynamics," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 116–131, 2012.
- [18] S. Kim and M. Kojima, "Second order cone programming relaxation of nonconvex quadratic optimization problems," *Optimization methods and software*, vol. 15, no. 3–4, pp. 201–224, 2001.
- [19] B. O’donoghue, E. Chu, N. Parikh, and S. Boyd, "Conic optimization via operator splitting and homogeneous self-dual embedding," *arXiv preprint arXiv:1312.3039*, 2013.