Convex Optimization of Nonlinear Feedback Controllers via Occupation Measures

Anirudha Majumdar, Ram Vasudevan, Mark M. Tobenkin, and Russ Tedrake
Computer Science and Artificial Intelligence Lab
Massachusetts Institute of Technology
Cambridge, MA, USA
Email: {anirudha,ramv,mmt,russt}@mit.edu

February 17, 2014

Abstract

The construction of feedback control laws for underactuated nonlinear robotic systems with input saturation limits is crucial for dynamic robotic tasks such as walking, running, or flying. Existing techniques for feedback control design are either restricted to linear systems, rely on discretizations of the state space, or require solving a non-convex optimization problem that requires feasible initialization. This paper presents a method for designing feedback controllers for polynomial systems that maximize the size of the time–limited backwards reachable set (BRS). In contrast to traditional approaches based on Lyapunov's criteria for stability, we rely on the notion of occupation measures to pose this problem as an infinite–dimensional linear program which can then be approximated in finite dimension via semidefinite programs (SDP)s. The solution to each SDP yields a polynomial control policy and an outer approximation of the largest achievable BRS which is well-suited for use in a trajectory library or feedback motion planning algorithm. We demonstrate the efficacy and scalability of our approach on six nonlinear systems. Comparisons to an infinite–horizon linear quadratic regulator approach and an approach relying on Lyapunov's criteria for stability are also included in order to illustrate the improved performance of the presented technique.

1 Introduction

Motion planning algorithms have begun to address robotic tasks that require simultaneously managing input saturation, nonlinear dynamics, and underactuation [LaValle, 2006, Karaman and Frazzoli, 2011, Kobilarov, 2012]. However due to the dynamic nature of such tasks, the open–loop motion plans constructed by applying such algorithms are inadequate since they are unable to correct for deviations from a planned path. As a result many techniques for feedback control synthesis have emerged such as those that rely on Dynamic Programming or the Hamilton–Jacobi Bellman Equation [Bertsekas, 2005a], [Ding and Tomlin, 2010, Mitchell et al., 2005], Nonlinear Model Predictive Control [Jadbabaie et al., 2001], feedback linearization [Sastry, 1999], or linearizing the dynamics about a nominal operating point in order to make Linear Quadratic Regulator (LQR) based techniques or Linear Model Predictive Control [Bemporad et al., 2002] applicable. Unfortunately, the Dynamic Programming and Hamilton–Jacobi Bellman Equation methods suffer from the curse of dimensionality and can require exorbitant grid resolution [Munos and Moore, 2002], while Nonlinear Model Predictive Control is still too computationally expensive for most real-world applications. Feedback linearization is untenable for underactuated systems [Spong, 1998] especially in the presence of actuation limits [Pappas et al., 1995] and other techniques that rely on linearizations generate locally valid controllers without providing any information about the neighborhood in which the controller is actually valid.

We address these shortcomings in this paper by describing an approach for designing feedback controllers for polynomial systems which maximize the set of points that reach a given target set at a specified finite time. We refer to this set as the time–limited *backwards reachable set* (BRS). Recently, a method relying on the notion of *occupation measures* was proposed in [Henrion and Korda, 2012] to compute the largest achievable BRS. In this paper, we extend this approach to perform feedback control synthesis by formulating the design of the controller that generates the largest BRS as an infinite–dimensional linear program (LP) over the space of nonnegative measures. To approximate this infinite–dimensional LP, we construct a sequence of finite–dimensional relaxations in terms of semidefinite programs (SDP)s. We prove that the finite–dimensional approximations satisfy two important convergence properties: first, each solution to this sequence of SDPs is an outer approximation to the largest possible BRS with asymptotically vanishing conservatism, and second, there exists a subsequence of the SDP solutions that weakly converges to an optimizing solution of the original infinite–dimensional LP.

1.1 Relationship to Lyapunov Techniques

The set of existing approaches to feedback control design that are most comparable to the one pursued herein are those that use Lyapunov's criteria for stability for simultaneous synthesis and maximization of the region of attraction of a particular target set. The criteria are checked for polynomial systems by using sums–of–squares (SOS) programming and result in a bilinear optimization problem that is solved using some form of alternation [Jarvis-Wloszek et al., 2005, Majumdar et al., 2013]. However, such methods are not guaranteed to converge to global optima (or necessarily even local optima) and require feasible initializations.

By examining the dual to the infinite–dimensional LP that we present, which is posed on the space of nonnegative continuous functions, one can appreciate the relationship between our approach and the Lyapunov–based approaches. The dual program, in particular, certifies that a certain set cannot reach the target set within a pre–specified time for any valid control law. The complement of this set is an outer approximation of the BRS. This subtle change transforms the non–convex feedback control synthesis problem written in terms of Lyapunov's criteria into the convex synthesis problem which we present. The convexity of the control synthesis approach we present also resembles the convexity observed in [Prajna et al., 2004] for the design of controllers to achieve global almost-everywhere asymptotic stability. Though certificates verifying global stability are of great utility, robotic systems are generally only locally stabilizable. As a result, having techniques that are able to generate regional certificates rather than global ones is critical.

1.2 Relationship to Occupation Measure Techniques

Our work extends the method described in [Henrion and Korda, 2012] which computes outer approximations to the largest possible BRS for a given target set. There are three important features of the approach presented in this paper that distinguish it from this prior work. First, in addition to computing outer approximations of the BRS, we are able to extract polynomial approximations to the feedback control law that maximizes the BRS. Next, as described formally in Section 4.2, the computational complexity of the algorithm presented here scales better with the number of control inputs of the system. Finally, as a consequence of this improved scaling, we are able to consider examples that approach the complexity of real-world robotic systems as we demonstrate in Section 6.

The occupation measure framework has been used in other contexts such as for the optimal control of piece-wise affine systems [Abdalmoaty et al., 2012], polynomial systems [Lasserre et al., 2008], and switched systems [Claeys et al., 2012, Henrion et al., 2013]. Of particular interest given the work presented here, is the approach for the generation of an optimal controller considered in [Claeys et al., 2012], which introduces the idea of interpreting the control inputs as signed measures rather than variables over which measures are defined. Though the control law construction we pursue is distinct from the optimal control law generation proposed in [Claeys et al., 2012], the idea of interpreting control inputs as signed measures is fundamental to our approach.

1.3 Contributions and Organization

The result of our analysis is an algorithm that could be used to augment existing feedback motion planning algorithms such as the LQR-Trees approach presented in [Tedrake et al., 2010] which computes and sequences together BRSs in order to drive a desired set of initial conditions to a predefined target set, or the method for performing robust online motion planning described in [Majumdar and Tedrake, 2012]. Our approach could be substituted for the local, linear control synthesis employed by the aforementioned papers with the benefit of selecting control laws that maximize the size of the BRS in the presence of input saturations. As a result, the number of trajectories required in a library in order to fill the space of possible initial conditions could be significantly reduced. A single nonlinear feedback controller could, in some cases, stabilize an entire set of initial conditions that previously required a library of locally-linear controllers.

The remainder of the paper is organized as follows: Section 2 introduces the notation used throughout the paper and briefly introduces infinite–dimensional LPs and SDPs; Section 3 formulates the feedback control synthesis problem as an infinite–dimensional LP and makes clear the relationship between traditional Lyapunov analysis and our approach; Section 4 constructs a sequence of finite–dimensional SDPs that approximate the infinite–dimensional LP, describes the scaling of these finite–dimensional approximations, and formalizes the convergence properties of this sequence; Section 5 details several straightforward extensions to the presented approach; Section 6 describes the performance of our approach on six examples, includes comparisons to a traditional infinite–horizon LQR–based approach, a method based on checking Lyapunov's criteria for stability via SOS programming, the method described in [Henrion and Korda, 2012], and also describes and illustrates a technique to take advantage of existing controllers to improve the performance of our approach; and Section 7 concludes the paper.

2 Background

In this section, we introduce the notation used throughout the remainder of the paper. We also include a brief introduction to infinite–dimensional LPs and SDPs. We make substantial use of measure theory in this paper, and the unfamiliar reader may wish to consult [Folland, 1999] for an introduction.

2.1 Notation

Given an element $y \in \mathbb{R}^{n \times m}$, let $[y]_{ij}$ denote the (i, j)–th component of y. We use the same convention for elements belonging to any multidimensional vector space. By $\mathbb{N}$ we denote the non-negative integers, and let $\mathbb{N}^n_k$ refer to those $\alpha \in \mathbb{N}^n$ with $|\alpha| = \sum_{i=1}^n [\alpha]_i \leq k$. Let $\mathbb{R}[y]$ denote the ring of real polynomials in the variable y.

2.2 Infinite Dimensional Linear Programs

Next we review infinite–dimensional LPs and their associated duality theory. The interested reader is encouraged to consult [Anderson and Nash, 1987] for a comprehensive introduction. Let X and Z be infinite–dimensional real vector spaces, A be a linear map from X to Z, and b be an element of Z. In addition, let c belong to the dual space of X, denoted X', whose elements are the linear functionals on X. A primal linear program expressed in standard form is written as:

$$\begin{aligned} \sup_{x \in X} \quad & \langle x, c \rangle \\ \text{s.t.} \quad & Ax = b, \\ & x \geq 0, \end{aligned} \tag{1}$$

where the inequality is interpreted element–wise. Note that LPs with constraints of the form $Ax \geq b$ can be written in standard form by introducing *slack variables*. More explicitly, the constraint $Ax \geq b$ can then be written as $Ax - z = b$, $z \geq 0$, with $z \in Z$.

An example of an infinite dimensional space X that is of particular interest to this paper is the space of measures supported on a compact domain of a Euclidean space. That is, for a compact set $K \subset \mathbb{R}^d$, let $\mathcal{M}(K)$ denote the space of signed Radon measures supported on K. The elements of $\mathcal{M}(K)$ can be identified with linear functionals acting on the space of continuous functions $C(K)$, that is, as elements of the dual space $C(K)'$ [Folland, 1999, Corollary 7.18]. Since $C(K) \subset \mathcal{M}(K)'$ [Folland, 1999, Theorem 5.8], the duality pairing of a measure $\mu \in \mathcal{M}(K)$ on a test function $v \in C(K)$ can be defined as:

$$\langle \mu, v \rangle = \int_K v(z) d\mu(z). \tag{2}$$

Similarly, the duality pairing of a vector of measures $\mu \in (\mathcal{M}(K))^p$ on a vector of test functions $v \in (C(K))^p$ is defined as:

$$\langle \mu, v \rangle = \sum_{i=1}^{p} \int_K [v]_i(z) d[\mu]_i(z). \tag{3}$$

As a result, a test function can be treated as a cost function over measures as in Program (1). In addition, the linear equality and non-negativity constraints appearing in Program (1) are able to describe linear equations on nonnegative measures.

Every primal LP has an associated *dual* LP. To formulate this dual LP, let Z' be the dual space of Z. Denote by A' the *adjoint* map from Z' to X' which is defined by:

$$\langle x, A'y \rangle = \langle Ax, y \rangle, \ \forall (x, y) \in X \times Z'. \tag{4}$$

The dual LP to the primal LP in Equation (1) expressed in standard form is:

$$\begin{aligned} \inf_{y \in Z'} \quad & \langle b, y \rangle \\ \text{s.t.} \quad & A'y - c \geq 0, \end{aligned} \tag{5}$$

For infinite–dimensional LPs there can be a difference between the primal and dual optimal values, which is referred to as the *duality gap*. Note the optimal value of a primal LP lower bounds the optimal value of its dual program. When there is no duality gap between the pair of programs *strong duality* is said to hold.

2.3 Semidefinite Programming

Infinite–dimensional LPs are generally not directly amenable to computation. Approximations to their solutions can be obtained by solving finite–dimensional semidefinite programs (SDPs). Details on this approximation are discussed in detail in Section 4. We briefly review some background on SDPs.

SDP problems are finite–dimensional convex optimization problems whose decision variables are symmetric matrices. Let $\mathbf{S}^n$ denote the space of symmetric $n \times n$ real–valued matrices. The objective function for SDPs is linear. The constraints are linear and semidefiniteness constraints on the decision variables. Recall that $Q \in \mathbf{S}^n$ is positive semidefinite, denoted $Q \succeq 0$, if $x^T Q x \geq 0, \forall x \in R^n$. A primal SDP expressed in standard form is written as:

$$\begin{aligned} \sup_{X \in \mathbf{S}^n} \quad & \langle C, X \rangle \\ \text{s.t.} \quad & \langle A_i, X \rangle = b_i \quad \forall i \in \{1, \dots, m\}, \\ & X \succeq 0, \end{aligned} \tag{6}$$

where $C, A_i \in \mathbf{S}^n$ and $\langle X, Y \rangle := \mathrm{Tr}(X^T Y)$. As in the LP case, there is a dual SDP associated to every primal SDP which when expressed in standard form is written as:

$$\begin{aligned} \inf_{y \in \mathbb{R}^m} \quad & b^T y \\ \text{s.t.} \quad & \sum_{i=1}^{m} A_i y_i - C \succeq 0. \end{aligned} \tag{7}$$

As in the infinite–dimensional LP case, the optimal value of a primal SDP lower bounds the optimal value of its dual program. Strong duality does not hold in general, but can be demonstrated if certain conditions are imposed on the SDPs.

Importantly SDPs can be used to check the non-negativity of polynomials. The decision problem associated with checking polynomial non-negativity is NP–hard in general [Parrilo, 2000]. However, the problem of determining whether a polynomial is a sum-of-squares (SOS), which is a sufficient condition for non–negativity, is amenable to efficient computation. A polynomial $p \in \mathbb{R}[x]$ is SOS if it can be written as $p(x) = \sum_{i=1}^{m} q_i^2(x)$ for a set of polynomials $\{q_i\}_{i=1}^{m} \subset \mathbb{R}[x]$. This condition is equivalent to the existence of a positive semidefinite symmetric matrix Q that satisfies:

$$p(x) = v(x)^T Q v(x), \tag{8}$$

where $v \in \mathbb{R}[x]$ is the vector of all monomials with degree less than or equal to half the degree of p [Parrilo, 2000]. The determination of the positive semidefinite matrix satisfying the linear constraints in the above equation can be posed as a SDP. To appreciate the connection of SDPs with the infinite–dimensional LPs over the space of measures, notice that the constraint in the dual infinite–dimensional LP in (5) requires checking the non–negativity of functions. This connection is formalized further in Section 4.

3 Problem Formulation

In this section, we formalize our problem of interest, construct an infinite–dimensional linear program (LP), and prove that the solution of this LP is equivalent to solving our problem of interest.

3.1 Problem Statement

Consider the control-affine system with feedback control

$$\dot{x}(t) = f\left(t, x(t)\right) + g\left(t, x(t)\right) u(t, x), \tag{9}$$

with state $x(t) \in \mathbb{R}^n$ and control action $u(t, x) \in \mathbb{R}^m$, such that the components of the vector f and the matrix g are polynomials. Our goal is to find a feedback controller, u, that maximizes the BRS for a given target set while respecting the *input constraint*

$$u(t, x) \in U = [a_1, b_1] \times \ldots \times [a_m, b_m], \tag{10}$$

where $\{a_1, \ldots, a_m\}, \{b_1, \ldots, b_m\} \subset \mathbb{R}$. Define the *bounding set*, and *target set* as:

$$\begin{aligned} X &= \left\{x \in \mathbb{R}^n \mid h_{X_i}(x) \geq 0, \forall i = \{1, \ldots, n_X\}\right\}, \\ X_T &= \left\{x \in \mathbb{R}^n \mid h_{T_i}(x) \geq 0, \forall i = \{1, \ldots, n_T\}\right\}, \end{aligned} \tag{11}$$

respectively, for given polynomials $h_{X_i}, h_{T_i} \in \mathbb{R}[x]$.

Given a finite final time $T > 0$, let the BRS for a particular control policy $u \in L^1([0, T] \times X, U)$, be defined as:

$$\begin{aligned} \mathcal{X}(u) = \Big\{ x_0 \in \mathbb{R}^n \mid & \dot{x}(t) = f\big(t, x(t)\big) + g\big(t, x(t)\big) u\big(t, x(t)\big) \\ & \text{a.e. } t \in [0, T],\ x(0) = x_0,\ x(T) \in X_T, \\ & x(t) \in X\ \forall t \in [0, T] \Big\}. \end{aligned} \tag{12}$$

$\mathcal{X}(u)$ is the set of initial conditions for solutions[1] to Equation (9) that remain in the bounding set and arrive in the target set at the final time when control law u is applied. Our aim is to find a controller $u^* \in L^1([0, T] \times X, U)$, that maximizes the volume of the BRS:

$$\lambda(\mathcal{X}(u^*)) \geq \lambda(\mathcal{X}(u)), \quad \forall u \in L^1([0, T] \times X, U), \tag{13}$$

where λ is the Lebesgue measure. u^* need not be unique. We denote the BRS corresponding to u^* by $\mathcal{X}^*$. To solve this problem, we make the following assumptions:

Assumption 1. *X and X_T are compact sets.*

Remark 1. *Without loss of generality, we assume that $U = \{u \in \mathbb{R}^m \mid -1 \leq u_j \leq 1\ \forall j \in \{1, \ldots, m\}\}$ (since g can be arbitrarily shifted and scaled). Assumption 1 ensures the existence of a polynomial $h_{X_i}(x) = C_X - \|x\|_2^2$ for a large enough $C_X > 0$.*

[1] Solutions in this context are understood in the Carathéodory sense, that is, as absolutely continuous functions whose derivatives satisfy the right hand side of Equation (9) almost everywhere [Aubin and Frankowska, 2008, Chapter 10]. Though absolutely continuous solutions exist even when u is just measurable, we focus our attention on absolutely integrable u since our goal is control synthesis which is accomplished via the Radon-Nikdoym Theorem [Folland, 1999, Theorem 3.8] in Theorem 3.

3.2 Liouville's Equation

We solve this problem by defining measures over $[0,T] \times X$ whose supports' model the evolution of *families of trajectories*. An initial condition and its relationship with respect to the terminal set can be understood via Equation (9), but the relationship between a family of trajectories and the terminal set must be understood through a different lens. First, define the linear operator $\mathcal{L}_f : C^1\big([0,T] \times X\big) \to C\big([0,T] \times X\big)$ on a test function v as:

$$\mathcal{L}_f v = \frac{\partial v}{\partial t} + \sum_{i=1}^{n} \frac{\partial v}{\partial x_i}[f]_i(t,x), \tag{14}$$

and its adjoint operator $\mathcal{L}'_f : C\big([0,T] \times X\big)' \to C^1\big([0,T] \times X\big)'$ by the adjoint relation:

$$\langle \mathcal{L}'_f \mu, v\rangle = \langle \mu, \mathcal{L}_f v\rangle = \int_{[0,T]\times X} \mathcal{L}_f v(t,x) d\mu(t,x) \tag{15}$$

for all $\mu \in \mathcal{M}\big([0,T] \times X\big)$ and $v \in C^1\big([0,T] \times X\big)$. Define the linear operator $\mathcal{L}_g : C^1\big([0,T] \times X\big) \to C\big([0,T] \times X\big)^m$ as:

$$[\mathcal{L}_g v]_j = \sum_{i=1}^{n} \frac{\partial v}{\partial x_i}[g]_{ij}(t,x), \tag{16}$$

for each $j \in \{1,\ldots,m\}$ and define its adjoint operator $\mathcal{L}'_g : \big(C\big([0,T] \times X\big)^m\big)' \to C^1\big([0,T] \times X\big)'$ according to its adjoint relation as in Equation (15). Note that $\mathcal{L}_f v(t,x) + (\mathcal{L}_g v(t,x))u(t,x)$ is the time-derivative $\dot{v}$ of a function v.

Given a test function, $v \in C^1\left([0,T] \times X\right)$, and an initial condition, $x(0) \in X$, it follows that:

$$v(T, x(T)) = v(0, x(0)) + \int_0^T \dot{v}\left(t, x(t|x_0)\right) dt. \tag{17}$$

The traditional approach to designing controllers that stabilize the system imposes Lyapunov conditions on the test functions and their derivatives. However, simultaneously searching for a controller and Lyapunov function results in a nonconvex optimization problem [Rantzer, 2001]. Instead we examine conditions on the space of measures–the dual to the space of functions–in order to arrive at a convex formulation.

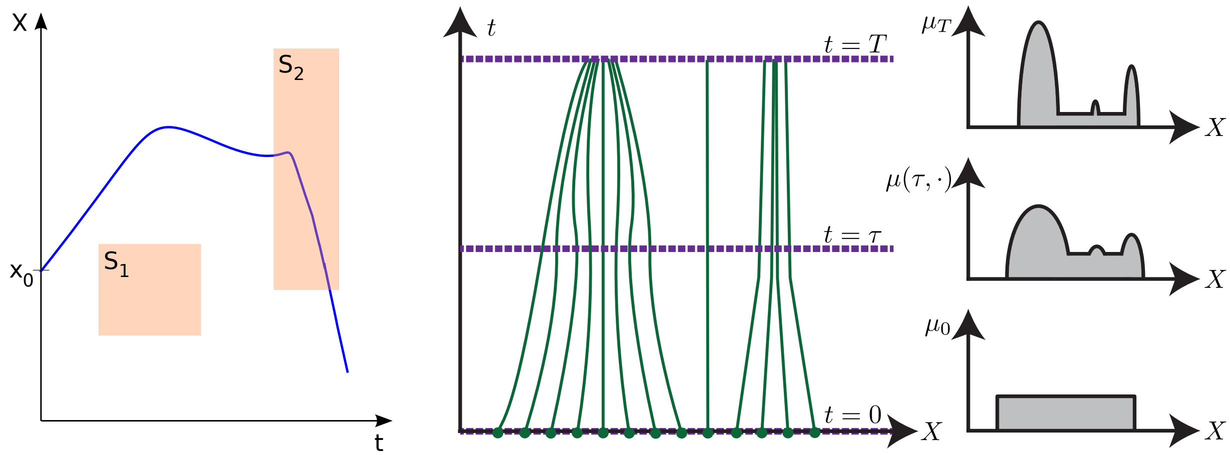
For a fixed control policy $u \in L^1\left([0,T] \times X, U\right)$ and an initial condition $x_0 \in \mathbb{R}^n$, let $x(\cdot|x_0) : [0,T] \to X$ be a solution to Equation (9). Define the *occupation measure* as:

$$\mu(A \times B|x_0) = \int_0^T I_{A\times B}\left(t, x(t|x_0))\right) dt, \tag{18}$$

for all subsets $A \times B$ in the Borel σ-algebra of $[0,T] \times X$, where $I_{A\times B}(\cdot)$ denotes the indicator function on a set $A \times B$. As a result of its definition, the occupation measure of a set $A \times B$ quantifies the amount of time the graph of the solution, $(t, x(t|x_0))$, spends in $A \times B$. In Figure 1(a), for example, the set S_1 has zero occupation measure though it has non-zero Lebesgue measure. The set S_2, in contrast, has non-zero occupation measure.

Equation (17) then becomes:

$$v(T, x(T)) = v(0, x(0)) + \int_{[0,T]\times X} \Big(\mathcal{L}_f v(t,x) + \mathcal{L}_g v(t,x)u(t,x)\Big) d\mu(t,x|x_0). \tag{19}$$



(a) For the initial condition x_0, the set S_1 has zero occupation measure, while the set S_2 has nonzero occupation measure.

(b) An illustration (left) of trajectories transforming according to Equation (9) and their corresponding occupation measures (right) at times $0, \tau$, and T transforming according to Equation (25).

Figure 1

When the initial state is not a single point, but is a distribution modeled by an *initial measure*, $\mu_0 \in \mathcal{M}(X)$, we define the *average occupation measure*, $\mu \in \mathcal{M}\left([0,T] \times X\right)$ by:

$$\mu(A \times B) = \int_X \mu(A \times B|x_0) d\mu_0(x_0), \tag{20}$$

and the *final measure*, $\mu_T \in \mathcal{M}\left(X_T\right)$ by:

$$\mu_T(B) = \int_X I_B(x(T|x_0)) d\mu_0(x_0). \tag{21}$$

Integrating with respect to μ_0 and introducing the initial, average occupation, and final measures, Equation (19) becomes:

$$\int\limits_{X_T} v(T,x) d\mu_T(x) = \int\limits_{X} v(0,x) d\mu_0(x) + \int_{[0,T]\times X} \Big(\mathcal{L}_f v(t,x) + \mathcal{L}_g v(t,x) u(t,x)\Big) d\mu(t,x). \tag{22}$$

It is useful to think of the measures μ_0, μ and μ_T as *unnormalized* probability distributions. The support of μ_0 models the set of initial conditions, the support of μ models the flow of trajectories, and the support of μ_T models the set of states at time T.

Next, we subsume $u(t,x)$ into a *signed* measure $\sigma^+ - \sigma^-$ defined by *nonnegative* measures[2] $\sigma^+, \sigma^- \in \left(\mathcal{M}\left([0,T] \times X\right)\right)^m$ such that:

$$\int_{A\times B} u_j(t,x) d\mu(t,x) = \int_{A\times B} d[\sigma^+]_j(t,x) - \int_{A\times B} d[\sigma^-]_j(t,x) \tag{23}$$

[2]Note that we can always decompose a signed measure into unsigned measures as a result of the Jordan Decomposition Theorem [Folland, 1999, Theorem 3.4].

for all subsets $A \times B$ in the Borel σ-algebra of $[0,T] \times X$ and for each $j \in \{1, \ldots, m\}$. This key step allows us to pose an infinite–dimensional LP over measures without explicitly parameterizing a control law while allowing us to "back out" a control law using Equation (23). Using the notation from Section 2.2, Equation (22) becomes:

$$\langle \mu_T, v(T, \cdot)\rangle = \langle \mu_0, v(0, \cdot)\rangle + \langle \mu, \mathcal{L}_f v\rangle + \langle \sigma^+ - \sigma^-, \mathcal{L}_g v\rangle \tag{24}$$

for all test functions $v \in C^1([0,T] \times X)$. Notice that this substitution renders Equation (24) linear in its measure components. To reflect the evaluation of the test function at a specific time, let δ_t denote the Dirac measure at a point t and let $\otimes$ denote the product of measures. Since Equation (24) must hold for all test functions, we obtain a linear operator equation:

$$\mathcal{L}'_f \mu + \mathcal{L}'_g \sigma^+ - \mathcal{L}'_g \sigma^- = \delta_T \otimes \mu_T - \delta_0 \otimes \mu_0, \tag{25}$$

called Liouville's Equation, which is a classical result in statistical physics that describes the evolution of a density of particles within a fluid [Arnold, 1989]. The occupation measures μ_0, μ and μ_T, along with Liouville's equation allow us to reason about *families* of trajectories of the dynamical system. This point of view is illustrated in Figure 1(b), which depicts the evolution of densities according to Liouville's Equation. This equation is satisfied by families of admissible trajectories starting from the initial distribution μ_0. The converse statement is true for control affine systems with a convex admissible control set, as we have assumed. We refer the reader to [Henrion and Korda, 2012, Appendix A] for an extended discussion of Liouville's Equation.

3.3 BRS via an Infinite Dimensional LP

The goal of this section is to use Liouville's Equation to formulate an infinite–dimensional LP, P, that maximizes the size of the BRS, modeled by $\mathrm{spt}(\mu_0)$, for a given target set, modeled by $\mathrm{spt}(\mu_T)$, where $\mathrm{spt}(\mu)$ denotes the support of a measure μ. Slack measures (denoted with "hats") are used to impose the constraints $\lambda \geq \mu_0$ and $\mu \geq [\sigma^+]_j + [\sigma^-]_j$ for each $j \in \{1, \ldots, m\}$, as was described in Section 2.2, where λ is the Lebesgue measure. The former constraint ensures that the optimal value of P is the Lebesgue measure of the largest achievable BRS (see Theorem 2). The latter constraint ensures that we are able to extract a bounded control law by applying Equation (23) (see Theorem 3). Define P as:

$$\begin{array}{lll}
\sup & \mu_0(X) & (P) \\
\text{s.t.} & \mathcal{L}'_f \mu + \mathcal{L}'_g(\sigma^+ - \sigma^-) = \delta_T \otimes \mu_T - \delta_0 \otimes \mu_0, & \\
 & [\sigma^+]_j + [\sigma^-]_j + [\hat{\sigma}]_j = \mu & \forall j \in \{1, \ldots, m\}, \\
 & \mu_0 + \hat{\mu}_0 = \lambda, & \\
 & [\sigma^+]_j, [\sigma^-]_j, [\hat{\sigma}]_j \geq 0 & \forall j \in \{1, \ldots, m\}, \\
 & \mu, \mu_0, \mu_T, \hat{\mu}_0 \geq 0, &
\end{array}$$

where the given data are f, g, X, X_T and the supremum is taken over a tuple of measures $(\sigma^+, \sigma^-, \hat{\sigma}, \mu, \mu_0, \hat{\mu}_0, \mu_T) \in \big(\mathcal{M}\big([0,T] \times X\big)\big)^m \times \big(\mathcal{M}\big([0,T] \times X\big)\big)^m \times \big(\mathcal{M}\big([0,T] \times X\big)\big)^m \times \mathcal{M}\big([0,T] \times X\big) \times \mathcal{M}(X) \times \mathcal{M}(X) \times \mathcal{M}(X_T)$. Given measures that achieve the supremum, the control law that maximizes the size of the BRS is then constructed by finding the $u \in L^1([0,T] \times X, U)$ whose components each satisfy Equation (23) for all subsets in the Borel σ-algebra of $[0,T] \times X$. Before proving that this two-step procedure computes $u^* \in L^1([0,T]\times X, U)$ as in Equation (13), define the dual program to P denoted D as:

$$\begin{aligned}
\inf \quad & \int_X w(x)d\lambda(x) && (D)\\
\text{s.t.} \quad & \mathcal{L}_f v + \Sigma_{i=1}^m [p]_i \le 0, \\
& [p]_i \ge 0, \quad [p]_i \ge |[\mathcal{L}_g v]_i| && \forall i \in \{1,\dots,m\},\\
& w \ge 0, \\
& w(x) \ge v(0,x) + 1 && \forall x \in X,\\
& v(T,x) \ge 0 && \forall x \in X_T
\end{aligned}$$

where the given data are f, g, X, X_T and the infimum is taken over $(v, w, p) \in C^1([0,T]\times X) \times C(X) \times (C([0,T]\times X))^m$. The dual allows us to obtain approximations of the BRS $\mathcal{X}^*$ (see Theorem 4).

Before continuing, we briefly describe the relationship between the dual program D and optimization problems that rely on Lyapunov's criteria for stability. Notice in particular the relationship between the function v in D and a Lyapunov function. The constraints $\mathcal{L}_f v + \sum_{i=1}^m [p]_i \le 0$, $[p]_i \ge 0$, and $[p]_i \ge |[\mathcal{L}_g v]_i|$, as we describe in Theorem 2, are equivalent to the constraint that v decrease along trajectories of the system for any valid control input. This, in addition to the constraint requiring $v(T,x) \ge 0$ for all $x \in \mathcal{X}_T$, implies that the 0-sublevel set of $v(0,x)$ for $x \in X$ is an inner approximation to the set of points that *cannot* reach the target set.

Therefore, in contrast to Lyapunov's criteria for stability which certifies that a given set of points reaches the target set, potentially in an asymptotic fashion, v certifies that a given set of points does not reach the target set. This in turn implies that the 1-super level set of w is an *outer approximation* to the BRS. The benefit of this reformulation of Lyapunov's criteria is the removal of the control input u from the set of decision variables for the program D. This removes the non–convexity that arises by requiring that the time derivative $\dot{V}(t,x) = \frac{\partial V(t,x)}{\partial t} + \nabla_x V(t,x)^T [f(x) + g(x)u]$ of a Lyapunov function V be non–positive, which is bilinear in the decision variables corresponding to V and u. Importantly, using our approach, we can still extract a controller by examining the primal problem P as we describe in the following theorems.

Theorem 1. *There is no duality gap between P and D.*

Proof. We use an argument similar to the one in [Henrion and Korda, 2012, Theorem 2]. To simplify the presentation we define:

$$\mathcal{C} = ((\mathcal{C}([0,T]\times X))^m)^3 \times \mathcal{C}([0,T]\times X) \times \mathcal{C}(X) \times \mathcal{C}(X) \times \mathcal{C}(X_T) \tag{26}$$

$$\mathcal{M} = ((\mathcal{M}([0,T]\times X))^m)^3 \times \mathcal{M}([0,T]\times X) \times \mathcal{M}(X) \times \mathcal{M}(X) \times \mathcal{M}(X_T), \tag{27}$$

and let $\mathcal{K}$ and $\mathcal{K}'$ denote the positive cones of $\mathcal{C}$ and $\mathcal{M}$, respectively. The cone K' is equipped with the weak*-topology [Folland, 1999, p. 169]. The LP problem (P) whose solution is denoted p^* can be rewritten as:

$$\begin{aligned}
\sup \quad & \langle \gamma, c\rangle \\
\text{s.t.} \quad & \mathcal{A}'\gamma = \beta, \\
& \gamma \in \mathcal{K}',
\end{aligned} \tag{28}$$

where the supremum is over the vector $\gamma = (\sigma^+, \sigma^-, \hat{\sigma}, \mu, \mu_0, \hat{\mu}_0, \mu_T)$, the linear operator $\mathcal{A}' : \mathcal{K}' \to C^1([0,T] \times X)' \times (\mathcal{M}([0,T] \times X))^m \times \mathcal{M}(X)$ is defined by $\mathcal{A}'\gamma = (-\mathcal{L}_f' \mu - \mathcal{L}_g'(\sigma^+ - \sigma^-) - \delta_0 \otimes \mu_0 + \delta_T \otimes \mu_T, [\sigma^+]_1 + [\sigma^-]_1 + [\hat{\sigma}]_1, \ldots, [\sigma^+]_m + [\sigma^-]_m + [\hat{\sigma}]_m, \mu_0 + \hat{\mu}_0)$, the right hand side of the equality constraint in Equation (28) is the vector of measures $\beta = (0, \underbrace{0, \ldots, 0}_{m \text{ times}}, \lambda)$, the vector function in the objective function is $c = (\underbrace{0, \ldots, 0}_{m \text{ times}}, 0, 1, 0, 0) \in \mathcal{C}$, so the objective function itself is:

$$\langle \gamma, c \rangle = \int_X d\mu_0 = \mu_0(X). \tag{29}$$

The LP problem in Equation (28) can be interpreted as the dual to the following LP problem:

$$\begin{aligned} \inf \quad & \langle \beta, z \rangle \\ \text{s.t.} \quad & \mathcal{A}z - c \in \mathcal{K}, \end{aligned} \tag{30}$$

where the infimum is over $z = (v, p, w) \in C^1([0,T] \times X) \times (C([0,T] \times X))^m \times C(X)$ and the linear operator $\mathcal{A} : C^1([0,T] \times X) \times (C([0,T] \times X))^m \times C(X) \to \mathcal{C}$ is defined by:

$$\mathcal{A}z = ([p]_1 - |[\mathcal{L}_g v]_1|, \ldots, [p]_m - |[\mathcal{L}_g v]_m|, -\mathcal{L}_f v - \sum_{i=1}^m [p]_i, w - v(0, \cdot), v(T, \cdot), w), \tag{31}$$

and satisfies the adjoint relation $\langle \mathcal{A}'\gamma, z \rangle = \langle \gamma, \mathcal{A}z \rangle$. The LP problem defined in Equation (30) is exactly dual to (D).

From [Anderson and Nash, 1987, Theorem 3.10] there is no duality gap between the LPs defined in Equations (28) and (30) if the supremum p^* is finite and the set $P = \{(\mathcal{A}'\gamma, \langle \gamma, c \rangle) \mid \gamma \in \mathcal{K}'\}$ is closed in the weak*-topology of $\mathcal{K}'$. The finiteness of p^* follows from the constraint $\mu + \hat{\mu}_0 = \lambda$, $\hat{\mu}_0 \geq 0$, and from the compactness of X. To prove that P is closed, notice that $\mathcal{A}'$ is weak*-continuous since $\mathcal{A}z \in \mathcal{C}$ for all $z \in C^1([0,T] \times X) \times (C([0,T] \times X))^m \times C(X)$. Next consider a sequence $\gamma_k \in \mathcal{K}'$ such that $\mathcal{A}'\gamma_k \to a$ and $\langle \gamma_k, c \rangle \to b$ as $k \to \infty$ for some $(a, b) \in C^1([0,T] \times X)' \times (\mathcal{M}([0,T] \times X))^m \times \mathcal{M}(X) \times \mathbb{R}$. Since the support of the measures is compact and $T < \infty$, the sequence γ_k is bounded. From the weak*-compactness of the unit ball which follows from the Banach-Alaoglu Theorem [Folland, 1999, Theorem 5.18], there is a subsequence γ_{k_i} that weak*-converges to an element $\gamma \in \mathcal{K}'$ so that $\lim_{i \to \infty}(\mathcal{A}'\gamma_{k_i}, \langle \gamma_{k_i}, c \rangle) \in P$. □

Theorem 2. *The optimal value of P is equal to $\lambda(\mathcal{X}^*)$, the Lebesgue measure of the BRS of the controller defined by Equation* (23)*.*

Proof. Since there is no duality gap between P and D, it is sufficient to show that the optimal value of D is equal to $\lambda(\mathcal{X}^*)$. We do this by demonstrating that D is equivalent to the dual LP defined in Equation (16) in [Henrion and Korda, 2012], whose optimal value is equal to $\lambda(\mathcal{X}^*)$ [Henrion and Korda, 2012, Theorem 1, Theorem 2]. Note that the constraints $w(x) \geq v(0,x) + 1$, $v(T,x) \geq 0$, and $w(x) \geq 0$ appear in both optimization problems. Since the objectives are also identical, it suffices to show that the first three constraints in D are equivalent to the constraint $\mathcal{L}_f v(t,x) + (\mathcal{L}_g v(t,x))u \leq 0 \ \forall (t,x,u) \in [0,T] \times X \times U$. Suppose that the former set of the three constraints holds. Given $u \in U$, note that $\mathcal{L}_f v + (\mathcal{L}_g v)u \leq \mathcal{L}_f v + \Sigma_{i=1}^m |[\mathcal{L}_g v]_i u_i|$. Hence, since $[p]_i \geq |[\mathcal{L}_g v]_i|$, $\mathcal{L}_f v + \Sigma_{i=1}^m [p]_i \leq 0$, and $|u_i| \leq 1$ (see Remark 1), we have the desired result.

To prove the converse, we illustrate the existence of $[p]_i(t,x) \geq 0$ that satisfies the three constraints appearing in D. Let $[p]_i(t,x) = |\mathcal{L}_g v(t,x)]_i|$, which is a non-negative continuous function. Clearly, $p_i \geq [\mathcal{L}_g v]_i$ and $[p]_i \geq -[\mathcal{L}_g v]_i$. To finish the proof, note:

$$\mathcal{L}_f v(t,x) + \Sigma_{i=1}^m [p]_i(t,x) = \sup_{u \in U} \mathcal{L}_f v(t,x) + \mathcal{L}_g v(t,x))u \leq 0$$

□

The solution to P can be used in order to construct the control law that maximizes the BRS:

Theorem 3. *Let $(\sigma^{+*}, \sigma^{-*}, \hat{\sigma}^*, \mu^*, \mu_0^*, \hat{\mu}_0^*, \mu_T^*)$ be the vector of measures that achieves the supremum of P. There exists a control law, $\tilde{u} \in L^1([0,T] \times X, U)$, that simultaneously satisfies Equation (23) when substituting in $(\sigma^{+*}, \sigma^{-*}, \hat{\sigma}^*, \mu^*, \mu_0^*, \hat{\mu}_0^*, \mu_T^*)$ and maximizes the size of the BRS, i.e. $\lambda(\mathcal{X}(\tilde{u})) \geq \lambda(\mathcal{X}(u)), \forall u \in L^1([0,T] \times X, U)$. Moreover, any two control laws constructed by applying Equation (23) to the vector of measures that achieves the supremum of P are equal μ^*-almost everywhere.*

Proof. Note that $[\sigma^{+*}]_j, [\sigma^{-*}]_j$, and μ^* are σ-finite for all $j \in \{1, \ldots, m\}$ since they are Radon measures defined over a compact set. Define $[\sigma^*]_j = [\sigma^{+*}]_j - [\sigma^{-*}]_j$ for each $j \in \{1, \ldots, m\}$ and notice that each $[\sigma^*]_j$ is also σ-finite. Since $[\sigma^{+*}]_j + [\sigma^{-*}]_j + [\hat{\sigma}^*]_j = \mu^*$ and $[\sigma^{+*}]_j, [\sigma^{-*}]_j, [\hat{\sigma}^*]_j \geq 0$, σ^* is absolutely continuous with respect to μ^*. Therefore as a result of the Radon–Nikodym Theorem [Folland, 1999, Theorem 3.8], there exists a $\tilde{u} \in L^1([0,T] \times X, U)$, which is unique μ^*-almost everywhere, that satisfies Equation (23) when plugging in the vector of measures that achieves the supremum of P. To see that $\lambda(\mathcal{X}(\tilde{u})) \geq \lambda(\mathcal{X}(u)), \forall u \in L^1([0,T] \times X, U)$, notice that by construction μ_T^*, μ_0^*, μ^*, and $\tilde{u}$ satisfy Equation (22) for all test functions $v \in C^1([0,T] \times X)$. Since μ_0^* describes the maximum BRS and Equation (22) describes all admissible trajectories, we have our result. □

Next, we prove that the w-component of feasible solutions to D converges from above to the indicator function on $\mathcal{X}^*$. As a result, the sequence of 1-superlevel sets of feasible w-components can be used as an outer approximation to the true BRS.

Theorem 4. *$\mathcal{X}^*$ is a subset of $\{x \mid w(x) \geq 1\}$, for any feasible w of the D. Furthermore, there is a sequence of feasible solutions to D such that the w-component converges from above to $I_{\mathcal{X}^*}$ in the L^1 norm and almost uniformly.*

Proof. Since D is equivalent to the dual LP defined in Equation (15) in [Henrion and Korda, 2012], this follows from Lemma 2 and Theorem 3 in [Henrion and Korda, 2012]. □

4 Numerical Implementation

The infinite–dimensional problems P and D are not directly amenable to computation. However, a sequence of finite–dimensional approximations in terms of SDPs can be obtained by characterizing measures in P by their *moments*, and restricting the space of functions in D to polynomials. The solutions to each of the SDPs in this sequence can be used to construct controllers and outer approximations that converge to the solution of the infinite–dimensional LP. A comprehensive introduction to such *moment relaxations* can be found in [Lasserre, 2010].

Measures on the set $[0,T] \times X$ are completely determined by their action (via integration) on a dense subset of the space $C^1([0,T] \times X)$ [Folland, 1999]. Since $[0,T] \times X$ is compact, the Stone–Weierstrass Theorem [Folland, 1999, Theorem 4.45] allows us to choose the set of polynomials as this dense subset. Every polynomial on $\mathbb{R}^n$, say $p \in \mathbb{R}[x]$ with $x = (x_1, \dots, x_n)$, can be expanded in the monomial basis via

$$p(x) = \sum_{\alpha \in \mathbb{N}^n} p_\alpha x^\alpha,$$

where $\alpha = (\alpha_1, \dots, \alpha_n)$ ranges over vectors of non-negative integers, $x^\alpha = x_1^{\alpha_1} \dots x_n^{\alpha_n}$, and $\text{vec}(p) = (p_\alpha)_{\alpha \in \mathbb{N}^n}$ is the vector of coefficients of p. By definition, the p_α are real and only finitely many are non-zero. We define $\mathbb{R}_k[x]$ to be those polynomials such that p_α is non-zero only for $\alpha \in \mathbb{N}_k^n$. The degree of a polynomial, $\deg(p)$, is the smallest k such that $p \in \mathbb{R}_k[x]$.

The moments of a measure μ defined over a real n-dimensional space are given by:

$$y_\mu^\alpha = \int x^\alpha d\mu(x). \tag{32}$$

Integration of a polynomial with respect to a measure ν can be expressed as a linear functional of its coefficients:

$$\langle \mu, p \rangle = \int p(x) d\mu(x) = \sum_{\alpha \in \mathbb{N}^n} p_\alpha y_\mu^\alpha = \text{vec}(p)^T y_\mu. \tag{33}$$

Integrating the square of a polynomial $p \in \mathbb{R}_k[x]$, we obtain:

$$\int p(x)^2 d\mu(x) = \text{vec}(p)^T M_k(y_\mu) \text{vec}(p), \tag{34}$$

where $M_k(y_\mu)$ is the *truncated moment matrix* defined by

$$[M_k(y_\mu)]_{(\alpha,\beta)} = y_\mu^{\alpha+\beta} \tag{35}$$

for $\alpha, \beta \in \mathbb{N}_k^n$. Note that for any positive measure μ, the matrix $M_k(y_\mu)$ must be positive semidefinite. Similarly, given $h \in \mathbb{R}[x]$ with $(h_\gamma)_{\gamma \in \mathbb{N}^n} = \text{vec}(h)$ one has

$$\int p(x)^2 h(x) d\mu(x) = \text{vec}(p)^T M_k(h, y_\mu) \text{vec}(p), \tag{36}$$

where $M_k(h, y)$ is a *localizing matrix* defined by

$$[M_k(h, y_\mu)]_{(\alpha,\beta)} = \sum_{\gamma \in \mathbb{N}^n} h_\gamma y_\mu^{\alpha+\beta} \tag{37}$$

for all $\alpha, \beta \in \mathbb{N}_k^n$. The localizing and moment matrices are symmetric and linear in the moments y.

4.1 Approximating Problems

Finite dimensional SDPs approximating P can be obtained by replacing constraints on measures with *constraints on moments*. All of the equality constraints of P can be expressed as an infinite–dimensional linear system of equations which the moments of the measures appearing in P must satisfy. This linear system is obtained by restricting to polynomial test functions (which we note are sufficient given our discussion above): $v(t, x) = t^\alpha x^\beta$, $[p]_j(t, x) = t^\alpha x^\beta$, and $w(x) = x^\beta$, $\forall \alpha \in$

$\mathbb{N}, \beta \in \mathbb{N}^n$. For example, the equality constraint corresponding to Liouville's Equation is obtained by examining:

$$0 = \int_{[0,T]\times X} \mathcal{L}_f(t^\alpha x^\beta) d\mu(t,x) + \int_{[0,T]\times X} \mathcal{L}_g(t^\alpha x^\beta) d[\sigma^+]_j(t,x)$$
$$- \int_{[0,T]\times X} \mathcal{L}_g(t^\alpha x^\beta) d[\sigma^-]_j(t,x) - \int_{X_T} T^\alpha x^\beta d\mu_T(x) + \int_X x^\beta d\mu_0(x).$$

A *finite–dimensional* linear system is obtained by truncating the degree of the polynomial test functions to $2k$. Let $\Gamma = \{\sigma^+, \sigma^-, \hat{\sigma}, \mu, \mu_0, \hat{\mu}_0, \mu_T\}$, then let $\mathbf{y}_k = (y_{k,\gamma}) \subset \mathbb{R}$ be a vector of sequences of moments truncated to degree $2k$ for each $\gamma \in \Gamma$. This is analogous to truncating the degree of a polynomial while performing SOS–programming to find a Lyapunov function. The finite–dimensional linear system is then represented by the linear system:

$$A_k(\mathbf{y}_k) = b_k. \tag{38}$$

Constraints on the support of the measures also need to be imposed (see [Lasserre, 2010] for details). Let the k-th relaxed SDP representation of P, denoted P_k, be defined as:

sup	y^0_{k,μ_0}	(P_k)
s.t.	$A_k(\mathbf{y}_k) = b_k,$	
	$M_k(y_{k,\gamma}) \succeq 0$	$\forall \gamma \in \Gamma,$
	$M_{k_{X_i}}(h_{X_i}, y_{k,\gamma}) \succeq 0$	$\forall (i,\gamma) \in \{1,\dots,n_X\} \times \Gamma \backslash \mu_T,$
	$M_{k_{T_i}}(h_{T_i}, y_{k,\mu_T}) \succeq 0$	$\forall i \in \{1,\dots,n_T\},$
	$M_{k-1}(h_\tau, y_{k,\gamma}) \succeq 0$	$\forall \gamma \in \Gamma \backslash \{\mu_0, \mu_T, \hat{\mu}_0\},$

where the given data are f, g, X, X_T and the supremum is taken over the sequence of moments, $\mathbf{y}_k = (y_{k,\gamma})$, $h_\tau = t(T-t)$, $k_{X_i} = k - \lceil \deg(h_{X_i})/2 \rceil$, $k_{T_i} = k - \lceil \deg(h_{T_i})/2 \rceil$, and $\succeq 0$ denotes positive semi-definiteness. For each $k \in \mathbb{N}$, let $\mathbf{y}^*_k$ denote the optimizer of P_k, with components $y^*_{k,\gamma}$ where $\gamma \in \Gamma$ and let p^*_k denote the supremum of P_k.

The dual of P_k is a sums-of-squares (SOS) program denoted D_k for each $k \in \mathbb{N}$, which is obtained by first restricting the optimization space in the D to the polynomial functions with degree truncated to $2k$ and by then replacing the non-negativity constraint D with an SOS constraint [Parrilo, 2000]. Define $Q_{2k}(h_{X_1}, \dots, h_{X_{n_X}}) \subset \mathbb{R}_{2k}[x]$ to be the set of polynomials $q \in \mathbb{R}_{2k}[x]$ (i.e. of total degree less than $2k$) expressible as

$$q = s_0 + \sum_{i=1}^{n_X} s_i h_{X_i}, \tag{39}$$

for some polynomials $\{s_i\}_{i=0}^{n_X} \subset \mathbb{R}_{2k}[x]$ that are sums of squares of other polynomials. Every such polynomial is clearly non-negative on X. Define $Q_{2k}(h_\tau, h_{X_1}, \dots, h_{X_{n_X}}) \subset \mathbb{R}_{2k}[t,x]$ and $Q_{2k}(h_{T_1}, \dots, h_{T_{n_T}}) \subset \mathbb{R}_{2k}[x]$, similarly. Employing this notation, the k-th relaxed SDP representation of D, denoted D_k, is defined as:

$$\begin{aligned} \inf \quad & l^T \text{vec}(w) && (D_k)\\ \text{s.t.} \quad & -\mathcal{L}_f v - \mathbf{1}^T p \in Q_{2k}(h_\tau, h_{X_1}, \ldots, h_{X_{n_X}}), \\ & p - (\mathcal{L}_g v)^T \in (Q_{2k}(h_\tau, h_{X_1}, \ldots, h_{X_{n_X}}))^m, \\ & p + (\mathcal{L}_g v)^T \in (Q_{2k}(h_\tau, h_{X_1}, \ldots, h_{X_{n_X}}))^m, \\ & w \in Q_{2k}(h_{X_1}, \ldots, h_{X_{n_X}}), \\ & w - v(0, \cdot) - 1 \in Q_{2k}(h_{X_1}, \ldots, h_{X_{n_X}}), \\ & v(T, \cdot) \in Q_{2k}(h_{T_1}, \ldots, h_{T_{n_T}}), \end{aligned}$$

where the given data are f, g, X, X_T, the infimum is taken over the vector of polynomials $(v, w, p) \in \mathbb{R}_{2k}[t, x] \times \mathbb{R}_{2k}[x] \times (\mathbb{R}_{2k}[t, x])^m$, and l is a vector of moments associated with the Lebesgue measure (i.e. $\int_X w \, d\lambda = l^T \text{vec}(w)$ for all $w \in \mathbb{R}_{2k}[x]$). For each $k \in \mathbb{N}$, let d_k^* denote the infimum of D_k.

Theorem 5. *For each $k \in \mathbb{N}$, there is no duality gap between P_k and D_k.*

Proof. This is a standard result from the theory of SDP duality so we do not include all of the proof details here. The proof involves noting that the moment vectors in SDP, P_k, are necessarily bounded because of the constraint $\mu_0 + \hat{\mu}_0 = \lambda$, and then arguing that the feasible set of the SDP, D_k, has an interior point. The existence of an interior point is sufficient to establish zero duality gap [Trnovská, 2005, Theorem 5]. □

Next, we present a technique to extract a polynomial control law $u_k(t, x)$ from the solution $\mathbf{y}_k$ of P_k. Given moment sequences truncated to degree $2k$, one can choose an approximate control law u_k with components $[u_k]_j \in \mathbb{R}_k[t, x]$ so that the truncated analogue of Equation (23) is satisfied. That is, by requiring:

$$\int_{[0,T]\times X} t^{\alpha_0} x^\alpha [u_k]_j(t, x) \, d\mu(t, x) = \int_{[0,T]\times X} t^{\alpha_0} x^\alpha d[\sigma^+ - \sigma^-]_j(t, x), \tag{40}$$

for (α_0, α) satisfying $\sum_{i=0}^n \alpha_i \le k$. When constructing a polynomial control law from the solution of P_k, these linear equations written with respect to the coefficients of $[u_k]_j$ are expressible in terms of $y^*_{k,\sigma^+}, y^*_{k,\sigma^-}$, and $y^*_{k,\mu}$. Direct calculation shows the linear system of equations is:

$$M_k(y^*_{k,\mu}) \text{vec}([u_k]_j) = y^*_{k,[\sigma^+]_j} - y^*_{k,[\sigma^-]_j}. \tag{41}$$

To extract the coefficients of the control law, one needs only to compute the generalized inverse of $M_k(y^*_{k,\mu})$ which exists since $M_k(y^*_{k,\mu})$ is positive semidefinite. Note that the degree of the extracted polynomial control law is dependent on the relaxation order k. Higher relaxation orders lead to higher degree controllers.

4.2 Computational Complexity

Our approach scales in a manner similar to Lasserre's hierarchy of SDP based relaxations for polynomial optimization [Lasserre, 2010]. In particular, since the measures appearing in the infinite–dimensional LP P are all supported on subsets of $[0, T] \times X$ with total dimension $n+1$, the number of moments associated with a given measure grows as $O((n+1)^k)$ for a fixed relaxation order k and scales as $O(k^{n+1})$ for fixed state space dimension n. The total number of measures in the program P scales linearly with the number of control inputs m.

We note that this is in contrast to the approach presented in [Henrion and Korda, 2012] for computing outer approximations of the BRS, wherein the occupation measures are supported on variables corresponding to time, state *and* the control input (dimension $1+n+m$). The number of moments in that formulation scales as $O((1+n+m)^k)$ for a given k and $O(k^{1+n+m})$ for fixed n and m. The reason for this difference in complexity stems from the fact that in our approach the control input is *not* treated as a separate variable u over which measures (and thus test functions) are defined. Instead, we introduce a signed measure $\sigma^+ - \sigma^-$ in equation (23) corresponding to the control input which depends only on t and x. As a result of this reduction in computational complexity, we are able to handle larger state and input spaces, as we illustrate in Section 6.

4.3 Convergence of Approximating Problems

Next, we prove the convergence properties of P_k and D_k and the corresponding controllers. We begin by proving that the polynomial w approximates the indicator function of the set $\mathcal{X}^*$. As we increase k, this approximation gets tighter. The following theorem, whose proof is a modified version of the proof of Theorem 5 in [Henrion and Korda, 2012], makes this statement precise.

Theorem 6. *For each $k \in \mathbb{N}$, let $w_k \in \mathbb{R}_{2k}[x]$ denote the w-component of the solution to D_k, and let $\bar{w}_k(x) = min_{i\leq k} w_i(x)$. Then, w_k converges from above to $I_{\mathcal{X}^*}$ in the L^1 norm, and $\bar{w}_k(x)$ converges from above to $I_{\mathcal{X}^*}$ in the L^1 norm and almost uniformly.*

Proof. From Theorem 4, for every $\epsilon > 0$, there exists a feasible tuple of functions $(v, w, p) \in C^1([0,T]\times X) \times C(X) \times (C([0,T]\times X))^m$ such that $w \geq I_{\mathcal{X}^*}$ and $\int_X (w - I_{\mathcal{X}^*}) d\lambda < \epsilon$. Let $\tilde{v}(t,x) := v(t,x) - 3\epsilon T + 3(T+1)\epsilon$, $\tilde{w}(x) := w(x) + 3(T+3)\epsilon$ and $[\tilde{p}]_i(t,x) = [p]_i(t,x) + (2\epsilon)/m, \forall i = \{1,\ldots,m\}$. Then, $\mathcal{L}_f \tilde{v} = \mathcal{L}_f v - 3\epsilon$, $\tilde{v}(t,x) = v(T,x) + 3\epsilon$, $\tilde{w}(x) - \tilde{v}(0,x) = 1 + 6\epsilon$, and $\mathcal{L}_g \tilde{v} = \mathcal{L}_g v$. Since the sets X and $[0,T]\times X$ are compact, and by a generalization of the Stone-Weierstrass theorem that allows for the simultaneous uniform approximation of a function and its derivatives by a polynomial [Hirsch, 1976, pp. 65-66], we are guaranteed the existence of polynomials $\hat{v}, \hat{w}, [\hat{p}]_i$ such that $\|\hat{v} - \tilde{v}\|_\infty < \epsilon$, $\|\mathcal{L}_f \hat{v} - \mathcal{L}_f \tilde{v}\|_\infty < \epsilon$, $\|\mathcal{L}_g \hat{v} - \mathcal{L}_g \tilde{v}\|_\infty < \epsilon/m$, $\|\hat{w} - \tilde{w}\|_\infty < \epsilon$ and $\|[\hat{p}]_i - [\tilde{p}]_i\|_\infty < \epsilon/m$. It is easily verified that these polynomials *strictly* satisfy the constraints of D_k. Hence, by Putinar's Positivstellensatz [Lasserre, 2010] and Remark 1, we are guaranteed that these polynomials are feasible for D_k for high enough degree of multiplier polynomials. We further note that $\hat{w} \geq w$. Then, $\int_X |\tilde{w} - \hat{w}| d\lambda \leq \epsilon\lambda(X)$, and thus $\int_X (\hat{w} - w) d\lambda \leq \epsilon\lambda(X)(3T+10)$. Hence, since $w \geq I_{\mathcal{X}^*}$ and $\int_X (w - I_{\mathcal{X}^*}) d\lambda < \epsilon$ by assumption, it follows that $\int_X (\hat{w} - I_{\mathcal{X}^*}) d\lambda < \epsilon(1 + \lambda(X)(3T+10))$ and $\hat{w} \geq I_{\mathcal{X}^*}$. This concludes the first part of the proof since ϵ was arbitrarily chosen.

The convergence of w_k to $I_{\mathcal{X}^*}$ in L^1 norm implies the existence of a subsequence w_{k_i} that converges almost uniformly to $I_{\mathcal{X}^*}$ [Ash, 1972, Theorems 2.5.2, 2.5.3]. Since $\bar{w}_k(x) \leq \min\{w_{k_i} : k_i \leq k\}$, this is sufficient to establish the second claim. □

Corollary 1. *$\{d_k^*\}_{k=1}^\infty$ and $\{p_k^*\}_{k=1}^\infty$ converge monotonically from above to the optimal value of D and P.*

Proof. This is a direct consequence of Theorems 1 and 6. □

Next, we prove that the 1-superlevel set of the polynomial w converges in Lebesgue measure to the largest achievable BRS $\mathcal{X}^*$. The proof of the statement is similar to Theorem 6 in [Henrion and Korda, 2012].

Theorem 7. *For each $k \in \mathbb{N}$, let $w_k \in \mathbb{R}_{2k}[x]$ denote the w-component of the solution to D_k, and let $\mathcal{X}_k := \{x \in \mathbb{R}^n \ : \ w_k(x) \geq 1\}$. Then, $\lim_{k\to\infty} \lambda(\mathcal{X}_k \backslash \mathcal{X}^*) = 0$.*

Proof. Using Theorem 4 we see $w_k \geq I_{\mathcal{X}_k} \geq I_{\mathcal{X}^*}$. From Theorem 6, we have $w_k \to I_{\mathcal{X}^*}$ in L^1 norm on X. Hence:

$$\lambda(\mathcal{X}^*) = \lim_{k\to\infty} \int_X w_k d\lambda \quad \geq \lim_{k\to\infty} \int_X I_{\mathcal{X}_k} d\lambda = \lim_{k\to\infty} \lambda(\mathcal{X}_k).$$

But since $\mathcal{X}^* \subset \mathcal{X}_k$ for all k, we must have $\lim_{k\to\infty} \lambda(\mathcal{X}_k) = \lambda(\mathcal{X}^*)$ and thus $\lim_{k\to\infty} \lambda(\mathcal{X}_k \backslash \mathcal{X}^*) = 0$. □

Finally, we prove a convergence result for the sequence of controllers generated by (41).

Theorem 8. *Let $\{y^*_{k,\gamma}\}_{\gamma\in\Gamma}$ be an optimizer of P_k and let $\{\gamma^*_k\}_{k=1}^\infty$ be any sequence of measures such that the truncated moments of γ^*_k match $y^*_{k,\gamma}$ for each $\gamma \in \Gamma$. In addition, for each $k \in \mathbb{N}$, let u^*_k denote the controller constructed by Equation (41) using $\{y^*_{k,\gamma}\}_{\gamma\in\Gamma}$. Then, there exists an optimizing vector of measures $(\sigma^{+*}, \sigma^{-*}, \hat{\sigma}^*, \mu^*, \mu^*_0, \hat{\mu}^*_0, \mu^*_T)$ for P, a $u^* \in L^1([0,T] \times X)$ generated using σ^{+*}, σ^{-*}, and μ^* according to Equation (23), and a subsequence $\{k_i\}_{i=1}^\infty \subset \mathbb{N}$ such that:*

$$\int\limits_{[0,T]\times X} v(t,x)[u^*_{k_i}]_j(t,x) d\mu^*_{k_i}(t,x) \xrightarrow{i\to\infty} \int\limits_{[0,T]\times X} v(t,x)[u^*]_j(t,x) d\mu^*(t,x), \tag{42}$$

for all $v \in C^1([0,T] \times X)$, and each $j \in \{1, \ldots, m\}$.

Proof. As a result of Theorem 4.3 in [Lasserre, 2010], if we complete each $y^*_{k,\gamma}$ with zeros to make it an infinite vector, then there exists a $y^*_\gamma \in l_\infty$ and a subsequence $\{k_i\}_{i\in\mathbb{N}}$ such that for each $\gamma \in \Gamma$, $\lim_{i\to\infty} y^*_{k_i,\gamma} = y^*_\gamma$. Moreover, as a result of Theorem 3.8(b) in [Lasserre, 2010] for each $\gamma \in \Gamma$ y^*_γ has a finite Borel representing measure, and this set of represented measures, which we denote by $(\sigma^{+*}, \sigma^{-*}, \hat{\sigma}^*, \mu^*, \mu^*_0, \hat{\mu}^*_0, \mu^*_T)$, is an optimizing vector of measures for P as a result of the proof of Theorem 4.3 in [Lasserre, 2010].

Next, note that the set of test functions v can be restricted to polynomials since the set of polynomials is dense in $C^1([0,T] \times X)$. Using Equations (23), (40), and (41):

$$\begin{aligned}\int\limits_{[0,T]\times X} v\Big([u^*_{k_i}]_j d\mu^*_{k_i} - [u^*]_j d\mu^*\Big) &= \int\limits_{[0,T]\times X} v\Big(d[\sigma^{+*}_{k_i}]_j - d[\sigma^{-*}_{k_i}]_j - d[\sigma^{+*}]_j + d[\sigma^{-*}]_j\Big) \\ &= \text{vec}(v)\Big(y^*_{k_i,\sigma^+_j} - y^*_{\sigma^+_j} + y^*_{\sigma^-_j} - y^*_{k_i,\sigma^-_j}\Big).\end{aligned}$$

where we have suppressed the dependence on (t,x) for convenience. The desired result follows from the previous equality since for each $\gamma \in \Gamma$, $\lim_{i\to\infty} y^*_{k_i,\gamma} = y^*_\gamma$. □

4.4 Implementation Details

The process of constructing SDPs P_k and D_k can be automated given the following problem data: polynomials f and g describing the dynamical system, a time-horizon $T > 0$, polynomials h_{X_i} and h_{T_i} describing the bounding set X and target set X_T, and a relaxation order k. This can be done in two different ways using freely available software. The first approach is to specify the primal problem over measures using the software toolbox GloptiPoly [Henrion et al., 2009]. GloptiPoly is a software package that features a user friendly interface for specifying and approximately solving the Generalized Problem of Moments (GPM) using SDP. The infinite dimensional LP P is an example of the GPM and can thus be specified using GloptiPoly, which can construct the SDPs P_k automatically from a high-level description of P. The second option is to directly specify the SOS programs D_k using a software package such as YALMIP [Löfberg, 2004]. This is the approach we took to construct the SDPs arising from the examples considered in Section 6 with an implementation that made use of a custom software package and the YALMIP toolbox. In both cases, the constructed SDP can be solved using a SDP solver such as SeDuMi [Sturm, 1999] or the commercially available MOSEK solver.

Note that both GloptiPoly and YALMIP allow one to extract dual variables from the specified programs (assuming a primal-dual solver such as SeDuMi or MOSEK has been used to solve the SDP). If one is using GloptiPoly, this allows one to extract the polynomials v and w describing outer approximations of the BRS. If on the other hand YALMIP is used to specify the SOS program D_k, one can extract the moment matrices in P_k required to construct the control law using Equation (41) from the solution.

5 Extensions

In this section, we describe two extensions to the approach presented in this paper.

5.1 Free Final Time

First, we examine the case where the task is to drive initial conditions to the target set within a pre-specified time $T < \infty$ rather than exactly at time T. Solutions are not required to remain within the target set once they reach it. This provides the controller with a degree of flexibility and a task specification that maybe more appropriate for rendezvous–type applications. We refer to this task as the *Free Final Time* problem.

The primal LP to address this problem is obtained from the LP P presented in Section 3 by modifying the support of the final measure μ_T to be $[0, T] \times X_T$. Each of the constraints are identical to the ones in P, but now $\mu_T \in \mathcal{M}\big([0, T] \times X_T\big)$. Consequently, the only modification to the original dual program D is that the non–negativity constraint on the function v is imposed for all time $t \in [0,T]$ instead of just at time T:

$$\begin{aligned}
\inf \quad & \int_X w(x)d\lambda(x) & & (43) \\
\text{s.t.} \quad & \mathcal{L}_f v + \Sigma_{i=1}^m [p]_i \le 0, & & \\
& [p]_i \ge 0, \quad [p]_i \ge |[\mathcal{L}_g v]_i| & & \forall i = \{1,\dots,m\}, \\
& w \ge 0, & & \\
& w(x) \ge v(0,x) + 1 & & \forall x \in X, \\
& v(t,x) \ge 0 & & \forall (t,x) \in [0,T] \times X_T
\end{aligned}$$

Each of the aforementioned corollaries and theorems extend to this problem with nearly identical proof and the numerical implementation follows in a straightforward manner.

5.2 Trigonometric variables

Our method can be readily adapted to handle dynamics with trigonometric dependence on states, x_i, so long as the dynamics are polynomial in $\sin(x_i)$ and $\cos(x_i)$. This is accomplished by introducing indeterminates c_i and s_i identified with $\sin(x_i)$ and $\cos(x_i)$ and modifying the SOS constraints to work over the quotient ring of polynomials associated with the ideal generated by the unit-circle constraints: $s_i^2 + c_i^2 = 1$. Details of the construction and implementation involved with this approach are discussed in [Parrilo, 2003, Section 4]. Note that in general, this ability to handle trigonometric variables comes at a cost. In particular, one needs to replace a variable x_i with *two* variables, thus increasing the number of states of the system.

6 Examples

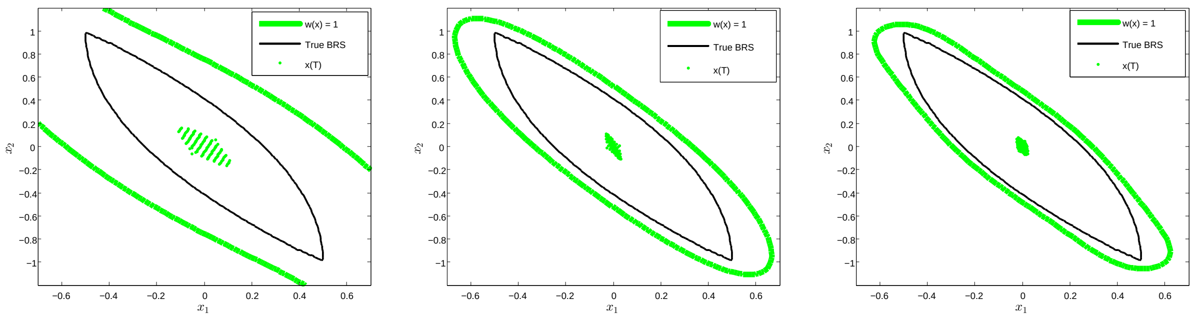
This section provides a series of numerical experiments on example systems of varying complexity. Each SDP was prepared using a custom software toolbox and the modeling tool YALMIP [Löfberg, 2004]. The programs are run with the freely available SeDuMi 1.3 [Sturm, 1999] and the commercial solver MOSEK on a machine with 8 Intel Xeon processors with a clock speed of 3.1 GHz and 32 GB RAM. During simulation, applied control inputs were taken taken to be the saturation of the synthesized feedback control law.

6.1 Double Integrator

The double integrator is a two state, single input system given by:

$$\begin{aligned}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= u,
\end{aligned} \tag{44}$$

with u restricted to the interval $U = [-1, 1]$. Setting the target set to the origin, $X_T = \{0\}$, and $T = 1$ the optimal BRS $\mathcal{X}^*$ can be computed analytically based on a minimum time "bang-bang controller" [Bertsekas, 2005b, pp. 136]. Note that this is a challenging system for grid based optimal control methods, since they require high resolution near the switching surface of the "bang-bang" control law.



(a) $k = 2$ (b) $k = 3$ (c) $k = 4$

Figure 2: An illustration of the convergence of outer approximations and the performance of controllers designed using our approach for increasing truncation degree, k, for the double integrator. Solid thick lines indicate the outer approximations, defined by $w_k = 1$. The boundary of the true BRS (which is analytically computable for this example) is depicted as the solid thinner line. Points indicate terminal states ($x(T)$) of controlled solutions with initial states $x(0)$ inside the true BRS using our generated feedback control laws u_k.

We take the bounding set to be $X = \{x \mid \|x\|^2 \leq 1.6^2\}$. Figure 2 compares the outer approximations of $\mathcal{X}^*$ for $k = 2, 3, 4$. The quality of the approximations increases quickly. Figure 2 also evaluates the performance of the control laws u_k by plotting the terminal states $x(T)$ for controlled solutions starting in $\mathcal{X}^*$. Even for $k = 3$, reasonable performance is achieved. The running times for $k = 2, 3, 4$ are 0.3, 0.7, and 4.2 seconds, respectively with the SeDuMi solver and 0.22, 0.28, and 0.51 seconds respectively with MOSEK.

As a concrete example of the output of our approach, we note that the time-varying feedback control law produced for $k = 2$ is:

$$u_2(t, x) = -1.541x_1 - 4.046x_1t - 1.099x_2 - 3.677x_2t.$$

6.2 Ground Vehicle Model

The Dubin's car [Dubins, 1957] is a popular model for autonomous ground and air vehicles and has been employed in a wide variety of applications [Bhatia et al., 2008, Chen and Ozguner, 2007, Gray et al., 2012]. Its dynamics are:

$$\begin{aligned} \dot{a} &= v\cos(\theta), \\ \dot{b} &= v\sin(\theta), \\ \dot{\theta} &= \omega, \end{aligned} \tag{45}$$

where the states are the x-position (a), y-position (b) and yaw angle (θ) of the vehicle and the control inputs are the forward speed (v) and turning rate (ω). A change of coordinates can be applied to this system in order to make the dynamics polynomial [DeVon and Bretl, 2007]. That is by considering the coordinate transform:

$$\begin{aligned} z_1 &= \theta \\ z_2 &= a\cos(\theta) + b\sin(\theta) \\ z_3 &= a\sin(\theta) - b\cos(\theta), \end{aligned} \tag{46}$$

with the input transformation:

$$\begin{aligned} u_1 &= \omega \\ u_2 &= v - z_3 u_1, \end{aligned} \tag{47}$$

the kinematics in Equation (45) can be expressed in power form as:

$$\begin{aligned} \dot{z}_1 &= u_1 \\ \dot{z}_2 &= u_2 \\ \dot{z}_3 &= z_2 u_1. \end{aligned} \tag{48}$$

By defining new state variables as:

$$\begin{aligned} x_1 &= z_1 \\ x_2 &= z_2 \\ x_3 &= -2z_3 + z_1 z_2, \end{aligned} \tag{49}$$

the system takes the form:

$$\begin{aligned} \dot{x}_1 &= u_1, \\ \dot{x}_2 &= u_2, \\ \dot{x}_3 &= x_1 u_2 - x_2 u_1. \end{aligned} \tag{50}$$

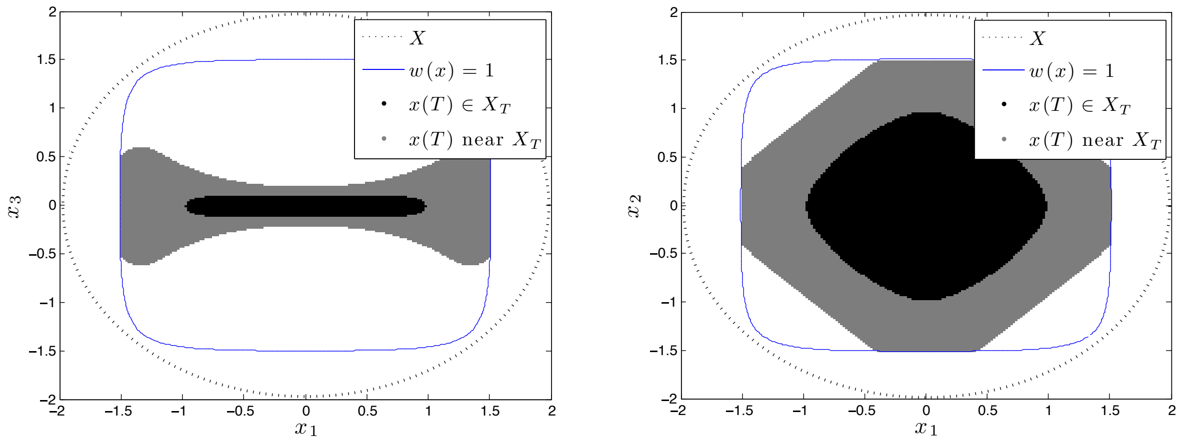
This system is also known as the Brockett integrator and is a popular benchmark since it is prototypical of many nonholonomic systems. Notice that the system has an uncontrollable linearization and does not admit a smooth time-invariant control law that makes the origin asymptotically stable [DeVon and Bretl, 2007]. Hence, this example illustrates the advantage of our method when compared to linear control synthesis techniques. We solve the "free final time" problem to construct a time-varying control law that drives the initial conditions in $X = \{x \mid \|x\|^2 \leq 4\}$ to the target set $X_T = \{x \mid \|x\|^2 \leq 0.1^2\}$ by time $T = 4$. In the Dubin's car coordinates, the target set is a neighborhood of the origin while being oriented along the positive a-axis. The control is restricted to $u_1, u_2 \in [-1, 1]$. Figure 3 plots outer approximations of the BRS for $k = 5$.

Figure 4 illustrates two sample trajectories generated using a feedback controller designed by our algorithm after transforming back to the original coordinate system. Note the nontrivial nature of the trajectories resulting from applying our nonlinear time-varying feedback law. The trajectory starting in the upper right hand corner executes a three-point turn to arrive at the desired position and orientation. Solving the SDP took 599 seconds with SeDuMi and 48 seconds with MOSEK.

6.3 Torque Limited Simple Pendulum

Next, we consider a two-state torque limited simple pendulum, described by the equation

$$I\ddot{\theta} = mgl\sin(\theta) - b\dot{\theta} + u, \tag{51}$$



(a) $k = 5$, (x_1, x_3) plane

(b) $k = 5$, (x_1, x_2) plane

Figure 3: The boundary of X (solid line) and the outer approximation of the BRS (dotted line) are shown in the (x_1, x_3) and (x_1, x_2) planes. In each plane, dark points indicate initial conditions of controlled solutions with $x(T) \in X_T$ (i.e. $\|x(T)\|^2 \leq 0.1^2$), and light points indicate initial conditions of solutions ending near the target set (specifically $\|x(T)\|^2 \leq 0.2^2$).

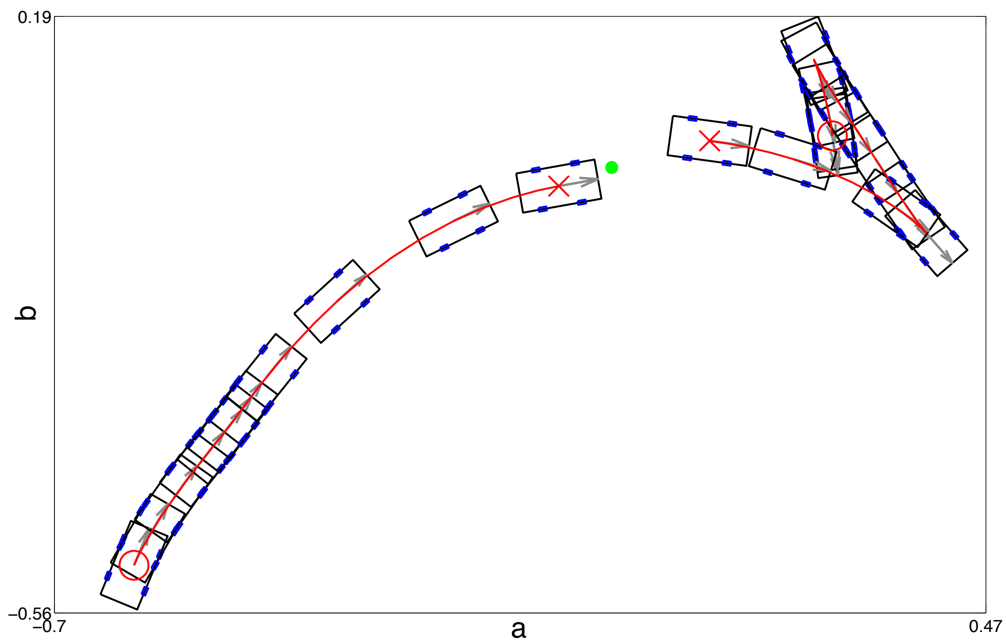


Figure 4: A pair of sample trajectories drawn as a line beginning with a circle and ending as a cross generated by our algorithm for the Dubin's car system. Each drawn time sample of the car includes an arrow indicating the forward–facing direction. The origin is marked by a filled in dot. The target set is a neighborhood of the origin with orientation pointing along the positive a-axis. The trajectory starting in the upper right hand corner executes a three-point turn to arrive at the desired position and orientation.

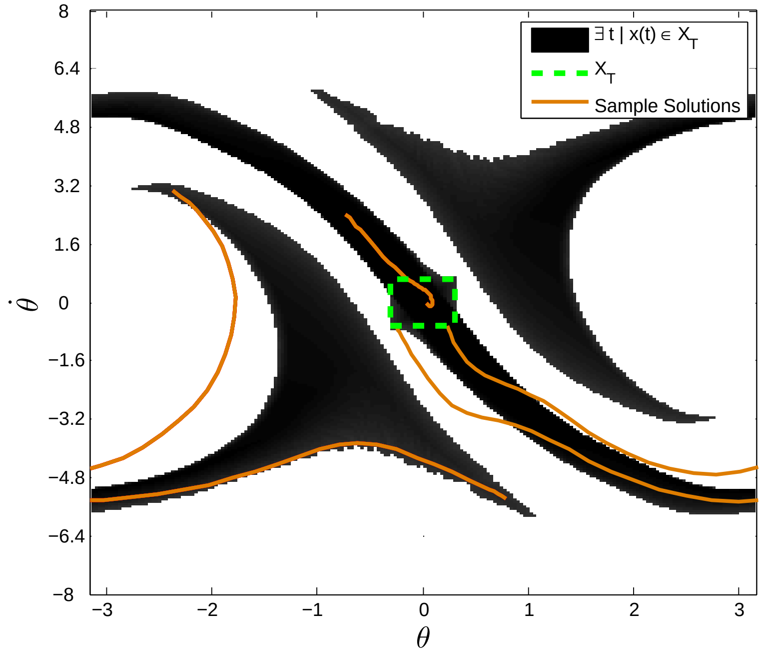


Figure 5: A depiction of the controller performance for $k = 5$ for the torque limited simple pendulum. Black regions indicate sampled initial conditions whose controlled solutions pass through the target set (dotted square). Three sample solutions are also plotted and each have terminal conditions in the target set. Note the solution starting near $(-2, 3.2)$ passes through zero velocity — the solution "pumps" to reach the upright position. While at first glance it may seem like the trajectories should remain within the black region at all times, this is not in fact the case due to the time-varying nature of the control law. Some points along the trajectories do not belong to the black region because the controller would succeed from those points only if it was applied using a starting time different from zero.

where θ represents the angle from upright and u represents a torque source at the pivot constrained to take values in $U = [-3, 3]$. We take $m = 1$, $l = 0.5$, $I = ml^2$, $b = 0.1$ and $g = 9.8$. The bounding set is defined by $\theta \in [-\pi, \pi)$ and $\dot{\theta} \in [-8, 8]$. We use the method described in Section 5.2 in order to directly handle the trigonometric dynamics of the system.

For this example, we solve the Free Final Time problem by taking $T = 1.5$, and defining the target set as $X_T = \{(\theta, \dot{\theta}) \mid \cos(\theta) \geq 0.95, \dot{\theta}^2 \leq 0.05\}$. The running time for the SDP is 680 seconds with SeDuMi and 156 secs with MOSEK for $k = 5$. Figure 5 plots sample solutions and summarizes the initial conditions that reach the target set. Notice that the controller is able to "swing-up" states close to the downright position to the upright configuration despite the stringent actuator limits and a short time-horizon. In particular, the trajectory starting close to $(-2, 3.2)$ in Figure 5 "pumps" energy into the system by first moving in one direction and then switching direction to swing up and reach the upright position.

6.4 Satellite Attitude Control

We now examine a six state system with three inputs describing attitude control of a satellite with thrusters applying torques. The dynamics are defined by

$$\begin{aligned} H\dot{\omega} &= -\Omega(\omega)H\omega + u, \\ \dot{\psi} &= \frac{1}{2}(I + \Omega(\psi) + \psi\psi^T)\omega, \end{aligned} \tag{52}$$

where $\omega \in \mathbb{R}^3$ are the angular velocities in the body-frame, $\psi \in \mathbb{R}^3$ represent the attitude as modified Rodriguez parameters (see [Prajna et al., 2004]), $\Omega : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ is the matrix defined so that $\Omega(\psi)\omega =$ is the cross product $\psi \times \omega$, and $H \in \mathbb{R}^{3\times 3}$ is the inertia matrix. We let H be diagonal with $[H]_{11} = 2$, $[H]_{22} = 1$ and $[H]_{33} = \frac{1}{2}$.

We take the input constraint set as $U = [-1, 1]^3$ and the origin as a target set. We apply the proposed control design methods with $k = 3$. Solving the SDP took approximately 6 hours with SeDuMi and 1 hour with MOSEK. Figure 6 examines the controller performance. A set of initial conditions are sampled from a hyperplane, and those whose solutions arrive near the target set are highlighted. We note that a SDP with $k = 2$ takes only 62.3 seconds, but yields a controller and BRS approximations that are slightly inferior, though potentially still useful in practice.

As mentioned in Section 4.2, one advantage of our approach over the method presented in [Henrion and Korda, 2012] for computing outer approximations of the BRS (besides the ability to extract a control law) is the computational scaling of the proposed algorithm with respect to the number of control inputs. On the satellite system, our implementation of the algorithm presented in [Henrion and Korda, 2012] is unable to run due to memory (RAM) constraints for a relaxation order of $k = 2$ or $k = 3$. For $k = 1$, using MOSEK the running time is 909.1 seconds. In contrast, our method using MOSEK takes 3.2 seconds to run.

6.5 Cart–Pole

Next, we consider the four state, single input cart–pole system illustrated in Figure 7. Its dynamics are defined by

$$\begin{aligned} (m_c + m_p)\ddot{x} - m_p l\ddot{\theta}\cos(\theta) &= -m_p l\dot{\theta}^2\sin(\theta) + u, \\ l\ddot{\theta} - \ddot{x}\cos(\theta) &= g\sin(\theta), \end{aligned} \tag{53}$$

where x is the position of the cart, θ is the angle of the pole, and u is the control input and corresponds to the force applied to the cart. The mass of the cart, m_c, is set equal to 10, the length of the pole, l, is set equal to 0.5, and the point mass at the end of the pole m_p is set equal to 1. The control input is restricted to the interval $[-35, 35]$.

We solve the Free Final Time problem with $T = 20$ and set the target set equal to the origin. The dynamics are expanded using the Taylor Series about the origin to degree three in order to obtain polynomial dynamics since the dynamics are rational rather than polynomial in the trigonometric variables. However when we simulate the system to verify the performance of our algorithm, we use the original dynamics rather than the Taylor Series expanded dynamics. The SDP takes 15.8 seconds to run using MOSEK when $k = 3$.

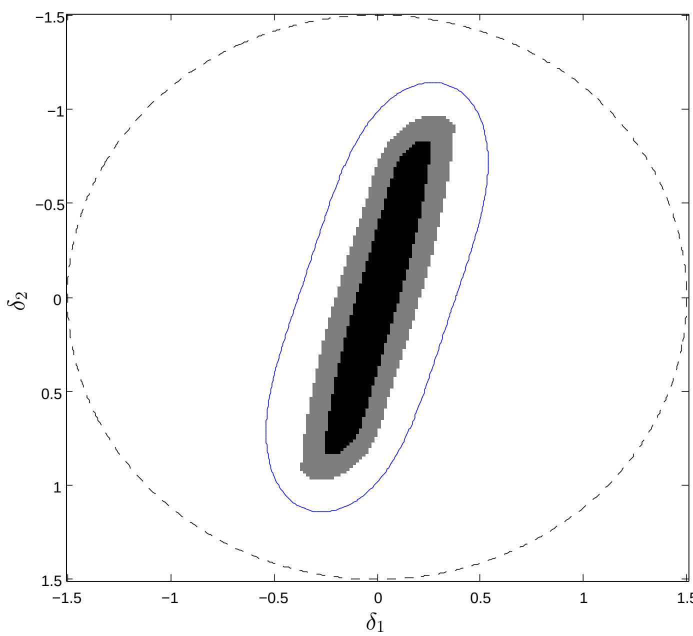


Figure 6: Demonstration of the controller performance for $k = 3$ for the satellite system. Points are sampled from the bounding set (dashed black) and the set where $w(x) \leq 1$ (boundary is solid line). To excite coupled dynamics between the angular velocities, initial conditions are chosen from the hyperplane with coordinates (δ_1, δ_2) given by $\delta_1 = (\psi_1 + \psi_2)/\sqrt{2}$ and $\delta_2 = (\dot{\omega}_1 + \dot{\omega}_2)/\sqrt{2}$. Black (resp. grey) points indicate initial conditions whose controller solution satisfies $\|x(T)\| \leq 0.1$ (resp. $\|x(T)\| \leq 0.2$).

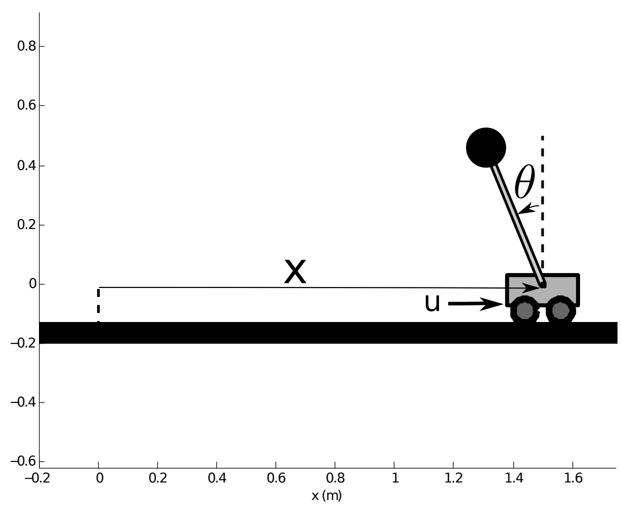


Figure 7: An illustration of the cart–pole system showing the configuration variables and the control input.

6.5.1 Comparison to LQR

We compare the performance of our controller to an infinite–horizon LQR controller designed by linearizing the dynamics about the origin [Kwakernaak and Sivan, 1972]. The state and input cost matrices are chosen to be a diagonal matrix $Q \in \mathbb{R}^{4\times 4}$ with $[Q]_{11} = 200$, $[Q]_{22} = 30$, $[Q]_{33} = 100$, and $[Q]_{44} = 30$, and a scalar $R = 0.5$, respectively. These cost matrices for the LQR controller were chosen by careful hand-tuning in order to obtain good performance.

Figure 8 illustrates the performance of the two controllers by simulating both closed loop systems forward for $T = 20$ from initial conditions densely sampled from two-dimensional slices of the state-space, i.e. simulating with initial conditions that have all but two of the states set to 0. Points that get close to the target (all coordinates less than 0.5 in absolute value) are plotted as circles for our controller and filled in dots for the LQR controller. In most of the slices, our controller is able to drive more initial conditions to the desired target set in comparison to the LQR controller. We also note that when the LQR cost matrices are designed without careful hand tuning the difference between the performance of our controller and the LQR controller is even more dramatic.

6.5.2 Improving Performance Using LQR

Though our controller does demonstrate better performance than the LQR controller on most of the slices, it is clear that in the θ–$\dot{\theta}$ slice our controller performs worse than the LQR one. We briefly describe how to take advantage of an existing well-tuned LQR controller (or any existing feedback controller in general) in order to further improve the performance of our approach. In particular, consider applying our method for designing a controller u to the following system:

$$\dot{x} = f(x) + g(x)[u + Kx], \tag{54}$$

where Kx is the feedback term from the LQR controller.

This modification of the system dynamics in the infinite–dimensional LP P could potentially allow our approach to take advantage of the local performance of the LQR controller. However, if the bound on the control input, u_j, of the original system are $[-1, 1]$, then the quantity $[u + Kx]_j$ must be bounded between $[-1, 1]$. This requires replacing the constraint $\mu \geq [\sigma^+]_j + [\sigma^-]_j$ in the infinite–dimensional LP P with the following two constraints:

$$\int_{[0,T]\times X} s_j d\mu \geq \int_{[0,T]\times X} s_j [d\sigma^+]_j - \int_{[0,T]\times X} s_j [d\sigma^-]_j + \int_{[0,T]\times X} s_j [Kx]_j d\mu \quad \forall j \in \{1, \dots m\}, \tag{55}$$

$$\int_{[0,T]\times X} r_j d\mu \geq - \int_{[0,T]\times X} r_j [d\sigma^+]_j + \int_{[0,T]\times X} r_j [d\sigma^-]_j - \int_{[0,T]\times X} r_j [Kx]_j d\mu \quad \forall j \in \{1, \dots, m\}, \tag{56}$$

for all $s_j, r_j \in C^1([0, T] \times X)$. These constraints can also be written as:

$$\mu \geq [\sigma^+]_j - [\sigma^-]_j + [Kx]_j \mu \quad \forall j \in \{1, \dots, m\}, \tag{57}$$

$$\mu \geq -[\sigma^+]_j + [\sigma^-]_j - [Kx]_j \mu \quad \forall j \in \{1, \dots, m\}. \tag{58}$$

The dual program corresponding to this modified program is derived in a manner similar to the dual program presented earlier, so we do not present it here.

Figure 9 compares the result of applying the controller obtained from this method to the original LQR controller on the aforementioned cart–pole system task. As before, we sample initial conditions

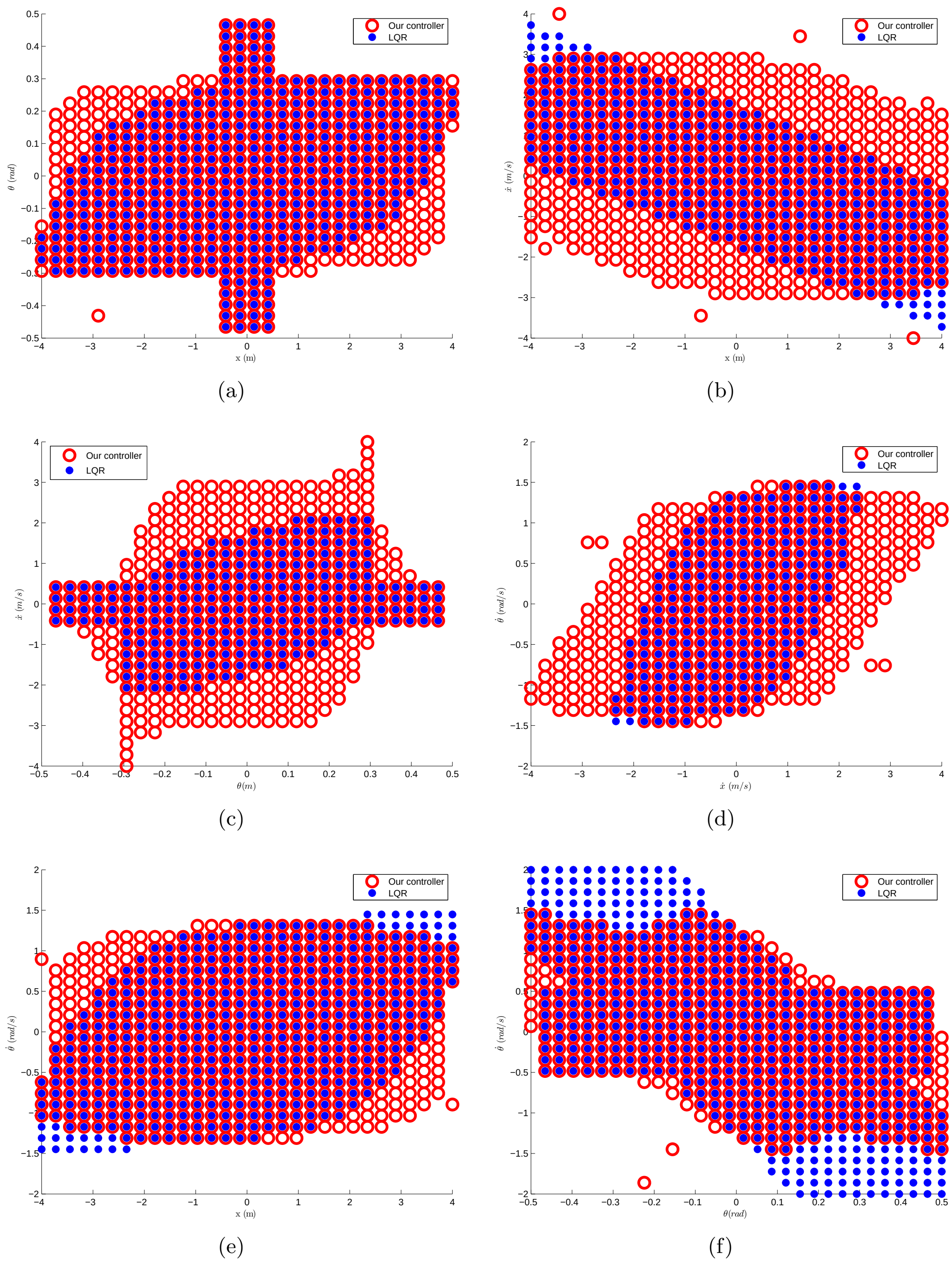


Figure 8: A comparison of the performance of the infinite–horizon LQR controller with the controller designed using our approach for the cart–pole system. Initial conditions are sampled from two–dimensional slices of the state space. Circles (dots) indicate points that pass near the target set when our controller (LQR controller) is applied. The plots illustrate that we are able to perform better than LQR on most portions of the sampled state space.

from slices of the state space and indicate which samples successfully get close to the target. The plots indicate that we are able to strictly improve the performance of the LQR controller using this approach except at two points, with significant improvements on certain slices.

6.5.3 Comparison to Lyapunov Based Approach

Next, we compare the performance of our controller generated on the unmodified dynamics (i.e., without the LQR feedback term) with a controller generated via Lyapunov's criteria for stability using SOS programming. We use the approach presented in [Majumdar et al., 2013, Section II] to design a controller that has *cubic* dependence on the state variables. This method employs an iterative algorithm and requires an initial guess for the controller, for which we use the LQR controller described in Section 6.5.1. As a side–note, we tried to design a linear controller using the Lyapunov function based approach but saw no improvement in the performance of the generated controller when compared to the LQR controller with which we initialized the algorithm.

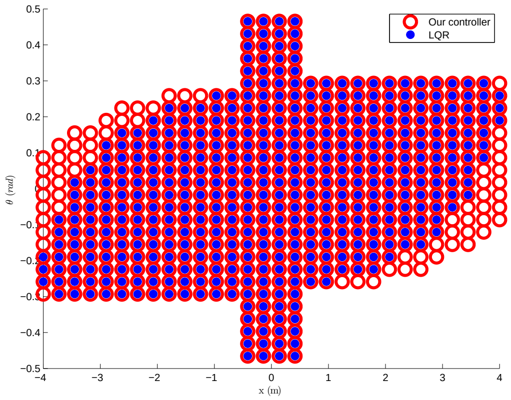
Figure 10 compares the performance of the two controllers. As in the earlier figures, we sample initial conditions from slices of the state space and indicate which points reach close to the target set. The approach presented in this paper demonstrates superior performance on the $\theta - \dot{x}$ and $\dot{x} - \dot{\theta}$ slices, while the performance of the controller from [Majumdar et al., 2013] is better on the $\dot{x} - x$ and $\theta - \dot{\theta}$ slices and only slightly more favorable on initial conditions drawn from the $x - \theta$ and $x - \dot{\theta}$ slices.

It is important to note that the method presented in this paper has several important advantages when compared to the approach presented in [Majumdar et al., 2013]. As mentioned in Section 1, the optimization problem that results from the formulation in [Majumdar et al., 2013] (and from other similar formulations based on Lyapunov conditions, see e.g. [Jarvis-Wloszek et al., 2005, Jarvis-Wloszek et al., 2003]) is non-convex and is solved by iteratively optimizing sets of decision variables. These iterations require a *feasible* initialization and involve solving three SOS programs at each iteration. In particular, the iterations for the example considered in this section were initialized with the LQR controller from Section 6.5.1 and required 24 iterations to converge (72 SOS programs in total). Further, the iterations are not guaranteed to converge to a global minimum (or necessarily even to a local minima) and can give different results based on the choice of initialization. The method presented in this paper in comparison does not require a feasible initialization and involves solving a single convex SOS program (albeit potentially a slightly more expensive one) and thus does not suffer from local minima. Moreover, our approach is amenable to theoretical analysis in the form of the convergence results presented in Section 3, which are currently unavailable for the method described in [Majumdar et al., 2013].

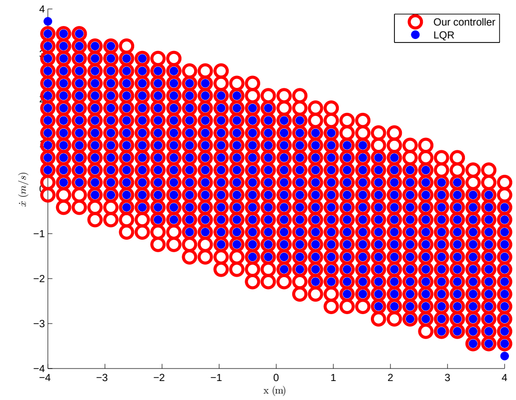
Finally, our method also scales well with the number of control inputs (m). In particular, the number of SOS constraints in the dual programs D_k grows linearly with m (for fixed k). In contrast, the number of SOS constraints for input constrained systems scales as $O(3^m)$ for the approach in [Majumdar et al., 2013]. This can lead to expensive optimization problems even for moderately sized input spaces.

6.6 uBot Mobile Manipulator

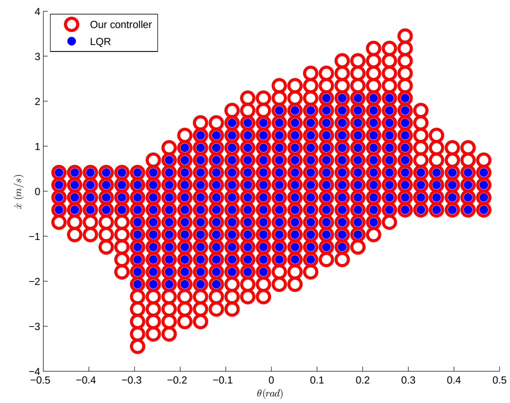
In our final example, we demonstrate the scalability of our algorithm by considering the eight state, three input system depicted in Figure 11(a). The model for the system, which we do not include here due to its complexity, is based on the uBot robot [Deegan et al., 2006]. We consider only the



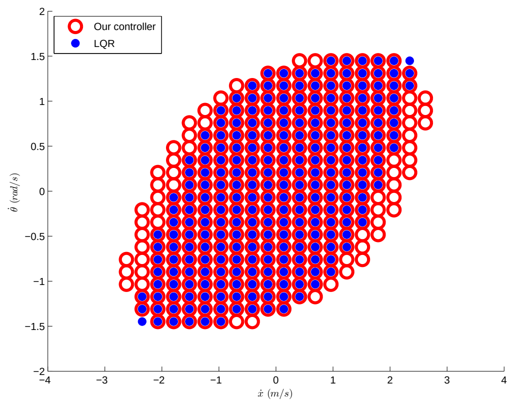
(a)



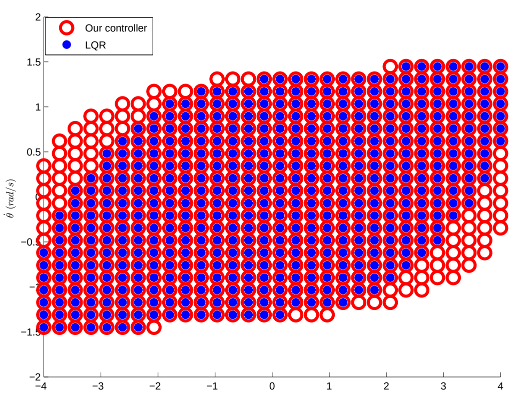
(b)



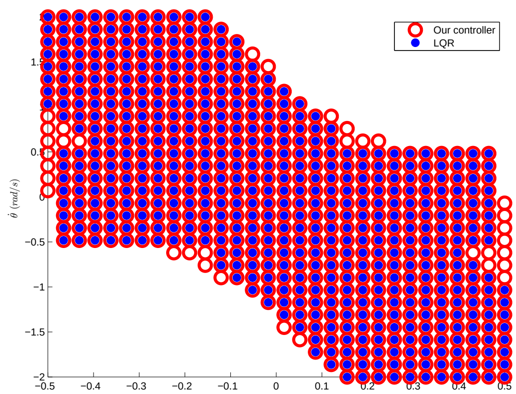
(c)



(d)



(e)



(f)

Figure 9: A comparison of the performance of the infinite–horizon LQR controller with the controller designed using our approach on the modified system dynamics described in Equation (54) for the cart–pole system. Initial conditions are sampled from two–dimensional slices of the state space. Circles (dots) indicate points that pass near the target set when our controller (LQR) is applied. We are able to strictly improve the performance of the LQR controller except at two points, with substantial improvements on certain slices.

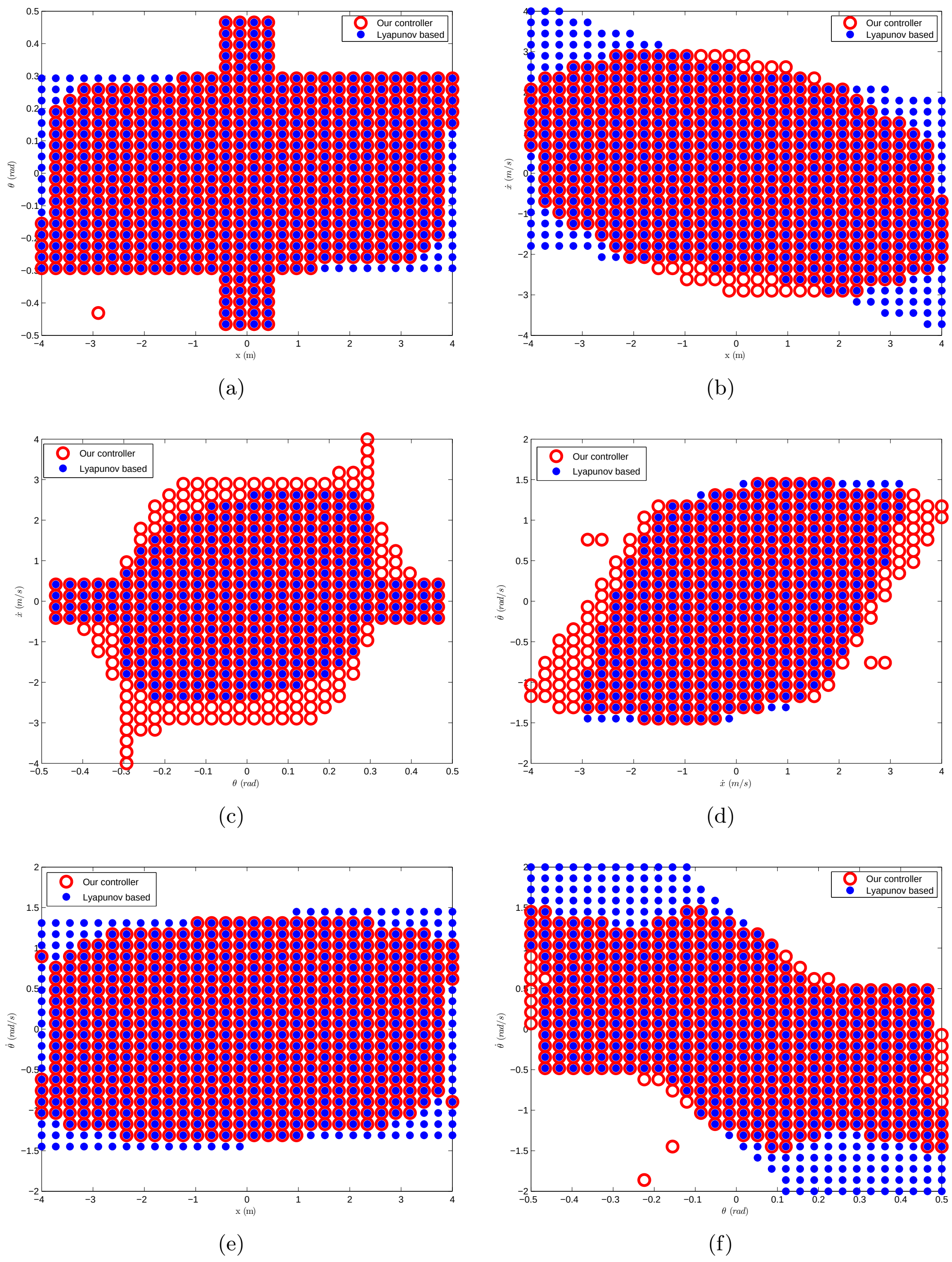
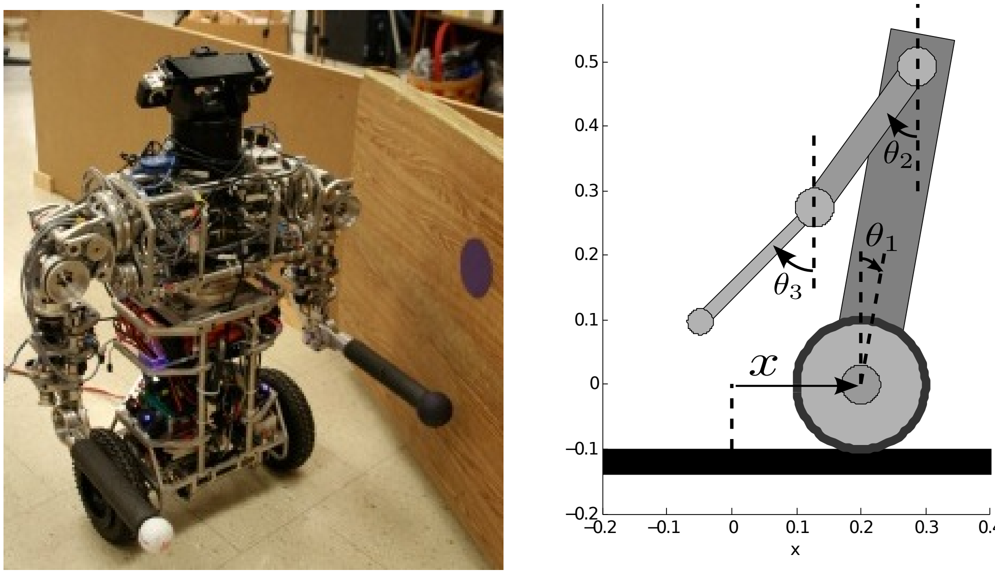


Figure 10: A comparison of the performance of the controller constructed using Lyapunov's criteria as described in [Majumdar et al., 2013] with the controller designed using our approach on the original system dynamics for the cart–pole system. Initial conditions are sampled from two–dimensional slices of the state space. Circles (dots) indicate points that pass near the target set when our controller (Lyapunov Based controller) is applied.



(a) An illustration of the uBot-5 hardware platform, developed at the University of Massachusetts, Amherst. Reproduced with permission from [Konidaris et al., 2011].

(b) An illustration of the planarized uBot model showing the configuration variables.

Figure 11: Depictions of the uBot system considered in Section 6.6.

dynamics in the sagittal plane by coupling the motion of the arms. An illustration of the planarized model and its associated configuration variables are shown in Figure 11(b). The state vector for the model includes the configuration variables and their derivatives.

The rotational inertia of the wheel is assumed to be negligible, and the system is modeled as a mobile cart with three links corresponding to the torso and two link arm. The masses of the cart, torso, and upper and lower arms are set to 0.6 kg, 12 kg, 2 kg, and 0.6 kg, respectively. The lengths of the torso link, upper arm link, and lower arm link are set to 0.5 m, 0.27 m and 0.27 m, respectively. The control inputs for the system are the force on the cart in the x-direction, denoted by F_x, and torques at the shoulder and elbow joints, denoted by τ_2 and τ_3, respectively. The θ_1 joint is not directly actuated. We impose the following input constraints on the system: $F_x \in [-1, 1]$ N and $\tau_i \in [-1, 1]$ N $\cdot$ m for $i = 2$ and 3.

We again solve the Free Final Time version of the control problem by setting $T = 7$ seconds and letting the target set to be the origin. The dynamics of the system are expanded using Taylor Series about the origin to degree three in order to obtain polynomial dynamics. We solve the SDP associated with $k = 2$, which takes 27 minutes with MOSEK. We were unable to run the SDP with SeDuMi for this example. Note that grid–based approaches such as those relying on Dynamic Programming or the Hamilton–Jacobi Bellman Equation are not able to address a problem of this dimensionality, which demonstrates the utility of our approach.

We apply the approach described in Section 6.5.2 to augment our dynamics with the LQR controller. The state and input cost matrices are taken to be the diagonal matrix $Q \in \mathbb{R}^{8\times 8}$ with $[Q]_{11} = 15$, $[Q]_{22} = 10$, $[Q]_{33} = 10$, $[Q]_{44} = 10$, $[Q]_{55} = 1$, $[Q]_{66} = 1$, $[Q]_{77} = 1$, and $[Q]_{88} = 1$ and diagonal matrix $R \in \mathbb{R}^{3\times 3}$ with $[R]_{11} = 2$, $[R]_{22} = 2$, and $[R]_{33} = 1$, respectively.

As before, we improve the performance of the LQR controller significantly. Figures 12(a) and 12(b) show initial conditions sampled from two slices of state space. Circles indicate initial conditions that get close to the target set (all coordinates less than 0.25 in absolute value) when our controller is applied and dots correspond to the application of the LQR controller. Figures 12(c) and 12(d) illustrate the performance of the controllers for two initial conditions for which our controller succeeds, but the LQR controller fails. The dashed lines in the figure represent the trajectory of the geometric center of the torso and the filled in circle represents the torso position at the goal state.

7 Discussion and Conclusion

We presented an approach for designing feedback controllers that maximize the size of the BRS by posing an infinite–dimensional LP over the space of non-negative measures. Finite–dimensional approximations to this LP in terms of SDPs can be used to obtain outer approximations of the largest achievable BRS and polynomial control laws that approximate the optimal control law. In contrast to previous approaches relying on Lyapunov's criteria for stability, our method is inherently convex and does not require feasible initialization.

7.1 Challenges

The most significant challenge confronting our approach at the moment is the scaling of our algorithm with the dimension of the system. The uBot example described in Section 6.6 with its eight states and three inputs is the largest example that existing solvers for SDPs are likely to manage without exploiting additional structure (e.g. symmetry) in the problem. However, recent developments, such as the release of MOSEK and SDPARA [Makoto et al., 2003], suggest that solvers that exploit sparsity or parallelization may allow our approach to scale to larger dimensions in the near future.

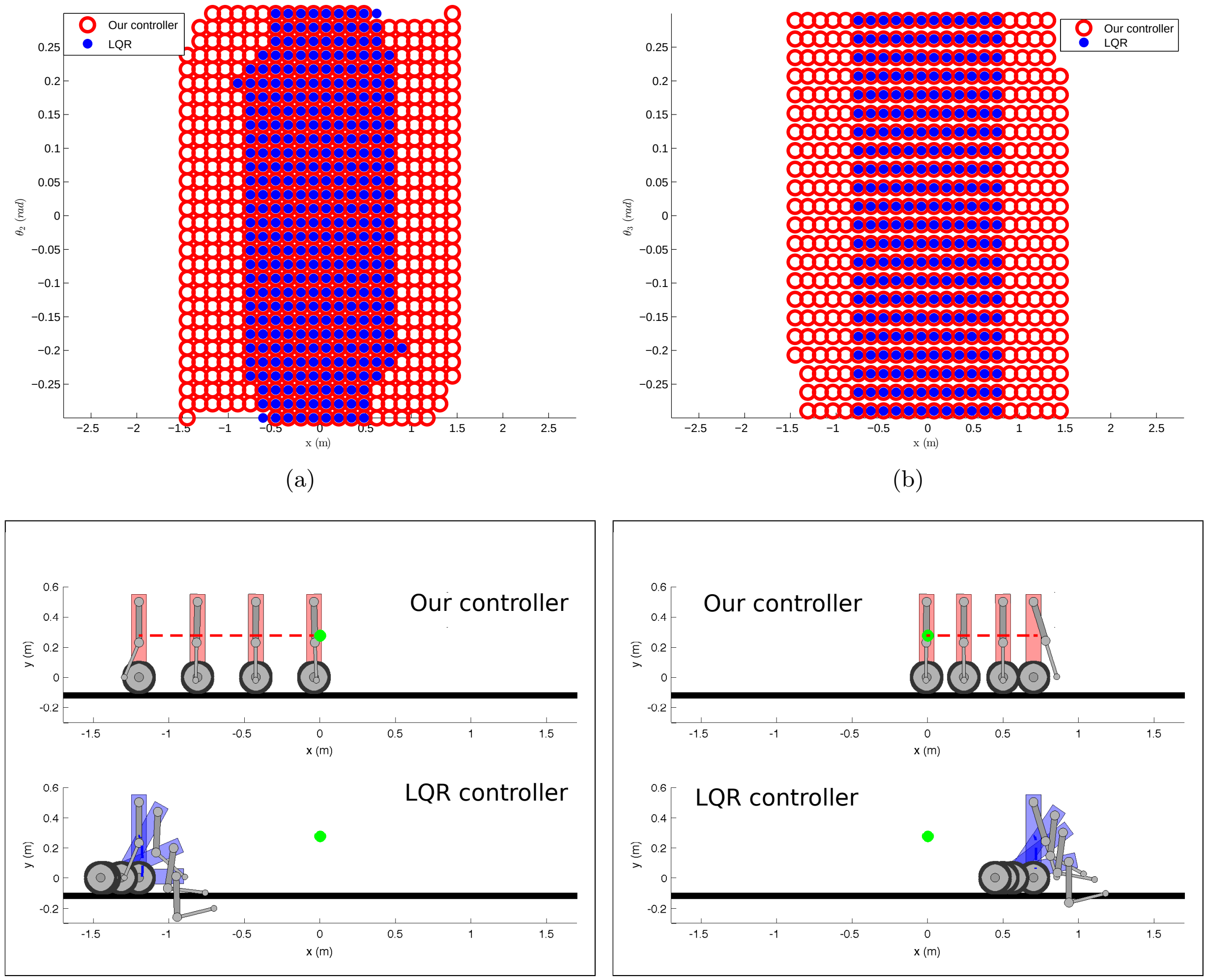
Additionally, the immaturity of existing SDP solvers means that numerical issues related to the scaling of problem data inevitably arise. Preprocessing of SDPs and SOS programs is still, in fact, an active area of research. We sometimes found it useful, for example, to scale the problem data to lie approximately within the unit ball.

7.2 Extensions

We are currently pursuing several extensions to our presented approach. Since we do not rely upon a linearization or a feasible initialization, our approach should be extendable to the case of hybrid dynamical systems. This could allow our approach to be applied to walking or running robots. Also, though we are constructing feedback control laws, during most robotic tasks, direct access to the state is unavailable. An extension that allows for output feedback would therefore be valuable.

We are also considering extensions that improve upon the presented convergence result. Inner approximations of the BRS are more useful than outer approximations during control tasks. The construction of control laws with associated inner approximations of the BRS is a problem we are currently pursuing. Ideally, a rate of convergence for the inner approximation and its associated feedback controller could be constructed.

As described in the Section 1, our method can be used to augment existing feedback motion planning techniques that rely on sequencing together BRSs in order to drive some desired set



(a) (b)

(c) (d)

Figure 12: A comparison of the performance of the infinite–horizon LQR controller with the controller designed using our approach on the modified system dynamics described in Equation (54) for the uBot system. Figures 12(a) and 12(b) show initial conditions sampled from two–dimensional slices of the state space. Circles (dots) represent points that pass near the target set when our controller (LQR) is applied. We are able to strictly improve the performance of the LQR controller except at two points. Figures 12(c) and 12(d) depict the performance of our controller and the LQR controller from two sample initial conditions. The dashed lines in the figure represent the trajectory of the geometric center of the torso and the filled in circle represents the torso position at the goal state.

of initial conditions to a given target set. We are currently investigating whether our proposed approach reduces the number of distinct controllers required for practical robotic motion planning tasks.

Finally, another promising direction for future research is to account for uncertainty and disturbances in the dynamics. This uncertainty could be stochastic or adversarial in nature and is an important consideration for many practical applications.

Acknowledgements

The authors are grateful to Milan Korda and Didier Henrion for many helpful discussions and to Scott Kuindersma for providing the dynamics for the uBot example considered in this paper.

Funding

This work was supported by the Office of Naval Research [MURI grant number N00014-10-1-095]; the National Science Foundation [Award numbers IIS-0915148, 724454, IIS-1161909]; and the Siebel Foundation [Siebel Scholarship].

References

[Abdalmoaty et al., 2012] Abdalmoaty, M. R., Henrion, D., and Rodrigues, L. (2012). Measures and LMIs for optimal control of piecewise-affine systems. *arXiv preprint arXiv:1211.3318*.

[Anderson and Nash, 1987] Anderson, E. and Nash, P. (1987). *Linear Programming in Infinite-Dimensional Spaces: Theory and Applications*. Wiley.

[Arnold, 1989] Arnold, V. (1989). *Mathematical Methods of Classical Mechanics*, volume 60. Springer.

[Ash, 1972] Ash, R. B. (1972). *Real Analysis and Probability*, volume 239. Academic Press New York.

[Aubin and Frankowska, 2008] Aubin, J. and Frankowska, H. (2008). *Set-Valued Analysis*. Birkhauser Boston.

[Bemporad et al., 2002] Bemporad, A., Borrelli, F., and Morari, M. (2002). Model predictive control based on linear programming–the explicit solution. *IEEE Transactions on Automatic Control*, 47(12):1974–1985.

[Bertsekas, 2005a] Bertsekas, D. P. (2005a). *Dynamic Programming & Optimal Control*, volume I and II. Athena Scientific, 3rd edition.

[Bertsekas, 2005b] Bertsekas, D. P. (2005b). *Dynamic Programming & Optimal Control*, volume I. 3rd edition.

[Bhatia et al., 2008] Bhatia, A., Graziano, M., Karaman, S., Naldi, R., and Frazzoli, E. (2008). Dubins trajectory tracking using commercial off-the-shelf autopilots. In *AIAA Guidance, Navigation, and Control Conference*, Honolulu, Hawaii.

[Chen and Ozguner, 2007] Chen, Q. and Ozguner, U. (2007). Intelligent off-road navigation algorithms and strategies of team desert buckeyes in the DARPA Grand Challenge 05. In *The 2005 DARPA Grand Challenge*, pages 183–203. Springer.

[Claeys et al., 2012] Claeys, M., Arzelier, D., Henrion, D., and Lasserre, J.-B. (2012). Measures and LMI for impulsive optimal control with applications to space rendezvous problems. In *American Control Conference (ACC), 2012*, pages 161–166. IEEE.

[Deegan et al., 2006] Deegan, P., Thibodeau, B., and Grupen, R. (2006). Designing a self-stabilizing robot for dynamic mobile manipulation. In *Robotics: Science and Systems Workshop on Manipulation for Human Environments*, Philadelphia, Pennsylvania.

[DeVon and Bretl, 2007] DeVon, D. and Bretl, T. (2007). Kinematic and dynamic control of a wheeled mobile robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4065–4070. IEEE.

[Ding and Tomlin, 2010] Ding, J. and Tomlin, C. (2010). Robust reach-avoid controller synthesis for switched nonlinear systems. In *49th IEEE Conference on Decision and Control*, pages 6481–6486. IEEE.

[Dubins, 1957] Dubins, L. (1957). On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3):497–516.

[Folland, 1999] Folland, G. B. (1999). *Real Analysis: Modern Techniques and Their Applications*, volume 2. John Wiley & Sons, second edition.

[Gray et al., 2012] Gray, A., Gao, Y., Lin, T., Hedrick, K. J., Tseng, H. E., and Borrelli, F. (2012). Predictive control for agile semi-autonomous ground vehicles using motion primitives. In *American Control Conference (ACC), 2012*, pages 4239 –4244.

[Henrion et al., 2013] Henrion, D., Daafouz, J., and Claeys, M. (2013). Optimal switching control design for polynomial systems: an LMI approach. *arXiv preprint arXiv:1303.1988*.

[Henrion and Korda, 2012] Henrion, D. and Korda, M. (2012). Convex computation of the region of attraction of polynomial control systems. *arXiv preprint arXiv:1208.1751*.

[Henrion et al., 2009] Henrion, D., Lasserre, J.-B., and Lofberg, J. (2009). Gloptipoly 3: moments, optimization and semidefinite programming. *Optimization Methods & Software*, 24(4-5):761–779.

[Hirsch, 1976] Hirsch, M. W. (1976). *Differential Topology*, volume 33. Springer-Verlag New York.

[Jadbabaie et al., 2001] Jadbabaie, A., Yu, J., and Hauser, J. (2001). Unconstrained receding-horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 46:776–783.

[Jarvis-Wloszek et al., 2003] Jarvis-Wloszek, Z., Feeley, R., Tan, W., Sun, K., and Packard, A. (2003). Some controls applications of sum of squares programming. In *42nd IEEE Conference on Decision and Control*, volume 5, pages 4676 – 4681.

[Jarvis-Wloszek et al., 2005] Jarvis-Wloszek, Z., Feeley, R., Tan, W., Sun, K., and Packard, A. (2005). Control applications of sum of squares programming. In Henrion, D. and Garulli, A., editors, *Positive Polynomials in Control*, volume 312 of *Lecture Notes in Control and Information Sciences*, pages 3–22. Springer Berlin / Heidelberg.

[Karaman and Frazzoli, 2011] Karaman, S. and Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *Int. Journal of Robotics Research*, 30:846–894.

[Kobilarov, 2012] Kobilarov, M. (2012). Cross-entropy motion planning. *The International Journal of Robotics Research*, 31(7):855–871.

[Konidaris et al., 2011] Konidaris, G., Kuindersma, S., Grupen, R., and Barto, A. (2011). Autonomous skill acquisition on a mobile manipulator. In *AAAI*.

[Kwakernaak and Sivan, 1972] Kwakernaak, H. and Sivan, R. (1972). *Linear Optimal Control Systems*. John Wiley & Sons, Inc.

[Lasserre, 2010] Lasserre, J. B. (2010). *Moments, Positive Polynomials and Their Applications*, volume 1. World Scientific.

[Lasserre et al., 2008] Lasserre, J. B., Henrion, D., Prieur, C., and Trelat, E. (2008). Nonlinear optimal control via occupation measures and lmi-relaxations. *SIAM Journal on Control and Optimization*, 47(4):1643–1666.

[LaValle, 2006] LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press.

[Löfberg, 2004] Löfberg, J. (2004). YALMIP : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan.

[Majumdar et al., 2013] Majumdar, A., Ahmadi, A. A., and Tedrake, R. (2013). Control design along trajectories with sums of squares programming. In *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*.

[Majumdar and Tedrake, 2012] Majumdar, A. and Tedrake, R. (2012). Robust online motion planning with regions of finite time invariance. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics*.

[Makoto et al., 2003] Makoto, Y., Katsuki, F., and Masakazu, K. (2003). SDPARA: Semidefinite programming algorithm parallel version. *Parallel Computing*, 29(8):1053–1067.

[Mitchell et al., 2005] Mitchell, I., Bayen, A., and Tomlin, C. (2005). A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957.

[Munos and Moore, 2002] Munos, R. and Moore, A. (2002). Variable resolution discretization in optimal control. *Machine Learning*, 49(2/3):291–323.

[Pappas et al., 1995] Pappas, G. J., Lygeros, J., and Godbole, D. N. (1995). Stabilization and tracking of feedback linearizable systems under input constraints. In *Proceedings of the IEEE Conference on Decision and Control*, pages 596–601. Citeseer.

[Parrilo, 2003] Parrilo, P. (2003). Exploiting structure in sum of squares programs. In *42nd IEEE Conference on Decision and Control*, volume 5, pages 4664–4669.

[Parrilo, 2000] Parrilo, P. A. (2000). *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology.

[Prajna et al., 2004] Prajna, S., Parrilo, P., and Rantzer, A. (2004). Nonlinear control synthesis by convex optimization. *IEEE Transactions on Automatic Control*, 49(2):310–314.

[Rantzer, 2001] Rantzer, A. (2001). A dual to Lyapunov's stability theorem. *Systems & Control Letters*, 42(3):161 – 168.

[Sastry, 1999] Sastry, S. (1999). *Nonlinear Systems: Analysis, Stability, and Control*, volume 10. Springer.

[Spong, 1998] Spong, M. W. (1998). Underactuated mechanical systems. In *Control Problems in Robotics and Automation*, pages 135–150. Springer.

[Sturm, 1999] Sturm, J. F. (1999). Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11(1-4):625 – 653.

[Tedrake et al., 2010] Tedrake, R., Manchester, I. R., Tobenkin, M. M., and Roberts, J. W. (2010). LQR-Trees: Feedback motion planning via sums of squares verification. *International Journal of Robotics Research*, 29:1038–1052.

[Trnovská, 2005] Trnovská, M. (2005). Strong duality conditions in semidefinite programming. *Journal of Electrical Engineering*, 56:87–89.