Noname manuscript No.
(will be inserted by the editor)

Robust Direct Trajectory Optimization Using Approximate Invariant Funnels

Zachary Manchester · Scott Kuindersma

Received: date / Accepted: date

Abstract Many critical robotics applications require robustness to disturbances arising from unplanned forces, state uncertainty, and model errors. Motion planning algorithms that explicitly reason about robustness require a coupling of trajectory optimization and feedback design, where the system's closed-loop response to disturbances is optimized. Due to the often-heavy computational demands of solving such problems, the practical application of robust trajectory optimization in robotics has so far been limited. Motivated by recent work on sums-of-squares verification methods for nonlinear systems, we derive a scalable robust trajectory optimization algorithm that optimizes approximate invariant funnels along the trajectory while planning. For the case of ellipsoidal disturbance sets and LQR feedback controllers, the state and input deviations along a nominal trajectory can be computed locally in closed form, permitting fast evaluation of robust cost and constraint functions and their derivatives. The resulting algorithm is a scalable extension of classical direct transcription that demonstrably improves tracking performance over non-robust formulations while incurring only a modest increase in computational cost. We evaluate the algorithm in several simulated robot control tasks.

1 Introduction

Motion planning is an active research area that has yielded several successes in recent years, from sampling-based algorithms that scale to large state spaces [17, 19] to nonlinear optimization methods capable of handling complex dynamic constraints [41, 42, 2] and contacts [43, 39, 40]. Despite this, the world's most advanced robots still struggle to perform robustly when subjected to disturbances caused by unplanned forces, state estimation errors, and model inaccuracies. Algorithms that explicitly

Z. Manchester and S. Kuindersma
School of Engineering and Applied Sciences
Harvard University, Cambridge, MA 02138
Email: zacmanchester@stanford.edu, scottk@seas.harvard.edu

reason about robustness require a coupling of motion planning and feedback design, frequently resulting in computationally expensive algorithms that have limited practical utility in robotics.

Verification methods have been successfully used to compute regions of finite-time invariance, or "funnels," around trajectories, using sums-of-squares (SOS) optimization [16, 44, 45]. In addition to providing performance certificates in the presence of disturbances [31], these methods can be used to build libraries of verified motions to construct reactive policies that switch between and modify funnels based on sensor feedback [26]. However, while these methods have been successfully used for trajectory *analysis*, their naive application to trajectory *optimization* leads to expensive nonlinear optimization problems that do not scale well to complex robotic systems with many degrees of freedom.

This paper builds on previous work on robust verification and trajectory optimization to derive a scalable robust trajectory optimization algorithm that reasons about disturbances by minimizing an analytical and differentiable cost function. The algorithm is derived using the observation that, under time-varying linear feedback (e.g., LQR) and ellipsoidal disturbance sets, bounds on state and input deviations can be computed locally along a trajectory in closed form. As a result, a tractable penalty function over the set of all disturbances can be defined and constraints on the perturbed states and inputs can be enforced.

We incorporate this robust penalty function into a direct transcription method (DIRTRAN), thereby inheriting many of the known benefits of these algorithms (e.g., easy handling of constraints and good numerical conditioning). The resulting algorithm, called DIRTREL (DIRect TRanscription with Ellipsoidal disturbances and Linear feedback), can be applied to the same class of systems as standard direct transcription, while significantly improving tracking performance in the presence of disturbances. In addition to demonstrating improved robustness on several simulated robots, we elucidate the theoretical connection between DIRTREL and robust verification methods based on sum-of-squares (SOS) optimization.

This paper is organized as follows: In Section 2 we summarize prior work related to robust control and verification in robotics. We then review the direct transcription algorithm and sum-of-squares programming in Section 3. The proposed robust direct transcription algorithm is presented in Section 4. We then show that DIRTREL can be viewed as an approximation of recently developed methods that solve SOS programs to find invariant funnels in Section 5. Section 6 describes several simulation experiments used to validate our new algorithm and analyze its computational performance. Finally, we discuss conclusions and future work in Section 7.

2 Related Work

There is a rich literature on robust control of linear dynamical systems that has developed over the past four decades [50]. In particular, a set of techniques collectively known as H_∞ control allows designers to optimize performance in the presence of disturbances by minimizing the L_2 gain of the closed-loop system. H_∞ techniques have also been extended to nonlinear systems [22, 24]. In robotics, these methods

have been used to improve trajectory tracking [49] and for optimization of limit cycle behaviors in simple walking models [4], but scaling these methods to general robust trajectory design has proved computationally challenging.

Several robust variants of differential dynamic programming [13] have been proposed for solving worst-case minimax problems [33], risk-sensitive optimizations for stochastic systems [8], and cooperative stochastic games [35]. Like the algorithm proposed in this paper, these methods consider system responses under linear feedback, but they use different robustness metrics and do not attempt to explicitly approximate invariant regions along the trajectory. State and input constraints require additional care, but could be handled approximately using penalty functions or exactly using augmented Lagrangian or quadratic programming approaches [21, 37]. The algorithm presented in this paper builds on direct formulations of trajectory optimization, so it naturally handles nonlinear constraints on the nominal and disturbed state and input trajectories.

Tube-based model-predictive control (MPC) algorithms [28] use online optimization to drive the system back to a central path in the presence of additive disturbances. However, for many systems, solving MPC problems in realtime remains challenging in practice. Kothare et al. [18] developed a robust MPC algorithm for linear systems based on online solution of semi-definite programs. Desaraju et al. [6] use online adaptation of linear models and semi-explicit optimization techniques to ease computational difficulties. Mordatch et al. [32] developed an ensemble trajectory optimization method that aims to reduce the expected cost under uncertain model parameters while also minimizing sample trajectory variance under fixed PD control. Ensemble methods have also been used in the context of linear-Gaussian control and sampling-based planners for identifying plans with low collision probabilities [46].

Connections between contraction analysis and motion planning have been explored using divergence metrics. The divergence costs described in [14] are similar in spirit to the robust cost function we propose, though they are based on sampling rather than analytical bounds. Lou and Hauser [23] combined robust motion planning with model estimation to optimize robust motions involving contact changes. However, their approach requires a kinematic plan to be given and only optimizes the timing of the motion.

Several authors have developed risk-sensitive optimal control methods [12, 48] for nonlinear stochastic systems with known models [8] or data-driven control learning approaches [47, 5, 20]. There has also been work on belief-space motion planning [38] in which uncertainty and information gathering were integrated as an explicit part of the planning objective.

Prior work has also augmented direct trajectory optimization methods with cost functionals that weight the tracking performance of linear feedback controllers [3, 10]. However, there are several important differences from our approach. Griffin and Grizzle [10], a fixed-gain proportional-integral controller is assumed, potentially creating limitations on closed-loop performance. Dai and Tedrake [3], the elements of the time-varying cost-to-go matrix associated with an LQR tracking controller are added as decision variables to the optimization problem, and disturbances are handled through a sampling scheme that scales poorly with the dimensionality of the disturbance vector. These design choices substantially increase the size and complexity of

the nonlinear program that must be solved, which limits the algorithm's applicability to higher-dimensional robot planning problems.

Verification approaches to feedback motion planning attempt to compute regions of finite-time invariance, or "funnels," [16, 45]. Moore et al. extended the LQR-Trees framework to include uncertainty in funnel estimates using a Common Lyapunov formulation of a sums-of-squares (SOS) program [31]. A related line of work led to the development of robust adaptive tracking controllers with guaranteed finite-time performance [30]. Majumdar and Tedrake extended these ideas to support robust online planning using pre-planned funnel libraries to construct a policy that can adjust funnels and tracking controllers based on sensor feedback [26]. The algorithm we describe is strongly connected to SOS verification approaches, but it aims to *design* robust trajectories, not just verify or switch between pre-computed trajectories. Naive integration of SOS constraints into trajectory optimization would result in intractably hard problems for most robots of interest. The key contribution of the proposed framework is that it reformulates this robust trajectory design as a direct transcription problem with additional cost and constraint functions that can be computed in closed form.

3 Background

The algorithm we propose builds upon direct trajectory optimization methods, with strong theoretical connections to modern verification algorithms for nonlinear dynamical systems. Below we provide a brief overview of these ideas to lay the groundwork for our subsequent development.

3.1 Direct Transcription

Direct transcription (DIRTRAN) methods solve optimal control problems by explicitly parameterizing the state and input trajectories and formulating a large, sparse nonlinear program [1]. Compared to shooting methods, these algorithms enable straightforward inclusion of state constraints and avoid numerical pitfalls such as the "tail wagging the dog" effect, at the expense of a larger problem size. The resulting nonlinear optimization problems can be solved using commercial sequential-quadratic programming (SQP) packages, such as SNOPT [9], that exploit the sparsity patterns in the linearized constraint matrix.

Given a nonlinear dynamical system, $\dot{x} = f(x, u)$, we discretize the system's state and input trajectories in time using N knot points, $x_{1:N} = \{x_1, \ldots, x_N\}$ and

$u_{1:N-1} = \{u_1, \ldots, u_{N-1}\}$, and solve the following NLP,

$$\begin{aligned} \underset{x_{1:N},\, u_{1:N-1},\, h}{\text{minimize}} \quad & g_N(x_N) + \sum_{i=1}^{N-1} g(x_i, u_i) \\ \text{subject to} \quad & x_{i+1} = x_i + f(x_i, u_i) \cdot h \qquad \forall i = 1 : N-1 \\ & u_i \in \mathcal{U} \qquad\qquad\qquad\quad \forall i = 1 : N-1 \\ & x_i \in \mathcal{X} \qquad\qquad\qquad\quad\;\; \forall i = 1 : N \\ & h_{\min} \leq h \leq h_{\max} \end{aligned} \tag{1}$$

where $g(\cdot, \cdot)$ and $g_N(\cdot)$ are cost functions, $\mathcal{X}$ is a set of feasible states, $\mathcal{U}$ is a set of feasible inputs, and h is the time step used for integration. For simplicity, we have assumed a forward Euler integration scheme, although other schemes such as backward Euler or midpoint interpolation can be used instead. By including h as a decision variable, we allow the solver to scale the duration of the trajectory. In what follows, we write the discrete-time dynamics as an iterated map, $x_{i+1} = f_h(x_i, u_i)$, for conciseness.

3.2 Invariant Funnels

The output of DIRTRAN is a feasible trajectory that locally minimizes cost. Given this trajectory, it is standard practice to design a local feedback controller to track the planned motion using techniques such as LQR. It is then natural to ask, "what stability guarantees can be made about the resulting closed-loop system?" To answer this question, it is often useful to compute an invariant funnel (also referred to as a "region of finite-time invariance"). Funnels are tube-like regions around a nominal trajectory within which all closed-loop trajectories of a system are guaranteed to remain if they begin inside (Figure 1) [16, 45]. This idea can be naturally extended to *robust* funnels that guarantee invariance in the presence of bounded disturbances [25].

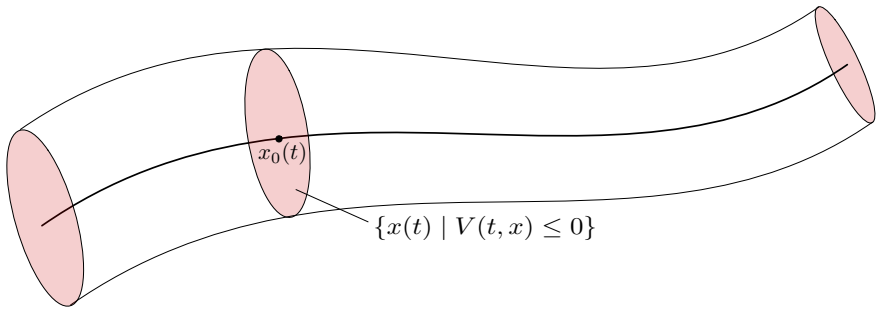

Fig. 1 Conceptual depiction of an invariant funnel around a nominal trajectory.

An invariant funnel can be represented mathematically as the sublevel set of a scalar function,

$$\{x(t) \mid V(t,x) \leq 0\}. \tag{2}$$

To guarantee that trajectories remain inside the funnel, the condition

$$\dot{V}(t,x,w) = \frac{\partial V}{\partial t} + \nabla V \cdot f_{cl}(x,w) \leq 0, \tag{3}$$

where $\dot{x} = f_{cl}(x,w)$ is the closed-loop system dynamics, must be satisfied on the boundary of the funnel (where $V(t,x) = 0$) for all disturbances $w \in \mathcal{W}$. We will also assume the disturbance set $\mathcal{W}$ can be represented as a sublevel set of a scalar function:

$$\mathcal{W} = \{w \mid S(w) \leq 0\}. \tag{4}$$

Combining these ideas, a sufficient condition for a robust funnel is

$$\begin{aligned} -\dot{V}(t,x,w) + L(t,x)V(t,x) + N(t,w)S(w) \geq 0, \\ N(t,w) \geq 0, \end{aligned} \tag{5}$$

where $L(t,x)$ and $N(t,w)$, known as multiplier functions, play a role similar to Lagrange multipliers. The intuition behind (5) is that V and S are negative inside the funnel, making the condition *more difficult* to satisfy. Outside the funnel, where $-\dot{V}$ may be negative, V and S are positive, allowing the first inequality in (5) to be satisfied. $N(w)$ must be strictly positive to guarantee that $-\dot{V} \geq 0$ holds for all $w \in \mathcal{W}$, while $L(x)$ can be positive or negative since $\dot{V}$ only needs to be negative on the boundary of the funnel (when $V(t,x) = 0$), and not necessarily in the interior of the funnel.

3.3 Sum-of-Squares Programming

Finding a function $V(t,x)$ that satisfies (5) is difficult in general. However, optimization algorithms have recently emerged that can search for polynomial V functions [36, 45]. Assuming all functions are polynomials, (5) can be converted into a numerical optimization problem by defining a vector z containing all monomials up to a given degree. The functions V, $\dot{V}$, S, L, and N can then be represented by symmetric matrices. For example, the polynomial $a^2 + b^2 - 2ab + 2a - 2b + 1$ can be written as $z^T C z$, where

$$z = \begin{bmatrix} a \\ b \\ 1 \end{bmatrix}, \tag{6}$$

$$C = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix}. \tag{7}$$

If the matrix C is positive semidefinite, the corresponding polynomial can be written as a sum of squares (SOS), and therefore must be globally non-negative [36]. Rewriting (5) in matrix form, the problem of finding a robust invariant funnel can be posed as the following optimization problem,

$$\begin{aligned} \text{minimize} \quad & \text{Vol}(\{x(t) \mid V(t,x) \leq 0\}) \\ \text{subject to} \quad & C \succeq 0, \end{aligned} \tag{8}$$

which is a semidefinite program (SDP). Since our goal is robustness to disturbances, we minimize the volume of the funnel to find the tightest bound on the disturbed state trajectories around the nominal trajectory.

While SDPs like (8) can be solved reliably in theory, the number of decision variables grows combinatorially in the dimensions of both the state and disturbance vectors, as well as the order of the polynomials [36]. As a result, only very simple low-dimensional systems can be handled in practice. Our goal is to reduce this computational burden so that robust invariant funnels can be used not only for analysis of closed-loop trajectories, but also inside the inner loop of trajectory optimization algorithms for complex high-dimensional robotic systems.

4 Direct Transcription with Ellipsoidal Disturbances

The following subsections describe how we extend the standard DIRTRAN problem to incorporate linear feedback, bounded disturbances, and a cost function that penalizes closed-loop deviations from the nominal trajectory.

4.1 State and Input Deviations

First, we assume disturbances, $w_i \in \mathcal{W}$, can enter into the dynamics in a general nonlinear way:

$$x_{i+1} = f_h(x_i, u_i, w_i). \tag{9}$$

Under this definition, w_i could, for example, correspond to model parameter uncertainty, unplanned external forces, or state estimation errors. A trajectory, $x_{1:N}$, $u_{1:N-1}$, that satisfies

$$x_{i+1} = f_h(x_i, u_i, 0) \tag{10}$$

is referred to as a *nominal trajectory*. Given a disturbance sequence, $w_{1:N-1}$, the deviations from the nominal state trajectory are calculated as

$$\delta x_{i+1} = f_h(x_i + \delta x_i, u_i + \delta u_i, w_i) - x_{i+1}, \tag{11}$$

and we assume that deviations from the nominal input sequence are computed using a linear feedback controller,

$$\delta u_i = -K_i \delta x_i. \tag{12}$$

Any linear controller can be used, but in the development that follows we define K_i to be the optimal time-varying linear quadratic regulator (TVLQR) gain matrix computed by linearizing the dynamics along the nominal trajectory and solving the dynamic Riccati equation,

$$\begin{aligned} K_i &= (R + B_i^T P_{i+1} B_i)^{-1} (B_i^T P_{i+1} A_i) \\ P_i &= Q + K_i^T R K_i + (A_i - B_i K_i)^T P_{i+1} (A_i - B_i K_i), \end{aligned} \tag{13}$$

where $A_i = \partial f_h / \partial x|_{x_i, u_i, 0}$, $B_i = \partial f_h / \partial u|_{x_i, u_i, 0}$, $P_N \equiv Q_N$, and $Q, Q_N \succeq 0$ and $R \succ 0$ are state and input cost matrices.

4.2 Robust Cost Function

To optimize robustness, our approach augments the NLP (1) with an additional cost term, $\ell_{\mathcal{W}}(x_{1:N}, u_{1:N-1})$. Intuitively, we want this function to penalize deviations of the closed-loop system from the nominal trajectory in the presence of disturbances, w_i, drawn from the set $\mathcal{W}$. To penalize these deviations, we assume a quadratic one-step cost, $\delta x_i^T Q^\ell \delta x_i + \delta u_i^T R^\ell \delta u_i$, where $Q^\ell \succeq 0$ and $R^\ell \succeq 0$ are positive semidefinite cost matrices.

In order to compute $\delta x_{1:N}$ and $\delta u_{1:N-1}$, we need a well-defined disturbance sequence, $w_{1:N-1}$. Instead of resorting to sampling or worst-case minimax optimization methods, we instead approximate the robust cost averaged over the entire disturbance set and summed along the trajectory:

$$\ell_{\mathcal{W}}(x_{1:N}, u_{1:N-1}) \approx \frac{1}{\text{Vol}(\mathcal{W})} \int_{\mathcal{W}} \left(\delta x_N^T Q_N^\ell \delta x_N + \sum_{i=1}^{N-1} \left(\delta x_i^T Q^\ell \delta x_i + \delta u_i^T R^\ell \delta u_i \right) \right) d\mathcal{W}. \quad (14)$$

For general nonlinear systems and disturbance sets, the integral in equation (14) cannot be easily computed. However, the assumption of an ellipsoidal disturbance set and a linearization of the dynamics about the nominal trajectory leads to a computationally tractable approximation. While linearization of the dynamics may seem limiting, we argue that the resulting local approximation has roughly the same region of validity as the LQR tracking controller. It therefore does not impose significant practical limitations beyond those already associated with the use of linear feedback.

We parameterize the ellipsoidal set $\mathcal{W}$ by a symmetric positive-definite matrix D, such that

$$w^T D^{-1} w \leq 1. \quad (15)$$

Note that the set of vectors describing the semi-axes of $\mathcal{W}$ are given by the columns of the principal square root of D, defined such that $D = D^{1/2} D^{1/2}$, where $D^{1/2}$ is also symmetric and positive definite. Using the fact that ellipsoids map to ellipsoids under linear transformations, approximate ellipsoidal bounds on the state deviations, δx_i, can be computed at each time step i. As in equation (15), we parameterize these ellipsoids by matrices $E_i \succ 0$:

$$\delta x_i^T E_i^{-1} \delta x_i \leq 1. \quad (16)$$

Assuming a bound on the initial state deviation parameterized by E_1, the matrices E_i can be found at all future time steps using the system dynamics. Linearizing equation (11) about the nominal trajectory gives a set of linear time-varying equations of the form,

$$\delta x_{i+1} \approx A_i \delta x_i + B_i \delta u_i + G_i w \,, \quad (17)$$

where

$$G_i = \left. \frac{\partial f_h}{\partial w} \right|_{x_i, u_i, 0}. \quad (18)$$

Interpreting the columns of $E_i^{1/2}$ as instantiations of δx_1,

$$E_i^{1/2} = \begin{bmatrix} \delta x_i^1 & \delta x_i^2 & \cdots & \delta x_i^n \end{bmatrix}, \tag{19}$$

equation (17) can be applied to each column individually to derive a a recursion for E_{i+1} in terms of E_i and D:

$$\begin{aligned} E_{i+1} &= \begin{bmatrix} \delta x_{i+1}^1 & \cdots & \delta x_{i+1}^n \end{bmatrix} \begin{bmatrix} \cdots \end{bmatrix}^T \\ &= \begin{bmatrix} (A_i - B_i K_i)\delta x_i^1 + G_i w_i & \cdots & (A_i - B_i K_i)\delta x_i^n + G_i w_i \end{bmatrix} \begin{bmatrix} \cdots \end{bmatrix}^T \\ &= (A_i - B_i K_i) E_i (A_i - B_i K_i)^T + G_i D G_i^T \\ &\quad + (A_i - B_i K_i) H_i G_i^T + G_i H_i^T (A_i - B_i K_i)^T, \end{aligned} \tag{20}$$

where H_i has been defined to account for $\delta x_i w_i$ cross terms. Equation (20) can be written compactly as,

$$M_{i+1} = F_i M_i F_i^T, \tag{21}$$

where M_i and F_i are defined as follows,

$$M_i = \begin{bmatrix} E_i & H_i \\ H_i^T & D \end{bmatrix} \tag{22}$$

$$F_i = \begin{bmatrix} (A_i - B_i K_i) & G_i \\ 0 & I \end{bmatrix}, \tag{23}$$

and M_1 is initialized with $H_1 = 0$ and $E_1 \succ 0$.

For completeness, we also mention that a continuous-time version of (21) can be derived. Assuming continuous-time dynamics,

$$\delta \dot{x}(t) \approx A_c(t)\delta x + B_c(t)\delta u + G_c(t) w, \tag{24}$$

we have,

$$\dot{M}(t) = F_c(t) M + M F_c(t)^T, \tag{25}$$

where M is defined as before, but the continuous-time $F_c(t)$ is,

$$F_c(t) = \begin{bmatrix} (A_c(t) - B_c(t) K_c(t)) & G_c(t) \\ 0 & 0 \end{bmatrix}, \tag{26}$$

and $K_c(t)$ is obtained by solving the continuous-time differential Riccati equation in-lieu of (13).

The propagation of E_i forward in time through the linearized dynamics bears some resemblance to the covariance propagation step in a Kalman Filter. However, there are some important differences. First, as a matter of interpretation, equation (21) is purely deterministic; E_i represents a strict bound rather than a statistical covariance. Second, equations (21)–(23) contain additional cross terms, signified by the presence of the H_i blocks in the matrix M_i, which are absent in the Kalman Filter due to the assumed statistical independence of noise at different time steps (the "Markov property").

Returning to the cost function $\ell_{\mathcal{W}}(x_{1:N}, u_{1:N-1})$, we replace the volume integral over the disturbance set in equation (14) with a sample mean calculated over the columns of $E_i^{1/2}$, which correspond to the semi-axis vectors of the ellipsoids bounding the state,

$$\ell_{\mathcal{W}}(x_{1:N}, u_{1:N-1}) = \\ \frac{1}{n_x} \sum_{\delta x_N \in \text{col}(E_N^{1/2})} \delta x_N^T Q_N^\ell \delta x_N + \frac{1}{n_x} \sum_{i=1}^{N-1} \sum_{\delta x_i \in \text{col}(E_i^{1/2})} \delta x_i^T Q^\ell \delta x_i + \delta u_i^T R^\ell \delta u_i \,, \tag{27}$$

where n_x is the state dimension. In the remainder of the paper we omit this constant factor, as it has no effect on the results and can be folded into the cost-weighting matrices.

The sum over the columns of $E_i^{1/2}$ can be computed by rewriting the quadratic forms in equation (14) using the trace operator,

$$\delta x_i^T Q \delta x_i = \text{Tr}(Q\, \delta x_i \delta x_i^T) \tag{28}$$

$$\delta u_i^T R \delta u_i = \text{Tr}(R\, \delta u_i \delta u_i^T), \tag{29}$$

and replacing the outer products $\delta x_i \delta x_i^T$ and $\delta u_i \delta u_i^T$ with suitable expressions involving E_i:

$$\ell_{\mathcal{W}}(x_{1:N}, u_{1:N-1}) = \text{Tr}\left(Q_N E_N\right) + \sum_{i=1}^{N-1} \text{Tr}\left((Q + K_i^T R K_i) E_i\right). \tag{30}$$

Equations (21)–(23) and (30) provide an easily computable and differentiable cost function that quantifies the system's closed-loop sensitivity to disturbances. The evaluation of $\ell_{\mathcal{W}}$ is summarized in Algorithm 1.

4.3 The DIRTREL Algorithm

We now develop a complete algorithm that outputs a feasible trajectory and feedback controller for the nominal ($w = 0$) system such that the sensitivity of the closed-loop system to disturbances is minimized. In addition to augmenting (1) with $\ell_{\mathcal{W}}(x_{1:N}, u_{1:N-1})$, we must also ensure that the closed-loop system obeys state and input constraints. To do so, we again use the columns of $E_i^{1/2}$, which give extreme values of δx_i on the boundary of the ellipsoid.

In particular, all state constraints on the nominal trajectory are also applied to the set of disturbed state vectors,

$$x_i^{\mathcal{W}} = x_i \pm \text{col}\left(E_i^{1/2}\right), \tag{31}$$

and all input constraints are also applied to the set of disturbed closed-loop inputs,

$$u_i^{\mathcal{W}} = u_i \pm \text{col}\left((K_i E_i K_i^T)^{1/2}\right). \tag{32}$$

Algorithm 1 Robust Cost Function

function $\ell_{\mathcal{W}}(x_{1:N}, u_{1:N-1}, D, E_1, Q^\ell, R^\ell, Q_N^\ell, Q, R)$
  for $i = 1 \dots N-1$ **do**
    $A_i \leftarrow \partial_{x=x_i} f_h(x,u,w)$
    $B_i \leftarrow \partial_{u=u_i} f_h(x,u,w)$
    $G_i \leftarrow \partial_{w=0} f_h(x,u,w)$
  end for
  $K_{1:N-1} \leftarrow TVLQR(A_{1:N-1}, B_{1:N-1}, Q, R)$
  $H_1 \leftarrow 0$
  for $i = 1 \dots N-1$ **do**
    $\ell \leftarrow \ell + \mathrm{Tr}\left((Q^\ell + K_i^T R^\ell K_i) E_i\right)$
    $E_{i+1} \leftarrow (A_i - B_i K_i) E_i (A_i - B_i K_i)^T$
      $+ (A_i - B_i K_i) H_i G_i^T$
      $+ G_i H^T (A_i - B_i K_i)^T$
      $+ G_i D G_i^T$
    $H_{i+1} = (A_i - B_i K_i) H_i + G_i D$
  end for
  $\ell \leftarrow \ell + \mathrm{Tr}\left(Q_N^\ell E_N\right)$
  return ℓ
end function

While not strictly equivalent to checking the entire boundary of the ellipsoid, enforcing constraints on the set of vectors $x_i^{\mathcal{W}}$ and $u_i^{\mathcal{W}}$ is much simpler computationally. The resulting optimization problem, referred to as DIRTREL, can be expressed as the following NLP:

$$\begin{aligned}
\underset{x_{1:N},\, u_{1:N-1},\, h}{\text{minimize}} \quad & \ell_{\mathcal{W}}(x_{1:N}, u_{1:N-1}) + g_N(x_N) + \sum_{i=1}^{N-1} g(x_i, u_i) \\
\text{subject to} \quad & x_{i+1} = f_h(x_i, u_i) && \forall i = 1:N-1 \\
& u_i \in \mathcal{U} && \forall i = 1:N-1 \\
& u_i^{\mathcal{W}} \in \mathcal{U} && \forall i = 1:N-1 \\
& x_i \in \mathcal{X} && \forall i = 1:N \\
& x_i^{\mathcal{W}} \in \mathcal{X} && \forall i = 1:N \\
& h_{\min} \le h \le h_{\max}
\end{aligned} \tag{33}$$

Unlike minimax approaches to robust control, $\ell_{\mathcal{W}}(x_{1:N}, u_{1:N-1})$ and the associated robust state and input constraints in (33) are smooth functions of the state and input trajectories. As a result, they can be differentiated in closed form and good convergence behavior can be achieved with standard NLP solvers based on Newton's method [34].

Since DIRTREL builds on the classic DIRTRAN algorithm, it inherits some of that algorithm's favorable computational properties [1]. Like DIRTRAN, the Jacobian matrices of the dynamics, state, and input constraints in (33) remain banded and sparse. The number of decision variables also remains the same as standard DIRTRAN. Unlike DIRTRAN, however, the Jacobians of the disturbed state and input constraints, as well as the Hessian of the robust cost function, are dense. It is possible to build a version of DIRTREL with sparse Jacobians and Hessians by making

the K_i and M_i matrices decision variables and adding corresponding equality constraints to enforce their dynamics. However, this would lead to a dramatic increase in the problem size. In practice, we solve (33) with a quasi-Newton method that builds a low-rank approximation of the Hessian from gradient vectors [9], mitigating some of the computational burden associated with these dense matrices.

A key computational difference between DIRTREL and the classic DIRTRAN algorithm lies in the differentiation of the system dynamics (9). To evaluate constraint Jacobians, implementations of DIRTRAN typically require first derivatives of the dynamics. In contrast, DIRTREL requires second derivatives of the dynamics to evaluate the Jacobians of the disturbed state and input constraints in (33) and to compute the gradient of the robust cost function described in Algorithm 1.

5 Connections to Sum-of-Squares Methods

At a high level, the ellipsoidal bounds around state and input trajectories computed by DIRTREL appear to have some similarities to the invariant funnels computed by SOS methods [45, 26]. This section explores these connections at a deeper level and shows that, in fact, these bounds are computationally tractable approximations of invariant funnels.

We now explicitly instantiate the polynomial equations defining a robust invariant funnel presented in Section 3.2. The ellipsoidal bounds of equations (15) and (16) give the following polynomial inequalities for the disturbance and state sets:

$$S(w) = w^T D^{-1} w - 1 \leq 0 \tag{34}$$

$$V(t, \delta x) = \delta x^T E(t)^{-1} \delta x - 1 \leq 0. \tag{35}$$

Using the linearized dynamics of equation (24), $\dot{V}$ can also be written explicitly as follows:

$$\begin{aligned} \dot{V}(t, \delta x, w) = \delta x^T E^{-1} G_c w + w^T G_c^T E^{-1} \delta x + \\ \delta x^T \left(\dot{E}^{-1} + E^{-1}(A_c - B_c K_c) + (A_c - B_c K_c)^T E^{-1} \right) \delta x. \end{aligned} \tag{36}$$

Putting (34)–(36) together, we can expand the inequality (5). We note that, since we are dealing only with quadratic functions, the result reduces to a standard linear matrix inequality (LMI), and the multiplier polynomials $L(\delta x)$ and $N(w)$ reduce to scalars ℓ and n:

$$\begin{aligned} \delta x^T \left(-\dot{E}^{-1} - (A_c - B_c K_c)^T E^{-1} - E^{-1}(A_c - B_c K_c) \right) \delta x \\ - \delta x^T E^{-1} G_c w - w^T G_c^T E^{-1} \delta x \\ + \ell(\delta x^T E^{-1} \delta x - 1) + n(w^T D^{-1} w - 1) \geq 0. \end{aligned} \tag{37}$$

To simplify the SOS program (8), we assume that we are given an initial value $E(0)$ at $t = 0$. Our goal is to propagate $E(t)$ forward in time while ensuring that it is

a tight bound. Therefore, we solve the following optimization problem at each instant in time,

$$\begin{aligned} \underset{\dot{E},\ell,n}{\text{minimize}} \quad & \text{Tr}(\dot{E}) \\ \text{subject to} \quad & -\dot{V} + \ell V + nS \geq 0 \\ & n \geq 0, \end{aligned} \tag{38}$$

where the first inequality constraint is (37).

Using the trace operator and the identity,

$$\frac{d}{dt}\left(E^{-1}\right) = -E^{-1}\dot{E}E^{-1}, \tag{39}$$

we can rewrite $-\dot{V} + \ell V + nS \geq 0$ as follows:

$$\begin{aligned} \text{Tr}\Big[& E^{-1}\dot{E}E^{-1}\delta x\delta x^T - (A_c - B_cK_c)^T E^{-1}\delta x\delta x^T \\ & - \delta x\delta x^T E^{-1}(A_c - B_cK_c) - E^{-1}G_c w\delta x^T \\ & - \delta x w^T G_c^T E^{-1} + \ell E^{-1}\delta x\delta x^T + nD^{-1}ww^T - \ell - n\Big] \geq 0. \end{aligned} \tag{40}$$

Realizing that on the boundary of an ellipsoid,

$$x^T A^{-1} x = 1 \implies A = \sum_{k=1}^{n} x_k x_k^T, \tag{41}$$

for a set of n orthogonal vectors x_k corresponding to the semi-axes of the ellipsoid, we use the linearity of the expression (40) to replace each outer product with its corresponding matrix ($\delta x\delta x^T \to E$, $ww^T \to D$, and $\delta x w^T \to H$) and multiply through by E to arrive at:

$$\text{Tr}\left[\dot{E} - E(A_c - B_cK_c)^T - (A_c - B_cK_c)E - G_cH^T - HG_c^T\right] \geq 0. \tag{42}$$

The minimization of $\text{Tr}(\dot{E})$ in the objective of (38) causes (42) to hold with equality. Also, for a symmetric positive-semidefinite matrix, $\text{Tr}(A) = 0 \implies A = 0$, allowing us to recover the following matrix ODE:

$$\dot{E} = E(A_c - B_cK_c)^T + (A_c - B_cK_c)E + G_cH^T + HG_c^T. \tag{43}$$

Equation (43) is precisely the continuous-time propagation rule for the ellipsoidal state bounds derived in Section 4. Therefore, these bounds can be interpreted as approximate robust invariant funnels computed with linearized system dynamics. Compared to the original SOS formulation, the approximation used in DIRTREL scales much more favorably with the size of the system's state and input vectors, making it particularly amenable to use inside the "inner loop" of other algorithms like trajectory optimizers.

6 Examples

We now present several examples to demonstrate the performance of DIRTREL. Comparisons are made to the standard approach of performing trajectory optimization with DIRTRAN followed by synthesis of a time-varying LQR tracking controller. All algorithms are implemented in MATLAB, and the SQP solver SNOPT [9] is used to solve the resulting NLPs. DIRTREL's running time on all examples is between two and four times that of standard DIRTRAN. Empirically, the increased running time is primarily attributed to the calculation of derivatives of the robust cost function (currently done in MATLAB). We anticipate substantial gains could be achieved with a more careful C++ implementation.

6.1 Pendulum with Uncertain Mass

In the first test case, a simple pendulum of unit length with an input torque is considered. The mass of the pendulum is bounded between 0.8 and 1.2, and the actuator has torque limits $-3 \leq u \leq 3$. The goal is to swing the pendulum from its downward stable equilibrium at $\theta = 0$ to the upward unstable equilibrium at $\theta = \pi$ in minimum time.

The following cost-weighting matrices are used in both the robust cost function $\ell_{\mathcal{W}}$ and the LQR tracking controller:

$$\begin{aligned} R = R^\ell &= 0.1 \\ Q = Q^\ell &= \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix} \\ Q_N = Q_N^\ell &= \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}. \end{aligned}$$

The matrix D corresponding to the ± 0.2 bound on the pendulum mass is $D = (0.2)^2$, and the algorithm is initialized with no initial disturbance on the state, $E_1 = 0_{2\times 2}$.

Figure 2 plots the approximate robust invariant funnel computed by DIRTREL in blue. In addition, the results of 25 full nonlinear simulations performed with mass values uniformly sampled over the range 0.8–1.2 are plotted in red, indicating the shape of the "true" funnel. Good agreement can be seen between the red sample trajectories and the blue funnel, indicating that the approximations made in DIRTREL are reasonable.

The state and input trajectories produced by DIRTRAN and DIRTREL are shown in Figure 3. Consistent with the minimum-time nature of the problem, DIRTRAN generates a bang-bang control policy that uses the maximum torque that the actuator is capable of producing. DIRTREL, on the other hand, produces a nominal trajectory that stays clear of the torque limits. Over several numerical simulations, the DIRTREL controller was able to perform successful swing-ups (in all trials) of pendulums with mass values up to $m \approx 1.3$, while the DIRTRAN controller was successful only up to $m \approx 1.1$. While tuning the cost function used in DIRTRAN through

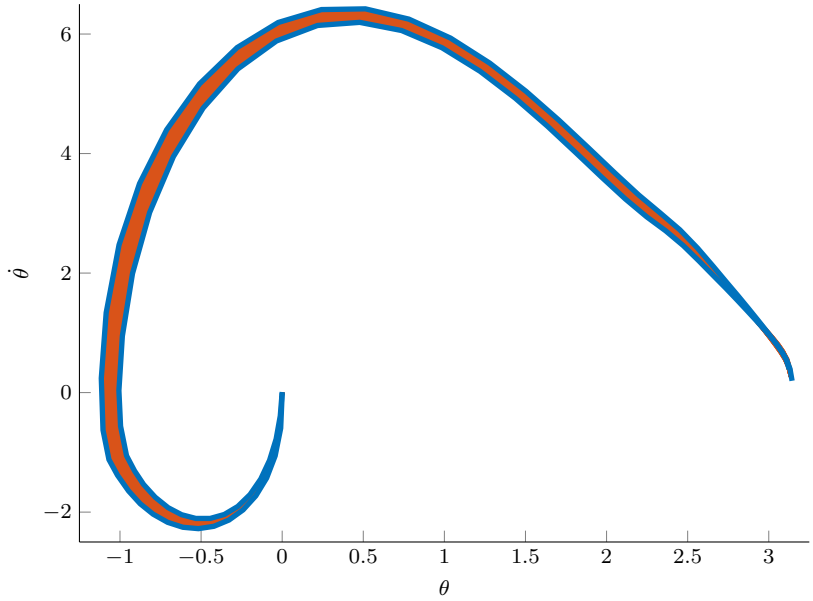

Fig. 2 Approximate robust invariant funnel computed by DIRTREL (blue) and 25 trajectories computed with uniformly sampled disturbances (red).

trial and error could likely result in more robust tracking performance, DIRTREL allows known bounds on the plant model to be incorporated in a principled and straightforward manner that eliminates the need for such "hand tuning."

6.2 Cart Pole with Unmodeled Friction

Motivated by the fact that friction is often difficult to model accurately in mechanical systems, we considered a swing-up problem for the cart pole system (Figure 4) with unmodeled friction. The system's state vector $x \in \mathbb{R}^4$ consists of the cart's position, the pendulum angle, and their corresponding first derivatives. The input $u \in \mathbb{R}$ consists of a force applied to the cart. Nonlinear Coulomb friction is applied between the cart and the ground. Once again, the goal is to swing the pendulum from the downward $\theta = 0$ position to the upward $\theta = \pi$ position.

In our simulations, the cart mass is taken to be $m_c = 1$, the pendulum mass is $m_p = 0.2$, and the pendulum length is $l = 0.5$. The cart's actuator is limited to $-10 \leq u \leq 10$. The nominal model used during trajectory optimization has no friction, while in simulation the following friction force is applied to the cart,

$$F_c = -\operatorname{sign}(\dot{x})\mu F_N, \tag{44}$$

where μ is a friction coefficient and F_N is the normal force exerted between the cart and the ground.

To account for unmodeled friction in DIRTREL, we make w an exogenous force input to the cart and bound it such that $-2 \leq w \leq 2$, corresponding to $D = 4$. We

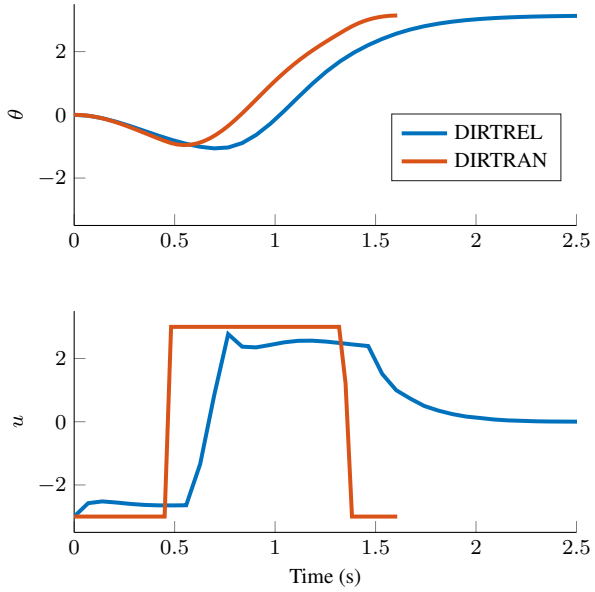

Fig. 3 Pendulum swing-up state and input trajectories

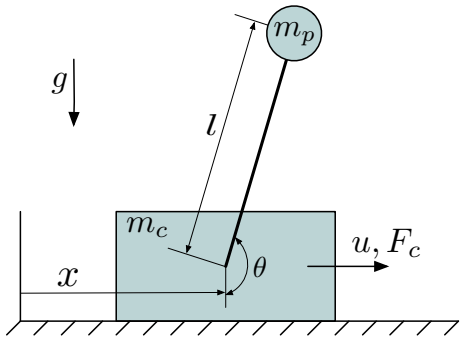

Fig. 4 The Cart Pole system with unmodeled friction with the ground.

use the following running cost in both DIRTRAN and DIRTREL:

$$g(x_i, u_i) = x_i^T x_i + 0.1 u_i^2 \,,$$

and a terminal constraint is enforced on the final state, $x_N = [0\ \pi\ 0\ 0]^T$. The following cost-weighting matrices are used in both the robust cost function $\ell_{\mathcal{W}}$ and the LQR tracking controller:

$$\begin{aligned} R = R^\ell &= 1 \\ Q = Q^\ell &= \begin{bmatrix} 10 I_{2\times 2} & 0 \\ 0 & I_{2\times 2} \end{bmatrix} \\ Q_N = Q_N^\ell &= 100 I_{4\times 4}. \end{aligned}$$

Figure 5 shows the nominal trajectories generated by DIRTRAN and DIRTREL. As in the pendulum example, the DIRTREL trajectory avoids saturating the actuator while taking longer to complete the swing up. In this case, however, the two trajectories are qualitatively different; the DIRTREL trajectory takes one additional swing to reach the vertical position.

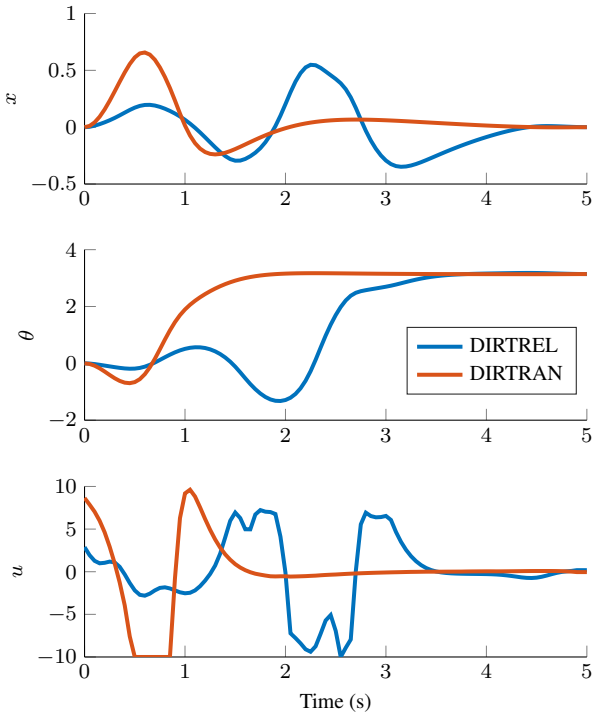

Fig. 5 Cart pole swing-up state and input trajectories

Numerous simulations were performed while varying the friction coefficient μ to characterize the robustness of the closed-loop systems. Successful swing up was observed using the DIRTRAN trajectory with LQR tracking in all trials over the range $0 \leq \mu \leq 0.064$, while the DIRTREL trajectory with LQR tracking was successful only over the range $0 \leq \mu \leq 0.295$.

6.3 Quadrotor with Wind Gusts

Next, we demonstrate DIRTREL on a quadrotor subjected to random wind gusts. The goal is for the aircraft to move from an initial position to a final position while navigating an obstacle field. The dynamics are described using the model of [29] and wind gusts are simulated by applying band-limited white noise accelerations to the system.

In both DIRTRAN and DIRTREL, constraints on the initial and terminal states of the quadrotor were applied, and a quadratic running cost of the following form was used:

$$g(x_i, u_i) = x_i^T Q x_i + u_i^T R u_i \,. \tag{45}$$

The same weighting matrices were used in the running cost, the robust cost $\ell_{\mathcal{W}}$, and the LQR tracking controllers:

$$\begin{aligned} R = R^\ell &= 0.1\, I_{4\times 4} \\ Q = Q^\ell &= \begin{bmatrix} 10\, I_{6\times 6} & 0 \\ 0 & I_{6\times 6} \end{bmatrix} \\ Q_N = Q_N^\ell &= \begin{bmatrix} 10\, I_{6\times 6} & 0 \\ 0 & I_{6\times 6} \end{bmatrix} \end{aligned}$$

Two different trajectories were computed with DIRTREL. In the first (DIRTREL-1), disturbances were bounded by ± 0.2 in the x and y axes and ± 0.05 in the z axis, corresponding to the following D matrix:

$$D = \begin{bmatrix} .2^2 & 0 & 0 \\ 0 & .2^2 & 0 \\ 0 & 0 & .05^2 \end{bmatrix}.$$

In the second trajectory (DIRTREL-2), the disturbance bounds were set to ± 0.4 in the x and y axes, corresponding to,

$$D = \begin{bmatrix} .4^2 & 0 & 0 \\ 0 & .4^2 & 0 \\ 0 & 0 & .05^2 \end{bmatrix}.$$

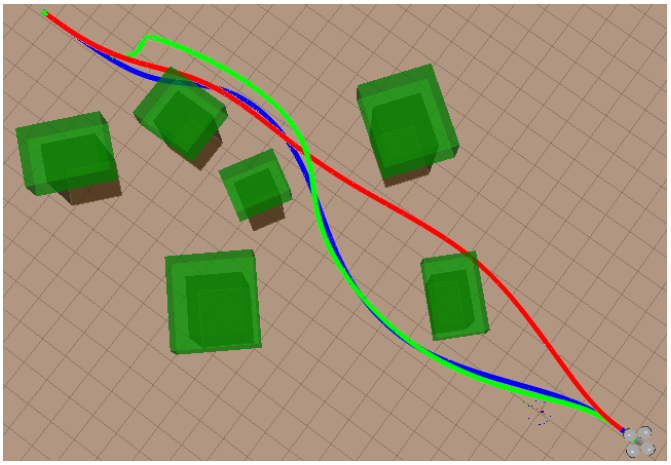

Fig. 6 DIRTRAN (red), DIRTREL-1 (blue), and DIRTREL-2 (green) quadrotor trajectories.

Figure 6 shows the nominal trajectories generated by DIRTRAN and DIRTREL. The DIRTRAN trajectory takes the shortest path to the goal state. However, it passes quite close to several obstacles. The DIRTREL trajectories, on the other hand, take longer paths to the goal but maintain greater distances from each obstacle.

We performed closed-loop simulations with random wind gusts of varying amplitudes. Disturbance inputs were generated by low-pass filtering white noise, then rescaling the resulting disturbance trajectory so that its maximum amplitude was equal to the desired value. Gust amplitudes in the x and y directions were varied from 0.1 to 0.5, while amplitudes in the z direction were held fixed at 0.05.

Table 1 Number of closed-loop quadrotor trajectories with obstacle collisions out of 100 trials.

Max Gust	DIRTRAN	DIRTRAN (Infl. Obs.)	DIRTREL-1	DIRTREL-2
0.1	37	0	0	0
0.2	65	1	0	0
0.3	77	4	3	0
0.4	82	21	5	0
0.5	90	27	9	1

Table 1 shows the results of 100 trials performed at each amplitude level. As expected, no collisions occurred using the DIRTREL controllers for gust amplitudes within the bounds imposed during planning (0.2 and 0.4 for for the first and second cases, respectively) result in no collisions, while the DIRTRAN controller was unable to avoid collisions with obstacles in many trials at every amplitude level.

To offer a more generous comparison, we calculated the closest distance between the quadrotor and an obstacle in the nominal DIRTREL-2 trajectory, inflated the obstacles by that distance, and re-planned a new trajectory with DIRTRAN. Obstacle inflation (also called constraint shrinking) techniques are a heuristic approach to improve robustness using traditional planning methods. The corresponding closed-loop simulation results are shown in the second column of Table 1. Due to its ability to explicitly reason about the closed-loop dynamics and how disturbances act on particular states, DIRTREL offers significantly better robustness than naive obstacle inflation.

6.4 Robot Arm with Fluid-Filled Container

Finally, we use DIRTREL to plan the motion of a robot arm carrying a fluid-filled container. The goal is to gently place the container on a shelf while avoiding collisions. A dynamics model of the seven-link Kuka IIWA arm was used, and bounds of ± 3 Newtons in the x and y directions and ± 10 Newtons in the z direction were placed on disturbance forces applied to the end effector in DIRTREL to account for both uncertain mass and un-modeled fluid-slosh dynamics inside the container.

In both algorithms, constraints were placed on the initial and final poses of the arm. The same quadratic running cost penalizing joint torque and quadratic terminal cost penalizing the final velocity of the end effector were also applied in both algorithms. The terminal cost was intended to encourage a gentle placement of the

container on the shelf. The following weighting matrices were used in both the robust cost function and LQR tracking controllers:

$$
\begin{aligned}
R = R^\ell &= 0.01\, I_{7\times 7} \\
Q = Q^\ell &= \begin{bmatrix} 100\, I_{7\times 7} & 0 \\ 0 & 10\, I_{7\times 7} \end{bmatrix} \\
Q_N = Q_N^\ell &= \begin{bmatrix} 500\, I_{7\times 7} & 0 \\ 0 & 50\, I_{7\times 7} \end{bmatrix}.
\end{aligned}
$$

The nominal trajectories produced by DIRTRAN and DIRTREL are depicted in Figure 7. As expected, the DIRTREL trajectory takes a wider path around the obstacle. However, as in the quadrotor example, this kinematic behavior can also be reproduced with DIRTRAN by inflating the obstacle.

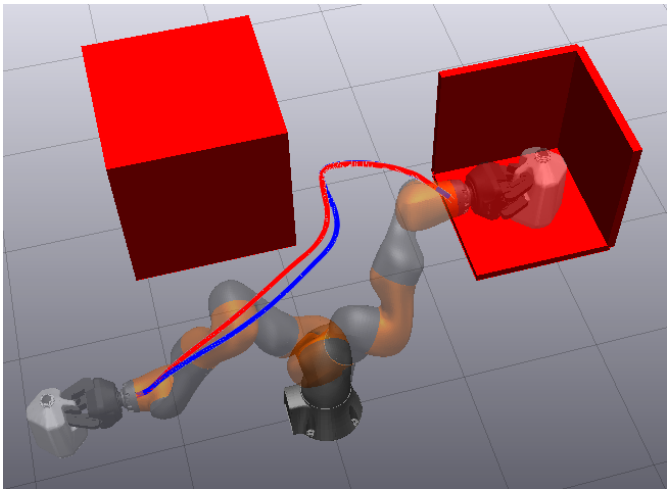

Fig. 7 Nominal end-effector trajectories produced by DIRTRAN (red) and DIRTREL (blue).

To compare dynamic performance, we compute the RMS deviations of the closed-loop system from the nominal trajectory with each controller. Ten trials were performed while varying the mass of the container and applying band-limited white noise disturbance forces to the end effector to simulate fluid slosh. DIRTRAN achieved an RMS state deviation of 0.1192 and an RMS input deviation of 4.478, while DIRTREL achieved an RMS state deviation of 0.0735 and an RMS input deviation of 4.422. The DIRTREL controller achieved nearly 40% better closed-loop tracking performance while using roughly the same control effort as the DIRTRAN controller.

7 Discussion

We have presented a new algorithm, DIRTREL, for robust feedback motion planning using approximate invariant funnels along the trajectory. The algorithm outputs

trajectories and closed-loop tracking controllers that outperform the standard combination of direct trajectory optimization followed by TVLQR synthesis. In addition to finding trajectories that locally minimize funnel volume through a closed-form cost function, DIRTREL easily handles constraints on both the nominal and disturbed system trajectories. Our simulation tests suggest that the approximations made in the algorithm are reasonable, and that robustness to nonlinear disturbances and model errors can be significantly improved for practical robotic systems.

Several interesting directions remain for future work. While we have focused on direct transcription methods in this paper due to their ease of implementation, it is also possible to derive similar robust versions of both collocation [11] and pseudospectral methods [7]. DIRTREL assumes that the system dynamics are linear near the nominal trajectory. While this approximation breaks down for large disturbances and highly nonlinear systems, we argue that it is consistent with the approximations inherent in the use of linear tracking controllers. However, this may not hold for large disturbances, and it would be interesting to more fully compare the approximate funnels computed by DIRTREL with nonlinear funnels computed using SOS techniques. It may also be possible to extend the DIRTREL algorithm to account for some nonlinearity by incorporating deterministic sampling ideas from the unscented Kalman filter [15, 27].

Acknowledgements

This work was supported by an Internal Research and Development grant from Draper. The authors would like to thank the reviewers and the members of the Harvard Agile Robotics Laboratory for their valuable feedback.

References

1. John T Betts. *Practical Methods for Optimal Control Using Nonlinear Programming*, volume 3 of *Advances in Design and Control*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001.
2. John T Betts. Survey of Numerical Methods for Trajectory Optimization. 21(2): 193–207, March–April 1998.
3. Hongkai Dai and Russ Tedrake. Optimizing Robust Limit Cycles for Legged Locomotion on Unknown Terrain. pages 1207–1213, 2012.
4. Hongkai Dai and Russ Tedrake. L2-Gain Optimization for Robust Bipedal Walking on Unknown Terrain. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2013.
5. Marc Peter Deisenroth and Carl Edward Rasmussen. PILCO: A Model-Based and Data-Efficient Approach to Policy Search. In *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, 2011.
6. Vishnu Desaraju, Alexander Spitzer, and Nathan Michael. Experience-driven Predictive Control with Robust Constraint Satisfaction under Time-Varying State Uncertainty. In *Robotics: Science and Systems (RSS)*, July 2017. ISBN 978-0-9923747-3-0. doi: 10.15607/RSS.2017.XIII.067.

7. Fariba Fahroo and I. Michael Ross. Direct Trajectory Optimization by a Chebyshev Pseudospectral Method. *Journal of Guidance, Control, and Dynamics*, 25 (1):160–166, January 2002. ISSN 0731-5090. doi: 10.2514/2.4862.
8. Farbod Farshidian and Jonas Buchli. Risk Sensitive, Nonlinear Optimal Control: Iterative Linear Exponential-Quadratic Optimal Control with Gaussian Noise. *arXiv:1512.07173 [cs]*, December 2015.
9. Philip E Gill, Waltar Murray, and Michael A Saunders. SNOPT: An SQP Algorithm for Large-scale Constrained Optimization. *SIAM Review*, 47(1):99–131, 2005.
10. B. Griffin and J. Grizzle. Walking gait optimization for accommodation of unknown terrain height variations. In *2015 American Control Conference (ACC)*, pages 4810–4817, July 2015. doi: 10.1109/ACC.2015.7172087.
11. C R Hargraves and S W Paris. Direct Trajectory Optimization Using Nonlinear Programming and Collocation. *J. Guidance*, 10(4):338–342, 1987.
12. D Jacobson. Differential dynamic programming methods for solving bang-bang control problems. *Automatic Control, IEEE Transactions on*, 13(6):661–675, December 1968. doi: 10.1109/TAC.1968.1099026.
13. D. H. Jacobson and D. Q. Mayne. *Differential Dynamic Programming*. Elsevier, 1970.
14. A. M. Johnson, J. King, and S. Srinivasa. Convergent Planning. *IEEE Robotics and Automation Letters*, 1(2):1044–1051, July 2016. ISSN 2377-3766. doi: 10.1109/LRA.2016.2530864.
15. S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.
16. A. Agung Julius and George J. Pappas. Trajectory Based Verification Using Local Finite-Time Invariance. In *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science, pages 223–236. Springer, Berlin, Heidelberg, April 2009. ISBN 978-3-642-00601-2 978-3-642-00602-9. doi: 10.1007/978-3-642-00602-9_16.
17. L E Kavraki, P Svestka, J C Latombe, and M H Overmars. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.
18. Mayuresh V. Kothare, Venkataramanan Balakrishnan, and Manfred Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–1379, October 1996. ISSN 0005-1098. doi: 10.1016/0005-1098(96)00063-5.
19. James J Kuffner Jr and Steven M LaValle. RRT-Connect : An Efficient Approach to Single-Query Path Planning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, April 2000.
20. Scott Kuindersma, Roderic Grupen, and Andrew Barto. Variable Risk Control via Stochastic Optimization. *International Journal of Robotics Research*, 32(7): 806–825, June 2013.
21. T. C. Lin and J. S. Arora. Differential dynamic programming technique for constrained optimal control. *Computational Mechanics*, 9(1):27–40, January 1991. ISSN 0178-7675, 1432-0924. doi: 10.1007/BF00369913.

22. Wei Lin and C. I. Byrnes. H infininty-control of discrete-time nonlinear systems. *IEEE Transactions on Automatic Control*, 41(4):494–510, April 1996. ISSN 0018-9286. doi: 10.1109/9.489271.
23. Jingru Lou and Kris Hauser. Robust Trajectory Optimization Under Frictional Contact with Iterative Learning. In *Robotics Science and Systems (RSS)*, 2015.
24. L. Magni, G. De Nicolao, R. Scattolini, and F. Allgöwer. Robust model predictive control for nonlinear discrete-time systems. *International Journal of Robust and Nonlinear Control*, 13(3-4):229–246, March 2003. ISSN 1099-1239. doi: 10.1002/rnc.815.
25. Anirudha Majumdar and Russ Tedrake. Robust online motion planning with regions of finite time invariance. In *Algorithmic Foundations of Robotics X*, pages 543–558. Springer, 2013.
26. Anirudha Majumdar and Russ Tedrake. Funnel Libraries for Real-Time Robust Feedback Motion Planning. *arXiv:1601.04037 [cs, math]*, January 2016.
27. Zachary Manchester and Scott Kuindersma. Derivative-Free Trajectory Optimization with Unscented Dynamic Programming. In *Proceedings of the 55th Conference on Decision and Control (CDC)*, Las Vegas, NV, 2016.
28. David Q. Mayne and Eric C. Kerrigan. Tube-Based Robust Nonlinear Model Predictive Control. In *Proceedings of the 7th IFAC Symposium on Nonlinear Control Systems*, pages 110–115, Pretoria, South Africa, 2007.
29. Daniel Mellinger, Nathan Michael, and Vijay Kumar. Trajectory generation and control for precise aggressive maneuvers with quadrotors. *The International Journal of Robotics Research*, 31(5):664–674, April 2012. ISSN 0278-3649, 1741-3176. doi: 10.1177/0278364911434236.
30. Joseph Moore and Russ Tedrake. Adaptive control design for underactuated systems using sums-of-squares optimization. In *Proceedings of the 2014 American Control Conference (ACC)*, 2014.
31. Joseph Moore, Rick Cory, and Russ Tedrake. Robust post-stall perching with a simple fixed-wing glider using LQR-Trees. *Bioinspiration & Biomimetics*, 9(2): 025013, June 2014. ISSN 1748-3182, 1748-3190. doi: 10.1088/1748-3182/9/2/025013.
32. Igor Mordatch, Kendall Lowrey, and Emanuel Todorov. Ensemble-CIO: Full-Body Dynamic Motion Planning that Transfers to Physical Humanoids. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2015.
33. Jun Morimoto, Garth Zeglin, and Christopher G Atkeson. Minimax Differential Dynamic Programming: Application to a Biped Walking Robot. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2003.
34. Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.
35. Yunpeng Pan, Evangelos Theodorou, and Kaivalya Bakshi. Robust Trajectory Optimization: A Cooperative Stochsatic Game Theoretic Approach. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, 2015.
36. Pablo Parillo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of

Technology, 2000.
37. Brian Plancher, Zachary Manchester, and Scott Kuindersma. Constrained Unscented Dynamic Programming. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, 2017.
38. Robert Platt, Leslie Kaelbling, Tomas Lozano-Perez, and Russ Tedrake. Non-gaussian belief space planning: Correctness and complexity. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, St. Paul, MN, May 2012.
39. Michael Posa, Cecilia Cantu, and Russ Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *International Journal of Robotics Research*, 33(1):69–81, January 2014.
40. Michael Posa, Scott Kuindersma, and Russ Tedrake. Optimization and stabilization of trajectories for constrained dynamical systems. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 1366–1373, Stockholm, Sweden, 2016. IEEE.
41. Nathan Ratliff, Matthew Zucker, J Andrew Bagnell, and Siddhartha Srinivasa. CHOMP: Gradient Optimization Techniques for Efficient Motion Planning. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2009.
42. John Schulman, Yan Duan, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, Jia Pan, Sachin Patil, Ken Goldberg, and Pieter Abbeel. Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research*, 33(9):1251–1270, August 2014. ISSN 0278-3649, 1741-3176. doi: 10.1177/0278364914528132.
43. Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and Stabilization of Complex Behaviors through Online Trajectory Optimization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.
44. Russ Tedrake, Ian R Manchester, Mark M Tobenkin, and John W Roberts. LQR-Trees: Feedback motion planning via sums of squares verification. *International Journal of Robotics Research*, 29:1038–1052, July 2010.
45. Mark Tobenkin, Ian Manchester, and Russ Tedrake. Invariant Funnels around Trajectories using Sums-of-Squares Programming. In *Proceedings of the 18th IFAC World Congress*, Milano, Italy, 2011.
46. Jur van den Berg, Pieter Abbeel, and Ken Goldberg. LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. *The International Journal of Robotics Research*, 30(7):895–913, June 2011. ISSN 0278-3649. doi: 10.1177/0278364911406562.
47. Bart van den Broek, Wim Wiegerinck, and Bert Kappen. Risk Sensitive Path Integral Control. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 615–622, 2010.
48. Peter Whittle. Risk-Sensitive Linear/Quadratic/Gaussian Control. *Advances in Applied Probability*, 13:764–777, 1981.
49. Je Sung Yeon and J. H. Park. Practical robust control for flexible joint robot manipulators. In *2008 IEEE International Conference on Robotics and Automation*, pages 3377–3382, May 2008. doi: 10.1109/ROBOT.2008.4543726.

50. Kemin Zhou. *Robust and Optimal Control*. Prentice Hall, Upper Saddle River, NJ, 1996. ISBN 978-0-13-456567-5.