DIRTREL: Robust Nonlinear Direct Transcription with Ellipsoidal Disturbances and LQR Feedback

Zachary Manchester and Scott Kuindersma
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138
Email: {zmanchester, scottk}@seas.harvard.edu

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 bounded 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. We derive a tractable robust optimization algorithm that combines direct transcription with linear-quadratic feedback to efficiently reason about closed-loop responses to disturbances. In the case of ellipsoidal disturbance sets, the state and input deviations along a nominal trajectory can be computed locally in closed form, thus allowing for fast evaluations of robust cost and constraint functions. The resulting algorithm, called DIRTREL, is an 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 planning and control tasks.

I. INTRODUCTION

Motion planning has been an active research topic that has yielded several successes in recent years, from sampling-based algorithms that scale to large state spaces [13, 14] to nonlinear optimization methods capable of handling complex dynamic constraints [31, 32, 2] and contacts [33, 29, 30]. 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 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. This paper aims to address this problem by extending a popular trajectory optimization method—direct transcription (DIRTRAN)—to optimize the closed-loop tracking performance of the system along the nominal planned trajectory.

This paper builds on previous work on robust motion planning, including methods based on direct trajectory optimization [25, 3], as well as indirect methods based on differential dynamic programming (DDP) [26, 6, 28]. While existing algorithms differ in the precise notion of robustness that each seeks to optimize, they all fall into one of two broad categories: deterministic minimax methods that seek to optimize performance given worst-case disturbances, and stochastic risk-sensitive methods that seek to optimize cost functionals that account for second-order statistics (e.g. variance) of system performance. To date, neither approach has resulted in an algorithm that can explicitly include known bounds on disturbances that enter the system dynamics in a nonlinear way, robustly handle state and input constraints, and scale to practical robotic systems with high dimensional state and input spaces.

In contrast, we propose a robust motion planning algorithm that reasons about disturbances by optimizing a robust cost function that is *differentiable* and can be computed *analytically*. The key to our approach is the observation that, by restricting the problem to time-varying linear feedback (e.g., LQR) and ellipsoidal disturbance sets, bounds on state and input deviations can be computed locally along a nominal trajectory in closed form. As a result, a computationally tractable penalty function over the set of all disturbances can be defined and constraints can be enforced on the disturbed trajectories. We incorporate this robust penalty function into a direct transcription method (DIRTRAN), and therefore inherit their well-known benefits, including straightforward handling of state and input constraints, good numerical conditioning, and favorable sparsity properties. 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 closed-loop robustness.

This paper is organized as follows: In Section II we summarize prior work related to robust control in robotics. We then review the classic direct transcription algorithm in Section III before proposing our robust extension, DIRTREL, in Section IV. Section V describes several simulation experiments used to validate our new algorithm and analyze its computational performance. Finally, we discuss conclusions and future work in Section VI.

II. RELATED WORK

The theory of robust control of linear dynamical systems has developed into a rich literature over the past four decades [37]. 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 investigated for

nonlinear systems [17, 19], but the resulting algorithms do not scale gracefully to most problems of interest in robotics. A notable exception is the work of Yeon and Park, who derived an H_∞ tracking controller for a 6-DoF manipulator for the case of a given trajectory and disturbances arising from flexible joints [36].

Several robust variants of differential dynamic programming [11] have been proposed for solving worst-case minimax problems [26], risk-sensitive optimizations for stochastic systems [6], and cooperative stochastic games [28]. Like the algorithm proposed in this paper, these methods consider closed-loop system responses under linear-quadratic feedback. However, they differ in the metric used to quantify robustness and lack the ability to explicitly incorporate known bounds on disturbances. Additionally, none of the existing DDP-based algorithms directly handles state or input constraints, although these could presumably be combined with constrained DDP variants [16]. In contrast, the algorithm presented in this paper naturally handles nonlinear constraints on the nominal and disturbed state and input trajectories.

Tube-based model-predictive control (MPC) algorithms [21] use MPC to reject additive disturbances along an optimized nominal trajectory. Mordatch et al. [25] developed an ensemble trajectory optimization method that aims to reduce the expected cost under random model parameters while reducing trajectory variation under PD control. Lou and Hauser [18] combined robust motion planning with model estimation to optimize robust motions involving contact changes. However, their approach required the kinematic plan to be given and only optimized the timing of the motion. Several authors have developed risk-sensitive optimal control methods [10, 35] for nonlinear stochastic systems with known models [6] or data-driven control learning approaches [34, 4, 15].

Verification approaches to feedback motion planning attempt to compute regions of finite-time invariance, or "funnels," which provide a certificate of stability. 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 [24]. A related line of work led to the development of robust adaptive tracking controllers with guaranteed finite-time performance [23]. 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 [20]. The algorithm we describe is complementary to these methods in that it aims to design robust trajectories that, if desired, can be subsequently tracked and verified using SOS-based methods.

The previous work most closely related to this paper is that of Griffin and Grizzle [8] and Dai and Tedrake [3]. These authors augment direct trajectory optimization methods with cost functionals that weight the tracking performance of linear feedback controllers. However, there are several important differences from our approach. In [8], a fixed-gain proportional-integral (PI) controller is assumed, placing severe limits on closed-loop performance and applicability of the method. In [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 constrained to satisfy a differential Riccati equation. In addition, disturbances must be sampled, which scales poorly with the dimensionality of the disturbance vector. This substantially increases the size and complexity of the nonlinear program that must be solved, limiting the practical utility of the algorithm.

III. DIRECT TRANSCRIPTION

Direct transcription methods solve optimal control problems by explicitly parameterizing the state and control 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 [7], that exploit the sparsity patterns in the linearized constraint matrix.

DIRTRAN is a particular implementation of direct transcription. Given a nonlinear dynamical system, $\dot{x} = f(x, u)$, we discretize the system trajectory 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,

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

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.

IV. 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.

A. State and Input Deviations

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

$$
x_{i+1} = f_h(x_i, u_i, w_i). \tag{2}
$$

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

$$x_{i+1} = f_h(x_i, u_i, 0) \qquad (3)$$

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

$$\delta x_{i+1} = f_h(x_i + \delta x_i, u_i + \delta u_i, w_i) - x_{i+1}, \qquad (4)$$

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

$$\delta u_i = -K_i \delta x_i. \qquad (5)$$

Any reasonable choice of 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,

$$K_i = (R + B_i^T P_{i+1} B_i)^{-1}(B_i^T P_{i+1} A_i)$$
$$P_i = Q + K_i^T R K_i^T + (A_i - B_i K_i)^T P_{i+1}(A_i - B_i K_i), \qquad (6)$$

where $A = \partial f_h / \partial x$, $B = \partial f_h / \partial u$, and $Q \succeq 0$ and $R \succ 0$ are user-specified state and input cost-weighting matrices.

B. A 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 some 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^ℓ and R^ℓ are positive semidefinite-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:

$$\ell_{\mathcal{W}}(x_{1:N}, u_{1:N-1}) \approx \frac{1}{\text{Vol}(\mathcal{W})} \int_{\mathcal{W}} \left(\delta x_N^T Q_N^\ell \delta x_N \right.$$
$$\left. + \sum_{i=1}^{N-1} \left(\delta x_i^T Q^\ell \delta x_i + \delta u_i^T R^\ell \delta u_i \right) \right) d\mathcal{W}. \quad (7)$$

For general nonlinear systems and disturbance sets, the integral in equation (7) 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 matrix $D \succ 0$, such that

$$w^T D^{-1} w \le 1. \qquad (8)$$

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 $D = D^{1/2} D^{1/2}$. Using the fact that ellipsoids map to ellipsoids under linear transformations, approximate ellipsoidal bounds on the state deviations, δx_i, at each time step i can be computed. As in equation (8), we parameterize these ellipsoids by matrices $E_i \succ 0$.

Assuming a bound on the initial state deviation δx_1 parameterized by E_1, the matrices E_i can be found at all future time steps using the system dynamics. Linearizing equation (4) about the nominal trajectory gives a set of linear time-varying equations of the form

$$\delta x_{i+1} \approx A_i \delta x_i + B_i \delta u_i + G_i w, \qquad (9)$$

where $G = \partial f_h / \partial w$. A recursion for E_{i+1} in terms of E_i and D can then be defined:

$$M_{i+1} = F_i M_i F_i^T, \qquad (10)$$

where M_i and F_i are defined as follows,

$$M_i = \begin{bmatrix} E_i & H_i \\ H_i^T & D \end{bmatrix} \qquad (11)$$

$$F_i = \begin{bmatrix} (A_i - B_i K_i) & G_i \\ 0 & I \end{bmatrix}, \qquad (12)$$

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

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 (10) is purely deterministic. E_i represents a strict bound rather than a statistical covariance. Second, equations (10)–(12) 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 statistical independence of noise at different time steps.

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 (7) 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,

$$\ell_{\mathcal{W}}(x_{1:N}, u_{1:N-1}) = \frac{1}{n_x} \sum_{\text{col}(E_N^{1/2})} \delta x_N^T Q_N^\ell \delta x_N$$
$$+ \frac{1}{n_x} \sum_{i=1}^{N-1} \sum_{\text{col}(E_i^{1/2})} \delta x_i^T Q^\ell \delta x_i + \delta u_i^T R^\ell \delta u_i, \quad (13)$$

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 (7) using the trace operator,

$$\delta x_i^T Q \delta x_i = \text{Tr}(Q \, \delta x_i \delta x_i^T) \tag{14}$$

$$\delta u_i^T R \delta u_i = \text{Tr}(R \, \delta u_i \delta u_i^T), \tag{15}$$

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:

$$\ell_{\mathcal{W}}(x_{1:N}, u_{1:N-1}) = $$
$$\text{Tr}(Q_N E_N) + \sum_{i=1}^{N-1} \text{Tr}\left((Q + K_i^T R K_i)E_i\right). \tag{16}$$

Equations (10)–(12) and (16) provide an easily computable cost function that quantifies the system's closed-loop sensitivity to disturbances. The evaluation of $\ell_{\mathcal{W}}(x_{1:N}, u_{1:N-1})$ is summarized in Algorithm 1.

Algorithm 1 Robust Cost Function

1: **function** $\ell_{\mathcal{W}}(x_{1:N}, u_{1:N-1}, D, E_1, Q^\ell, R^\ell, Q_N^\ell, Q, R)$
2:     **for** $i = 1 \ldots N - 1$ **do**
3:         $A_i \leftarrow \partial_{x=x_i} f_h(x, u, w)$
4:         $B_i \leftarrow \partial_{u=u_i} f_h(x, u, w)$
5:         $G_i \leftarrow \partial_{w=0} f_h(x, u, w)$
6:     **end for**
7:     $K_{1:N-1} \leftarrow TVLQR(A_{1:N-1}, B_{1:N-1}, Q, R)$
8:     $H_1 \leftarrow 0$
9:     **for** $i = 1 \ldots N - 1$ **do**
10:        $\ell \leftarrow \ell + \text{Tr}\left((Q^\ell + K_i^T R^\ell K_i)E_i\right)$
11:        $E_{i+1} \leftarrow (A_i - B_i K_i)E_i(A_i - B_i K_i)^T$
                $+ (A_i - B_i K_i)H_i G_i^T$
                $+ G_i H^T (A_i - B_i K_i)^T$
                $+ G_i D G_i^T$
12:        $H_{i+1} = (A_i - B_i K_i)H_i + G_i D$
13:     **end for**
14:     $\ell \leftarrow \ell + \text{Tr}(Q_N^\ell E_N)$
15:     **return** ℓ
16: **end function**

C. 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 the extreme values of δx_i on the boundary of the ellipsoid.

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

$$x_i^{\mathcal{W}} = x_i \pm \text{col}\left(E_i^{1/2}\right), \tag{17}$$

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

$$u_i^{\mathcal{W}} = u_i \pm \text{col}\left((K_i E_i K_i^T)^{1/2}\right). \tag{18}$$

The resulting optimization problem, referred to as DIRTREL, can be expressed as the following NLP:

$$\begin{aligned}
\underset{x_{1:N}, \, u_{1:N-1}, \, h}{\text{minimize}} \quad & \ell_{\mathcal{W}}(x_{1:N}, u_{1:N-1}) + g_N(x_N) + \sum_{i=0}^{N-1} g(x_i, u_i) \\
\text{subject to} \quad & x_{i+1} = f_h(x_i, u_i) \quad \forall i = 1 : N-1 \\
& u_i \in \mathcal{U} \quad\quad\quad\;\; \forall i = 1 : N-1 \\
& u_i^{\mathcal{W}} \in \mathcal{U} \quad\quad\quad \forall i = 1 : N-1 \\
& x_i \in \mathcal{X} \quad\quad\quad\;\; \forall i = 1 : N \\
& x_i^{\mathcal{W}} \in \mathcal{X} \quad\quad\quad \forall i = 1 : N \\
& h_{\min} \leq h \leq h_{\max}
\end{aligned} \tag{19}$$

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 (19) 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 [27]. Also, since DIRTREL combines the classic DIRTRAN algorithm with a robust cost function based on the LQR Riccati recursion, it inherits the favorable linear computational complexity of those algorithms with respect to the number of knot points in the trajectory [1]. Finally, we note that the number of decision variables has remained the same as standard DIRTRAN.

V. 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 commercial SQP solver SNOPT [7] 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.

A. 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:

$$R = R^\ell = 0.1$$

$$Q = Q^\ell = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}$$

$$Q_N = Q_N^\ell = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$$

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:

$$E_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{20}$$

The state and input trajectories produced by DIRTRAN and DIRTREL are shown in Figure 1. Consistent with the minimum-time nature of the problem, DIRTRAN produces 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 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 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 straight-forward manner that eliminates the need for such "hand tuning."
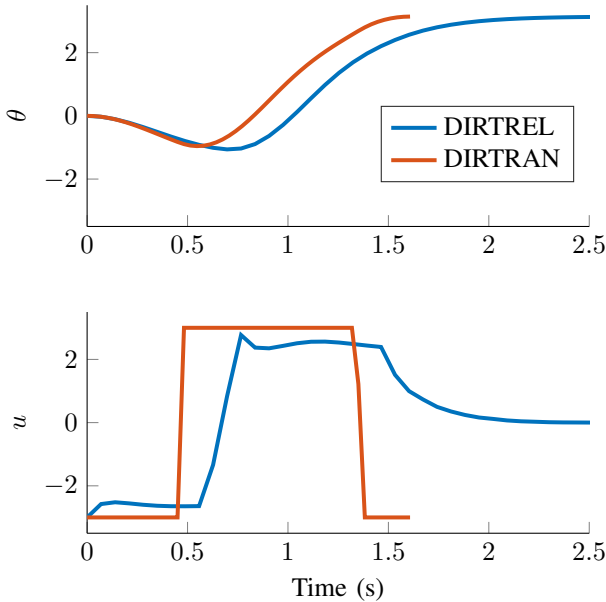

Fig. 1. Pendulum swing-up trajectories

B. Cart Pole with Unmodeled Friction

Friction is typically difficult to model in mechanical systems. DIRTREL allows uncertain friction forces to be accounted for without assuming any particular functional form.

To demonstrate this, we now consider a swing-up problem for the cart pole system depicted in Figure 2. 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.
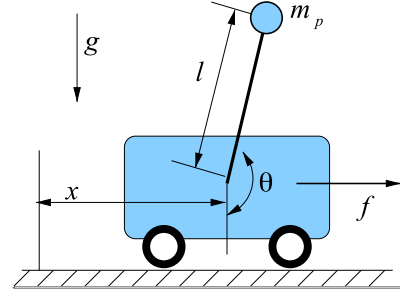

Fig. 2. Cart pole

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 $-10 \le u \le 10$. The nominal model used during trajectory optimization has no friction, while in simulation the following friction force is applied to the cart,

$$F_c = -\operatorname{sign}(\dot{x})\mu F_N, \tag{21}$$

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 $-2 \le w \le 2$, corresponding to $D = 4$. We 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:

$$R = R^\ell = 1$$

$$Q = Q^\ell = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Q_N = Q_N^\ell = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}.$$

Figure 3 shows the nominal trajectories generated by both 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

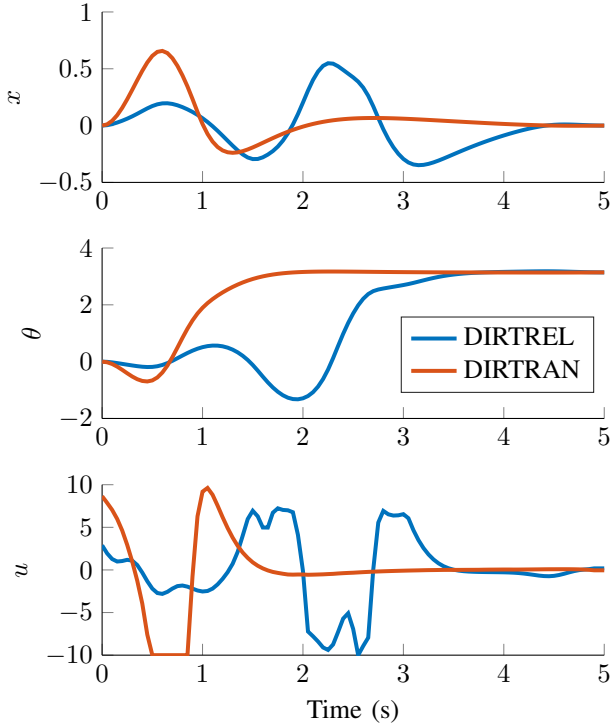trajectory takes one additional swing to reach the vertical position.

Fig. 3. Cart pole swing-up 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 over the range $0 \leq \mu \leq 0.064$, while the DIRTREL trajectory with LQR tracking was successful over the range $0 \leq \mu \leq 0.295$.

C. 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 [22] 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:

$$g(x_i, u_i) = x_i^T Q x_i + u_i^T R u_i. \tag{22}$$

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

$$R = R^\ell = 0.1\, I_{4\times4}$$
$$Q = Q^\ell = \begin{bmatrix} 10\, I_{6\times6} & 0 \\ 0 & I_{6\times6} \end{bmatrix}$$
$$Q_N = Q_N^\ell = \begin{bmatrix} 10\, I_{6\times6} & 0 \\ 0 & I_{6\times6} \end{bmatrix}$$

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,

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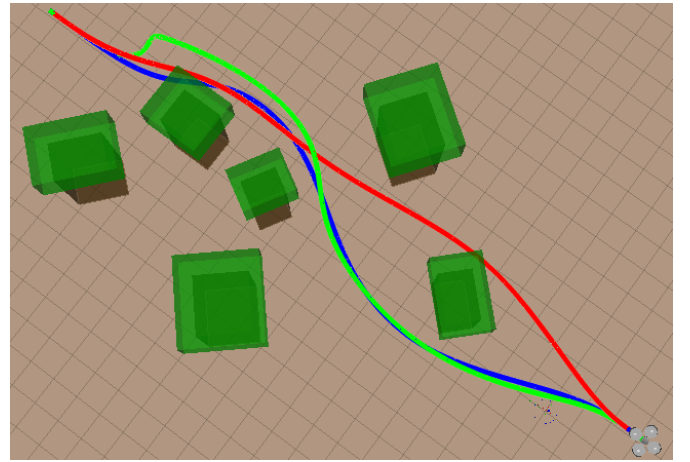

Fig. 4. DIRTRAN (red), DIRTREL-1 (blue), and DIRTREL-2 (green) quadrotor trajectories.

Figure 4 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 I
NUMBER OF CLOSED-LOOP QUADROTOR TRAJECTORIES WITH OBSTACLE COLLISIONS OUT OF 100 TRIALS.

Max Gust	DIRTRAN	DIRTREL-1	DIRTREL-2
0.1	37	0	0
0.2	65	0	0
0.3	77	3	0
0.4	82	5	0
0.5	90	9	1

Table I 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 Table II. 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.

TABLE II
NUMBER OF CLOSED-LOOP QUADROTOR TRAJECTORIES WITH OBSTACLE COLLISIONS OUT OF 100 TRIALS AFTER PLANNING WITH INFLATED OBSTACLES.

Max Gust	DIRTRAN (INFLATED)
0.1	0
0.2	1
0.3	4
0.4	21
0.5	27

D. 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:

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

The nominal trajectories produced by DIRTRAN and DIRTREL are depicted in Figure 5. 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.
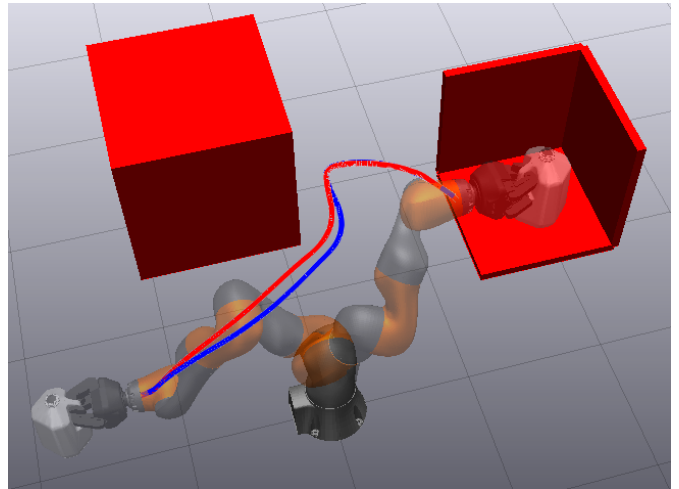

Fig. 5. Nominal end-effector trajectories produced by DIRTRAN (red) and DIRTREL (blue).

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 0.0142 and an RMS input deviation of 20.05, while DIRTREL achieved an RMS state deviation of 0.0054 and an RMS input deviation of 19.55. The DIRTREL controller achieved substantially better closed-loop tracking performance while using roughly the same control effort as the DIRTRAN controller.

VI. DISCUSSION

We have presented a new algorithm, DIRTREL, for robust feedback motion planning. By bounding disturbance sets with ellipsoids and assuming linear closed-loop feedback control, we have derived a computationally tractable robust penalty function and a method to account for state and input constraints along disturbed system trajectories. The resulting algorithm produces nominal trajectories and closed-loop tracking controllers which, together, outperform the standard combination of direct trajectory optimization followed by TVLQR controller synthesis. DIRTREL inherits all of the advantages of direct trajectory optimization methods, including the ability to handle constraints on both the nominal and disturbed system trajectories.

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 [9] and pseudospectral methods [5]. 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 and does not entail a practical disadvantage. It may, however, be possible to extend the algorithm to account

for some nonlinearity by incorporating deterministic sampling ideas from the Unscented Kalman Filter [12].

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. In *Proceedings of the 51st IEEE Conference on Decision and Control (CDC)*, pages 1207–1213, 2012.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] C R Hargraves and S W Paris. Direct Trajectory Optimization Using Nonlinear Programming and Collocation. *J. Guidance*, 10(4):338–342, 1987.

[10] 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.

[11] D. H. Jacobson and D. Q. Mayne. *Differential Dynamic Programming*. Elsevier, 1970.

[12] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[13] 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.

[14] 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.

[15] Scott Kuindersma, Roderic Grupen, and Andrew Barto. Variable Risk Control via Stochastic Optimization. *International Journal of Robotics Research*, 32(7):806–825, June 2013.

[16] 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.

[17] Wei Lin and C. I. Byrnes. H infinity-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.

[18] Jingru Lou and Kris Hauser. Robust Trajectory Optimization Under Frictional Contact with Iterative Learning. In *Robotics Science and Systems (RSS)*, 2015.

[19] 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.

[20] Anirudha Majumdar and Russ Tedrake. Funnel Libraries for Real-Time Robust Feedback Motion Planning. *arXiv:1601.04037 [cs, math]*, January 2016.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] Peter Whittle. Risk-Sensitive Linear/Quadratic/Gaussian Control. *Advances in Applied Probability*, 13:764–777, 1981.

[36] 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.

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