Chance-Constrained Trajectory Optimization for Non-linear Systems with Unknown Stochastic Dynamics

Onur Celik[1], Hany Abdulsamad[2] and Jan Peters[2,3]

***Abstract*— Iterative trajectory optimization techniques for non-linear dynamical systems are among the most powerful and sample-efficient methods of model-based reinforcement learning and approximate optimal control. By leveraging time-variant local linear-quadratic approximations of system dynamics and reward, such methods can find both a target-optimal trajectory and time-variant optimal feedback controllers. However, the local linear-quadratic assumptions are a major source of optimization bias that leads to catastrophic greedy updates, raising the issue of proper regularization. Moreover, the approximate models' disregard for any physical state-action limits of the system causes further aggravation of the problem, as the optimization moves towards unreachable areas of the state-action space. In this paper, we address the issue of constrained systems in the scenario of online-fitted stochastic linear dynamics. We propose modeling state and action physical limits as probabilistic chance constraints linear in both state and action and introduce a new trajectory optimization technique that integrates these probabilistic constraints by optimizing a relaxed quadratic program. Our empirical evaluations show a significant improvement in learning robustness, which enables our approach to perform more effective updates and avoid premature convergence observed in state-of-the-art algorithms.**

I. Introduction

Model-based reinforcement learning has played an important role in the latest surge of popular research interest in learning-control of autonomous systems [1]. More specifically, trajectory-centric optimization techniques of non-linear dynamics have proven to be extremely sample efficient in comparison to model-free policy search approaches [2]–[4].

With the notable exception of [5], model-based trajectory optimization techniques [6], [7] are closely related to differential dynamic programming methods (DDP), initially presented in [8] and further generalized in [9]. DDP is a powerful approach for generating optimal trajectories with optimal time-variant feedback controllers. By relying on linear-quadratic approximations of the dynamics and reward around a nominal trajectory, DDP-based methods can leverage the local approximations to iteratively optimize both the trajectory and tracking feedback controllers in closed-form via dynamic programming [10]. This view of control has a computational advantage over direct optimization techniques such as collocation methods, which solve large optimization problems directly in the trajectory space and generally result only in open-loop control sequences [11].

However, despite the overwhelming success of DDP, it still suffers from multiple shortcomings. On the one hand, the greedy exploitation of poor local approximations of dynamics is a major problem that leads to premature convergence. This issue has been effectively addressed in recent research by proposing different schemes of regularization [2], [6], [7]. On the other hand, state and action constraints present a serious challenge, as they introduce hard non-linearities, that cannot be straightforwardly incorporated into the dynamic programming framework. The effect of constraints becomes more severe in settings where a global model is not available for automatic differentiation, hence requiring the linear approximation of the dynamics to be fitted online from samples.

We view these issues of DDP as interlocked. The inability of time-variant local linear models to consider state and action constraints results in updates that exploit unreachable parts of the state-actions space, leading to catastrophically poor linear-quadratic approximations in regions subject to hard non-linearities. Moreover, considering constraints becomes more challenging in scenarios with stochastic dynamics, in that the true state of the system is hidden and only available through sufficient statistics. Another crucial aspect in a stochastic setting is the infinite support of the noisy measurements, which results in the constraints being active over the whole state-action space.

To address these issues, we propose an augmented view of DDP that introduces the physical limits as probabilistic chance constraints linear in state and action. When considering time-variant linear-Gaussian approximations of the dynamic, we can relax the generally non-convex chance constraints by applying Boole's inequality. This relaxation allows us to formulate an additional quadratic program that forces the optimized nominal trajectory to stay in a feasible state-action region with high probability, all while considering the feedback gains optimized by DDP.

Several approaches to trajectory optimization for non-linear systems address the problem of constrained dynamics on different levels. In the domain of deterministic environments, Tassa et al. considered action box-constraints in [12], while the authors in [13] introduce soft state-action limits via a Lagrange function augmentation. More sophisticated integration of constraints is presented in [14], in which the authors formulate a quadratic program to determine the active set of constraints at every iteration. In a stochastic setting, the work by Van Den Berg et. al [15] introduces probabilistic constraints as direct penalty terms on the cost function.

*This work has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement # 640554.

[1]Onur Celik is with the Department of Computer Science, Universtität Tübingen. mevluet-onur.celik@uni-tuebingen.de

[2]Hany Abdulsamad and Jan Peters are with the Department of Computer Science, Intelligent Autonomous Systems, Technische Universtität Darmstadt. {abdulsamad, peters}@ias.tu-darmstadt.de

[3]Jan Peters is with the Max Planck Institute for Intelligent Systems.

Furthermore, probabilistic constraints are considered in the context of linear optimal control. In [16] the authors optimally handle probabilistic constraints by ellipsoidal relaxation for finite-horizon open-loop scenarios, while in [17] a similar problem is tackled by applying Boole's inequality. In [18] Vitus et al. propose an algorithm to extend the work in [17] and [16] by considering closed-loop uncertainty and optimizing the risk allocation. Finally, in [19] the problem of infeasible initial solutions is addressed by progressively introducing the constraint into the objective.

We situate our contribution in the class of differential dynamic programming for stochastic non-linear systems subject to probabilistic constraints in state and action. We empirically show that our proposed approach can deal with highly non-linear constrained dynamic environments, leading to better overall performance and a robust learning process by virtue of improved online-fitted local approximations.

II. Chance-Constrained Optimization

Chance constraints arise naturally in different fields of optimization when considering stochastic systems. For an overview, we refer to [20]. Dealing with such probabilistic constraints proves to be challenging, as they are often non-convex and hard to evaluate without resolving to computationally expensive sampling techniques. These difficulties have motivated further research into tractable forms of chance constraints, which led to several convex approximations [21]. This work will focus on using Boole's inequality for constraint relaxation. A detailed description in the context of trajectories will follow in Section II-B.

A. Problem Formulation

Consider the constrained optimal control problem with probabilistic state and action constraints and unknown stochastic time-discrete transition dynamics

$$\begin{aligned} \max_{\boldsymbol{A}} \quad & J(\boldsymbol{s}, \boldsymbol{A}), \\ \text{s.t.} \quad & \boldsymbol{s}_{t+1} \sim \mathcal{P}(\boldsymbol{s}_{t+1} | \boldsymbol{s}_t, \boldsymbol{a}_t), \\ & \Pr(\boldsymbol{s}_{0:T} \in \mathcal{S}) \geq 1 - \theta, \\ & \Pr(\boldsymbol{a}_{0:T-1} \in \mathcal{A}) \geq 1 - \vartheta, \end{aligned}$$

where $\mathcal{S}$ and $\mathcal{A}$ are the feasible state and action spaces respectively. The probability levels θ, ϑ are hyperparameters that influence the risk behavior in terms of violating the constraints. The goal of this constrained optimization is to maximize the objective by finding the optimal action sequence $\boldsymbol{A}$. In general, we consider the expected cumulative reward for a trajectory of length T in the quadratic form

$$\begin{aligned} J(\boldsymbol{s}, \boldsymbol{A}) = -\mathbb{E}\Big[& \sum_{t=0}^{T-1} (\boldsymbol{s}_t - \boldsymbol{s}_{g,t})^\intercal \boldsymbol{M}_t (\boldsymbol{s}_t - \boldsymbol{s}_{g,t}) + \boldsymbol{a}_t^\intercal \boldsymbol{D}_t \boldsymbol{a}_t \\ & + (\boldsymbol{s}_T - \boldsymbol{s}_{g,T})^\intercal \boldsymbol{M}_T (\boldsymbol{s}_T - \boldsymbol{s}_{g,T}) \Big], \end{aligned} \tag{1}$$

where $\boldsymbol{M}$ and $\boldsymbol{D}$ are positive-definite weight matrices of appropriate dimensions and $\boldsymbol{s}_g$ is the target state. Note that a quadratic objective is not necessarily required, and non-quadratic objectives can be locally approximated by quadratic forms.

B. Relaxation of Chance Constraints

Chance constraints can be conservatively relaxed by applying Boole's inequality [22]–[24]. For the purpose of brevity, only upper-bound state constraints are considered. However, the same relaxation procedure can be straightforwardly applied to obtain a lower-bound and to relax the action constraints. Generally, the state-linear joint chance constraint for a whole trajectory is formulated as

$$\begin{aligned} \Pr(\boldsymbol{s}_{0:T} \in \mathcal{S}) &= \Pr(\bigcap_{t=0}^{T} \boldsymbol{s}_t \in \mathcal{S}) \geq 1 - \theta, \\ &= \Pr(\bigcap_{t=0}^{T} \boldsymbol{h}_t^\intercal \boldsymbol{s}_t \leq b_t) \geq 1 - \theta. \end{aligned} \tag{2}$$

where $\boldsymbol{h}_t$ and b_t parameterize the half-plane defined by the constraints. Consequently, the probability of a trajectory to be within a feasible set is constrained to be higher than a probability $1 - \theta$. In the framework of stochastic programming, it is usually beneficial to reformulate Equation (2) into separate inequalities over individual constraints [20], which is achieved by transforming the intersection operator into a union operator according to rules of probability.

$$\begin{aligned} \Pr(\bigcap_{t=0}^{T} \boldsymbol{h}_t^\intercal \boldsymbol{s}_t \leq b_t) &= 1 - \Pr(\bigcup_{t=0}^{T} \boldsymbol{h}_t^\intercal \boldsymbol{s}_t > b_t), \\ &\geq 1 - \sum_{t=0}^{T} 1 - \Pr(\boldsymbol{h}_t^\intercal \boldsymbol{s}_t \leq b_t). \end{aligned} \tag{3}$$

The sum in Inequality (3) can now be treated as a collection of single probabilities per time-step

$$\begin{aligned} \sum_{t=0}^{T} 1 - \Pr(\boldsymbol{h}_t^\intercal \boldsymbol{s}_t \leq b_t) \leq \theta, \\ \Pr(\boldsymbol{h}_t^\intercal \boldsymbol{s}_t \leq b_t) \geq 1 - \theta_t, \end{aligned} \tag{4}$$

where $\sum_{t=0}^{T} \theta_t = \theta$. By assuming a Gaussian probability density, a common assumption in control applications, Equation (4) is rewritten using the cumulative density function

$$\begin{aligned} \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{b_t - \boldsymbol{h}_t^\intercal \boldsymbol{\mu}_{\boldsymbol{s}_t}}{\sqrt{2\boldsymbol{h}_t^\intercal \boldsymbol{\Sigma}_{\boldsymbol{s}_t} \boldsymbol{h}_t}}\right)\right] \geq 1 - \theta_t, \\ b_t - \boldsymbol{h}_t^\intercal \boldsymbol{\mu}_{\boldsymbol{s}_t} - \sqrt{2\boldsymbol{h}_t^\intercal \boldsymbol{\Sigma}_{\boldsymbol{s}_t} \boldsymbol{h}_t}\ \operatorname{erf}^{-1}(1 - 2\theta_t) \geq 0, \end{aligned} \tag{5}$$

where $\boldsymbol{\mu}_{\boldsymbol{s}_t}$ and $\boldsymbol{\Sigma}_{\boldsymbol{s}_t}$ are the state mean and covariance respectively. Moreover, due to properties of the error function, the inequality $\sum_{t=0}^{T} \theta_t \leq \theta < 0.5$ is conservatively enforced by setting $\theta_t = \theta / T$ and requiring $\theta < 0.5$, as in [24].

C. Iterative Linear Quadratic Gaussian Control (iLQG)

We base our trajectory optimization technique on DDP/iLQG methods. This section provides a short overview on the principles of DDP [8] and iLQG [2]. For any arbitrary time-index reward function R_t, the trajectory optimization objective is the expected cumulative reward

$$J(\boldsymbol{s}, \boldsymbol{A}) = \mathbb{E}\left[\sum_{t=0}^{T-1} R_t(\boldsymbol{s}_t, \boldsymbol{a}_t) + R_T(\boldsymbol{s}_{\boldsymbol{T}})\right].$$

DDP and iLQG leverage the principle of dynamic programming to simplify the optimization over a complete sequence of actions $\boldsymbol{a}_{0:T-1}$ to an optimization over single actions $\boldsymbol{a}_t$ for each time-step. For this purpose the time-indexed state-value function is introduced

$$V_t(\boldsymbol{s}) = \max_{\boldsymbol{a}_t} \left[R_t(\boldsymbol{s}_t, \boldsymbol{a}_t) + \sum_{\boldsymbol{s}_{t+1}} V_{t+1}(\boldsymbol{s}_{t+1}) \mathcal{P}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t) \right],$$

over which the dynamic programming backward recursion is performed. By assuming linear transitions dynamics and a quadratic rewards along a nominal trajectory, optimal feedback controllers can be derived in closed-form. DDP and iLQG consider the perturbed state-action-value function $Q_t(\delta\boldsymbol{s}, \delta\boldsymbol{a}) = R_t(\boldsymbol{s}_t + \delta\boldsymbol{s}, \boldsymbol{a}_t + \delta\boldsymbol{a}) - R_t(\boldsymbol{s}_t, \boldsymbol{a}_t) + V_{t+1}\left(\mathcal{P}(\boldsymbol{s}_t + \delta\boldsymbol{s}, \boldsymbol{a}_t + \delta\boldsymbol{a})\right) - V_{t+1}\left(\mathcal{P}(\boldsymbol{s}_t, \boldsymbol{a}_t)\right)$, resulting from a second order Taylor approximation

$$Q_t(\delta\boldsymbol{s}, \delta\boldsymbol{a}) \approx \frac{1}{2} \begin{bmatrix} 1 \\ \delta\boldsymbol{s} \\ \delta\boldsymbol{a} \end{bmatrix}^{\intercal} \begin{bmatrix} 0 & \boldsymbol{Q}_{s,t}^{\intercal} & \boldsymbol{Q}_{a,t}^{\intercal} \\ \boldsymbol{Q}_{s,t} & \boldsymbol{Q}_{ss,t} & \boldsymbol{Q}_{sa,t} \\ \boldsymbol{Q}_{a,t} & \boldsymbol{Q}_{as,t} & \boldsymbol{Q}_{aa,t} \end{bmatrix} \begin{bmatrix} 1 \\ \delta\boldsymbol{s} \\ \delta\boldsymbol{a} \end{bmatrix}.$$

The subscripts s and a stand for the first and second order approximations. The entries of $Q_t(\delta\boldsymbol{s}, \delta\boldsymbol{a})$ are given as

$$\begin{aligned}
\boldsymbol{Q}_{s,t} &= \boldsymbol{R}_{s,t} + \mathcal{P}_{s,t}^{\intercal} \boldsymbol{V}_{s,t+1}, \\
\boldsymbol{Q}_{a,t} &= \boldsymbol{R}_{a,t} + \mathcal{P}_{a,t}^{\intercal} \boldsymbol{V}_{s,t+1}, \\
\boldsymbol{Q}_{ss,t} &= \boldsymbol{R}_{ss,t} + \mathcal{P}_{s,t}^{\intercal} \boldsymbol{V}_{ss,t+1} \mathcal{P}_{s,t} + \boldsymbol{V}_{s,t+1} \mathcal{P}_{ss,t}, \\
\boldsymbol{Q}_{aa,t} &= \boldsymbol{R}_{aa,t} + \mathcal{P}_{a,t}^{\intercal} \boldsymbol{V}_{ss,t+1} \mathcal{P}_{a,t} + \boldsymbol{V}_{s,t+1} \mathcal{P}_{aa,t}, \\
\boldsymbol{Q}_{as,t} &= \boldsymbol{R}_{as,t} + \mathcal{P}_{a,t}^{\intercal} \boldsymbol{V}_{ss,t+1} \mathcal{P}_{s,t} + \boldsymbol{V}_{s,t+1} \mathcal{P}_{as,t}.
\end{aligned}$$

The main difference of iLQG compared to DDP is in neglecting the second order derivatives of the dynamics in iLQG. Given these approximations the optimal feedback controller is given as $\delta\boldsymbol{a}^* = -\boldsymbol{Q}_{aa,t}^{-1}(\boldsymbol{Q}_a + \boldsymbol{Q}_{as,t}\delta\boldsymbol{s}) = \boldsymbol{K}_t\delta\boldsymbol{s} + \boldsymbol{k}_t$. Inserting $\delta\boldsymbol{a}^*$ into $Q_t(\delta\boldsymbol{s}, \delta\boldsymbol{a})$ returns the update equations of the state-value function per time-step

$$\begin{aligned}
\Delta V_t &= -\frac{1}{2} \boldsymbol{Q}_{a,t} \boldsymbol{Q}_{aa,t}^{-1} \boldsymbol{Q}_{a,t}, \\
\boldsymbol{V}_{s,t} &= \boldsymbol{Q}_{s,t} - \boldsymbol{Q}_{a,t} \boldsymbol{Q}_{aa,t}^{-1} \boldsymbol{Q}_{as,t}, \\
\boldsymbol{V}_{ss,t} &= \boldsymbol{Q}_{ss,t} - \boldsymbol{Q}_{sa,t} \boldsymbol{Q}_{aa,t}^{-1} \boldsymbol{Q}_{as,t}.
\end{aligned}$$

During the forward pass, new trajectories of the stochastic non-linear dynamics are sampled by propagating the actions through the real system

$$\begin{aligned}
\boldsymbol{a}_t &= \boldsymbol{a}_{r,t} + \boldsymbol{k}_t + \boldsymbol{K}_t(\boldsymbol{s}_t - \boldsymbol{s}_{r,t}), \\
\boldsymbol{s}_{t+1} &\sim \mathcal{P}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t), \quad \boldsymbol{s}_0 = \boldsymbol{s}_{r,0},
\end{aligned} \tag{6}$$

where $\boldsymbol{s}_{r,t}, \boldsymbol{a}_{r,t}$ denote the mean state and action at time t from the last iteration and are also referred to as the nominal or reference trajectory, here denoted by the subscript r.

Special care has to be taken during the backward pass of DDP and iLQG to ensure that $\boldsymbol{Q}_{aa,t}$ is negative-definite, which has inspired different regularization schemes. In DDP, this regularization is commonly applied to $\boldsymbol{Q}_{aa,t}$ as $\tilde{\boldsymbol{Q}}_{aa,t} = \boldsymbol{Q}_{aa,t} - \mu\boldsymbol{I}$, with $\mu \geq 0$. However, other regularizations directly affecting the value function have been shown to be more effective [2], and will be used throughout this work.

D. Augmented Linearized Closed-Loop System

To formulate the chance-constrained optimization problem, we first introduce the notation and system description of the online-fitted time-variant linear system. Following [19], our approach optimizes the feedforward terms of the control, while satisfying the constraints for the linearized dynamics and maintains the feedback gains computed during the backward pass of DDP/iLQG.

Given N trajectories from the non-linear system as described in Equation (6), we fit linear-Gaussian models to the sampled data via regularized linear regression. Consequently we obtain the transition and control matrices $\boldsymbol{A}_t, \boldsymbol{B}_t$, as well as the bias vector $\boldsymbol{c}_t$ for each time-step. The resulting time-variant linear dynamics $\boldsymbol{s}_{t+1} = \boldsymbol{A}_t\boldsymbol{s}_t + \boldsymbol{B}_t\boldsymbol{a}_t + \boldsymbol{c}_t + \boldsymbol{w}_t$, with $\boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{0}, \Sigma_t)$, and the controller $\boldsymbol{a}_t = \boldsymbol{K}_t(\boldsymbol{s}_t - \boldsymbol{s}_{r,t}) + \boldsymbol{k}_t + \boldsymbol{a}_{r,t}$ are used to formulate the closed-loop linear system $\boldsymbol{s}_{t+1} = \hat{\boldsymbol{A}}_t\boldsymbol{s}_t + \boldsymbol{B}_t\boldsymbol{k}_t + \boldsymbol{d}_t + \boldsymbol{w}_t$, where $\hat{\boldsymbol{A}}_t = \boldsymbol{A}_t + \boldsymbol{B}_t\boldsymbol{K}_t$ and $\boldsymbol{d}_t = \boldsymbol{c}_t - \boldsymbol{B}_t\boldsymbol{K}_t\boldsymbol{s}_{r,t} + \boldsymbol{B}_t\boldsymbol{a}_{r,t}$.

To represent the closed-loop system over an entire trajectory we use the augmented notation

$$\tilde{\boldsymbol{s}} = \begin{bmatrix} \boldsymbol{s}_0 \\ \boldsymbol{s}_1 \\ \vdots \\ \boldsymbol{s}_T \end{bmatrix}, \tilde{\boldsymbol{k}} = \begin{bmatrix} \boldsymbol{k}_0 \\ \boldsymbol{k}_1 \\ \vdots \\ \boldsymbol{k}_{T-1} \end{bmatrix}, \tilde{\boldsymbol{w}} = \begin{bmatrix} \boldsymbol{w}_0 \\ \boldsymbol{w}_1 \\ \vdots \\ \boldsymbol{w}_{T-1} \end{bmatrix}, \tilde{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{I} \\ \hat{\boldsymbol{A}}_0 \\ \vdots \\ \hat{\boldsymbol{A}}_{T-1} \cdots \hat{\boldsymbol{A}}_0 \end{bmatrix},$$

$$\tilde{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{B}_0 & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \hat{\boldsymbol{A}}_1\boldsymbol{B}_0 & \boldsymbol{B}_1 & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\boldsymbol{A}}_{T-1} \cdots \hat{\boldsymbol{A}}_1\boldsymbol{B}_0 & \hat{\boldsymbol{A}}_{T-1} \cdots \hat{\boldsymbol{A}}_2\boldsymbol{B}_1 & \dots & \boldsymbol{B}_{T-1} \end{bmatrix},$$

$$\tilde{\boldsymbol{d}} = \begin{bmatrix} \boldsymbol{d}_0 \\ \boldsymbol{d}_1 \\ \vdots \\ \boldsymbol{d}_{T-1} \end{bmatrix}, \tilde{\boldsymbol{G}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{I} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \hat{\boldsymbol{A}}_1 & \boldsymbol{I} & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\boldsymbol{A}}_{T-1} \cdots \hat{\boldsymbol{A}}_1 & \hat{\boldsymbol{A}}_{T-1} \cdots \hat{\boldsymbol{A}}_2 & \cdots & \boldsymbol{I} \end{bmatrix},$$

The augmented weighting matrices for the quadratic objective take the form

$$\begin{aligned}
\tilde{\boldsymbol{M}} &= \mathrm{diag}(\boldsymbol{M}_0, \dots, \boldsymbol{M}_T), \ \tilde{\boldsymbol{D}} = \mathrm{diag}(\boldsymbol{D}_0, \dots, \boldsymbol{D}_{T-1}), \\
\tilde{\boldsymbol{M}}_C &= \mathrm{diag}(\boldsymbol{M}_0 + \boldsymbol{K}_0^{\intercal}\boldsymbol{D}_0\boldsymbol{K}_0, \dots, \\
&\qquad \boldsymbol{M}_{T-1} + \boldsymbol{K}_{T-1}^{\intercal}\boldsymbol{D}_{T-1}\boldsymbol{K}_{T-1}, \boldsymbol{M}_T), \\
\tilde{\boldsymbol{K}} &= \mathrm{diag}(\boldsymbol{K}_0, \dots, \boldsymbol{K}_{T-1}),
\end{aligned}$$

and the closed-loop linearized stochastic dynamics is written in terms of the augmented notation as

$$\tilde{\boldsymbol{s}} = \tilde{\boldsymbol{A}}\boldsymbol{s}_0 + \tilde{\boldsymbol{B}}\tilde{\boldsymbol{k}} + \tilde{\boldsymbol{G}}\tilde{\boldsymbol{w}} + \tilde{\boldsymbol{G}}\tilde{\boldsymbol{d}}, \tag{7}$$

which in turn can be decomposed to the mean and covariance of a Gaussian state density

$$\begin{aligned}
\boldsymbol{\mu}_{\tilde{\boldsymbol{s}}} &= \tilde{\boldsymbol{A}}\boldsymbol{s}_0 + \tilde{\boldsymbol{B}}\tilde{\boldsymbol{k}} + \tilde{\boldsymbol{G}}\tilde{\boldsymbol{d}}, \\
\tilde{\boldsymbol{\Sigma}}_{\tilde{\boldsymbol{s}}} &= \tilde{\boldsymbol{A}}\boldsymbol{\Sigma}_{\boldsymbol{s}_0}\tilde{\boldsymbol{A}}^{\intercal} + \tilde{\boldsymbol{G}}\tilde{\boldsymbol{\Sigma}}_{\tilde{\boldsymbol{w}}}\tilde{\boldsymbol{G}}^{\intercal},
\end{aligned}$$

where $\tilde{\boldsymbol{\Sigma}}_{\tilde{\boldsymbol{w}}}$ are the stacked estimates of the covariance for each time-step, taken under the N samples drawn during the

last forward pass. Furthermore, given the feedback gains, we compute the action covariance along the trajectory

$$\tilde{\boldsymbol{\Sigma}}_{\tilde{\boldsymbol{a}}} = \tilde{\boldsymbol{K}}\tilde{\boldsymbol{A}}\boldsymbol{\Sigma}_{\boldsymbol{s}_0}\tilde{\boldsymbol{A}}^\intercal\tilde{\boldsymbol{K}}^\intercal + \tilde{\boldsymbol{K}}\tilde{\boldsymbol{G}}\tilde{\boldsymbol{\Sigma}}_{\tilde{\boldsymbol{w}}}\tilde{\boldsymbol{G}}^\intercal\tilde{\boldsymbol{K}}^\intercal.$$

E. Augmented Objective and Relaxed Chance Constraints

We simplify Objective (1) by using the stacked notation and the closed-loop matrices from Section II-D

$$\begin{aligned}
J(\tilde{\boldsymbol{s}}, \tilde{\boldsymbol{a}}) &= -\mathbb{E}[\tilde{\boldsymbol{s}}^\intercal\tilde{\boldsymbol{M}}_C\tilde{\boldsymbol{s}}] + \mathbb{E}[2\tilde{\boldsymbol{s}}_g^\intercal\tilde{\boldsymbol{M}}\tilde{\boldsymbol{s}}] - \mathbb{E}[\tilde{\boldsymbol{s}}_g^\intercal\tilde{\boldsymbol{M}}\tilde{\boldsymbol{s}}_g]... \\
&... + \mathbb{E}[2\tilde{\boldsymbol{s}}_r^\intercal\tilde{\boldsymbol{K}}^\intercal\tilde{\boldsymbol{D}}\tilde{\boldsymbol{K}}\tilde{\boldsymbol{s}}] - \mathbb{E}[2\tilde{\boldsymbol{a}}_r^\intercal\tilde{\boldsymbol{D}}\tilde{\boldsymbol{K}}\tilde{\boldsymbol{s}}] - \mathbb{E}[2\tilde{\boldsymbol{k}}^\intercal\tilde{\boldsymbol{D}}\tilde{\boldsymbol{K}}\tilde{\boldsymbol{s}}]... \\
&... - \mathbb{E}[\tilde{\boldsymbol{s}}_r^\intercal\tilde{\boldsymbol{K}}^\intercal\tilde{\boldsymbol{D}}\tilde{\boldsymbol{K}}\tilde{\boldsymbol{s}}_r] + \mathbb{E}[2\tilde{\boldsymbol{a}}_r^\intercal\tilde{\boldsymbol{D}}\tilde{\boldsymbol{K}}\tilde{\boldsymbol{s}}_r] + \mathbb{E}[2\tilde{\boldsymbol{k}}^\intercal\tilde{\boldsymbol{D}}\tilde{\boldsymbol{K}}\tilde{\boldsymbol{s}}_r]... \\
&... - \mathbb{E}[\tilde{\boldsymbol{a}}_r^\intercal\tilde{\boldsymbol{D}}\tilde{\boldsymbol{a}}_r] - \mathbb{E}[2\tilde{\boldsymbol{k}}^\intercal\tilde{\boldsymbol{D}}\tilde{\boldsymbol{a}}_r] - \mathbb{E}[\tilde{\boldsymbol{k}}^\intercal\tilde{\boldsymbol{D}}\tilde{\boldsymbol{k}}].
\end{aligned}$$

Given that the expectations are of linear-quadratic quantities under Gaussian densities, it is possible to evaluate this objective in closed-form. This objective depends only on the forward terms $\tilde{\boldsymbol{k}}$ and can be reformulated as $\tilde{J}(\tilde{\boldsymbol{k}})$.

Following the relaxation presented in Section II-B and using the stacked notation we can write the upper and lower state-linear chance constraints as

$$\tilde{\boldsymbol{b}}_u - \tilde{\boldsymbol{h}}_u^\intercal\boldsymbol{\mu}_{\tilde{\boldsymbol{s}}} - \sqrt{2\tilde{\boldsymbol{h}}_u^\intercal\tilde{\boldsymbol{\Sigma}}_{\tilde{\boldsymbol{s}}}\tilde{\boldsymbol{h}}_u} \odot \mathbf{erf}^{-1}(\mathbf{1} - 2\tilde{\boldsymbol{\theta}}_u) \geq \mathbf{0}, \quad (8)$$

$$-\tilde{\boldsymbol{b}}_l + \tilde{\boldsymbol{h}}_l^\intercal\boldsymbol{\mu}_{\tilde{\boldsymbol{s}}} + \sqrt{2\tilde{\boldsymbol{h}}_l^\intercal\tilde{\boldsymbol{\Sigma}}_{\tilde{\boldsymbol{s}}}\tilde{\boldsymbol{h}}_l} \odot \mathbf{erf}^{-1}(2\tilde{\boldsymbol{\theta}}_l - \mathbf{1}) \geq \mathbf{0}, \quad (9)$$

where $\tilde{\boldsymbol{h}}$ and $\tilde{\boldsymbol{b}}$ parameterize the upper and lower half-planes of the state constraints and $\tilde{\boldsymbol{\theta}}_u$ and $\tilde{\boldsymbol{\theta}}_l$ denote the probability values per time-step, all stacked and indexed by u and l respectively. Analogously, the action constraints of the closed-loop system can be formulated

$$\tilde{\boldsymbol{z}}_{\mathrm{u}} - \tilde{\boldsymbol{f}}_u^\intercal(\tilde{\boldsymbol{K}}(\boldsymbol{\mu}_{\tilde{\boldsymbol{s}}} - \tilde{\boldsymbol{s}}_r) + \tilde{\boldsymbol{a}}_r + \tilde{\boldsymbol{k}}) - \boldsymbol{\lambda}_u \geq \mathbf{0}, \quad (10)$$

$$-\tilde{\boldsymbol{z}}_l + \tilde{\boldsymbol{f}}_l^\intercal(\tilde{\boldsymbol{K}}(\boldsymbol{\mu}_{\tilde{\boldsymbol{s}}} - \tilde{\boldsymbol{s}}_r) + \tilde{\boldsymbol{a}}_r + \tilde{\boldsymbol{k}}) + \boldsymbol{\lambda}_l \geq \mathbf{0}, \quad (11)$$

where $\boldsymbol{\lambda}_u = \sqrt{2\tilde{\boldsymbol{f}}_u^\intercal\tilde{\boldsymbol{\Sigma}}_{\tilde{\boldsymbol{a}}}\tilde{\boldsymbol{f}}_u} \odot \mathbf{erf}^{-1}(\mathbf{1} - 2\tilde{\boldsymbol{\vartheta}}_u)$ and $\boldsymbol{\lambda}_l = \sqrt{2\tilde{\boldsymbol{f}}_l^\intercal\tilde{\boldsymbol{\Sigma}}_{\tilde{\boldsymbol{a}}}\tilde{\boldsymbol{f}}_l} \odot \mathbf{erf}^{-1}(2\tilde{\boldsymbol{\vartheta}}_l - \mathbf{1})$, $\tilde{\boldsymbol{f}}$ and $\tilde{\boldsymbol{z}}$ are the stacked half-plane parameters of the action constraints and $\tilde{\boldsymbol{\vartheta}}_u, \tilde{\boldsymbol{\vartheta}}_l$ are the stacked upper and lower bound probabilities per time-step. The operator $\odot$ denotes the element-wise multiplication.

F. Chance-Constrained Trajectory Optimization

Based on the formulations introduced in Section II-D and Section II-E, it is possible to construct an optimization problem around the reference trajectory to find a sequence of feedforward terms $\tilde{\boldsymbol{k}}$ that maintain the Constraints (8-11).

The resulting optimization is a quadratic program with linear constraints in $\tilde{\boldsymbol{k}}$. Thus, the probabilistic problem reduces to a deterministic one, which can be solved efficiently with many numerical solvers, for example, qpOASES [25] within the CasADi framework [26]. The complete dynamic programming and optimization loop is described in Algorithm 1 and is summarized as follows: During an initial forward pass, we obtain N trajectory samples, around which the dynamics is linearized for each time-step. The linearized dynamics is used to perform the backward pass of iLQG and obtain the feedback and feedforward controllers along the reference trajectory. These controllers are then used to formulate the closed-loop linearized system with the stacked notation and to warm-start the quadratic program. The solution of the constrained program returns the optimal feedforward sequence $\boldsymbol{k}_t$, which is used to perform the next forward pass and linearization. Following [2], we also use the hyperparameter α that scales the feedforward control in order to keep the next forward pass of the non-linear system in a valid trust-region around the linear-quadratic approximations.

Algorithm 1 Chance-Constrained Trajectory Opt. (CCTO)

Input: $\theta_{u,t},\ \theta_{l,t},\ \vartheta_{u,t},\ \vartheta_{l,t},\ \alpha, N$
Output: $\boldsymbol{K}_t,\ \boldsymbol{k}_t,\ \boldsymbol{s}_{r,t},\ \boldsymbol{a}_{r,t}$

```
a_t^{1:N}, s_t^{1:N} ← forwardPass(a_{r,t}, s_{r,t}, K_t, k_t, α)
while not converged do
    a_{r,t}, s_{r,t} ← meanTraj(a_t^{1:N}, s_t^{1:N})
    A_t, B_t, c_t ← fitDynamics(a_t^{1:N}, s_t^{1:N})
    K_t, k_t^⋆ ← backwardPass(A_t, B_t)
    k_t ← solveQP(A_t, B_t, c_t, K_t, k_t^⋆, θ_{u,t}, θ_{l,t}, ϑ_{u,t}, ϑ_{l,t})
    a_t^{1:N}, s_t^{1:N} ← forwardPass(a_{r,t}, s_{r,t}, K_t, k_t, α)
end while
```

III. Empirical Evaluation

We evaluate our approach on two highly non-linear dynamical tasks, the Furuta pendulum [27] and a Cart-Pole environment. Both problems are under-actuated and have state and actions constraints. We consider quadratic reward functions for both experiments and set the probability values for violating the constraints to $\theta_u = \theta_l = \vartheta_u = \vartheta_l = 0.01$.

a) Furuta Pendulum Swing-Up: In the Furuta pendulum the state is represented by the angles of both links and the corresponding angular velocities. Only the horizontal link is actuated and is subject to both state and the action constraints. To make the environment stochastic, we introduce both action and process noise. We run our experiment under identical conditions for CCTO and iLQG. We fix the feedforward scalar α to 0.05 for both algorithms and perform 20 seeded trials, each with 45 iterations, 50 rollouts per iteration. The resulting performance curve of both algorithms can be seen in Figure 1. Furthermore, we present the planned nominal trajectories, as well as the planned nominal actions of both algorithms for one trial. The filled space is the area between the minimum and maximum values of states and actions and should not be confused with a probability distribution over trajectories. The advantage of our approach is clear. CCTO reaches better overall performance with a higher final reward and smaller standard deviation, Table I. iLQG plans frequently and consistently to violate the constraints, while CCTO keeps the state and action trajectories within a feasible space. This consideration leads to an improved approximation of the non-linear system dynamic and allows CCTO to perform robust improvement steps during the optimization. This result is affirmed by the low regularization values of CCTO, Table II.

b) Cart-Pole Swing-Up: For the well-known Cart-Pole environment, we consider constraints on the position of the cart as well as on the action. To make the task more

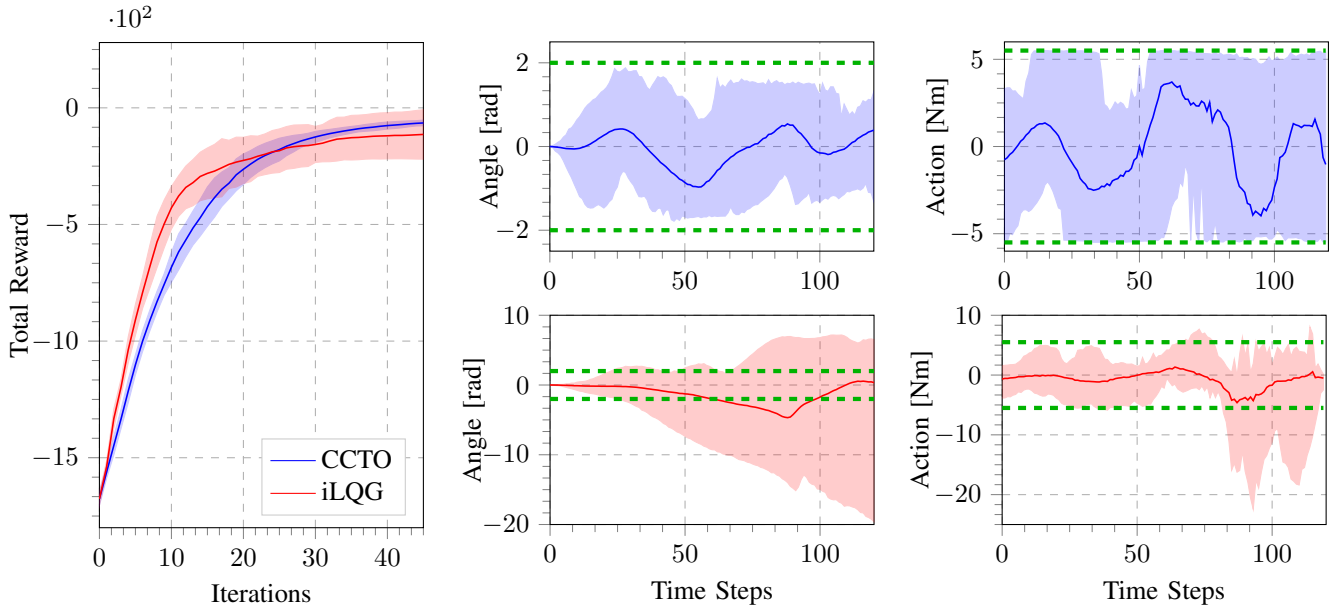

Fig. 1: Total-reward curve reflecting the performance of iLQG and CCTO for the Furuta pendulum swing-up task (left). In addition, we show the space (min. and max.) of planned nominal trajectories of the constrained angle (middle) and the corresponding executed actions (right), CCTO (blue), iLQG (red). CCTO obeys the physical limits of the system, while iLQG drives the dynamics against the constraints (green). These violations lead to poor linear approximations of the dynamics and an overall slightly lower mean and higher variance performance of iLQG.

Iteration	10	30	45
CCTO	$-6.8(\pm 0.32)$	$\mathbf{-1.3(\pm 0.11)}$	$\mathbf{-0.65(\pm 0.6)}$
iLQG	$\mathbf{-4.3(\pm 0.46)}$	$-1.6(\pm 0.39)$	$-1.1(\pm 0.53)$

TABLE I: Mean total reward and standard deviation of the Furuta swing-up task scaled by $1\mathrm{e}{-2}$.

Iteration	10	30	45
CCTO	**0**	$\mathbf{2.5\mathrm{e}{-8}}$	$\mathbf{1\mathrm{e}{-4}}$
iLQG	**0**	2.85e38	5e80

TABLE II: Mean regularization in the Furuta task over all trials for different iterations. CCTO needs less regularization due to avoidance of hard non-linearities.

Iteration	20	30	55
CCTO	$\mathbf{-2.3(\pm 0.32)}$	$\mathbf{-1.2(\pm 0.32)}$	$\mathbf{-0.31(\pm 0.06)}$
iLQG	$-9.3(\pm 0.10)$	$-9.3(\pm 0.10)$	$-9.3(\pm 0.10)$

TABLE III: Mean total reward and standard deviation of the Cart-Pole swing-up task scaled by $1\mathrm{e}{-2}$.

Iteration	20	30	55
CCTO	**0**	**0**	**0**
iLQG	5.7e39	1e80	1e80

TABLE IV: Mean regularization in the Cart-Pole task over all trials for different iterations. CCTO needs less regularization due to avoidance of hard non-linearities.

challenging, we again apply action and process noise, enforce harsh action constraints and limit the time horizon to 100 time steps, the equivalent of 2 seconds. We evaluate iLQG and CCTO on 20 seeded trials, each with 55 iterations and 50 rollouts per iteration. We set the feedforward scaling parameter α to 0.1. Analogously to the last experiment, Figure 2 depicts the performance curve of iLQG and CCTO, as well as the spaces of planned nominal trajectories for the cart's position and the corresponding actions. In this experiment, iLQG moves very quickly towards a local optimum and does not manage to swing the Cart-Pole up. In contrast, CCTO performs the swing-up by finding a suitable nominal trajectory in the feasible constrained space. Tables III and IV reflect the performance discrepancy between both algorithms, in terms of total rewards and needed regularization.

IV. Conclusion and Future Research

We have proposed a new trajectory optimization technique, based on the framework of differential dynamic programming, that takes into consideration probabilistic chance constraints in stochastic environments with unknown non-linear dynamics. We used Boole's inequality to conservatively relax the non-convex chance constraints, enabling us to formulate a constrained quadratic program and optimize the nominal trajectory to stay inside the feasible set defined by the probabilistic linear state and action limits. We have provided a thorough derivation of our approach and empirically demonstrated the advantage of enforcing physical limits on two simulated highly dynamical and stochastic non-linear systems. The results indicate that incorporating the chance constraints leads to higher fidelity in the online-fitted local linear-quadratic approximations, and consequently greatly influences the robustness of the iterative optimization process. This observation is reflected in very low regularizations in comparison to standard iLQG.

In future research, we will extend our optimization to include not only the nominal trajectory but also the feedback gains, and we will consider optimizing the probabilistic

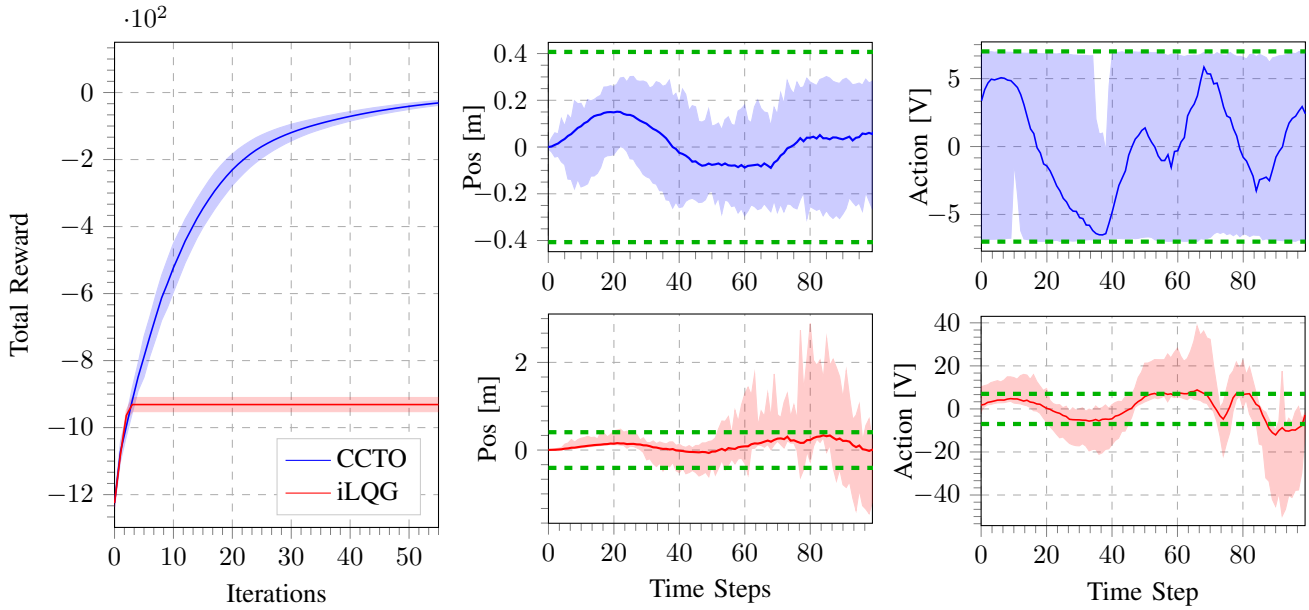

Fig. 2: Total-reward curve reflecting the performance of iLQG and CCTO for the Cart-Pole swing-up task (left). Furthermore, we show the space (min. and max.) of planned nominal trajectories of the constrained position (middle) and the corresponding executed actions (right), CCTO (blue), iLQG (red). CCTO obeys the physical limits of the system, while iLQG drives the dynamics against the constraints (green). These violations, especially those of the action constraint cause iLQG to get stuck in a poor local optimum, while CCTO is able to solve the task and perform the swing-up.

constraint bounds via risk allocation to achieve dynamic risk measures across time and iterations. In addition, we plan to move to the fully stochastic optimization framework of maximum-entropy iLQG [6] to avoid regularization heuristics of the DDP framework.

References

[1] M. P. Deisenroth, G. Neumann, and J. Peters, "A survey on policy search for robotics," *Foundations and Trends® in Robotics*, 2013.

[2] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012.

[3] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control," *IEEE transactions on pattern analysis and machine intelligence*, 2015.

[4] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, 2016.

[5] M. Deisenroth and C. E. Rasmussen, "PILCO: A model-based and data-efficient approach to policy search," in *Proceedings of the 28th International Conference on machine learning*, 2011.

[6] S. Levine and P. Abbeel, "Learning neural network policies with guided policy search under unknown dynamics," in *Advances in Neural Information Processing Systems*, 2014.

[7] H. Abdulsamad, O. Arenz, J. Peters, and G. Neumann, "State-regularized policy search for linearized dynamical systems," in *International Conference on Automated Planning and Scheduling*, 2017.

[8] D. H. Jacobson and D. Q. Mayne, "Differential dynamic programming," 1970.

[9] E. Todorov and W. Li, "A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *American Control Conference*. IEEE, 2005.

[10] B. D. Anderson and J. B. Moore, *Optimal control: Linear quadratic methods*. Courier Corporation, 2007.

[11] O. Von Stryk and R. Bulirsch, "Direct and indirect methods for trajectory optimization," *Annals of operations research*, 1992.

[12] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *IEEE International Conference on Robotics and Automation*. IEEE, 2014.

[13] B. Plancher, Z. Manchester, and S. Kuindersma, "Constrained unscented dynamic programming," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2017.

[14] Z. Xie, C. K. Liu, and K. Hauser, "Differential dynamic programming with nonlinear constraints," in *IEEE International Conference on Robotics and Automation*. IEEE, 2017.

[15] J. Van Den Berg, S. Patil, and R. Alterovitz, "Efficient approximate value iteration for continuous gaussian pomdps," in *AAAI Conference on Artificial Intelligence*, 2012.

[16] D. H. Van Hessem, "Stochastic inequality constrained closed-loop model predictive control–with application to chemical process operation," 2004.

[17] L. Blackmore and M. Ono, "Convex chance constrained predictive control without sampling," in *AIAA Guidance, Navigation, and Control Conference*, 2009.

[18] M. P. Vitus and C. J. Tomlin, "Closed-loop belief space planning for linear, gaussian systems," in *IEEE International Conference on Robotics and Automation*. IEEE, 2011.

[19] G. Kurz, M. Dolgov, and U. D. Hanebeck, "Progressive closed-loop chance-constrained control," in *International Conference on Information Fusion*. IEEE, 2016.

[20] A. Prékopa, *Stochastic programming*. Springer Science & Business Media, 2013.

[21] A. Nemirovski and A. Shapiro, "Convex approximations of chance constrained programs," *SIAM Journal on Optimization*, 2006.

[22] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[23] M. P. Vitus and C. J. Tomlin, "On feedback design and risk allocation in chance constrained control," in *IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011.

[24] S. Jha and V. Raman, "On optimal control of stochastic linear hybrid systems," in *International Conference on Formal Modeling and Analysis of Timed Systems*. Springer, 2016.

[25] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, 2014.

[26] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, 2018.

[27] K. Furuta, M. Yamakita, and S. Kobayashi, "Swing-up control of inverted pendulum using pseudo-state feedback," *Journal of Systems and Control Engineering*, 1992.