Chance-Constrained Convex MPC for Quadruped Locomotion

Ananya Trivedi[1], Sarvesh Prajapati[1], Mark Zolotas[1,2], Taşkın Padır[1,3]

***Abstract*— Recent advances in quadrupedal locomotion have highlighted the complexities of maintaining stability and performance in diverse and challenging environments. To address these issues, we model the variability in payload and terrain as distributions of parametric uncertainties and additive disturbances in the single rigid body dynamics (SRBD) model. By solving a stochastic optimal control problem with chance-constrained friction cone constraints, we determine the ground reaction forces required for dynamic stability using convex quadratic programming. This approach achieves an update rate of approximately 500 Hz, which is comparable to deterministic linear Model Predictive Control (LMPC). In 1000 Monte Carlo simulations, our method outperformed LMPC and MPC with hand-tuned safety margins in maintaining stability over uneven terrain, reducing foot slippage, and accurately tracking the center of mass trajectory, even with additional payloads. Hardware experiments on the Unitree Go1 robot showed that, with the same controller parameters, the robot successfully performed blind locomotion across varied terrains while carrying unknown loads exceeding 50% of its body weight.**

I. Introduction

Driven by significant hardware advances, quadrupedal robots are increasingly used in industrial applications and search and rescue missions [1]. These robots enhance productivity and safety by transporting heavy loads and navigating uneven surfaces. However, they rely on sensors like RGB-D cameras and LiDAR, which can be imperfect due to resolution limits, occlusions, and environmental complexities. Additionally, uncertainties from state-estimation errors, model mismatches, and varying payloads can lead to unintended contact locations, causing potential slippage or collisions. As shown in Section IV, classical MPC methods for quadrupedal locomotion [2] are insufficient at addressing these issues.

Stochastic Model Predictive Control (SMPC) incorporates probabilistic uncertainties into the control design. Unlike classical MPC, SMPC defines chance constraints using probabilistic models, ensuring constraints are met with a user-specified probability [3]. This approach balances control objectives and constraint satisfaction, crucial for high-performance operations near system limits [4]. The probabilistic framework also regulates the distribution of system states and control actions over the MPC horizon, maintaining robust performance and meeting safety requirements [5].

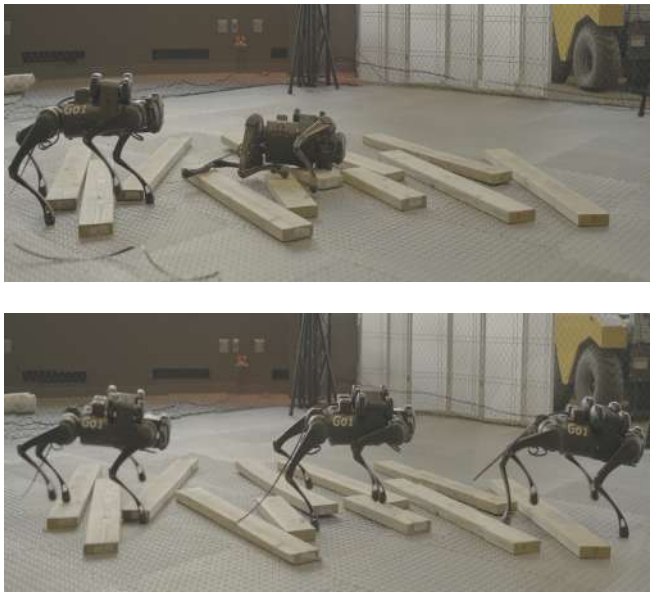


Fig. 1: Snapshots comparing LMPC (top) and CCMPC (bottom) as a quadrupedal robot navigates randomly placed wooden blocks while carrying an unmodeled 6 kg payload.

In this work, we develop an SMPC approach to generate optimal ground reaction forces for quadrupedal robots, enabling them to navigate challenging terrains and carry substantial loads. By treating mass, inertia, and contact sequences as stochastic variables and including additive noise in the dynamics, we capture the complexities of real-world environments. This approach is computationally efficient, adding minimal overhead compared to deterministic or heuristic MPC with hand-tuned friction cone constraints. Extensive simulations show our framework enables traversal over random wavefields, stairs, slopes, and different gaits more effectively than traditional methods. We validate our approach with similar hardware experiments on the torque-controlled Unitree Go1 robot, demonstrating its effectiveness in handling inertial and contact location uncertainties. The following section reviews related work and highlights the differences between existing methods and our approach.

A. Related Works

The utilization of convex MPC with the SRBD model has greatly enhanced the real-time implementation of diverse walking gaits [2]. However, these approaches require a highly

[1]Institute for Experiential Robotics, Northeastern University, Boston, Massachusetts, USA. {trivedi.ana, prajapati.s, m.zolotas, t.padir}@northeastern.edu

[2]Mark Zolotas is currently at Toyota Research Institute (TRI), Cambridge, MA, USA. This paper describes work performed at Northeastern University and is not associated with TRI.

[3]Taşkın Padır holds concurrent appointments as a Professor of Electrical and Computer Engineering at Northeastern University and as an Amazon Scholar. This paper describes work performed at Northeastern University and is not associated with Amazon.

accurate dynamics model, making them less effective when the real-world physics deviate from the designed controller model [6]. There have been efforts to account for uncertainties in trajectory optimization for legged locomotion using Differential Dynamic Programming (DDP) [7]. However, these approaches handle inequality constraints implicitly, making it difficult to address uncertainty impacts on constraint satisfaction [8]. Unlike DDP, our method uses chance constraints to address this issue explicitly.

Methods like Expected Residual Minimization (ERM) [9] and Stochastic Linear Complementarity Problem (SLCP) [10] aim to optimize stochastic contact-implicit trajectories. However, they are computationally demanding and prone to local minima, making them less suitable for real-time applications [11]. Conversely, our approach reformulates the stochastic optimal control problem into a deterministic convex quadratic programming (QP) problem, which can be solved efficiently [12].

In contrast to model-based control, model-free reinforcement learning (RL) techniques eliminate the need for an accurate robot model by using domain randomization during training [13]. This approach exposes the control policies to a wide range of scenarios, enhancing their robustness to diverse environments [14]. However, these controllers face challenges such as the sim-to-real gap, potentially leading to conservative strategies or failure to handle out-of-distribution disturbances [15]. They are also less intuitive to tune and require more engineering effort compared to model-based approaches [16]. Our method, with minimal parameter tuning, demonstrates strong generalization capabilities, as evidenced by extensive Monte Carlo simulations.

Adaptive MPC techniques serve as a middle ground between model-free RL and classical MPC approaches. Existing methods for quadrupedal locomotion estimate residual model uncertainties offline using simulation data [17] or online using L1 adaptive control [18]. However, proper initialization of estimated parameters is crucial to avoid instability before online model convergence [19]. Additionally, these methods assume constant uncertainty throughout the MPC horizon. By propagating state and control variance along the MPC horizon, our method captures evolving uncertainty, thereby enhancing prediction accuracy.

Recent work on bipedal locomotion has achieved robust walking in simulation by using tube MPC to account for additive polytopic uncertainties in the dynamics [20]. However, this approach assumed a constant center of mass (CoM) height and zero angular momentum, resulting in a limited range of movements. Xu et al. [6] addressed this limitation by employing robust min-max MPC for quadrupedal walking. However, they had to assign different parameters for each disturbance realization due to the inherent conservativeness of robust MPC (RMPC) approaches [21]. In contrast, SMPC offers greater flexibility than RMPC by allowing small, user-defined probabilities of constraint violation, which mitigates this issue without significantly sacrificing performance [3]. Despite its advantages, existing work on SMPC for quadrupedal robots is mostly confined to simulations and

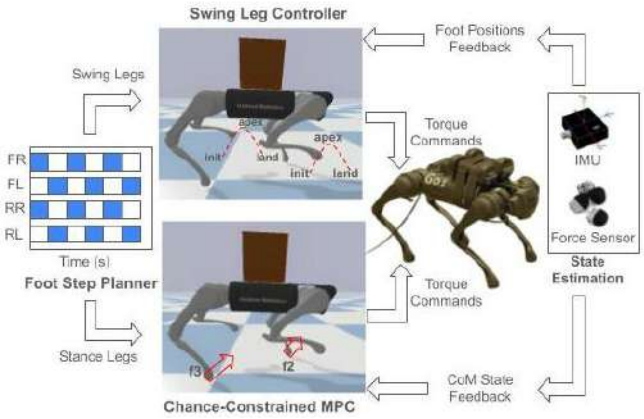


Fig. 2: A typical modular control architecture for quadrupedal locomotion.

is too computationally intensive for real-time deployment in a receding horizon MPC scheme [22]–[24]. In this paper, we overcome these challenges by deploying our approach on a hardware platform and leveraging a distribution of parameters to account for uncertainties, thereby avoiding the need for hand-tuning like in RMPC.

B. Contributions

The key contributions of this paper are:

1) We develop a chance-constrained model predictive control (CCMPC) algorithm that incorporates parametric uncertainties and additive disturbances into the SRBD model, ensuring consistent performance across terrains, gaits, and payloads.
2) Our method reformulates the stochastic optimal control problem into a deterministic convex quadratic programming problem. This achieves update rates comparable to linear MPC, supporting loads of more than 50% of the robot's weight while maintaining real-time performance with minimal computational overhead.
3) We validate our approach through extensive Monte Carlo simulations and hardware experiments on the Unitree Go1 robot. Our method demonstrates superior performance on varied terrains, including muddy slopes, stairs, grass, and gravel. To the best of our knowledge, this is the first implementation of SMPC for quadrupedal robots on hardware.

II. Chance-Constrained Foot Force MPC

As illustrated in Fig. 2, the footstep planner determines whether each foot should enter the swing (white) or stance (blue) phase. For the swing foot, the Raibert heuristic [2] calculates the required motor torques, planning the trajectory from initiation through apex to landing using a cubic spline. The chance-constrained MPC, which is the focus of this paper, optimizes ground reaction forces for the stance feet. The state estimator fuses IMU and foot force sensor data using an Extended Kalman Filter (EKF) to estimate the center of mass states and leg end positions.

The whole-body dynamics can be expressed using manipulator equations [25] as follows:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}^T \boldsymbol{\tau}_{\text{motors}} + \mathbf{J}(\mathbf{q})^T \mathbf{f}_{\text{contact}} \quad (1)$$

Here, $\mathbf{q}$ represents the generalized robot pose, $\mathbf{M}$ is the mass matrix, and $\mathbf{N}$ encapsulates other non-linear effects. The selection matrix $\mathbf{S}$ captures the actuation from the 12 motors on board the Unitree Go1 robot. As discussed in [26], the foot contact forces $\mathbf{f}_{\text{contact}}$ can be mapped to desired motor torques $\boldsymbol{\tau}_{\text{motors}}$ via the contact Jacobian $\mathbf{J}(\mathbf{q})$. MPC is an effective strategy to determine these foot forces due to its ability to solve constrained optimal control problems [25]. However, MPC struggles under significant modeling errors due to diverse payloads and terrains. This often leads to instability and deviations from desired trajectories [6].

To address this, we present the stochastic variation of the SRBD model [2] for quadrupedal robots. The state vector $\mathbf{x} = [\boldsymbol{\Theta}^T, \mathbf{p}^T, \boldsymbol{\omega}^T, \dot{\mathbf{p}}^T, g]^T$ includes the robot's orientation, position, velocities, and gravity term. The control inputs are the ground reaction forces at each leg, $\mathbf{u} = [\mathbf{f}_1^T, \mathbf{f}_2^T, \mathbf{f}_3^T, \mathbf{f}_4^T]^T$. At time instant i, the equations of motion are thus expressed as:

$$\mathbf{x}_{i+1} = \mathbf{A}_i \mathbf{x}_i + \mathbf{B}_i(\boldsymbol{\delta}_i)\mathbf{u}_i + \mathbf{w}_i, \quad (2)$$

$$\text{where} \quad \boldsymbol{\delta}_i \sim \mathcal{N}(\mathbb{E}[\boldsymbol{\delta}_i], \boldsymbol{\Sigma}_{\boldsymbol{\delta}}), \quad \mathbf{w}_i \sim \mathcal{N}(\mathbb{E}[\mathbf{w}], \boldsymbol{\Sigma}_{\mathbf{w}})$$

Here, $\mathbb{E}[\boldsymbol{\delta}_i] = \overline{\boldsymbol{\delta}_i} = [m, \text{diag}(\mathbf{I}), \mathbf{r}_{i,1}^T, \ldots, \mathbf{r}_{i,4}^T]$ represents the mean of the distribution of parametric uncertainties, where m is the nominal robot mass, and $\text{diag}(\mathbf{I})$ represents the diagonal entries of the nominal inertia matrix, respectively. At time instant i, $\mathbf{r}_{i,1}, \mathbf{r}_{i,2}, \mathbf{r}_{i,3}$, and $\mathbf{r}_{i,4}$ are the foot locations planned using the Raibert heuristic as shown in Fig. 2. $\boldsymbol{\Sigma}_{\boldsymbol{\delta}}$ is the user-tunable covariance matrix representing the variations in these parameters due to unknown payloads and uneven terrain. Similarly, $\mathbb{E}[\mathbf{w}] = \mathbf{0}$ and $\boldsymbol{\Sigma}_{\mathbf{w}}$ are the mean and covariance matrix of the residual model nonlinearities [17] not accounted for by the nominal SRBD model.

To manage these uncertainties, we use CCMPC, which incorporates probabilistic descriptions of uncertainties into a stochastic optimal control problem [5]. With uncertainties, guaranteeing constraint satisfaction at all times is impractical. Chance constraints ensure state and control constraints are satisfied with a specified probability, allowing controlled levels of constraint violation [27]. This balance between performance and robustness is crucial for stability and safety in uncertain environments.

Specifically, we enforce the linearized friction cone constraints and the unilateral force constraints as chance constraints on the ground reaction forces. Mathematically, these constraints can be expressed as:

$$\Pr(\mathbf{C}_i \mathbf{u}_i \leq \mathbf{0}) \geq \epsilon \quad (3)$$

Here, $\mathbf{C}_i \in \mathbb{R}^{20\times 12}$ encapsulates the friction cone and unilateral force constraints, and ϵ represents the acceptable thresholds of constraint satisfaction. Additionally, we aim to minimize the expected cost of deviation from a desired CoM trajectory and the norm of control effort:

$$\mathbb{E}[J(\mathbf{x}, \mathbf{u})] = \mathbb{E}\left[\sum_{i=0}^{N-1} \|\mathbf{x}_{i+1} - \mathbf{x}_{\text{ref},i+1}\|_{\mathbf{Q}}^2 + \|\mathbf{u}_i\|_{\mathbf{R}}^2\right] \quad (4)$$

Over the MPC horizon N, the positive semi-definite matrix $\mathbf{Q}$ weighs the tracking error between the state $\mathbf{x}$ and the reference state $\mathbf{x}_{\text{ref}}$, and the positive definite matrix $\mathbf{R}$ weighs the control effort. Finally, we consider the following constraint, which ensures that forces from legs not in contact with the ground are zeroed out:

$$\mathbf{D}_i \mathbf{u}_i = 0 \quad (5)$$

The resultant form of the chance-constrained foot force MPC is thus expressed as:

$$\begin{aligned} &\underset{\mathbf{x}_i, \mathbf{u}_i}{\text{minimize}} \quad \mathcal{L}^S(\mathbf{x}_i, \mathbf{u}_i) \\ &\text{subject to} \quad \mathbf{g}_{\text{ineq}}^S(\mathbf{u}_i) \leq 0 \text{ and } \mathbf{h}_{\text{eq}}^S(\mathbf{x}_i, \mathbf{u}_i) = 0 \end{aligned} \quad (6)$$

Here, $\mathcal{L}^S(\mathbf{x}_i, \mathbf{u}_i)$ is the stochastic cost function (Equation 4), $\mathbf{g}_{\text{ineq}}^S(\mathbf{u}_i)$ represents the friction cone chance-constraints (Equation 3), and $\mathbf{h}_{\text{eq}}^S(\mathbf{x}_i, \mathbf{u}_i)$ represents the stochastic SRBD model (Equation 2) and the non-contact force constraint (Equation 5).

III. Convex QP Reformulation of CCMPC

Solving the original CCMPC problem, as expressed in Equation 6, is computationally infeasible due to the stochastic dynamics and the integration of multi-dimensional Gaussian probability density functions required to resolve the chance constraints [4]. In this section, we derive an efficient deterministic reformulation of the stochastic optimal control problem.

A. Uncertainty Propagation

Due to the parametric uncertainties in the system dynamics and additive disturbances, future predicted states result in a stochastic distribution. Similar to [23], we parameterize the control law as a state-dependent feedback policy. As a result, future control actions also exhibit uncertainty. Specifically, the control law is of the form:

$$\mathbf{u}_i = \mathbf{v}_i + \mathbf{K}_i(\mathbf{x}_i - \bar{\mathbf{x}}_i) \quad (7)$$

Here, $\mathbf{v}_i$ is the feedforward control input and $\bar{\mathbf{x}}_i$ is the predicted mean of the state distribution. The feedforward action, $\mathbf{v}_i$, provides the primary control effort to achieve the desired trajectory based on the predicted system behavior. The feedback action, $\mathbf{K}_i(\mathbf{x}_i - \bar{\mathbf{x}}_i)$, adjusts for any deviations from the predicted trajectory due to disturbances or model inaccuracies. Optimizing both the feedback gains $\mathbf{K}_i$ and the control inputs $\mathbf{v}_i$ leads to a bi-level optimization problem, which is non-convex and computationally expensive for real-time MPC applications [28]. Instead, we precompute the gains $\mathbf{K}_i$ using efficient solvers for the Discrete Algebraic Riccati Equation (DARE) [29]. Thus, the decision variables of our optimization problem are $\mathbf{v}_i$ and $\bar{\mathbf{x}}_i$.

Next, we present the expressions for the first and second moments of the trajectory distributions. For detailed derivations, interested readers are referred to [5]. Applying the expectation operator to Equation 2, we get:

$$\mathbb{E}[\mathbf{x}_{i+1}] = \bar{\mathbf{x}}_{i+1} = \mathbf{A}_i\bar{\mathbf{x}}_i + \mathbf{B}_i(\bar{\boldsymbol{\delta}}_i)\bar{\mathbf{u}}_i \tag{8}$$

The covariance of the state distribution at the next time step can be expressed as:

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{x}_{i+1}} &= \mathbb{E}[(\mathbf{x}_{i+1} - \bar{\mathbf{x}}_{i+1})(\mathbf{x}_{i+1} - \bar{\mathbf{x}}_{i+1})^T] \\ &= \mathbf{A}_{cl}\boldsymbol{\Sigma}_{\mathbf{x}_i}\mathbf{A}_{cl}^T + \mathbf{P}_i\boldsymbol{\Sigma}_{\boldsymbol{\delta}}\mathbf{P}_i^T + \boldsymbol{\Sigma}_{\mathbf{w}}\end{aligned} \tag{9}$$

Here, $\mathbf{A}_{cl} = \mathbf{A}_i + \mathbf{B}_i(\bar{\boldsymbol{\delta}}_i)\mathbf{K}_i$ represents the closed-loop system dynamics, and $\mathbf{P}_i$ is the Jacobian of $\bar{\mathbf{x}}_{i+1}$ with respect to the mean of the parameters $\bar{\boldsymbol{\delta}}_i$:

$$\mathbf{P}_i = \frac{\partial\bar{\mathbf{x}}_{i+1}}{\partial\bar{\boldsymbol{\delta}}_i} = \frac{\partial(\mathbf{A}_i\bar{\mathbf{x}}_i + \mathbf{B}_i(\bar{\boldsymbol{\delta}}_i)\bar{\mathbf{u}}_i)}{\partial\bar{\boldsymbol{\delta}}_i} \tag{10}$$

Based on these equations, the mean and covariance of the control distribution can be expressed as:

$$\mathbb{E}[\mathbf{u}_i] = \mathbf{v}_i \quad \text{and} \quad \boldsymbol{\Sigma}_{\mathbf{u}_i} = \mathbf{K}_i\boldsymbol{\Sigma}_{\mathbf{x}_i}\mathbf{K}_i^T \tag{11}$$

We now show how the simplicity offered by using these Gaussian-distributed trajectories allows us to analytically derive a deterministic counterpart of the chance control constraints.

B. Friction Cone Constraint Tightening

To ensure there is no slippage between the robot's feet and the ground, the reaction forces must stay within the linearized friction pyramid and satisfy the unilaterality constraint [30]. This results in five chance-constraints per foot that must be jointly satisfied [23]. The Boole-Bonferroni inequality [31] allows us to conservatively approximate the probability of satisfying these joint chance constraints by summing the probabilities of each individual constraint for all $n = 4$ feet.

$$\sum_{j=1}^{5n} \Pr(\mathbf{C}_i^j\mathbf{u}_i > 0) \leq 1 - \epsilon, \implies \text{Equation 3} \tag{12}$$

where, $\mathbf{C}_i^j$ is the j^{th} row of the $\mathbf{C}_i$ matrix.

While optimizing risk allocation for each constraint could be more effective, it involves a two-stage optimization problem which can be computationally expensive [32]. To get around this, we assign uniform risks α to each constraint where, $\alpha = (1 - \epsilon)/5n$. Effectively Equation 12, becomes:

$$\Pr(\mathbf{C}_i^j\mathbf{u}_i > 0) \leq \alpha \implies \Pr(\mathbf{C}_i^j\mathbf{u}_i \leq 0) \leq 1 - \alpha \tag{13}$$

Finally, since the control actions follow a normal distribution, $\mathbf{u}_i \sim \mathcal{N}(\mathbf{v}_i, \boldsymbol{\Sigma}_i^u)$, individual chance constraints in Equation 13 can be deterministically reformulated as follows [33]:

$$\mathbf{C}_i^j\mathbf{v}_i + \phi^{-1}(1-\alpha)\sqrt{\mathbf{C}_i^j\,\boldsymbol{\Sigma}_i^u\,(\mathbf{C}_i^j)^T} \leq 0 \tag{14}$$

where ϕ^{-1} is the inverse of the cumulative distribution function of a standard normal distribution [31]. Applying Equation 14 to all rows of $\mathbf{C}_i$, we get the following constraints on the mean of control actions:

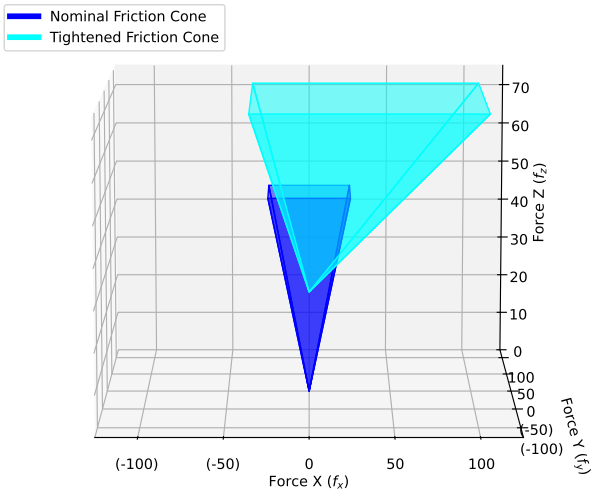


Fig. 3: Illustration of friction cone constraint tightening.

$$\mathbf{C}_i\mathbf{v}_i \leq \mathbf{c}_i, \tag{15}$$

$\mathbf{c}_i = \text{col}(c_i^1, \ldots, c_i^{5n})$ and $c_i^k = -\phi^{-1}(1-\alpha)\sqrt{\mathbf{C}_i^k\,\boldsymbol{\Sigma}_i^u\,(\mathbf{C}_i^k)^T}$

This modification of the original friction cone constraints based on the computed $\mathbf{c}_i$ values in Eqn. 15 is commonly referred to as constraint tightening [3]. Its magnitude is directly correlated to the constraint satisfaction threshold ϵ and the control covariance propagated along the MPC horizon $\boldsymbol{\Sigma}_i^u$. Fig. 3 illustrates the effect of constraint tightening on the feasible contact force set in MPC. The nominal cone depicts the standard force limits, while the distorted cone represents the tightened constraints, indicating a more conservative and restrictive set of allowable forces.

Alternatively, we could have also heuristically chosen a constant value for $\mathbf{c}_i$ [23]. In Section IV, we demonstrate how this heuristic-based approach results in increased violations of the friction cone constraints, leading to greater instability. Despite this added computation in chance-constrained MPC, the resultant stance-leg control problem can still be solved efficiently at 500 Hz, suitable for real-time implementation.

C. Reduction to Quadratic Program

In this section, we consolidate the setup of the MPC optimization problem. Following the approach in [34], we use a mean-equivalent approximation of the expected cost along the nominal trajectory:

$$\text{Eqn. 4} \approx J(\bar{\mathbf{x}}, \mathbf{v}) = \sum_{i=0}^{N-1} \|\bar{\mathbf{x}}_{i+1} - \mathbf{x}_{\text{ref},i+1}\|_{\mathbf{Q}}^2 + \|\mathbf{v}_i\|_{\mathbf{R}}^2 \tag{16}$$

Finally, we apply the constraint Eqn. 5 to the mean of the contact forces, that is

$$\mathbb{E}[\mathbf{D}_i \mathbf{u}_i] = \mathbf{D}_i \mathbf{v}_i = 0 \tag{17}$$

For real-time application, we base the friction cone constraint tightening on the previous MPC solution, allowing these factors to be pre-computed and fixed during the current optimization [4].

In summary, given a nominal contact plan and a desired CoM trajectory, CCMPC involves solving the quadratic programming problem below

$$\begin{aligned} \{\bar{\mathbf{x}}^*, \mathbf{v}^*\} = \underset{\bar{\mathbf{x}}_i, \mathbf{v}_i}{\text{minimize}} \quad & \mathcal{L}^D(\bar{\mathbf{x}}_i, \mathbf{v}_i) \\ \text{subject to} \quad & \mathbf{g}^D_{\text{ineq}}(\mathbf{v}_i) \leq 0 \text{ and } \mathbf{h}^D_{\text{eq}}(\bar{\mathbf{x}}_i, \mathbf{v}_i) = 0 \end{aligned} \tag{18}$$

Here, $\mathcal{L}^D(\bar{\mathbf{x}}_i, \mathbf{v}_i)$ is the quadratic cost function (Equation 16), $\mathbf{g}^D_{\text{ineq}}(\mathbf{v}_i)$ represents the linearized friction cone constraints (Equation 15), and $\mathbf{h}^D_{\text{eq}}(\bar{\mathbf{x}}_i, \mathbf{v}_i)$ represents the two linear equality constraints, namely the dynamics constraint (Equation 8) and non-contact force constraint (Equation 17). Using single shooting [35], we further reduce the number of decision variables in Eqn. 18 to only include the mean of the control actions. The resultant optimization problem is solved using the qpOASES solver [12] . The first element of the optimal ground reaction force sequence $\mathbf{v}^*$ is then converted to desired foot motor torques using the contact Jacobian, $\mathbf{J}(\mathbf{q})$. Algorithm 1 outlines the steps necessary to implement the proposed stance-leg controller for quadrupedal locomotion.

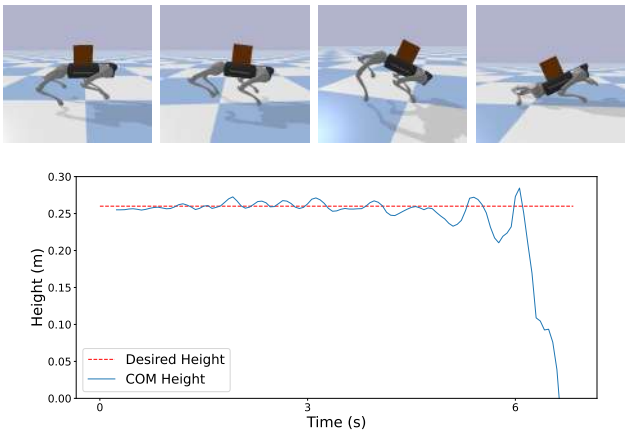
Algorithm 1 Chance-Constrained MPC

```
Initialize c_i ← 0
while goal configuration is not reached, do
    for i = 0 to N − 1 do                          ▷ MPC Loop
        Compute δ̄_i, A_i, B_i(δ̄_i), D_i using Eqn. 2
        Add cost function and constraints to Eqn. 18
    end for

    Output: x̄*, v* ← Solve Eqn. 18                 ▷ MPC Output

    Initialize Σ_x ← 0                  ▷ Constraint Tightening Loop
    for i = 0 to N − 1 do
        K_i ← DARE(A_i, B_i(δ̄_i), Q, R)
        Σ_u ← Propagate Control Variance, Eqn. 11
        c_i ← Compute Factors, Eqn. 15
        Σ_x ← Propagate State Variance, Eqns. 9, 10
    end for
end while
```

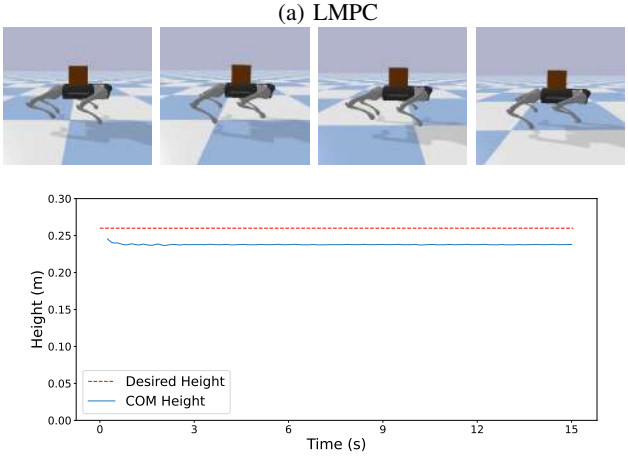
IV. Experiments

A. Implementation Details

We validate our approach on the Unitree Go1-Edu robot. The legs are relatively massless in comparison to the torso, supporting the assumptions of the linearized SRBD model [2]. However, this introduces modeling uncertainties which we mitigate to some extent using CCMPC.



(a) LMPC



(b) CCMPC

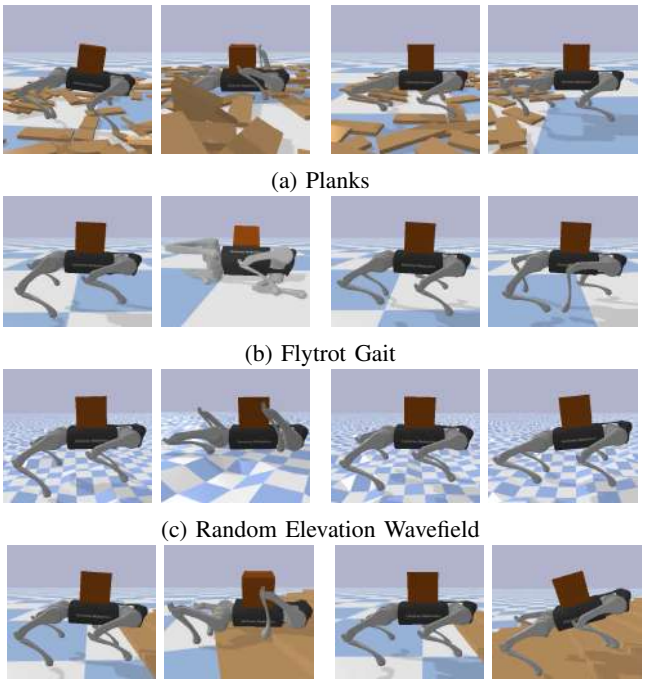
Fig. 4: Simulated height tracking performance with an unmodeled 6 kg payload.

The robot's pose and velocities are obtained using onboard IMU and motor encoders. All algorithms are executed on a Legion 5 Pro laptop with an Intel i7-12700H processor and 32 GB RAM. Communication between the robot and the laptop is facilitated using Lightweight Communications and Marshalling (LCM) [36]. In the following sections, we present detailed results from our simulation and hardware experiments.

B. Simulation Analysis

All simulation experiments are conducted using the PyBullet simulator [37] with a high-fidelity dynamics model provided by the robot supplier. We set a single set of parameters across all the experiments in this section, demonstrating the versatility of our approach. These hyperparameters include state and control penalty weights for both the MPC and DARE solvers, foot height, stepping frequency, robot mass, and inertia, as well as the covariance of parametric and additive uncertainties.

We conduct 1000 Monte Carlo simulations to demonstrate that our control policy can stabilize quadrupedal motion across various payloads and terrains without parameter ad-



(a) Planks

(b) Flytrot Gait

(c) Random Elevation Wavefield

(d) Blind Stair Climbing

Fig. 5: Comparison of LMPC (left two images, fails) and CCMPC (right two images, succeeds) across various gaits and terrains.

justments.We compare our CCMPC against two benchmarks: Linear MPC (LMPC), which lacks friction cone constraint tightening, and Heuristic MPC (HMPC), which uses hand-computed constraint tightening factors.

In all these experiments, the robot was commanded to move forward at 0.25 m/s, performing blind locomotion over wooden planks with varying payloads. The payloads (1 to 10 kg) and plank heights (0 to 5 cm) were sampled from uniform distributions, introducing variability in inertial properties and contact locations. Given the robot's mass of 12 kg and a foot raise height of 8 cm, these values represent significant variations from nominal conditions. For heuristic constraint tightening, we considered a maximum unmodeled payload of 10 kg and calculated the necessary gravity compensation in the z direction from the stance feet. To accommodate the maximum commanded acceleration of 0.2 m/s^2, we ensured the robot could generate sufficient force in both the x and y directions, even with the additional payload. This method allowed us to heuristically determine the necessary constraint tightening factors for all directions. For CCMPC the constraint satisfaction threshold ϵ was set to 0.95, ensuring that the chance constraints were respected more than 95% of the time.

We assessed the performance of the controllers using three primary metrics, logged in Tables I, II, and III. The success rate (Table I) was determined based on several failure criteria: deviations in the height of the base by more than 30% from its desired value, the orientation of the robot with respect to the ground normal dropping below 0.8 (indicating a significant tilt), or any infeasibility in solving the QP due to constraint violations. These thresholds were identified during hardware tuning as the points where the robot physically fell. The slippage ratio (Table II) was calculated as $(f_x^2+f_y^2)/f_z^2$, representing the degree to which the friction cone was saturated, with higher values indicating worse performance. For the normalized MPC cost (Table III), CCMPC was normalized to 1, and the other methods were compared relative to it; a higher cost indicates increased actuation and poorer tracking of the desired CoM trajectory. Both the slippage ratio and the normalized MPC cost (Table III) were obtained by averaging their respective values over all iterations of the simulations. Our results demonstrate that LMPC performed the worst, HMPC showed improvements due to heuristic constraint tightening, but CCMPC, with its adaptive constraint tightening, consistently achieved the best results.

TABLE I: Success Rate

Gait	LMPC	HMPC	CCMPC
Trot	38.1%	77.7%	**100%**
Flytrot	40.3%	73.7%	**100%**

TABLE II: Slippage Ratio

Gait	LMPC	HMPC	CCMPC
Trot	0.26	0.12	**0.11**
Flytrot	0.25	0.17	**0.15**

TABLE III: Normalized MPC Cost

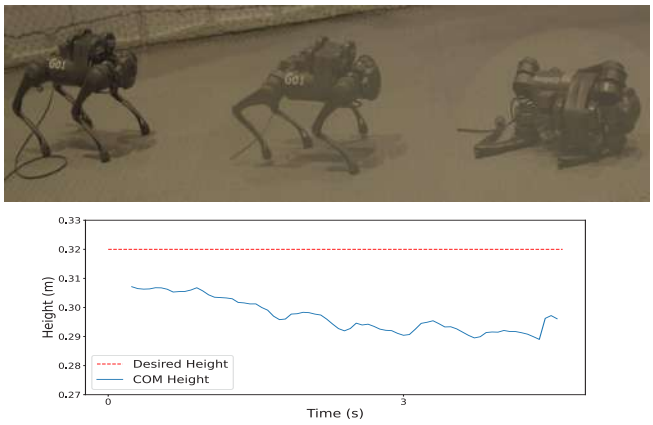
Gait	LMPC	HMPC	CCMPC
Trot	3.30	1.37	**1.0**
Flytrot	5.06	3.96	**1.0**

As shown in Fig. 5, in addition to the Monte Carlo experiments, we conducted simulations on various challenging terrains. The robot traversed an uneven terrain with randomly varied ground heights, navigated slopes, climbed stairs, and performed height tracking tests. For all these experiments, the robot performed blind locomotion while carrying a fixed payload of 6 kg at a commanded velocity of 0.25 m/s. Additionally, we conducted high-speed tests at 1.75 m/s using the flytrot gait, which has an underactuated phase with all four feet in the air. Video demonstrations of these experiments can be found in the supplemental video. These tests showed that while the robot fell using LMPC, it successfully maintained stability and tracking with CCMPC.

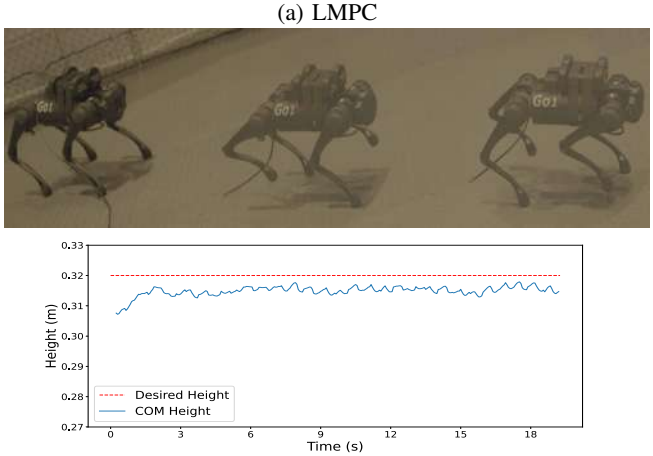
C. Hardware Validation

To validate our approach on hardware, we repeated all the simulation experiments using the real robot. We kept the parameters consistent across all hardware experiments, demonstrating the robustness of our control strategy without the need for parameter adjustments.

The first key hardware experiment, illustrated in Fig. 1, involved loading the robot with a 6 kg payload and commanding it to walk over randomly placed wooden planks at a speed of 0.5 m/s. This experiment was conducted under blind locomotion conditions, where the controller was unaware of the existence of the planks or the additional mass. While LMPC failed to navigate this terrain, CCMPC successfully guided the robot across the planks.



(a) LMPC



(b) CCMPC

Fig. 6: Hardware height tracking performance with an unmodeled 6 kg payload.

To demonstrate the repeatability of our control strategy, we conducted a series of progressive load tests, as shown in Fig. 6. In these experiments, the robot was commanded to move forward at a speed of 0.25 m/s while carrying increasing payloads from 0 kg up to 6 kg. The results showed that while LMPC failed to navigate when the payload exceeded 3 kg, CCMPC maintained stability and successfully navigated the terrain with all tested payloads.

We demonstrated the versatility of our control algorithm through various challenging terrain experiments, as illustrated in Fig. 7. The robot, loaded with additional unknown mass, successfully navigated muddy slopes, grass, and gravel, and climbed stairs. The robot also performed tasks such as pushing and pulling an unknown 5 kg payload. In another experiment, the robot walked over a whiteboard coated with cooking oil, significantly reducing the friction between the robot's feet and the surface. For this test, we reduced the robot's speed to 0.1 m/s and adjusted the coefficient of friction in the MPC from 0.4 to 0.2 to account for the slippery surface. Despite these adverse conditions, the robot successfully navigated all these surfaces, showcasing its ability to handle diverse ground textures and maintain stability on slippery and uneven surfaces.

More hardware demonstrations can be found in the supplemental video, which provides a comprehensive overview of the robot's performance across these varied conditions. These experiments highlight the effectiveness of our CCMPC algorithm in handling a wide range of scenarios and maintaining stable locomotion across various conditions.

V. Conclusion

In this paper, we presented a Chance-Constrained Model Predictive Control framework for quadrupedal locomotion aimed at enhancing stability and adaptability in varied and challenging terrains. Our approach combines stochastic optimal control with real-time implementation, ensuring robust performance despite payload and terrain uncertainties. Extensive simulations and hardware experiments demonstrated that our formulation outperforms Linear MPC (LMPC) and Heuristic MPC (HMPC) in terms of stability, success rate, and foot slippage. The ability to manage a wide range of payloads and terrains without parameter adjustments highlights the versatility of our method. Our experiments included diverse scenarios, such as uneven terrains, slopes, stairs, and high-speed gaits. The results confirm that our approach maintains stable locomotion across various conditions, showcasing its potential for practical real-world applications.

However, some limitations were observed, particularly with heavier weights. Merely tightening control constraints in anticipation of added mass, rather than estimating its dynamic effects, can render the QP infeasible. Future work could explore Bayesian machine learning methods, as proposed in [38], [39], to estimate uncertainties and adjust constraints accordingly. Additionally, future work will focus on extending this approach to loco-manipulation tasks for bipedal robots handling unknown payloads [40].

References

[1] P. Biswal and P. K. Mohanty, "Development of quadruped walking robots: A review," *Ain Shams Engineering Journal*, vol. 12, no. 2, pp. 2017–2031, 2021.

[2] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 1–9.

[3] A. Mesbah, "Stochastic model predictive control: An overview and perspectives for future research," *IEEE Control Systems Magazine*, vol. 36, no. 6, pp. 30–44, 2016.

[4] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using gaussian process regression," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2736–2743, 2019.

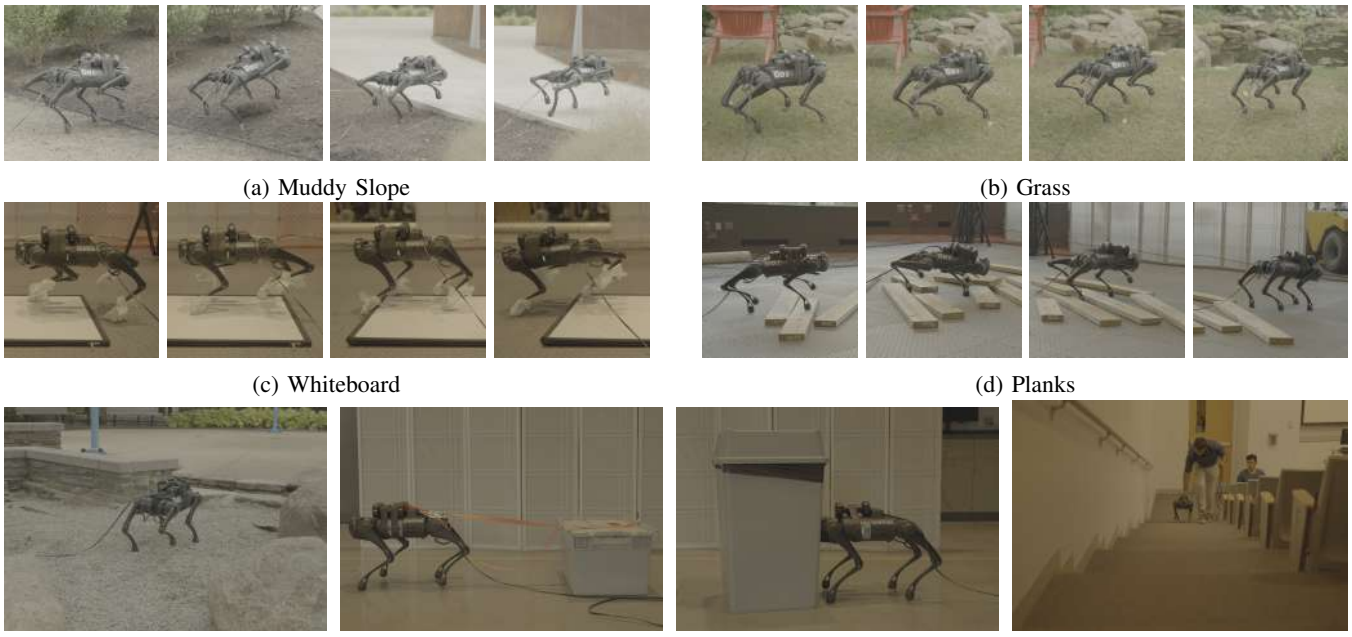
[5] T. Lew, R. Bonalli, and M. Pavone, "Chance-constrained sequential convex programming for robust trajectory optimization," in *2020 European Control Conference (ECC)*. IEEE, 2020, pp. 1871–1878.

[6] S. Xu, L. Zhu, H.-T. Zhang, and C. P. Ho, "Robust convex model predictive control for quadruped locomotion under uncertainties," *IEEE Transactions on Robotics*, 2023.

[7] B. Hammoud, M. Khadiv, and L. Righetti, "Impedance optimization for uncertain contact interactions through risk sensitive optimal control," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4766–4773, 2021.

[8] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1168–1175.

[9] L. Drnach, J. Z. Zhang, and Y. Zhao, "Mediating between contact feasibility and robustness of trajectory optimization through chance complementarity constraints," *Frontiers in Robotics and AI*, vol. 8, p. 785925, 2022.



(a) Muddy Slope (b) Grass

(c) Whiteboard (d) Planks

(e) Gravel (f) Pulling Unmodeled Box (g) Pushing Unmodeled Box (h) Blind Stair Climbing

Fig. 7: Hardware validation of CCMPC across different locomotion tasks and terrains.

[10] Y. Shirai, D. K. Jha, A. U. Raghunathan, and D. Romeres, "Chance-constrained optimization in contact-rich systems," in *2023 American Control Conference (ACC)*. IEEE, 2023, pp. 14–21.

[11] Y. Shirai, D. K. Jha, and A. U. Raghunathan, "Covariance steering for uncertain contact-rich systems," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7923–7929.

[12] 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*, vol. 6, pp. 327–363, 2014.

[13] J. Lee, M. Bjelonic, A. Reske, L. Wellhausen, T. Miki, and M. Hutter, "Learning robust autonomous navigation and locomotion for wheeled-legged robots," *Science Robotics*, vol. 9, no. 89, p. eadi9641, 2024.

[14] S. Chen, B. Zhang, M. W. Mueller, A. Rai, and K. Sreenath, "Learning torque control for quadrupedal locomotion," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*. IEEE, 2023, pp. 1–8.

[15] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: a survey," in *2020 IEEE symposium series on computational intelligence (SSCI)*. IEEE, 2020, pp. 737–744.

[16] T. Eimer, M. Lindauer, and R. Raileanu, "Hyperparameters in reinforcement learning and how to tune them," in *International Conference on Machine Learning*. PMLR, 2023, pp. 9104–9149.

[17] A. Pandala, R. T. Fawcett, U. Rosolia, A. D. Ames, and K. A. Hamed, "Robust predictive control for quadrupedal locomotion: Learning to close the gap between reduced-and full-order models," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6622–6629, 2022.

[18] M. Sombolestan and Q. Nguyen, "Adaptive force-based control of dynamic legged locomotion over uneven terrain," *IEEE Transactions on Robotics*, 2024.

[19] I. D. Landau, R. Lozano, M. M'Saad *et al.*, *Adaptive control*. Springer New York, 1998, vol. 51.

[20] A. Gazar, M. Khadiv, A. Del Prete, and L. Righetti, "Stochastic and robust mpc for bipedal locomotion: A comparative study on robustness and performance," in *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2021, pp. 61–68.

[21] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in identification and control*. Springer, 2007, pp. 207–226.

[22] S. Katayama, N. Takasugi, M. Kaneko, N. Nagatsuka *et al.*, "Robust locomotion via zero-order stochastic nonlinear model predictive control with guard saltation matrix," *arXiv preprint arXiv:2403.14159*, 2024.

[23] A. Gazar, M. Khadiv, A. Del Prete, and L. Righetti, "Multi-contact stochastic predictive control for legged robots with contact locations uncertainty," *arXiv preprint arXiv:2309.04469*, 2023.

[24] A. Gazar, M. Khadiv, S. Kleff, A. Del Prete, and L. Righetti, "Nonlinear stochastic trajectory optimization for centroidal momentum motion generation of legged robots," in *The International Symposium of Robotics Research*. Springer, 2022, pp. 420–435.

[25] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. Del Prete, "Optimization-based control for dynamic legged robots," *IEEE Transactions on Robotics*, 2023.

[26] A. Herzog, S. Schaal, and L. Righetti, "Structured contact force optimization for kino-dynamic motion generation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2703–2710.

[27] A. Majumdar and M. Pavone, "How should a robot assess risk? towards an axiomatic theory of risk in robotics," in *Robotics Research: The 18th International Symposium ISRR*. Springer, 2020, pp. 75–84.

[28] S. Dempe and A. Zemkoho, "Bilevel optimization," in *Springer optimization and its applications*. Springer, 2020, vol. 161.

[29] R. Tedrake *et al.*, "Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems," 2014.

[30] P.-B. Wieber, R. Tedrake, and S. Kuindersma, "Modeling and control of legged robots," in *Springer handbook of robotics*. Springer, 2016, pp. 1203–1234.

[31] A. Prékopa, *Stochastic programming*. Springer Science & Business Media, 2013, vol. 324.

[32] J. Pilipovsky and P. Tsiotras, "Chance-constrained optimal covariance steering with iterative risk allocation," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 2011–2016.

[33] G. C. Calafiore and L. E. Ghaoui, "On distributionally robust chance-constrained linear programs," *Journal of Optimization Theory and Applications*, vol. 130, pp. 1–22, 2006.

[34] L. Hewing and M. N. Zeilinger, "Stochastic model predictive control for linear systems using probabilistic reachable sets," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 5182–5188.

[35] J. B. Rawlings, D. Q. Mayne, M. Diehl *et al.*, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.

[36] A. S. Huang, E. Olson, and D. C. Moore, "Lcm: Lightweight communications and marshalling," in *2010 IEEE/RSJ International*

Conference on Intelligent Robots and Systems. IEEE, 2010, pp. 4057–4062.

[37] E. Coumans and Y. Bai, "Pybullet quickstart guide," 2021.

[38] A. Trivedi, S. Bazzi, M. Zolotas, and T. Padır, "Probabilistic dynamic modeling and control for skid-steered mobile robots in off-road environments," in *2023 IEEE International Conference on Assured Autonomy (ICAA)*. IEEE, 2023, pp. 57–60.

[39] A. Trivedi, M. Zolotas, A. Abbas, S. Prajapati, S. Bazzi, and T. Padır, "A probabilistic motion model for skid-steer wheeled mobile robot navigation on off-road terrains," *arXiv preprint arXiv:2402.18065*, 2024.

[40] J. Li, J. Ma, O. Kolt, M. Shah, and Q. Nguyen, "Dynamic loco-manipulation on hector: Humanoid for enhanced control and open-source research," *arXiv preprint arXiv:2312.11868*, 2023.