IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Chance-Constrained Convex MPC for Robust Quadruped Locomotion Under Parametric and Additive Uncertainties

Ananya Trivedi, *Graduate Student Member, IEEE*, Sarvesh Prajapati, *Member, IEEE*, Mark Zolotas, *Member, IEEE*, Michael Everett, *Member, IEEE*, and Taşkın Padır, *Senior Member, IEEE*

***Abstract*—Recent advances in quadrupedal locomotion have focused on improving stability and performance across diverse environments. However, existing methods often lack adequate safety analysis and struggle to adapt to varying payloads and complex terrains, typically requiring extensive tuning. To overcome these challenges, we propose a Chance-Constrained Model Predictive Control (CCMPC) framework that explicitly models payload and terrain variability as distributions of parametric and additive disturbances within the single rigid body dynamics model. Our approach ensures safe and consistent performance under uncertain dynamics by expressing the model's friction cone constraints, which define the feasible set of ground reaction forces, as chance constraints. Moreover, we solve the resulting stochastic control problem using a computationally efficient quadratic programming formulation. Extensive Monte Carlo simulations of quadrupedal locomotion across varying payloads and complex terrains demonstrate that CCMPC significantly outperforms two competitive benchmarks: Linear MPC and MPC with hand-tuned safety margins to maintain stability, reduce foot slippage, and track the center of mass. Hardware experiments on the Unitree Go1 robot show successful locomotion across various indoor and outdoor terrains with unknown loads exceeding 50% of the robot's body weight, despite no additional parameter tuning.**

***Index Terms*—Robust/adaptive control, planning under uncertainty, legged robots, optimization and optimal control.**

I. Introduction

QUADRUPEDAL robots have demonstrated significant potential in various industrial applications and search and rescue missions [1]. However, preventing falls in dynamic environments remains a critical challenge [2]. Inaccurate system models, external disturbances, and unpredictable payload variations can cause deviations from planned motions, resulting in contact location or timing errors [3]. As shown in Fig. 1, classical Model Predictive Control (MPC) methods struggle with these discrepancies, leading to instability, foot slippage, and eventually falls.

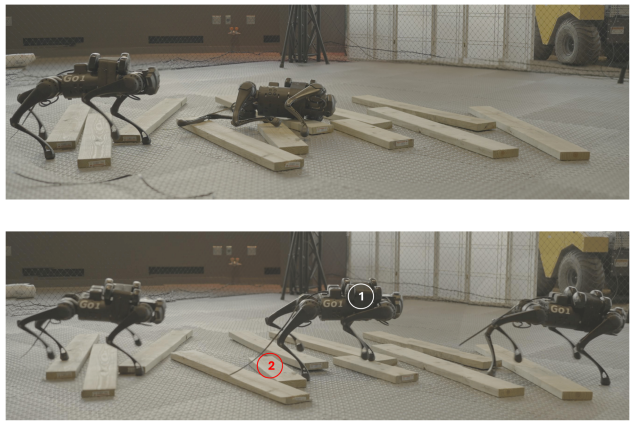


Fig. 1. Chance-Constrained MPC (bottom) stabilizes the robot despite inertial uncertainty from 6 kg dumbbells (1) and contact uncertainty from planks (2). Linear MPC (top), however, relies purely on feedback and lacks the anticipatory behavior needed to maintain balance under these conditions.

Optimization-based control has driven significant progress in quadrupedal locomotion, but often relies on accurate robot models and well-characterized contact dynamics [4]. Model-free reinforcement learning (RL) techniques aim to overcome these limitations by training control policies that can generalize across diverse terrains and robot configurations [2]. While these approaches have demonstrated strong empirical performance, they often lack interpretability and may require extensive re-training for deployment in new environments [5].

Stochastic MPC (SMPC) methods offer a middle ground by incorporating probability distributions of uncertainty into model-based control design [6]. Unlike deterministic MPC, SMPC permits a small probability of constraint violation. This formulation allows the controller to balance conflicting objectives, such as following a desired trajectory while mitigating unstable behaviors across a range of real-world disturbances. However, existing SMPC methods for quadrupedal robots are typically evaluated only in simulation and rely on offline trajectory optimization for warm-starting, due to the non-convexity and computational demands of the underlying problem [3], [7].

Received 4 November 2024; accepted 10 June 2025. Date of publication 2 July 2025; date of current version 10 July 2025. This article was recommended for publication by Associate Editor J. Carpentier and Editor A. Kheddar upon evaluation of the reviewers' comments. This work was supported by Impact Engines Program at Northeastern University.

Ananya Trivedi, Sarvesh Prajapati, Michael Everett, and Taşkın Padır are with Northeastern University, Boston, MA 02115 USA (e-mail: trivedi.ana@northeastern.edu; prajapati.s@northeastern.edu; m.everett@northeastern.edu; t.padir@northeastern.edu).

Mark Zolotas is with Northeastern University, Boston, MA 02115 USA, and also with Toyota Research Institute (TRI), Northeastern University, Cambridge, MA 02139 USA (e-mail: m.zolotas@northeastern.edu).

A video of the results and accompanying code can be found at: https://cc-mpc.github.io/.

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3585315, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3585315

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

In this work, we propose a novel Chance-Constrained MPC (CCMPC) algorithm—a specific form of SMPC—to generate ground reaction forces for quadrupedal robots. Our approach models mass, inertia, and contact sequences as stochastic variables. By formulating the control problem as a quadratic programming (QP) problem [8], we achieve real-time solve rates at ∼500 Hz, comparable to Linear MPC (LMPC). We validate CCMPC through extensive simulations and hardware experiments on the Unitree Go1 robot. Our approach achieves superior performance over traditional methods in maintaining stability, reducing foot slippage, and supporting payloads exceeding 50% of the robot's weight across muddy slopes, stairs, grass, and gravel. This is accomplished using a unified control policy that effectively handles different terrain conditions and payload variations without the need for parameter re-tuning. The paper's key contributions are:

- We develop a CCMPC algorithm tailored for quadrupedal robots to handle disturbances from variable payloads and unknown terrain dynamics.
- The control problem is cast as a quadratic program, enabling fast solve times suitable for real-time execution.
- We validate our method through simulations and hardware experiments. In simulations, CCMPC achieves a 100% success rate across multiple gaits, compared to 48.9% for LMPC and 80.9% for hand-tuned MPC. To the best of our knowledge, this is the first SMPC implementation for quadrupedal robots on hardware.

II. Related Works

Using convex MPC with the single rigid body dynamics (SRBD) model has enabled fast computation of diverse quadrupedal walking gaits [9], [10]. These approaches require an accurate dynamics model, making them less effective when the real-world physics deviates from the designed controller [11], [12]. Several recent methods address model and contact uncertainty explicitly through stochastic trajectory optimization [13], [14], [15]. While their deterministic counterparts have been successfully deployed on hardware [16], [17], these stochastic extensions remain confined to simulation. Moreover, they typically require warm-starting or a carefully chosen initial guess due to the non-convexity of the underlying problem. By instead approximating the robot dynamics and reformulated chance constraints as linear, our method reduces to a convex quadratic program amenable to real-time execution.

In contrast to model-based control, model-free RL eliminates the need for accurate models by using domain randomization during training [18]. This approach exposes the control policies to a wide range of scenarios, enhancing their robustness to diverse environments [19]. However, these methods often struggle with the sim-to-real gap, leading to conservative behaviors or failure to handle out-of-distribution disturbances [20]. They also require significant engineering effort and frequent re-tuning [5]. In comparison, our method achieves demonstrable generalization across terrain and payload variations with no parameter re-tuning.

Adaptive MPC enhances classical MPC by updating model parameters online to compensate for uncertain system dynamics. Existing approaches estimate residual model discrepancies offline using simulation [21] or online using L1 adaptive control [22]. However, proper initialization of estimated parameters is crucial to avoid instability before convergence [23]. Additionally, these methods assume constant uncertainty throughout the MPC horizon. By incrementally propagating the state and control variance, our method captures evolving uncertainty and improves prediction accuracy.

Gazar et al. [24] achieved bipedal walking in simulation using tube MPC to handle additive polytopic uncertainties, but the motion was limited to planar translation. Xu et al. [11] addressed this limitation by employing robust min-max MPC (RMPC) for quadrupedal walking. Their method handles uncertainties in payload and friction coefficients. As in typical RMPC formulations, the resultant optimization problem accounts for the worst-case realization of bounded disturbances [25]. For large disturbance sets, this can lead to infeasibility due to significant shrinkage of the feasible set [26]. Consequently, Xu et al. [11] used one set of parameters for experiments involving payload variations and a different set for experiments involving slippery surfaces. In contrast, our method accounts for uncertainties through contact force constraint tightening, guided by the propagation of state and control covariances. As shown in Section IV-A, these covariances grow gradually along the prediction horizon, resulting in progressively tighter feasible sets without assuming fixed worst-case effects. This allowed all CCMPC experiments—simulation and hardware—to use the same parameters.

SMPC offers a principled trade-off between performance and robustness by allowing user-specified probabilities of constraint violation [6], [27]. This reduces the conservativeness inherent in RMPC, which must satisfy constraints under all possible disturbances even at the cost of feasibility or performance [26]. Despite the advantages of SMPC, existing work on quadrupedal robots is mostly confined to simulation and is computationally intensive for real-time deployment [3], [7]. The method in [3] performs constraint tightening over the x–y contact plane based on additive disturbances, but does not explicitly handle vertical (z-direction) disturbances or payload variations. In contrast, the method in [7] applies chance constraints to both the friction cone and the robot's kinematic reachability. However, it relies on centroidal momentum dynamics and requires optimizing over angular momentum, resulting in a non-convex formulation that must be warm-started using a whole-body planner. A side-by-side comparison is provided in the supplementary video between our method and [7]. In contrast to existing SMPC methods, our approach employs a linearized SRBD model [9] tailored for quadrupeds with lightweight legs, enabling a fully convex approximation that operates in real-time without warm-starting.

III. Chance-Constrained Foot Force MPC

In this section, we introduce the CCMPC framework, which optimizes ground reaction forces to ensure stability under uncertain conditions. As illustrated in Fig. 2, the framework uses a footstep planner that determines when each leg should enter the swing (white) or stance (blue) phase based on a predefined contact schedule [10]. For the swing foot, the Raibert heuristic [9] calculates the required motor torques, planning the trajectory from initiation through apex to landing using a cubic spline. CCMPC then optimizes ground reaction forces for the stance feet. The state estimator fuses IMU and foot force sensor data using an Extended Kalman Filter to estimate center of mass (CoM) states and leg end positions.

We use the linearized SRBD model for MPC, which maps the ground reaction forces, $\mathbf{u} \in \mathbb{R}^{12} = [\mathbf{f}_1^\top, \mathbf{f}_2^\top, \mathbf{f}_3^\top, \mathbf{f}_4^\top]^\top$, to the system state [9]. The state vector, $\mathbf{x} \in \mathbb{R}^{13} = [\boldsymbol{\Theta}^\top, \mathbf{p}^\top, \boldsymbol{\omega}^\top, \dot{\mathbf{p}}^\top, g]^\top$, comprises the robot's orientation,

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

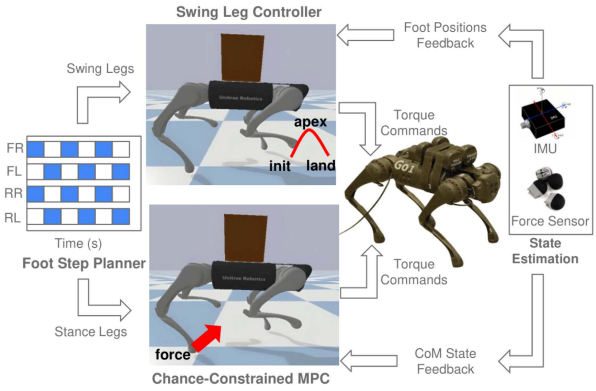


Fig. 2. Architecture used for quadrupedal locomotion.

position, angular velocity, linear velocity, and gravity magnitude. The primary objective of this work is to design a controller capable of stabilizing the robot's motion under disturbances from unknown payloads and uncertain terrain contacts. To achieve this, we group the robot's mass, inertia, and predefined foot contact locations into a random variable $\boldsymbol{\delta}_i \sim \mathcal{N}(\mathbb{E}[\boldsymbol{\delta}_i], \boldsymbol{\Sigma}_{\boldsymbol{\delta}})$, representing known distributions over these uncertain parameters. The mean values $\mathbb{E}[\boldsymbol{\delta}_i]$ correspond to the nominal mass m, rotational inertia matrix $\mathbf{I}$, and foot contact locations $\mathbf{r}_{i,1}, \ldots, \mathbf{r}_{i,4}$, compactly denoted as $\bar{\boldsymbol{\delta}}_i = [m, \mathrm{diag}(\mathbf{I}), \mathbf{r}_{i,1}^\top, \ldots, \mathbf{r}_{i,4}^\top]$. Thus, the stochastic linear time-varying equations of motion are:

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

where $\mathbf{A}_i$ is the state transition matrix and $\mathbf{B}_i(\boldsymbol{\delta}_i)$ is the control input matrix, with the nominal form $\mathbf{B}_i(\bar{\boldsymbol{\delta}}_i)$ evaluated at the mean as:

$$\mathbf{B}_i(\bar{\boldsymbol{\delta}}_i) = \begin{bmatrix} \mathbf{0}_{3\times3} & \cdots & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \cdots & \mathbf{0}_{3\times3} \\ \mathbf{I}^{-1}\mathbf{S}(\mathbf{r}_{i,1}) & \cdots & \mathbf{I}^{-1}\mathbf{S}(\mathbf{r}_{i,4}) \\ m^{-1}\mathbf{1}_{3\times3} & \cdots & m^{-1}\mathbf{1}_{3\times3} \\ \mathbf{0}_{1\times3} & \cdots & \mathbf{0}_{1\times3} \end{bmatrix} \Delta t \tag{2}$$

Here, $\mathbf{S}(\cdot)$ denotes the skew-symmetric matrix for the cross product $\mathbf{r} \times \mathbf{f}$, Δt is the discretization timestep, and $\mathbf{1}_{3\times3}$ is a 3×3 identity matrix. The term $\mathbf{w}_i$ captures linearization errors and unmodeled dynamics, and is treated as a zero-mean Gaussian with known covariance, $\mathbf{w}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{w}})$.

To ensure that ground reaction forces remain physically admissible and prevent slippage, they must satisfy linearized friction cone constraints at each stance foot [4]. Since the disturbances $\boldsymbol{\delta}_i$ and $\mathbf{w}_i$ are unbounded, it is generally not possible to guarantee strict constraint satisfaction [7]. Instead, chance constraints enforce that safety-critical constraints, such as friction cone constraints, are satisfied with a high probability $\epsilon < 1$. This allows for a principled balance between optimization objectives. In Section V, we use $\epsilon = 0.95$ and observe that the robot maintains stability without falls under realistic payload and terrain disturbances. Hence, for a given friction coefficient μ, the chance constraints are enforced as:

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

where $\mathbf{C} \in \mathbb{R}^{20\times12} = \mathrm{blkdiag}(\mathbf{C}_\mu, \mathbf{C}_\mu, \mathbf{C}_\mu, \mathbf{C}_\mu)$, and

$$\mathbf{C}_\mu = \begin{bmatrix} -1 & 0 & -\mu \\ 1 & 0 & -\mu \\ 0 & -1 & -\mu \\ 0 & 1 & -\mu \\ 0 & 0 & -1 \end{bmatrix} \tag{4}$$

The last row of (4) enforces that the normal component of the ground reaction force remains non-negative. We minimize the expected cost $\mathbb{E}[J(\mathbf{x}, \mathbf{u})]$, which penalizes deviations from the desired CoM trajectory $\mathbf{x}_{\text{ref}}$, as well as a regularization term on the control inputs:

$$\mathbb{E}[J(\mathbf{x}, \mathbf{u})] = \mathbb{E}\left[\sum_{i=0}^{N-1} \|\mathbf{x}_{i+1} - \mathbf{x}_{\text{ref},i+1}\|_{\mathbf{Q}}^2 + \|\mathbf{u}_i\|_{\mathbf{R}}^2\right], \tag{5}$$

Here, the expectation is taken over $\mathbf{x} \sim (\bar{\mathbf{x}}, \boldsymbol{\Sigma}_x)$ and $\mathbf{u} \sim (\mathbf{v}, \boldsymbol{\Sigma}_u)$. The matrix $\mathbf{Q} \succeq 0$ weighs tracking error and $\mathbf{R} \succ 0$ penalizes control effort. Finally, we include a constraint using matrix $\mathbf{D}_i$ to zero out contact forces for the swing legs:

$$\mathbf{D}_i \mathbf{u}_i = 0 \tag{6}$$

The resultant form of the Chance-Constrained Foot Force MPC, incorporating stochastic dynamics and chance inequality constraints, is expressed as:

$$\begin{aligned} &\underset{\mathbf{x}_i, \mathbf{u}_i}{\text{minimize}} \quad \text{Eqn. (5)} \\ &\text{subject to} \quad \text{Eqn. (1)}, \ \text{Eqn. (3)}, \ \text{Eqn. (6)} \end{aligned} \tag{7}$$

IV. Convex QP Reformulation of CCMPC

Solving the CCMPC problem in (7) is computationally intractable, as enforcing chance constraints entails high-dimensional Gaussian integrals. These integrals lack a closed-form solution owing to the nonlinear dependence of the matrix $\mathbf{B}$ on $\boldsymbol{\delta}$ [6]. In this section, we derive a tractable deterministic approximation for real-time implementation.

A. Uncertainty Propagation

Due to parametric and additive uncertainties in the system dynamics, future states follow a stochastic distribution. Following the approach in [3], we parameterize the control law as a state-dependent feedback policy. Consequently, future control actions also possess inherent uncertainty. Specifically, we define the control law as:

$$\mathbf{u}_i = \mathbf{v}_i + \mathbf{K}_i(\mathbf{x}_i - \bar{\mathbf{x}}_i), \tag{8}$$

where $\mathbf{v}_i$ denotes the feedforward control input and $\bar{\mathbf{x}}_i$ is the predicted mean of the state distribution. The term $\mathbf{v}_i$ drives the system along the reference trajectory, while the feedback term $\mathbf{K}_i(\mathbf{x}_i - \bar{\mathbf{x}}_i)$ corrects deviations caused by disturbances and model inaccuracies. Jointly optimizing $\mathbf{v}_i$ and $\mathbf{K}_i$ leads to a bi-level optimization problem, which is non-convex and computationally expensive for real-time control [28]. Instead, we pre-compute $\mathbf{K}_i$ prior to each MPC iteration by solving the Discrete Algebraic Riccati Equation (DARE) at each timestep using the system matrices $\mathbf{A}_i$ and $\mathbf{B}_i(\bar{\boldsymbol{\delta}}_i)$. Thus, the MPC decision variables are limited to $\mathbf{v}_i$ and $\bar{\mathbf{x}}_i$.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Next, we derive how the first and second moments of the stochastic trajectory propagate over time. For detailed derivations, we refer the reader to [6]. Applying the expectation operator to (1), the resulting distribution at the next timestep has mean $\bar{\mathbf{x}}_{i+1}$ and covariance $\mathbf{\Sigma}_{\mathbf{x}_{i+1}}$ given by:

$$\bar{\mathbf{x}}_{i+1} \approx \mathbf{A}_i \bar{\mathbf{x}}_i + \mathbf{B}_i(\bar{\boldsymbol{\delta}}_i)\mathbf{v}_i \tag{9}$$

$$\mathbf{\Sigma}_{\mathbf{x}_{i+1}} \approx \mathbf{A}_{cl}\mathbf{\Sigma}_{\mathbf{x}_i}\mathbf{A}_{cl}^\top + \mathbf{P}_i\mathbf{\Sigma}_{\boldsymbol{\delta}}\mathbf{P}_i^\top + \mathbf{\Sigma}_{\mathbf{w}}, \tag{10}$$

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

$$\mathbf{P}_i = \frac{\partial \bar{\mathbf{x}}_{i+1}}{\partial \bar{\boldsymbol{\delta}}_i} = \frac{\partial(\mathbf{A}_i\bar{\mathbf{x}}_i + \mathbf{B}_i(\bar{\boldsymbol{\delta}}_i)\mathbf{v}_i)}{\partial \bar{\boldsymbol{\delta}}_i} \tag{11}$$

The control distribution is then given by:

$$\mathbb{E}[\mathbf{u}_i] = \mathbf{v}_i, \quad \mathbf{\Sigma}_{\mathbf{u}_i} = \mathbf{K}_i\mathbf{\Sigma}_{\mathbf{x}_i}\mathbf{K}_i^\top \tag{12}$$

These trajectory distributions will be used in the next section to derive deterministic counterparts of the chance constraints.

B. Contact Force Constraint Tightening

To incorporate (3) into a tractable optimization problem, we seek a deterministic surrogate that conservatively approximates the original joint chance constraint. The Boole-Bonferroni inequality allows us to do so by bounding the sum of the individual violation probabilities $\Pr(\mathbf{C}^j\mathbf{u}_i > 0)$, where $\mathbf{C}^j$ denotes the jth row of $\mathbf{C} \in \mathbb{R}^{20\times 12}$:

$$\sum_{j=1}^{20} \Pr(\mathbf{C}^j\mathbf{u}_i > 0) \le 1-\epsilon \Rightarrow \Pr(\mathbf{C}\mathbf{u}_i \le \mathbf{0}) \ge \epsilon \tag{13}$$

While optimizing risk allocation for each constraint could be more effective, it involves a two-stage optimization problem that can be computationally expensive [7]. To circumvent this problem, we assign uniform risks α to each constraint, where $\alpha = (1-\epsilon)/20$. Hence, each constraint is approximated as:

$$\Pr(\mathbf{C}^j\mathbf{u}_i > 0) \le \alpha \Leftrightarrow \Pr(\mathbf{C}^j\mathbf{u}_i \le 0) \ge 1-\alpha \tag{14}$$

Finally, since the control actions follow a normal distribution, $\mathbf{u}_i \sim \mathcal{N}(\mathbf{v}_i, \mathbf{\Sigma}_i^u)$, individual chance constraints in (14) can be deterministically reformulated as [6]:

$$\mathbf{C}^j\mathbf{v}_i + \phi^{-1}(1-\alpha)\sqrt{\mathbf{C}^j\mathbf{\Sigma}_i^u(\mathbf{C}^j)^\top} \le 0 \tag{15}$$

where ϕ^{-1} is the inverse of the cumulative distribution function of a standard normal distribution [3]. Applying (15) to all rows of $\mathbf{C}$, we get the following constraints on the mean of control actions:

$$(3) \approx \mathbf{C}\mathbf{v}_i \le \mathbf{c}_i, \tag{16}$$

$$\mathbf{c}_i = \mathrm{col}(c_i^1, \ldots, c_i^{20}) \text{ and } c_i^j = -\phi^{-1}(1-\alpha)\sqrt{\mathbf{C}^j\mathbf{\Sigma}_i^u(\mathbf{C}^j)^\top}$$

The tightened form of the unilateral contact force constraint $f^z \ge 0$ thus becomes:

$$f^z \ge \phi^{-1}(1-\alpha)\sqrt{\mathbf{C}^z\mathbf{\Sigma}_i^u(\mathbf{C}^z)^\top} \tag{17}$$

where $\mathbf{C}^z$ corresponds to the last row of $\mathbf{C}_\mu$. As Fig. 3 shows, (16) similarly narrows the longitudinal-lateral force bounds.

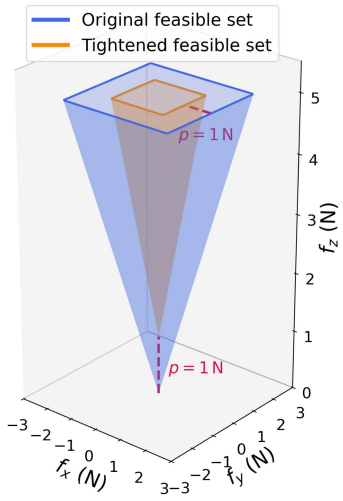


Fig. 3. Reduced feasible set after uniform $p = 1$ N tightening of a friction-cone and unilateral force constraint ($\mu = 0.4$).

Specifically, $|f^x|, |f^y| \le \mu f^z$ becomes $|f^x|, |f^y| \le \mu f^z - p$ with $p \ge 0$.

Unmodeled contact-location errors, such as an early touchdown, introduce impulsive disturbances that can drive the actual ground-reaction forces outside the friction cone, leading to foot slippage. By tightening the friction cone constraints with the back-off terms $\mathbf{c}_i$ in (16), we create a buffer that absorbs these disturbance forces. This buffer lets the controller tolerate moderate contact-location uncertainty without modeling it explicitly as is done in [29]. Similarly, the additional bias in (17) encourages the optimizer to generate higher vertical forces for unmodeled payload support.

Under higher uncertainty, this tightening procedure leads to slower speeds and less aggressive maneuvers, trading off agility for safety. It is worth noting that reducing the friction coefficient μ also acts as a form of friction cone tightening, limiting allowable f^x and f^y forces. However, it does not impose any lower bound on f^z, which is essential for maintaining feasibility under unknown payloads.

C. Reduction to Quadratic Program

We now consolidate the setup of the optimization problem. We adopt a certainty-equivalent formulation, where the cost and constraints are evaluated using the mean state and control trajectories. This improves computational efficiency while remaining consistent with the underlying stochastic dynamics [30]. The resulting objective function and the constraint enforcing zero contact forces for swing legs are:

$$\text{Eqn. (5)} \approx J(\bar{\mathbf{x}}, \mathbf{v}) = \sum_{i=0}^{N-1} \|\bar{\mathbf{x}}_{i+1} - \mathbf{x}_{\text{ref},i+1}\|_{\mathbf{Q}}^2 + \|\mathbf{v}_i\|_{\mathbf{R}}^2 \tag{18}$$

$$\text{Eqn. (6)} \approx \mathbb{E}[\mathbf{D}_i\mathbf{u}_i] = \mathbf{D}_i\mathbf{v}_i = \mathbf{0} \tag{19}$$

As shown in Algorithm 1, the control loop alternates between two phases. In the first phase, the tightening factors $\mathbf{c}_i$ are computed using the propagated state and control covariances. These factors are then used in the second phase to define the tightened constraints for contact force optimization.

Algorithm 1: Chance-Constrained MPC (CCMPC).

Initialize $\mathbf{c}_0 \leftarrow \mathbf{0}$, $\boldsymbol{\Sigma}_{\mathbf{x}_0} \leftarrow \mathbf{0}$, $\boldsymbol{\Sigma}_{\mathbf{u}_0} \leftarrow \mathbf{0}$, $\bar{\mathbf{x}}^* \leftarrow \mathbf{0}$, $\mathbf{v}^* \leftarrow \mathbf{0}$
while goal configuration not reached
  /* Constraint Tightening Phase */
  for $i = 0$ to $N-2$
    Compute $\bar{\boldsymbol{\delta}}_i$, $\mathbf{A}_i$, $\mathbf{B}_i(\bar{\boldsymbol{\delta}}_i)$, $\mathbf{D}_i$
    $\mathbf{K}_i \leftarrow \text{DARE}(\mathbf{A}_i, \mathbf{B}_i(\bar{\boldsymbol{\delta}}_i), \mathbf{Q}, \mathbf{R})$
    $\boldsymbol{\Sigma}_{\mathbf{x}_{i+1}} \leftarrow$ (10), (11)
    $\boldsymbol{\Sigma}_{\mathbf{u}_{i+1}} \leftarrow$ (12)
    $\mathbf{c}_{i+1} \leftarrow$ (16)
  end for
  /* Trajectory Optimization Phase */
  for $i = 0$ to $N-1$
    Cost: (17), Constraints: (9), (16), (18)
  end for
  Solve QP: $\bar{\mathbf{x}}^*, \mathbf{v}^* \leftarrow$ Solve (19)
end While

In summary, given a nominal contact plan and a desired CoM trajectory, CCMPC solves the deterministic reformulation of the original stochastic optimization problem as:

$$\{\bar{\mathbf{x}}^*, \mathbf{v}^*\} = \underset{\bar{\mathbf{x}}_i, \mathbf{v}_i}{\text{minimize}} \quad \text{Eqn. (18)}$$
$$\text{subject to} \quad \text{Eqn. (9)}, \ \text{Eqn. (16)}, \ \text{Eqn. (19)} \qquad (20)$$

Using single shooting, we further reduce the number of decision variables in (20) to only include the mean of the control actions [9]. Since the objective is quadratic and the constraints are linear, the resulting convex QP can be efficiently solved in real-time using the qpOASES solver [31]. Finally, the first element of the optimal ground reaction force sequence $\mathbf{v}^*$ is converted to desired foot motor torques using the contact Jacobian, $\mathbf{J}(\mathbf{q})$.

V. Experiments and Results

This section presents simulation and hardware evaluations of CCMPC on the Unitree Go1 robot across varied, unmodeled payloads and terrains. We benchmark it against an LMPC controller with no constraint tightening and a Heuristic MPC (HMPC) controller that uses fixed, hand-tuned tightening margins. We conclude the section with an analysis explaining why the three approaches exhibit different performance levels.

A. Simulation Experiments

The simulation experiments are conducted using the PyBullet simulator [32] with a high-fidelity robot model. All algorithms are executed on a Legion 5 Pro laptop with an Intel i7-12700H processor and 32 GB RAM. The reported runtime of 500 Hz includes all steps in the control pipeline, i.e., the footstep planner, the inverse kinematics-based swing leg controller, the calculation of constraint tightening factors, and solving the resulting QP.

We conduct Monte Carlo simulations to demonstrate that our control policy can stabilize quadrupedal motion across various payloads and terrains without parameter re-tuning. The robot was commanded to move forward at 0.25 m/s, performing blind locomotion over wooden planks under varying payloads. Payloads were uniformly sampled from 0 kg (no load) to 10 kg (maximal load), and plank heights ranged from 0 to 5 cm.

TABLE I
MPC Specific Parameters Used Across All Methods

Parameter	Value
Stepping Frequency	2.5 Hz
Foot Height	0.08 m
Planning Horizon	10 steps
Planning Timestep	0.025 s
Position Weight (Z)	500
Velocity Weights (X, Y)	20, 5
Angular Velocity Weights (X, Y, Z)	0.2, 0.2, 1.0
Roll and Pitch Weights	0.2, 0.2
Control Penalty Weights	1e-6

TABLE II
CCMPC Variance Parameters

Parameter	Value
Mass (kg^2)	15.0
Inertia (X, Y, Z) ($\text{kg}^2 \cdot \text{m}^4$)	0.02, 0.06, 0.06
Angular Velocity (X, Y, Z) (rad^2/s^2)	0.5, 0.2, 0.01
Linear Velocity (X, Y, Z) (m^2/s^2)	0.5, 0.2, 0.01
Contact Location (X, Y, Z) (m^2)	0.36, 0.36, 0.36

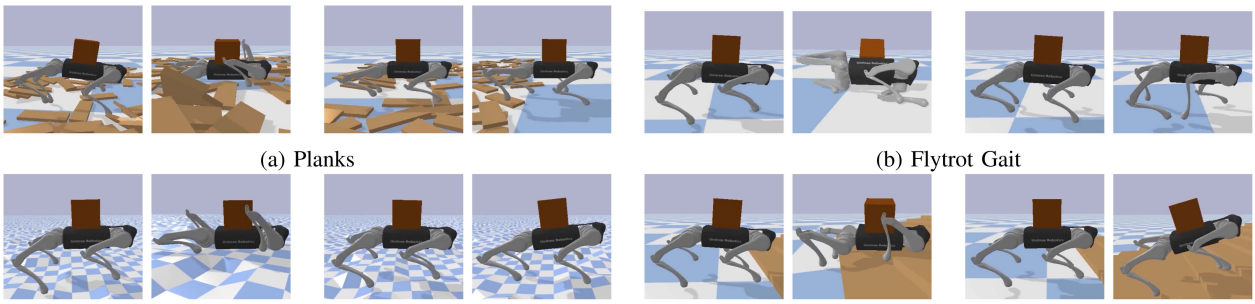
TABLE III
Results From 1000 Monte-Carlo Simulations

Metric	Gait	LMPC	HMPC	CCMPC
Success Rate (%) ↑	Trot	48.4	80.8	**100**
	Flytrot	49.5	81.0	**100**
Average Slippage Ratio ↓	Trot	0.24	0.11	**0.08**
	Flytrot	0.24	0.16	**0.14**
Normalized Tracking Cost ↓	Trot	1.3±0.1	1.1±0.2	**1.0±0.1**
	Flytrot	2.0±0.3	1.8±0.2	**1.0±0.2**
Normalized Effort Cost ↓	Trot	1.3±0.1	1.2±0.2	**1.0±0.2**
	Flytrot	1.6±0.2	1.4±0.2	**1.0±0.2**

A total of 1000 samples were generated to induce variability in inertial properties and contact locations. Given the robot's nominal mass of 12 kg and a foot raise height of 8 cm, these values represent significant variations from standard conditions. For heuristic constraint tightening, we assumed a maximum unmodeled payload of 10 kg and computed the required gravity compensation force f^z from the stance feet. To accommodate the maximum commanded acceleration of 0.2 m/s^2, we ensured sufficient horizontal forces f^x and f^y could be generated under load. This approach was used to heuristically find the constraint tightening factors in (16).

The shared MPC settings for CCMPC, HMPC, and LMPC are listed in Table I. The CCMPC-specific covariances $\boldsymbol{\Sigma}_\delta$ and $\boldsymbol{\Sigma}_w$ appear in Table II. Because payload changes primarily affect vertical dynamics, we logarithmically swept the z-position weight from 0.5 to 5000. This was performed on a terrain with randomly placed 5 cm planks while varying payloads from 0 kg to 10 kg for all three controllers. A value of 500 yielded the fewest falls. Other gains in Table I were co-tuned in the same sweep and then fixed for all simulation and hardware runs. The HMPC constraint-tightening factors were tuned in a similar grid search.

We assessed each controller's performance using four metrics, logged in Table III. Success rate was determined based on three failure criteria: a drop in base height exceeding 30% of the



(a) Planks (b) Flytrot Gait

(c) Random Elevation Wavefield (d) Blind Stair Climbing

Fig. 4. Comparison of LMPC (left two, fails) and CCMPC (right two, succeeds) across various gaits and terrains. Similar failures that occurred with LMPC were also observed with HMPC in these experiments.

(a) Grass (b) Gravel (c) Muddy Slope (d) Planks (e) Bouncy Playground

(f) Pull Task (g) Push Task (h) Concrete Slope (i) Blind Stair Climbing (j) Slippery Whiteboard

Fig. 5. Hardware validation of CCMPC. Video demonstrations can be found at https://cc-mpc.github.io/.

desired value, orientation relative to the ground normal falling below 0.8 (where 1 indicates perfect upright alignment), and any QP infeasibility due to constraint violations. These thresholds were identified as points where the robot consistently failed to remain upright. The slippage ratio, defined as the magnitude of tangential to normal contact force, indicates how close the contact force is to saturating the friction cone, with higher values implying increased risk of slippage. For normalized costs, CCMPC was used as the reference and assigned a value of 1. The standard deviations are similarly normalized by dividing the raw standard deviation by the corresponding mean. Lastly, normalized tracking cost refers to CoM tracking performance, while normalized effort cost relates to actuation effort, where higher values indicate worse tracking and greater effort, respectively. To ensure fair comparison, these metrics were computed only for successful runs to avoid skew from outlier conditions. Quantitative results in Table III indicate that CCMPC outperforms both baselines across all the evaluation metrics.

With a fixed 6 kg unmodeled payload, we benchmarked all controllers on challenging terrains, such as planks, stairs, and random elevation fields, as well as at 1.75 m/s in the flytrot gait. The results are visualized in Fig. 4.

B. Hardware Validation

To validate our approach in real-world conditions, we conducted hardware experiments using the Unitree Go1 robot, replicating key scenarios from the simulations. The first 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 planks or additional mass. While LMPC failed to navigate this terrain, CCMPC successfully guided the robot across the planks.

We explored the versatility of our control algorithm through various challenging terrain experiments, as shown in Fig. 5. The robot, loaded with additional 6 kg mass, successfully climbed stairs and navigated muddy slopes, grass, and gravel. The robot also performed tasks on objects, 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 tested surfaces, showcasing its ability to handle diverse ground textures and maintain stability on slippery and uneven terrains.

To demonstrate the repeatability of our control strategy, we conducted a series of progressive load tests, as shown in Fig. 6. The robot was commanded to move forward at a speed of 0.25 m/s while carrying increasing payloads: 1.3 kg, 2.2 kg, 3.0 kg, 4.3 kg, 6.0 kg, and 7.3 kg. The robot traversed flat ground during all tests. The results showed that LMPC maintained stability up to 4.3 kg but failed to navigate with a 6.0 kg payload. In contrast, CCMPC successfully handled both 6.0 kg

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

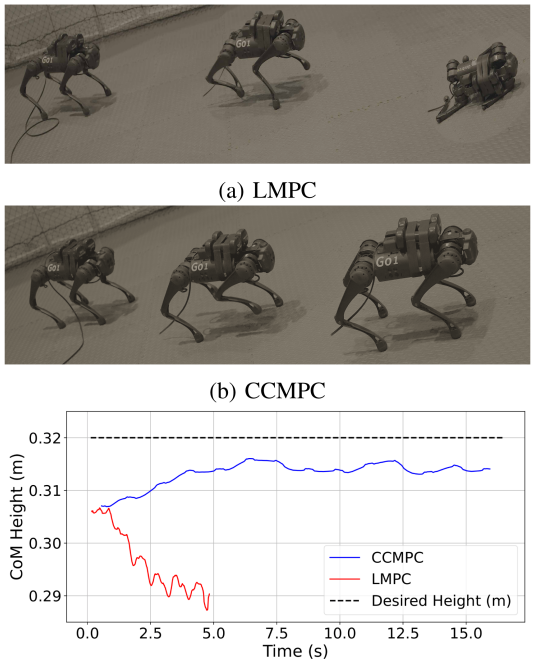


Fig. 6. Hardware height tracking comparison with an unmodeled 6 kg payload.

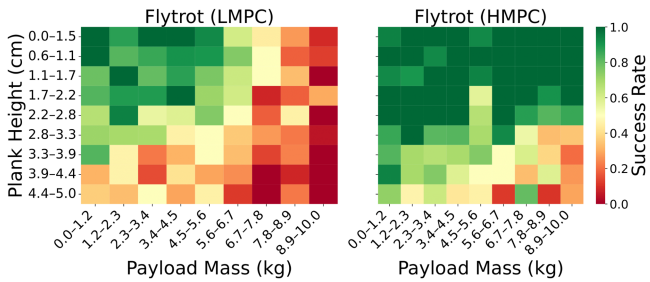


Fig. 7. Monte Carlo success rate heatmaps for different payload-plank height combinations under the flytrot gait.

and 7.3 kg payloads, even exceeding the manufacturer's recommended maximum payload of 5.0 kg.

C. Discussion

We now discuss the differences between control strategies to better understand sources of performance disparity.

Linear MPC: LMPC leverages its inherent feedback mechanism to recompute control actions based on the observed system state [9]. This reactive strategy is effective under mild disturbances, corresponding to the top-left region in Fig. 7. However, it breaks down when either mass or contact disturbances become severe, resulting in failure.

Heuristic MPC: Unlike LMPC, HMPC incorporates a degree of anticipatory behavior through fixed constraint tightening. It elevates the required f^z values based on the worst-case expected payload, which we set to 10 kg, and adjusts the allowable f^x and f^y accordingly. Fig. 7 shows that HMPC achieves higher success rates than LMPC across most payloads and plank heights, particularly in the midrange. Nevertheless, failures tend to concentrate in the bottom-right region. These events are associated with both large unmodeled payloads and significant contact variability.

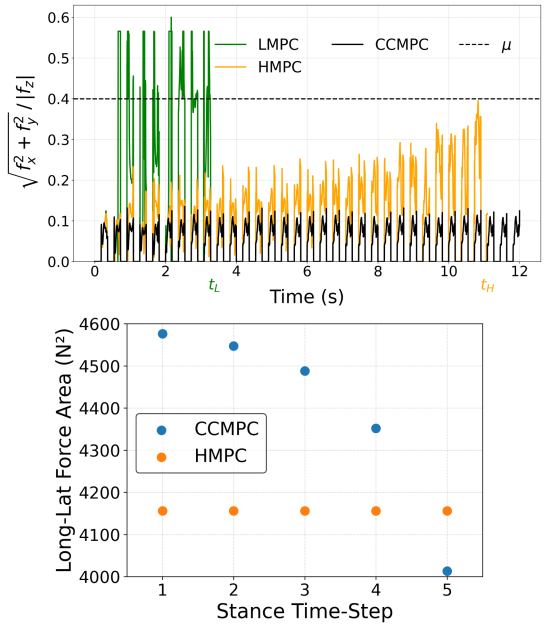


Fig. 8. Time-evolution of the friction-cone saturation ratio (top) and tightened longitudinal-lateral feasible force area (bottom) for the right-hind foot under a 6 kg unmodeled load.

HMPC follows a conservative strategy that assumes worst-case uncertainty at all times. As a result, the robot was frequently driven into configurations where the tightened constraints left no feasible solution for recovery. This limitation mirrors the behavior observed in similar RMPC formulations [11], [25].

CCMPC Strengths: Fig. 8 contrasts the three controllers for the right-hind leg during a flat-ground trot under a 6 kg payload. The top panel shows the friction-cone saturation ratio $\sqrt{(f^x)^2+(f^y)^2}/|f^z|$, with zeros corresponding to swing phases. This ratio indicates proximity to friction-cone limits, where exceeding μ occurs from either large horizontal forces or insufficient vertical support f_z. LMPC, lacking lower-bound enforcement on f_z, quickly drives the ratio above μ, causing infeasibility at t_L. HMPC imposes a fixed buffer for f_z, initially maintaining feasibility. However, as the robot enters challenging configurations, HMPC lacks state-dependent tightening and eventually loses feasibility at t_H. By contrast, CCMPC maintains a well-bounded ratio throughout.

The bottom panel of Fig. 8 plots feasible force areas in the f_x–f_y plane at $f_z = 60\,\text{N}$. This value corresponds to half the robot's total weight (12 kg) supported by a single leg during stance. Both HMPC and CCMPC yield comparable initial areas ($\sim 4\,\text{kN}^2$), confirming HMPC is not overly conservative. However, HMPC maintains this area constant throughout the horizon. In contrast, CCMPC smoothly contracts the feasible area as uncertainty grows across successive stance phases within the short MPC horizon ($N = 10$). This state and covariance dependent tightening allows CCMPC to preserve feasibility over realistic payload and terrain variations.

CCMPC Limitations: One limitation arises when increasing the uncertainty parameters Σ_θ and Σ_w to reflect heavier payloads and more variable contact conditions than those evaluated. In such cases, the resulting constraint tightening often led to QP infeasibility. A possible remedy is to integrate Bayesian machine learning techniques [33], [34] for more accurate estimation of

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

unknown dynamics. The estimated mean could then feed into the equality-constrained dynamics, while its variance informs the tightening of inequality constraints. Additionally, CCMPC does not account for full-body dynamics or joint torque limits. In scenarios with significant actuation constraints, a full-body nonlinear MPC formulation may be required [35]. Our framework supports such models, though solver integration is left for future work.

VI. Conclusion

In this letter, we presented a Chance-Constrained Model Predictive Control (CCMPC) framework to generate ground reaction forces for quadrupedal robots. CCMPC produces a unified control policy that stabilizes motion across a wide range of unmodeled payloads and varying terrain conditions in real-time, without re-tuning for different disturbance realizations. The effectiveness of CCMPC was validated through extensive simulation and hardware experiments. Looking ahead, we aim to extend this framework to loco-manipulation tasks in bipedal robots handling unknown payloads [36].

Acknowledgment

This letter describes that the work of Mark Zolotas was performed at Northeastern University and is not associated with TRI. The author Taşkın Padır is also an Amazon Scholar and this letter describes that the work of Taşkın Padır was performed at Northeastern University and is not associated with Amazon.

References

[1] P. Biswal and P. K. Mohanty, "Development of quadruped walking robots: A review," *Ain Shams Eng. J.*, vol. 12, no. 2, pp. 2017–2031, 2021.
[2] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Sci. Robot.*, vol. 5, no. 47, 2020, Art. no. eabc5986.
[3] A. Gazar, M. Khadiv, A. Del Prete, and L. Righetti, "Multi-contact stochastic predictive control for legged robots with contact locations uncertainty," 2023, *arXiv:2309.04469*.
[4] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard, and A. Del Prete, "Optimization-based control for dynamic legged robots," *IEEE Trans. Robot.*, vol. 40, pp. 43–63, 2024.
[5] J. Ibarz, J. Tan, C. Finn, M. Kalakrishnan, P. Pastor, and S. Levine, "How to train your robot with deep reinforcement learning: Lessons we have learned," *Int. J. Robot. Res.*, vol. 40, no. 4/5, pp. 698–721, 2021.
[6] T. Lew, R. Bonalli, and M. Pavone, "Chance-constrained sequential convex programming for robust trajectory optimization," in *Proc. Eur. Control Conf.*, 2020, pp. 1871–1878.
[7] 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 *Proc. Int. Symp. Robot. Res.*, 2022, pp. 420–435.
[8] S. Boyd, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[9] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–9.
[10] Y. Yang, T. Zhang, E. Coumans, J. Tan, and B. Boots, "Fast and efficient locomotion via learned gait transitions," in *Proc. Conf. Robot Learn.*, 2022, pp. 773–783.
[11] S. Xu, L. Zhu, H.-T. Zhang, and C. P. Ho, "Robust convex model predictive control for quadruped locomotion under uncertainties," *IEEE Trans. Robot.*, vol. 39, no. 6, pp. 4837–4854, Dec. 2023.
[12] M. Wang, M. Wonsick, X. Long, and T. Padr, "In-situ terrain classification and estimation for NASA's humanoid robot valkyrie," in *Proc. 2020 IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, 2020, pp. 765–770.
[13] B. Hammoud, M. Khadiv, and L. Righetti, "Impedance optimization for uncertain contact interactions through risk sensitive optimal control," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 4766–4773, Jul. 2021.
[14] L. Drnach, J. Z. Zhang, and Y. Zhao, "Mediating between contact feasibility and robustness of trajectory optimization through chance complementarity constraints," *Front. Robot. AI*, vol. 8, 2022, Art. no. 785925.
[15] Y. Shirai, D. K. Jha, A. U. Raghunathan, and D. Romeres, "Chance-constrained optimization in contact-rich systems," in *Proc. Amer. Control Conf.*, 2023, pp. 14–21.
[16] S. Le Cleac'h et al., "Fast contact-implicit model predictive control," *IEEE Trans. Robot.*, vol. 40, pp. 1617–1629, 2024.
[17] G. Kim, D. Kang, J.-H. Kim, S. Hong, and H.-W. Park, "Contact-implicit model predictive control: Controlling diverse quadruped motions without pre-planned contact modes or trajectories," *Int. J. Robot. Res.*, vol. 44, pp. 486–510, 2024, doi: 10.1177/02783649241273645.
[18] J. Lee, M. Bjelonic, A. Reske, L. Wellhausen, T. Miki, and M. Hutter, "Learning robust autonomous navigation and locomotion for wheeled-legged robots," *Sci. Robot.*, vol. 9, no. 89, 2024, Art. no. eadi9641.
[19] S. Chen, B. Zhang, M. W. Mueller, A. Rai, and K. Sreenath, "Learning torque control for quadrupedal locomotion," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2023, pp. 1–8.
[20] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: A survey," in *Proc. IEEE Symp. Ser. Comput. Intell.*, 2020, pp. 737–744.
[21] 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 Robot. Automat. Lett.*, vol. 7, no. 3, pp. 6622–6629, Jul. 2022.
[22] M. Sombolestan and Q. Nguyen, "Adaptive force-based control of dynamic legged locomotion over uneven terrain," *IEEE Trans. Robot.*, vol. 40, pp. 2462–2477, 2024.
[23] I. D. Landau et al. *Adaptive Control*, vol. 51. New York, NY, USA: Springer, 1998.
[24] 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 *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2021, pp. 61–68.
[25] J. Morimoto and C. Atkeson, "Minimax differential dynamic programming: An application to robust biped walking," in *Proc. 16th Int. Conf. Neural Inf. Process. Syst.*, 2002, pp. 1563–1570.
[26] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in Identification and Control*. Landon, U.K.: Springer, 2007, pp. 207–226.
[27] S. Katayama et al., "Robust locomotion via zero-order stochastic nonlinear model predictive control with guard saltation matrix," 2024, *arXiv:2403.14159*.
[28] S. Dempe and A. Zemkoho, *Bilevel Optimization* (Springer Optimization and its Applications), vol. 161. Cham, Switzerland: Springer, 2020.
[29] H. Zhao and M. Khadiv, "Trajectory optimization under contact timing uncertainties," in *Proc. IEEE-RAS 23rd Int. Conf. Humanoid Robots (Humanoids)*, 2024, pp. 1064–1071.
[30] A. Trivedi, S. Prajapati, A. Shirgaonkar, M. Zolotas, and T. Padir, "Data-driven sampling based stochastic MPC for skid-steer mobile robot navigation," 2024, *arXiv:2411.03289*.
[31] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOASES: A parametric active-set algorithm for quadratic programming," *Math. Program. Comput.*, vol. 6, pp. 327–363, 2014.
[32] E. Coumans and Y. Bai, "Pybullet quickstart guide," *ed: PyBullet Quickstart Guide*, 2021. [Online]. Available: https://docs.google.com/document/u/1/d
[33] A. Trivedi, S. Bazzi, M. Zolotas, and T. Padir, "Probabilistic dynamic modeling and control for skid-steered mobile robots in off-road environments," in*Proc. IEEE Int. Conf. Assured Autonomy*, 2023, pp. 57–60.
[34] A. Trivedi, M. Zolotas, A. Abbas, S. Prajapati, S. Bazzi, and T. Padir, "A probabilistic motion model for skid-steer wheeled mobile robot navigation on off-road terrains," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 12599–12605.
[35] A. Meduri, P. Shah, J. Viereck, M. Khadiv, I. Havoutis, and L. Righetti, "Biconmp: A nonlinear model predictive control framework for whole body motion planning," *IEEE Trans. Robot.*, vol. 39, no. 2, pp. 905–922, Apr. 2023.
[36] J. Li, J. Ma, O. Kolt, M. Shah, and Q. Nguyen, "Dynamic loco-manipulation on hector: Humanoid for enhanced control and open-source research," 2023, *arXiv:2312.11868*.