

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

Ananya Trivedi¹, Sarvesh Prajapati¹, Mark Zolotas^{1,2}, Michael Everett¹ and Taşkın Padır^{1,3}

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 (SRBD) 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 (LMPC) 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. A video of the results and accompanying code can be found at: <https://cc-mpc.github.io/>.

I. INTRODUCTION

Quadrupedal robots have demonstrated significant potential in various industrial applications and search and rescue missions. These robots enhance productivity by transporting heavy loads and traversing diverse terrains [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 unintended contact locations or timing errors [3]. As shown in Fig. 1, classical Model Predictive Control (MPC) methods often struggle with these discrepancies, leading to instability, foot slippage, or even falls.

Conventional model-based control methods for quadrupedal locomotion either fail to account for dynamics uncertainties, as in Differential Dynamic Programming (DDP) [4], or become

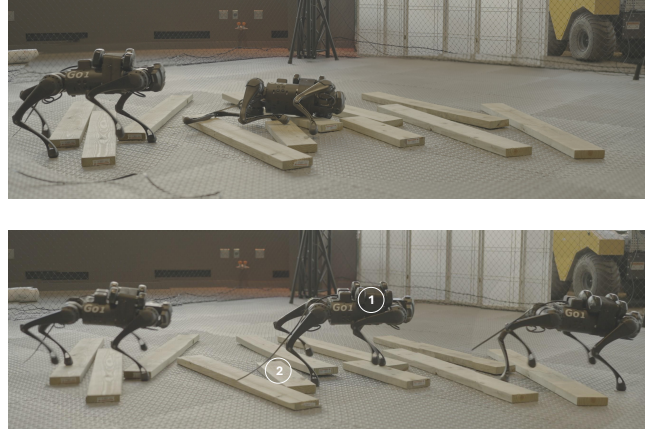


Fig. 1: Chance-Constrained MPC (bottom) stabilizes the robot by handling a distribution of inertial uncertainties from 6 kg dumbbells (1) and contact uncertainties, from planks (2). Linear MPC (top) fails under these conditions.

computationally infeasible when doing so, as with Stochastic Linear Complementarity Problems (SLCPs) [5]. Model-free Reinforcement Learning (RL) aims to generalize robot locomotion strategies across diverse environments through offline training, followed by online deployment. While these neural network policies work well in practice, they often lack interpretability and may require frequent re-training to ensure reliable deployment [6].

In contrast, Stochastic Model Predictive Control (SMPC) directly incorporates uncertainties into the control design by modeling them as probability distributions of disturbances [7], [8]. Unlike deterministic MPC, SMPC permits a small probability of constraint violation. In our experiments, we set this probability to 5%, corresponding to a 2σ confidence level. As a result, 95% of outcomes are expected to remain within constraints under Gaussian disturbances [9], [10]. 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 terrain disturbances.

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 [11], we achieve real-time solve rates at ~ 500 Hz, comparable to Linear MPC (LMPC). We validate CCMPC through extensive simulations and hardware

¹Institute for Experiential Robotics, Northeastern University, Boston, Massachusetts, USA.

²Mark 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.

³Taşkın 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.

Corresponding author: trivedi.ana@northeastern.edu

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 tuning. The key contributions of this paper are summarized as follows:

- We develop a CCMPC algorithm tailored for quadrupedal robots to handle disturbances from variable payloads and complex terrain dynamics.
- The control problem is formulated as a quadratic program, achieving fast solve times suitable for real-time application.
- We validate our method through simulations and hardware experiments. In simulations, CCMPC achieves a 100% success rate across multiple gaits, compared to 39.2% for Linear MPC (LMPC) and 75.7% for hand-tuned MPC. To our knowledge, this is the first SMPC implementation for quadrupedal robots on hardware.

II. RELATED WORKS

Recent advances in trajectory optimization have improved the reliability of quadrupedal locomotion, but managing real-world uncertainties remains challenging. Using convex MPC with the single rigid body dynamics (SRBD) model has enabled fast computation of diverse walking gaits for quadrupedal robots [12]. These approaches require an accurate dynamics model, making them less effective when the real-world physics deviates from the designed controller model [13]. Prior work has used DDP to account for uncertainties in trajectory optimization for legged locomotion [14]. Employing DDP handles inequality constraints implicitly, making it difficult to address uncertainty impacts on constraint satisfaction [15]. In contrast, our method uses chance constraints [16] to address this issue explicitly.

Methods for optimizing stochastic contact-implicit trajectories in legged robots, such as Expected Residual Minimization (ERM) and SLCP, are computationally demanding and prone to local minima, making them less suitable for real-time applications [17], [18]. Conversely, our approach reformulates the stochastic optimal control problem into a deterministic convex QP problem, which can be solved at 500 Hz [19].

In contrast to model-based control, model-free RL techniques eliminate the need for an accurate robot model by using domain randomization during training [20]. This approach exposes the control policies to a wide range of scenarios, enhancing their robustness to diverse environments [21]. Nevertheless, these controllers face challenges such as the sim-to-real gap, potentially leading to conservative strategies or failure to handle out-of-distribution disturbances [22]. They are also less intuitive to tune and require more engineering effort compared to model-based approaches [6]. 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

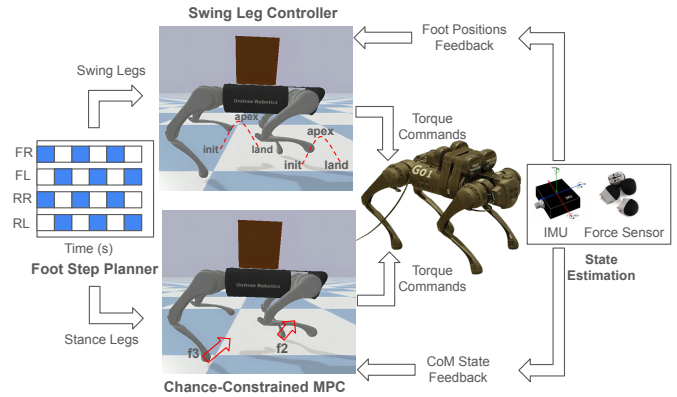


Fig. 2: The modular control architecture adopted in this work for quadrupedal locomotion.

methods for quadrupedal locomotion estimate residual model uncertainties offline using simulation data [23] or online using L1 adaptive control [24]. In this instance, proper initialization of estimated parameters is crucial to avoid instability before online model convergence [25]. 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.

Gazar et al. [26] achieved bipedal walking in simulation despite random forces applied to the robot’s CoM. This was accomplished by using tube MPC to handle additive polytopic uncertainties in the dynamics. However, their approach assumed a constant CoM height and zero angular momentum, limiting movement to planar motion without rotation. Xu et al. [13] addressed this limitation by employing robust min-max MPC for quadrupedal walking. Their method considers bounded disturbances and always plans for the worst-case scenario, resulting in overly conservative behavior [27]. As a result, they had to assign one set of MPC weights for frictional disturbances and another for payload uncertainties. In contrast, SMPC offers greater flexibility than RMPC by allowing small, user-defined probabilities of constraint violation, which mitigates the issue of conservativeness without sacrificing performance [8]. Despite the advantages of SMPC, existing work on quadrupedal robots is mostly confined to simulations and is computationally intensive for real-time deployment [3], [28]–[30]. In this paper, we overcome this computational challenge through an efficient QP implementation. This allows us to deploy the approach on a hardware platform and demonstrate its real-time processing capabilities without the need for hand-tuning, as required in RMPC.

III. CHANCE-CONSTRAINED FOOT FORCE MPC

In this section, we introduce the Chance-Constrained Foot Force MPC 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 [31]. For the swing foot, the Raibert heuristic [12]

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 (EKF) to estimate the center of mass states and leg end positions.

The whole-body dynamics can be expressed using manipulator equations [32]:

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

Here, \mathbf{q} represents the generalized robot pose, \mathbf{M} is the mass matrix, and \mathbf{N} encapsulates other nonlinear terms. The selection matrix \mathbf{S} captures the actuation from the motors, mapping the joint torques to the robot's dynamics. As discussed in [33], 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 [32]. Nevertheless, MPC faces challenges when dealing with substantial modeling errors, which can arise from various factors including diverse payloads and terrains. These modeling errors often lead to instability and deviations from desired trajectories [13].

To address this, we present a stochastic variation of the SRBD model [12] 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 , equations of motion are thus expressed as:

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

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

Here, \mathbf{A}_i is the state transition matrix and $\mathbf{B}_i(\boldsymbol{\delta}_i)$ is a selection matrix that maps the control inputs to the state, where $\boldsymbol{\delta}_i$ represents the distribution of parametric uncertainties. Its mean is given by $\bar{\boldsymbol{\delta}}_i = [m, \text{diag}(\mathbf{I}), \mathbf{r}_{i,1}^T, \dots, \mathbf{r}_{i,4}^T]$, where m is the nominal robot mass and $\text{diag}(\mathbf{I})$ represents the diagonal entries of the nominal inertia matrix. 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 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 [23] that are not accounted for by the nominal SRBD model.

We use CCMPC to capture these sources of uncertainty by incorporating probabilistic descriptions of uncertainties into a stochastic optimal control problem [10]. Though by modeling uncertainties, guaranteeing constraint satisfaction at all times becomes impractical. Chance constraints instead ensure state and control constraints are satisfied with a specified probability, allowing controlled levels of constraint violation [10]. This balance between effective performance and reliability is essential for maintaining consistent operation in uncertain environments.

Specifically, we enforce the linearized friction cone constraints and the unilateral force constraints [34] 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 for the four feet of a quadrupedal robot. Each foot contributes four linearized constraints to approximate the friction cone and one constraint for the unilateral contact force, resulting in a total of 20 constraints. The matrix \mathbf{C}_i maps these constraints to the 12-dimensional control input vector representing ground reaction forces. The term ϵ represents acceptable probability thresholds for constraint satisfaction, allowing a controlled level of constraint violation. Furthermore, we minimize the expected cost, which includes deviations from the desired CoM trajectory and a regularization term on the control inputs to reflect control effort:

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

Here, the expectation is taken with respect to the distributions of the robot state and control actions: $\mathbf{x} \sim (\bar{\mathbf{x}}, \boldsymbol{\Sigma}_x)$ and $\mathbf{u} \sim (\mathbf{v}, \boldsymbol{\Sigma}_u)$, where $\bar{\mathbf{x}}$ and \mathbf{v} denote the means, and $\boldsymbol{\Sigma}_x$ and $\boldsymbol{\Sigma}_u$ denote the covariances. 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}_{ref} , and the positive definite matrix \mathbf{R} weighs the control effort. Finally, we include a constraint using \mathbf{D}_i to ensure that forces are zeroed out for legs not in contact with the ground, as determined by the footstep planner:

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

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

$$\begin{aligned} & \underset{\mathbf{x}_i, \mathbf{u}_i}{\text{minimize}} && \text{Eqn. 4} \\ & \text{subject to} && \text{Eqn. 2, Eqn. 3, Eqn. 5} \end{aligned} \quad (6)$$

IV. CONVEX QP REFORMULATION OF CCMPC

Solving the original CCMPC problem, as expressed in Eqn. 6, is computationally infeasible. This is primarily due to the need for integrating multi-dimensional Gaussian probability density functions to evaluate the chance constraints, which becomes intractable for high-dimensional systems [9]. 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 [3], 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 [35]. Instead, we precompute the gains \mathbf{K}_i using efficient solvers for the Discrete Algebraic Riccati Equation (DARE) [36]. Thus, the decision variables of our optimization problem are \mathbf{v}_i and $\bar{\mathbf{x}}_i$.

Next, we present expressions for the first and second moments of the trajectory distributions. For detailed derivations, interested readers are referred to [10]. Applying the expectation operator to Eqn. 2, we obtain:

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

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

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

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 $\bar{\boldsymbol{\delta}}_i$:

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

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

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

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

B. Friction Cone Constraint Adjustment

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 [32]. This results in five chance constraints per foot that must be jointly satisfied [3]. The Boole-Bonferroni inequality [29] 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.

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

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 that can be computationally expensive [37]. To circumvent this problem, we assign uniform risks α to each constraint, where $\alpha = (1 - \epsilon)/5n$. Hence, Eqn. 12 becomes:

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

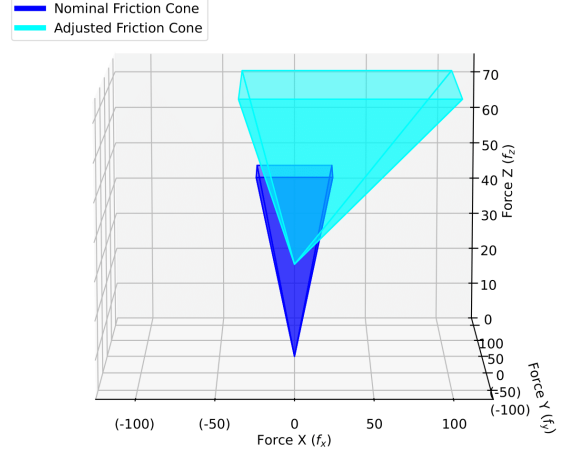


Fig. 3: Illustration of friction cone constraint adjustment.

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 Eqn. 13 can be deterministically reformulated as [38]:

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

where ϕ^{-1} is the inverse of the cumulative distribution function of a standard normal distribution [3]. Applying Eqn. 14 to all rows of \mathbf{C}_i , we get the following constraints on the mean of control actions:

$$\mathbf{C}_i \mathbf{v}_i \leq \mathbf{c}_i, \quad (15)$$

$\mathbf{c}_i = \text{col}(c_i^1, \dots, c_i^{5n})$ 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 inequality constraints, based on the computed \mathbf{c}_i values in Eqn. 15, adjusts the feasible set of ground reaction forces to capture variations in operating conditions and robot dynamics [8]. The extent of this adjustment is determined by the constraint satisfaction threshold ϵ and the control covariance $\boldsymbol{\Sigma}_i^u$ propagated along the MPC horizon. Fig. 3 illustrates this schematically.

Alternatively, we could have also heuristically chosen a constant value for \mathbf{c}_i [3]. In Section V, we demonstrate how this heuristic-based approach results in increased constraint violations, leading to greater instability. Despite the additional computation required to evaluate the \mathbf{c}_i factors for constraint modification, the resultant stance-leg control problem can still be solved efficiently at 500 Hz, making it 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 [39], we use a mean-equivalent approximation of the expected cost along the nominal trajectory:

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

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

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

Algorithm 1 Chance-Constrained MPC

```

Initialize  $\mathbf{c}_i \leftarrow \mathbf{0}$ 
while goal configuration is not reached, do
  for  $i = 0$  to  $N - 1$  do ▷ MPC Loop
    Compute  $\bar{\delta}_i, \mathbf{A}_i, \mathbf{B}_i(\bar{\delta}_i), \mathbf{D}_i$  using Eqn. 2
    Add cost function and constraints to Eqn. 18
  end for
  Output:  $\bar{\mathbf{x}}^*, \mathbf{v}^* \leftarrow$  Solve Eqn. 18 ▷ MPC Output
  Initialize  $\Sigma_{\mathbf{x}} \leftarrow \mathbf{0}$  ▷ Constraint Tightening Loop
  for  $i = 0$  to  $N - 1$  do
     $\mathbf{K}_i \leftarrow$  DARE( $\mathbf{A}_i, \mathbf{B}_i(\bar{\delta}_i), \mathbf{Q}, \mathbf{R}$ )
     $\Sigma_{\mathbf{u}} \leftarrow$  Propagate Control Variance, Eqn. 11
     $\mathbf{c}_i \leftarrow$  Compute Factors, Eqn. 15
     $\Sigma_{\mathbf{x}} \leftarrow$  Propagate State Variance, Eqns. 9, 10
  end for
end while

```

Friction cone constraint tightening is based on the previous MPC solution, allowing these factors to be pre-computed and fixed for real-time optimization [9].

In summary, given a nominal contact plan and a desired CoM trajectory, CCMPC involves solving the deterministic reformulation of the original stochastic optimization problem as the following quadratic program:

$$\begin{aligned} \{\bar{\mathbf{x}}^*, \mathbf{v}^*\} = \underset{\bar{\mathbf{x}}_i, \mathbf{v}_i}{\text{minimize}} \quad & \text{Eqn. 16} \\ \text{subject to} \quad & \text{Eqn. 8, Eqn. 15, Eqn. 17} \end{aligned} \quad (18)$$

Using single shooting [40], 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 [19]. 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.

V. EXPERIMENTS

In this section, we present simulation and hardware experiments evaluating our CCMPC framework. CCMPC effectively tracks the desired height with unmodeled 6 kg payloads and successfully handles loads up to 7.3 kg, exceeding the manufacturer’s recommended payload limit. Extensive indoor and outdoor tests across diverse terrains—including grass, gravel, muddy slopes, stairs, and slippery surfaces—confirm its stability under various payloads. Additional results are available in the supplementary video.

A. Implementation Details

We validate our approach on the Unitree Go1 robot, which has a total mass of 12 kg. The legs contribute around 10% of the total weight, allowing us to approximate the robot’s dynamics with a SRBD model, as in Eqn. 2 [12]. 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 laptop is facilitated using Lightweight Communications and Marshalling

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	15.0
Inertia (X, Y, Z)	0.02, 0.06, 0.06
Angular Velocity (X, Y, Z)	0.5, 0.2, 0.01
Linear Velocity (X, Y, Z)	0.5, 0.2, 0.01
Contact Location (X, Y, Z)	0.36, 0.36, 0.36

(LCM) [41]. The reported runtime of 500 Hz accounts for all the computational steps in the control pipeline. This includes the footstep planner, the inverse kinematics-based swing leg controller, the calculation of constraint adjustment factors, and solving the resulting QP.

We compare our CCMPC approach against two benchmarks: Linear MPC (LMPC), which lacks friction cone constraint tightening, and Heuristic MPC (HMPC), which uses hand-computed constraint tightening factors. The MPC-specific parameters used for these three methods are detailed in Table I. The parametric and additive uncertainties for CCMPC, corresponding to Σ_{δ} and $\Sigma_{\mathbf{w}}$ in Eqn. 2, are shown in Table II.

B. Simulation Analysis

The simulation experiments are conducted using the Py-Bullet simulator [42] with a high-fidelity dynamics model. We conduct Monte Carlo simulations to demonstrate that our control policy can stabilize quadrupedal motion across various payloads and terrains without parameter adjustments. The robot was commanded to move forward at 0.25 m/s, performing blind locomotion over wooden planks with varying payloads, as shown in Fig. 4(a). The payloads were uniformly sampled within a range of 1 to 10 kg, while the plank heights were uniformly sampled from 0 to 5 cm. A total of 1000 samples were generated to introduce sufficient 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/s², 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.

We assessed each controller’s performance using four metrics, logged in Table III. Success rate was determined based on

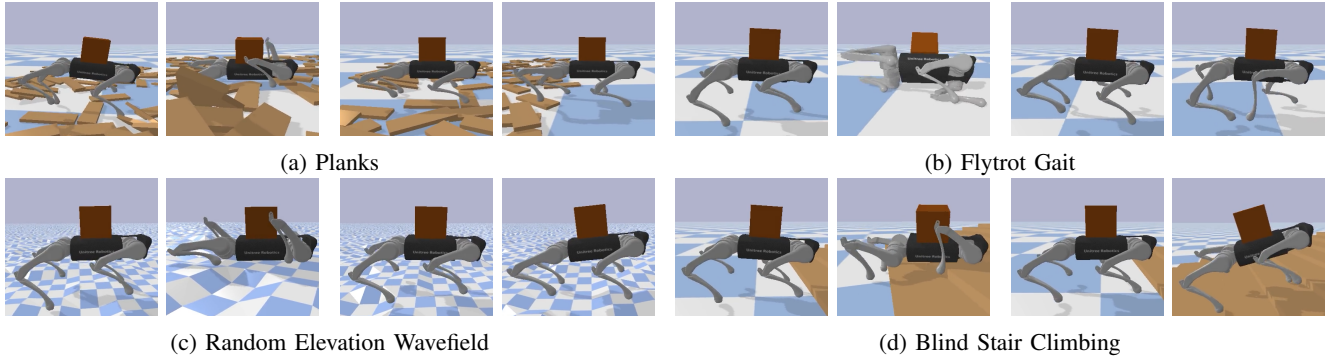


Fig. 4: Comparison of LMPC (left, fails) and CCMPC (right, succeeds) across various gaits and terrains. Similar failures were also observed with HMPC in these experiments.

TABLE III: Results from 1000 Monte-Carlo Simulations.

Metric	Gait	LMPC	HMPC	CCMPC
Success Rate (%) \uparrow	Trot	38.1	77.7	100
	Flytrot	40.3	73.7	100
Average Slippage Ratio \downarrow	Trot	0.26	0.12	0.11
	Flytrot	0.25	0.17	0.15
Normalized Tracking Cost \downarrow	Trot	1.4 ± 0.1	1.1 ± 0.2	1.0 ± 0.1
	Flytrot	2.2 ± 0.4	1.9 ± 0.2	1.0 ± 0.3
Normalized Effort Cost \downarrow	Trot	1.5 ± 0.2	1.3 ± 0.2	1.0 ± 0.2
	Flytrot	1.9 ± 0.3	1.5 ± 0.4	1.0 ± 0.2

three failure criteria: deviations in the base’s height by more than 30% from its desired value, orientation of the robot with respect to the ground normal dropping below 0.8 (indicating a significant tilt), and any infeasibility in solving the QP due to constraint violations. These thresholds were identified during hardware tuning as points where the robot physically fell. The slippage ratio was calculated as $(f_x^2 + f_y^2)/f_z^2$, where f_x , f_y , and f_z represent the ground reaction force components. This ratio represents the degree to which the friction cone was saturated, with higher values indicating worse performance. For the normalized costs, CCMPC was used as the baseline and normalized to 1. The other methods were compared relative to it in two categories: normalized tracking cost, where a higher value indicates poorer tracking of the desired CoM trajectory, and normalized effort cost, where a higher value indicates increased actuation effort. To ensure a fair comparison, the normalized cost and slippage ratio metrics were computed only for successful simulations. Failed simulations were excluded from these metrics to prevent skewed results due to failure conditions.

Our quantitative results show that LMPC had the lowest success rate, highest slippage ratio, and highest normalized cost. HMPC showed improvements over LMPC due to hand-tuned constraint tightening. However, as HMPC maintains a constant level of constraint tightening throughout the MPC horizon, it was less effective in adapting to disturbances, like unmodeled payloads and uneven planks. CCMPC obtained the best performance across both gaits, dynamically adjusting constraints in real-time to balance safety and tracking, thereby minimizing slippage and maintaining stability [3].

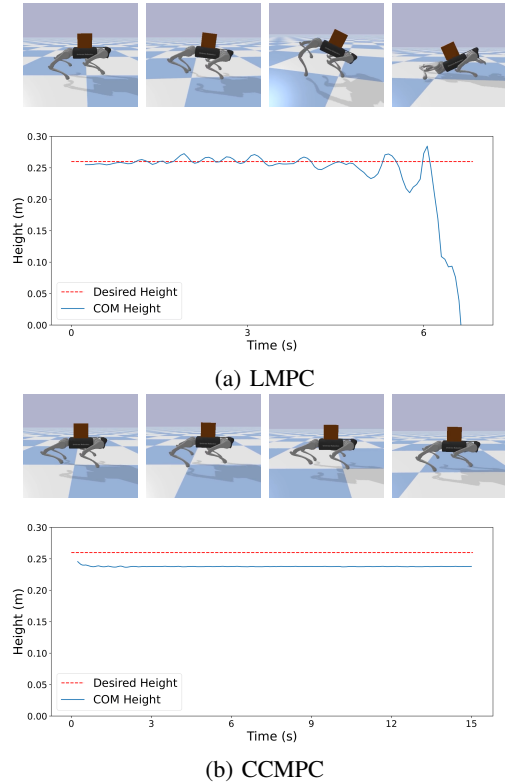


Fig. 5: Simulated height tracking performance with an unmodeled 6 kg payload.

Fig. 5 illustrates the height tracking performance, where unlike CCMPC, LMPC struggles to maintain stability and results in a fall. The slight offset in CoM height under CCMPC arises from prioritizing safety constraints, such as preventing foot slippage, allowing minor deviations in height to ensure robust performance across various conditions.

Fig. 4 demonstrates simulations on various challenging terrains, where the robot traversed random elevation wavefield, randomly placed planks, and performed blind stair climbing. High-speed tests at 1.75 m/s were also conducted using the flytrot gait. In all these scenarios, the robot performed blind locomotion while carrying a fixed 6 kg payload. Both LMPC and HMPC struggled to maintain stability, while CCMPC successfully maintained stability and tracking, effectively handling unmodeled terrain and payload uncertainties.

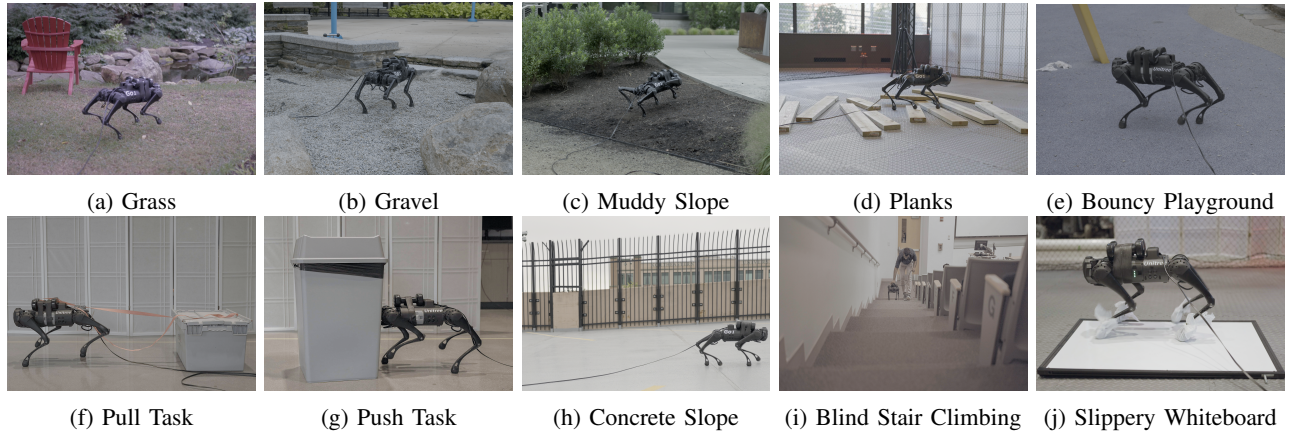


Fig. 6: Hardware validation of CCMPC across different locomotion tasks and terrains. Video demonstrations can be found at <https://cc-mpc.github.io/>.

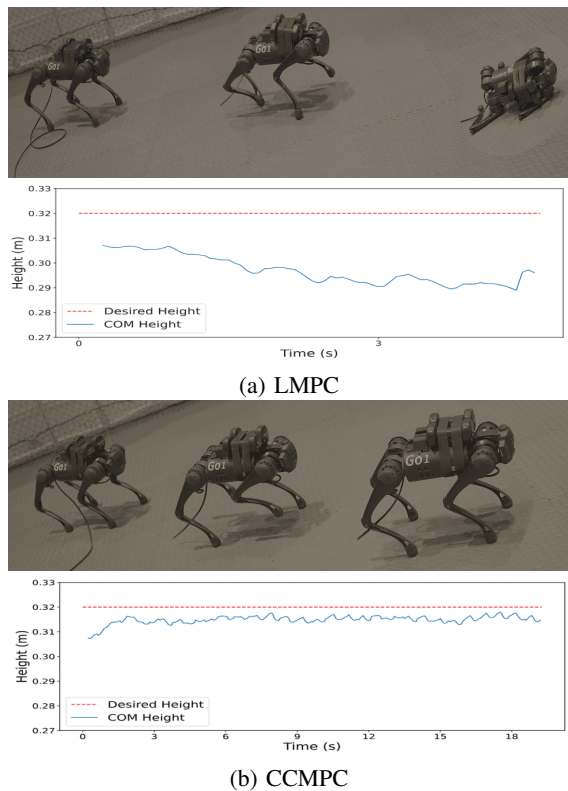


Fig. 7: Hardware height tracking performance with an unmodeled 6 kg payload.

C. 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. 6. The robot, loaded with additional unknown 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 these surfaces, showcasing its ability to handle diverse ground textures and maintain stability on slippery and uneven surfaces.

To demonstrate the repeatability of our control strategy, we conducted a series of progressive load tests, as shown in Fig. 7. In these experiments, 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 and 7.3 kg payloads, even exceeding the manufacturer’s recommended maximum payload of 5.0 kg. More hardware demonstrations can be found in the supplementary video, which provides a comprehensive overview of the robot’s performance across these varied conditions.

VI. CONCLUSION

In this paper, we presented a Chance-Constrained Model Predictive Control (CCMPC) framework to generate optimal ground reaction forces for quadrupedal locomotion. CCMPC generates a unified control policy to stabilize quadrupedal locomotion across a wide range of unmodeled payloads and varying terrain conditions in real-time with minimal parameter tuning. The effectiveness of our approach was validated through extensive simulation and hardware experiments.

However, some limitations were observed, particularly with heavier payloads. Tightening control constraints in anticipation

of added mass can lead to QP infeasibility. To address this, future work will explore machine learning methods [43], [44] to actively learn payload dynamics. Additionally, we aim to extend this approach to loco-manipulation tasks in bipedal robots handling unknown payloads [45].

VII. ACKNOWLEDGMENTS

This work is partially supported by the Impact Engines program at Northeastern University.

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. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.
- [3] 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.
- [4] J. N. Nganga and P. M. Wensing, "Accelerating second-order differential dynamic programming for rigid-body systems," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7659–7666, 2021.
- [5] 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.
- [6] 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.
- [7] A. Prékopa, *Stochastic programming*. Springer Science & Business Media, 2013, vol. 324.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 1999.
- [12] J. Di Carlo, P. M. Wensing, B. Katz, G. Bleidt, 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.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] M. Sombolstan and Q. Nguyen, "Adaptive force-based control of dynamic legged locomotion over uneven terrain," *IEEE Transactions on Robotics*, 2024.
- [25] I. D. Landau, R. Lozano, M. M'Saad *et al.*, *Adaptive control*. Springer New York, 1998, vol. 51.
- [26] 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.
- [27] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in identification and control*. Springer, 2007, pp. 207–226.
- [28] 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.
- [29] 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.
- [30] H. Zhao and M. Khadiv, "Trajectory optimization under contact timing uncertainties," *arXiv preprint arXiv:2407.11478*, 2024.
- [31] Y. Yang, T. Zhang, E. Coumans, J. Tan, and B. Boots, "Fast and efficient locomotion via learned gait transitions," in *Conference on robot learning*. PMLR, 2022, pp. 773–783.
- [32] 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.
- [33] 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.
- [34] P.-B. Wieber, R. Tedrake, and S. Kuindersma, "Modeling and control of legged robots," in *Springer handbook of robotics*. Springer, 2016, pp. 1203–1234.
- [35] S. Dempe and A. Zemkoho, "Bilevel optimization," in *Springer optimization and its applications*. Springer, 2020, vol. 161.
- [36] R. Tedrake *et al.*, "Drake: A planning, control, and analysis toolbox for nonlinear dynamical systems," 2014.
- [37] 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.
- [38] 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.
- [39] 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.
- [40] 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.
- [41] 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.
- [42] E. Coumans and Y. Bai, "Pybullet quickstart guide," 2021.
- [43] A. Trivedi, S. Bazzi, M. Zolotas, and T. Padir, "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.
- [44] 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," *arXiv preprint arXiv:2402.18065*, 2024.
- [45] 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.