

# Robustifying Model-Based Locomotion by Zero-order Stochastic Nonlinear Model Predictive Control with Guard Saltation Matrix

Sotaro Katayama<sup>1</sup>, Noriaki Takasugi<sup>1</sup>, Mitsuhiro Kaneko<sup>2</sup>, Norio Nagatsuka<sup>3</sup>, and Masaya Kinoshita<sup>1</sup>

**Abstract**—This paper presents a stochastic/robust nonlinear model predictive control (NMPC) to enhance the robustness of model-based legged locomotion against contact uncertainties. We integrate the contact uncertainties into the covariance propagation of stochastic/robust NMPC framework by leveraging the guard saltation matrix and an extended Kalman filter-like covariance update. We achieve fast stochastic/robust NMPC computation by utilizing the zero-order algorithm with additional improvements in computational efficiency concerning the feedback gains. We conducted numerical experiments and demonstrate that the proposed method can accurately forecast future state covariance and generate trajectories that satisfies constraints even in the presence of the contact uncertainties. Hardware experiments on the perceptive locomotion of a wheeled-legged robot were also carried out, validating the feasibility of the proposed method in a real-world system with limited on-board computation.

## I. INTRODUCTION

A fundamental challenge in model-based legged locomotion is ensuring robustness against uncertainties that arise in contacts with the environment. When traversing uneven terrains, robots must determine actions based on information acquired from exteroceptive sensors such as RGB-D cameras or LiDAR. However, the accuracy of this environmental information, such as the terrain height surrounding the robot, is imperfect due to limitations of sensor resolution, occlusions, or non-flat terrains. Model-based control methods, such as model predictive control (MPC), have struggled with estimation inaccuracies in environmental information. For instance, if the estimated terrain height is higher than the actual height, a swinging foot may fail to establish contact, leading to the robot's instability and potential falls.

In contrast to model-based control methods, model-free reinforcement learning (RL) offers the ability to incorporate various environmental uncertainties into control policies. In a study by Miki et al. [1], the authors propose explicitly encoding uncertainties in exteroceptive sensors, such as the terrain height map around the robot, into the training process. By employing domain randomization to account for perception uncertainties and environmental variations, the policy achieves the capability to traverse challenging real-world environments. This RL approach to mitigating uncertainties prompts a question: How can model-based control methods achieve robustness against contact uncertainties?

<sup>1</sup>Sony Group Corporation, Minato-ku, Tokyo, Japan, 108-0075  
sotaro.katayama@sony.com

<sup>2</sup>Sony Global Manufacturing and Operations Corporation, Minato-ku, Tokyo, Japan, 108-0075

<sup>3</sup>Sony Interactive Entertainment Inc., Sony City 1-7-1, Konan, Minato-ku, Tokyo, 108-0075 Japan

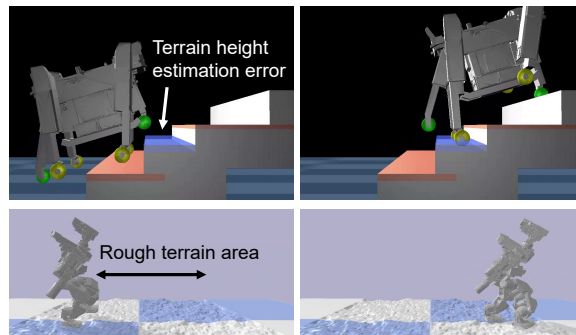


Fig. 1: Simulations utilizing the proposed stochastic nonlinear model predictive control. Top row: Tachyon 3, a six-telescopic-wheeled-legged robot, ascends stairs under terrain height estimation errors. Bottom row: EVAL-03, a miniature humanoid robot, traverses a rough terrain area.

In this paper, we explore a model-based control approach that explicitly considers contact uncertainties in legged locomotion. Specifically, we incorporate the contact uncertainties into stochastic/robust nonlinear MPC (NMPC) by leveraging the guard saltation matrix introduced by Payne et al. [2] and the extended Kalman filter (EKF)-like covariance propagation proposed by Zhang and Ohtsuka [3]. We then apply the zero-order stochastic/robust NMPC algorithm [4], [5], [3] with additional enhancements to computational efficiency regarding the feedback gains. Our stochastic/robust NMPC ensures robustness to contact uncertainty via online optimization, guaranteeing the satisfaction of constraints such as the friction cone, collision avoidance, and joint limits even in the presence of disturbances in the contact events.

### A. Related Works

Model-based control methods for legged robots, predominantly MPC, inevitably involve uncertainties. Even besides the contact-related ones, there are a variety of uncertainties such as model approximations (e.g., single rigid-body approximation), model parameters (e.g., inertias), and unmodeled effects (e.g., actuator dynamics). Stochastic/robust MPC is a promising framework to cope with these uncertainties [6]. Stochastic/robust MPC predicts the uncertainties in the system state over the horizon and tightens constraints to ensure that the constraints are satisfied even in the presence of disturbances. Despite the potential computational burden, the advent of zero-order algorithms has paved the way for practical applications [4], [5], [3].

In [7], [8], the stochastic/robust MPC techniques have

been successfully applied for legged locomotion. Xu et al. [7] employ linear robust MPC to account for uncertainties arising from single rigid-body and linear approximations of the system dynamics. Gazar et al. [8] also utilize stochastic NMPC to enhance robustness against uncertainties in the continuous-time system dynamics. However, these studies cannot capture the uncertainties involved in the contact events. Incorporating uncertainties related to contacts into stochastic NMPC is not straightforward compared to simple process noise in continuous-time system dynamics.

The saltation matrix, a mathematical tool to analyze the sensitivity of the state with respect to discrete events, can be a promising tool to capture the uncertainties in the contact events [9]. It has been utilized in offline trajectory optimization [10], (deterministic) contact-implicit MPC [9], and state estimation [2]. Notably, Payne et al. [2] propose the guard saltation matrix, an extension of the original saltation matrix that can incorporate uncertainty in the guard condition, such as terrain height in the case of legged locomotion.

## B. Contribution

In this paper, we present a stochastic/robust NMPC against contact uncertainties for legged locomotion. Our contributions can be summarized as follows:

- A novel covariance propagation for stochastic/robust NMPC against contact uncertainties leveraging a priori covariance update using the guard saltation matrix and EKF-like a posteriori covariance update.
- An efficient numerical algorithm for stochastic/robust NMPC utilizing the zero-order algorithm and interior point method.
- Application of the proposed method to the full-centroidal NMPC formulation.
- Performance comparison between different NMPC methods through simulations on two robotic platforms.
- Hardware experiments on perceptive locomotion over uneven terrains with on-board implementation of the proposed stochastic/robust NMPC.

This paper is organized as follows. Section II briefly summarizes the saltation matrices for hybrid systems. Section III proposes a stochastic/robust NMPC framework leveraging the saltation matrices and EKF-like covariance update for contact uncertainties. Section IV introduces an efficient computation strategy to implement the proposed stochastic/robust NMPC. Section V investigates the effectiveness of the proposed approach through simulations and hardware experiments. Finally, Section VI concludes this paper and discusses future works.

## II. HYBRID SYSTEM AND GUARD SALTATION MATRIX

In this section, we summarize the guard saltation matrix introduced in [2] that characterizes a certain aspect of legged locomotion. To begin, we introduce a mode index  $m$  to represent the combination of contact flags (on/off) for all feet. For the state  $x(t) \in \mathbb{R}^{n_x}$  and control input  $u(t) \in \mathbb{R}^{n_u}$ ,

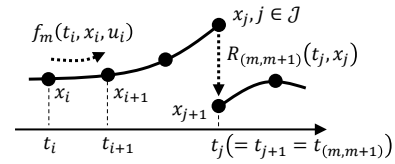


Fig. 2: Discretization of the optimal control problem.

the continuous-time state equation, or *flow map*, for mode  $m$  is defined as follows:

$$\dot{x}(t) = f_m(t, x(t), u(t)). \quad (1)$$

The switch from mode  $m$  to mode  $m+1$  occurs when the system satisfies the *guard condition*:

$$g_{(m,m+1)}(t_{(m,m+1)}^-, x(t_{(m,m+1)}^-)) = 0, \quad (2)$$

where  $t_{(m,m+1)}^-$  denotes the instant immediately before the switch from mode  $m$  to mode  $m+1$ . At the instance of the switch, the state undergoes an instantaneous change, or a state jump, represented by the *reset map*:

$$x(t_{(m,m+1)}^+) = R_{(m,m+1)}(t_{(m,m+1)}^-, x(t_{(m,m+1)}^-)), \quad (3)$$

where  $t_{(m,m+1)}^+$  is the instant immediately after the switch. The guard condition (2) represents, for example, the distance between a foot and the ground, while the reset map (3) can capture impulsive dynamics due to collisions [11].

The original saltation matrix [11] describes the first-order sensitivity of  $x(t_{(m,m+1)}^+)$  with respect to  $x(t_{(m,m+1)}^-)$  in cases where (2) is one-dimensional, i.e., (2) is a scalar-valued function. Specifically, the saltation matrix for the switch from mode  $m$  to  $m+1$  is defined as [11]:

$$\Xi_{x,(m,m+1)} := \nabla_x R_{(m,m+1)} + \frac{f_{m+1}^+ - \nabla_x R_{(m,m+1)} f_m^- - \nabla_t R_{(m,m+1)} \nabla_x g_{(m,m+1)}}{\nabla_t g_{(m,m+1)} + \nabla_x g_{(m,m+1)} f_m^-}, \quad (4)$$

where  $f_m^- := f_m(t_{(m,m+1)}^-, x(t_{(m,m+1)}^-), u(t_{(m,m+1)}^-))$  and  $f_{m+1}^+ := f_{m+1}(t_{(m,m+1)}^+, x(t_{(m,m+1)}^+), u(t_{(m,m+1)}^+))$ , and we drop the obvious arguments from  $g_{(m,m+1)}$  and  $R_{(m,m+1)}$ . Similarly, the guard saltation matrix [2] describes the first-order sensitivity of  $x(t_{(m,m+1)}^+)$  with respect to the guard condition and is defined as:

$$\Xi_{g,(m,m+1)} := \frac{\nabla_x R_{(m,m+1)} f_m^- + \nabla_t R_{(m,m+1)} - f_{m+1}^+}{\nabla_t g_{(m,m+1)} + \nabla_x g_{(m,m+1)} f_m^-}. \quad (5)$$

Note that, to define these saltation matrices, the following transversality condition must be hold:

$$\nabla_t g_{(m,m+1)} + \nabla_x g_{(m,m+1)} f_m^- < 0. \quad (6)$$

## III. STOCHASTIC/ROBUST MODEL PREDICTIVE CONTROL WITH SALTATION MATRIX

### A. Problem Settings

We formulate the optimal control problem (OCP) for a legged robot, assuming a predetermined contact sequence

composed of a mode sequence  $1, 2, \dots, M$  and corresponding switching times  $t_{(1,2)}, \dots, t_{(M-1,M)}$ . The finite-time horizon is discretized into a  $N + 1$  ( $N > 0$ ) nodes whose instants are given by  $[t_0, t_1, \dots, t_N]$ . We define  $\mathcal{J}$  as the set of jump nodes  $j$  where a state jump occurs between  $j$  and  $j + 1$ , i.e.,  $t_j = t_{j+1} = t_{(m,m+1)}$ , with  $m$  as the corresponding mode index. Additionally, we define  $\mathcal{I} := \{0, 1, \dots, N - 1\} \setminus \mathcal{J}$ . Employing the direct multiple-shooting method [12], we introduce state trajectory  $x_0, \dots, x_N \in \mathbb{R}^{n_x}$  and control inputs trajectory  $u_i \in \mathbb{R}^{n_u}$  for all  $i \in \mathcal{I}$ . The definition of the variables in our discretization are summarized in Fig. 2.

In stochastic/robust settings, we account for process noise in the discrete-time state equation. With a process noise trajectory  $w_0, \dots, w_{N-1} \in \mathbb{R}^{n_w}$ , the continuous-time state equation (1) is discretized into the following discrete-time state equation:

$$x_{i+1} = F_i(t_i, x_i, u_i, w_i), \quad i \in \mathcal{I}. \quad (7)$$

For instance, using the forward Euler method, the discrete-time state equation is:

$$F_i(t_i, x_i, u_i, w_i) := x_i + f_m(t_i, x_i, u_i) \Delta \tau_i + \Gamma_i w_i, \quad i \in \mathcal{I},$$

where  $\Gamma_i \in \mathbb{R}^{n_x \times n_w}$  is a constant matrix,  $m$  is the appropriate mode index, and  $\Delta \tau > 0$  is the time step. We also consider disturbances in the state jump:

$$x_{j+1} = F_j(t_j, x_j, w_j) := R_{(m,m+1)}(t_j, x_j) + \Gamma_j w_j, \quad j \in \mathcal{J}. \quad (8)$$

Following the literature on stochastic/robust NMPC [6], [4], [5], we assume that the state is represented by a Gaussian distribution. We introduce the mean and covariance of the state trajectory  $\bar{x}_0, \dots, \bar{x}_N \in \mathbb{R}^{n_x}$  and  $P_0, \dots, P_N \in \mathbb{R}^{n_x \times n_x}$ . The covariance propagation is approximated as:

$$\begin{aligned} P_{i+1} &= (A_i + B_i K_i) P_i (A_i + B_i K_i)^T + \Gamma_i W_i \Gamma_i^T \\ &:= \Phi_i(P_i, K_i, \bar{x}_i, u_i), \quad i \in \mathcal{I}, \end{aligned} \quad (9)$$

where  $A_i := \nabla_x F_i(t_i, \bar{x}_i, u_i, 0)$ ,  $B_i := \nabla_u F_i(t_i, \bar{x}_i, u_i, 0)$ ,  $\Gamma_i := \nabla_w F_i(t_i, \bar{x}_i, u_i, 0)$ ,  $W_i \in \mathbb{R}^{n_w \times n_w}$  is a symmetric positive semi-definite matrix, and  $K_i \in \mathbb{R}^{n_u \times n_x}$  is a state feedback gain. Equation (9) refers to closed-loop covariance propagation with an appropriate feedback gain  $K_i$ , and reduces to open-loop propagation with  $K_i = O$ . Closed-loop propagation can mitigate the overgrowth of state covariance and result in less conservative constraints [5], but requires optimization of the feedback gain  $K_i$ . In robust MPC,  $W_i$  describes an ellipsoid containing the process noise  $w_i$  [4], [5]. In stochastic MPC,  $W_i$  describes the noise distribution covariance, such as  $w_i \sim \mathcal{N}(0, W_i)$  [6].

### B. Covariance Propagation Between a Contact Event

Covariance propagation between mode switches (contact events) is crucial for handling contact uncertainties. We propose a novel covariance update consisting of an *a priori* update using saltation matrices and an *a posteriori* update akin to the EKF.

1) *A priori covariance update using Guard Saltation Matrices*: We utilize saltation matrices (4) and (5) in covariance propagation to introduce contact uncertainties. When the guard condition (2) is one-dimensional, as in EKF applications [2], we update the covariance by

$$\begin{aligned} \hat{P}_{j+1} &= \Xi_{x,(m,m+1)} P_j \Xi_{x,(m,m+1)}^T \\ &\quad + \Xi_{g,(m,m+1)} C_g \Xi_{g,(m,m+1)}^T, \end{aligned} \quad (10)$$

where  $C_g \in \mathbb{R}$  denotes the covariance of  $g_{(m,m+1)}$ . However, typical gaits such as quadrupedal trotting, several feet make contact simultaneously at the same instant, which requires multi-dimensional guard conditions.

Unfortunately, literature on the saltation matrices such as [2] only treat the cases where the guard condition (2) is one-dimensional. To apply the saltation matrix even with the multi-dimensional guard conditions whose dimension is  $n_g > 0$ , we update the covariance as:

$$\begin{aligned} \hat{P}_{j+1} &= \sum_{i=1}^{n_g} \left\{ \Xi_{x,(m,m+1)}^{(i)} P_j \Xi_{x,(m,m+1)}^{(i)T} \right. \\ &\quad \left. + \Xi_{g,(m,m+1)}^{(i)} C_{g^{(i)}} \Xi_{g,(m,m+1)}^{(i)T} \right\}, \end{aligned} \quad (11)$$

where  $\Xi_{x,(m,m+1)}^{(i)}$  and  $\Xi_{g,(m,m+1)}^{(i)}$  are the saltation matrices for the  $i$ -th element of  $g_{(m,m+1)}$ , and  $C_{g^{(i)}} \in \mathbb{R}$  is the covariance of the  $i$ -th element of the guard condition  $g_{(m,m+1)}$ . This update represents a summation of the one-dimensional update (10) for each element of the guard condition. In practice,  $g_{(m,m+1)}^{(i)}$  does not necessarily satisfy the transversality condition (6). We omit such index  $i$  from the summation of (11). While this approach was effective in our experiments, theoretically-supported methods such as using the Bouligand derivative [13] are left for future work.

Note that computing (4) and (5) requires values of  $f_m^-$  and  $f_{m+1}^+$ . We approximate them by using variables on the nearest discretization nodes, i.e., by  $f_m(t_{j-1}, x_{j-1}, u_{j-1})$  and  $f_{m+1}(t_{j+1}, x_{j+1}, u_{j+1})$ , respectively.

2) *A posteriori covariance update using output-feedback MPC*: The update (11) is an open-loop covariance update and may lead to overgrowth of the covariance. To address this, we apply an a posteriori covariance update from output-feedback stochastic MPC [3]. For the stochastic process whose belief dynamics are (7) and (8), the guard condition (2), i.e., the contact positions, can be regarded an output equation, as is well-known in the EKF-based state estimation for the legged robots [14]. Treating the guard condition (2) as an output equation, we apply the a posteriori covariance update law proposed in [3], resulting in:

$$\begin{aligned} L_{j+1} &:= \hat{P}_{j+1} \nabla_x g_{(m,m+1)}^T \\ &\quad (\nabla_x g_{(m,m+1)} \hat{P}_{j+1} \nabla_x g_{(m,m+1)}^T + C_g)^{-1}, \end{aligned} \quad (12)$$

$$P_{j+1} = (I - L_{j+1} \nabla_x g_{(m,m+1)}) \hat{P}_{j+1}, \quad (13)$$

where  $\hat{P}$  is the covariance after the update by (11). We summarize the covariance update from  $P_j$  to  $P_{j+1}$  using (11), (12), and (13) into  $P_{j+1} = \Phi_j(P_j, \bar{x}_j)$ . This a posteriori

update can also be applied for  $i \in \mathcal{I}$  to suppress the covariance overgrowth, although this paper focuses on the update between a contact event.

### C. Stochastic/Robust Optimal Control Problem

With the settings described, the stochastic/robust OCP is formulated as:

$$\min_{\substack{\bar{x}_0, \dots, \bar{x}_N, \\ \bar{u}_0, \dots, \bar{u}_{N-1}, \\ P_0, \dots, P_N}} J := V_N(\bar{x}_N) + \sum_{i \in I} l_i(\bar{x}_i, \bar{u}_i) + \sum_{j \in J} l_j(\bar{x}_j) \quad (14a)$$

$$\text{s.t. } \bar{x}_0 = \bar{x}, \quad (14b)$$

$$\bar{x}_{i+1} = F_i(t_i, \bar{x}_i, \bar{u}_i, 0), \quad i \in I, \quad (14c)$$

$$\bar{x}_{j+1} = F_j(t_j, \bar{x}_j, 0), \quad j \in J, \quad (14d)$$

$$P_{i+1} = \Phi_i(P_i, K_i, \bar{x}_i, \bar{u}_i), \quad i \in I, \quad (14e)$$

$$P_{j+1} = \Phi_j(P_j, \bar{x}_j), \quad j \in J, \quad (14f)$$

$$g_j(\bar{x}_j) = 0, \quad j \in J, \quad (14g)$$

$$h_i(\bar{x}_i, \bar{u}_i) + \beta_i(P_i, K_i, \bar{x}_i, \bar{u}_i) \geq 0, \quad i \in I, \quad (14h)$$

$$h_j(\bar{x}_j) + \beta_j(P_j, \bar{x}_j) \geq 0, \quad j \in J, j = N, \quad (14i)$$

where  $V_N(\cdot)$  denotes the terminal cost,  $l_i(\cdot)$  and  $l_j(\cdot)$  denote the stage costs, and  $h_i(\cdot)$  and  $h_j(\cdot)$  denote the inequality constraint, respectively.  $\beta_i(\cdot)$  and  $\beta_j(\cdot)$  are backoff terms introduced to ensure that the original inequality constraints  $h_i(\cdot) \geq 0$  and  $h_j(\cdot) \geq 0$  are satisfied under the predicted state covariances  $P_i$  and  $P_j$ , respectively, and are defined as:

$$\beta_i(P_i, K_i, \bar{x}_i, \bar{u}_i) := \gamma_i \sqrt{\nabla_{x,u} h_i(x_i, u_i)^T \begin{bmatrix} I \\ K_i \end{bmatrix} P_i \begin{bmatrix} I \\ K_i \end{bmatrix}^T \nabla_{x,u} h_i(x_i, u_i)}, \quad (15)$$

$$\beta_j(P_j, \bar{x}_j) := \gamma_j \sqrt{\nabla_x h_j(x_j)^T P_j \nabla_x h_j(x_j)}. \quad (16)$$

In robust MPC,  $\gamma_i$  is typically set to 1.0. In stochastic MPC, for a chance constraint:

$$\Pr[h_i(x_i, u_i) \geq 0] > p_i, \quad (17)$$

where,  $1.0 > p_i > 0.5$  is the probability of constraint satisfaction, the constraint is approximated as a deterministic constraint with  $\gamma_i := \Psi^{-1}(p_i)$ , where  $\Psi^{-1}(\cdot)$  is the inverse cumulative density function of a standard normal Gaussian distribution [3]. Note that  $\Psi^{-1}(\cdot)$  can be computed numerically offline.

## IV. EFFICIENT STOCHASTIC/ROBUST MODEL PREDICTIVE CONTROL ALGORITHM

### A. Zero-Order Algorithm for Covariance Propagation

The zero-order algorithm of stochastic/robust NMPC [4], [5], [3] simplifies the Newton-type iteration for the OCP (14) by disregarding the sensitivities of the covariance propagation equations (14e) and (14f) and the sensitivities of the backoff terms  $\beta(\cdot)$ . The computational burden per Newton-type iteration is competitive with the nominal NMPC.

A limitation of the zero-order algorithm is that its convergence rate may slow down when the process noise covariance  $W_i$  is substantial, as the backoff terms have a large effect on the convergence. We empirically found that for efficient optimization,  $W_i$  should be carefully tuned to moderate the covariance propagation rather than accurately reflecting the process noise covariance of the actual system.

### B. Interior Point Method and Feedback Gain

We employ the nonlinear primal-dual interior point method to handle hard inequality constraints [15]. It should be noted that state-only inequality constraints need to be treated as soft constraints such as [16] to prevent the backoff term (15) from becoming excessively large, which leads to no feasible solution satisfying the tightened constraints (14h)–(14i) [4]. In practice, we also might have to clip the maximum values of the backoff terms to avoid ill-conditioned optimization problems.

The Riccati recursion [17], used in conjunction with the interior point method and soft constraints, allows for efficient computation of the Newton step and provides the feedback gain trajectory  $K_0, \dots, K_{N-1}$  as a by-product. This method facilitates the efficient online update of the feedback gain during MPC application, particularly without additional computations than the Newton-step computation solely to retrieve the state feedback gains or backoff terms, which are required in the previous studies [5], [7].

In applying the proposed method in MPC, the optimal feedback gain trajectory  $K_0, \dots, K_{N-1}$  is stored after each MPC iteration. At the next time step, the feedback gain trajectory is estimated (i.e., warmstarted) using the stored feedback gain trajectory as well as the state and control input trajectories, for example, through linear interpolation.

### C. Algorithm Summary

Algorithm 1 outlines the proposed zero-order stochastic/robust NMPC with the interior point method. The algorithm begins by initializing (warmstarting) state, control input, and feedback gain trajectories using linear interpolation of the previous NMPC solution (line 1). It then iterates (lines 3–10) until termination criteria are met (line 2).

When computational time of the online optimization is limited, the Newton-type iteration (lines 3–10) may be executed only once per sampling time, as in the real-time iteration scheme [18]. We also empirically found that fixing the barrier parameter of the interior point method and penalty parameters of the soft constraints such as [16] can enable warmstart of the solution and improve the computational speed, albeit potentially at a slight cost to control performance. Throughout the following simulation and hardware experiments, the Newton-type iteration was performed only once per sampling time and the barrier and penalty parameters of hard and soft constraints, respectively, were fixed.

## V. EXPERIMENTS

### A. Full-Centroidal Formulations

To evaluate the effectiveness of the proposed stochastic/robust NMPC, we conducted simulation and hardware

---

**Algorithm 1** Zero-Order Stochastic/Robust NMPC

---

**Input:** Current time  $t$  and current state  $x(t)$ **Output:** Optimal trajectories  $x_0, \dots, x_N$  and  $u_0, \dots, u_{N-1}$ 

- 1: Initialize  $x_0, \dots, x_N$ ,  $u_0, \dots, u_{N-1}$ , and  $K_0, \dots, K_{N-1}$  by the linear interpolation of the previous NMPC solution.
  - 2: **while** termination criteria are not satisfied **do**
  - 3:   Compute the values and partial derivatives of  $V_N(\cdot)$ ,  $l_i(\cdot)$ ,  $l_j(\cdot)$ ,  $F_i(\cdot)$ ,  $F_j(\cdot)$ ,  $g_j(\cdot)$ ,  $h_i(\cdot)$ , and  $h_j(\cdot)$ .
  - 4:   Propagate the state covariances by (14e) and (14f).
  - 5:   Compute backoff terms  $\beta_i(\cdot)$  by (15) and (16).
  - 6:   Reduce the inequality constraints into the cost terms, e.g., via nonlinear interior point method.
  - 7:   Perform the Riccati recursion to compute the search directions  $\Delta x_0, \dots, \Delta x_N$ ,  $\Delta u_0, \dots, \Delta u_{N-1}$  and feedback gains  $K_0, \dots, K_{N-1}$ .
  - 8:   Find a step-size via line-search (optional).
  - 9:   Update and store  $x_0, \dots, x_N$  and  $u_0, \dots, u_{N-1}$ .
  - 10:   Update the barrier parameter (optional).
  - 11: **end while**
- 

experiments on two distinct robotic platforms: the six-telescopic-wheeled-legged robot Tachyon 3<sup>12</sup> [19] and the miniature humanoid robot EVAL-03<sup>3</sup> [20]. Both platforms are depicted in Fig. 1. Within the stochastic/robust NMPC framework, we employed the full-centroidal NMPC formulation [21], [22], [23], which offers a practical trade-off between model fidelity and computational tractability. Let  $n_J$  be the number of joints of the robot. The state  $x$  and control input  $u$  are defined as  $x := [q^T \ h^T]^T$  and  $u := [\dot{q}_J^T \ f^T]^T$ , respectively, where  $q \in \mathbb{R}^{6+n_J}$  is the configuration,  $h \in \mathbb{R}^6$  is the normalized centroidal momentum,  $\dot{q}_J \in \mathbb{R}^{n_J}$  is the joint velocities, and  $f$  denotes the stack of the contact forces.

Since  $\dot{q}_J$  is included in the control input, state jumps (i.e., impulsive changes of generalized velocity) are not assumed, i.e., the reset map and its derivative are given as:

$$x(t_{(m,m+1)}^+) = x(t_{(m,m+1)}^-), \quad \nabla_x R_{(m,m+1)} = I_{n_x \times n_x}. \quad (18)$$

Note that, although the reset map is the identity, the saltation matrix (4) is not the identity, and the guard saltation matrix (5) is not zero, due to the different values of  $f_m^-$  and  $f_{m+1}^+$ . The guard condition consists of the distances between the swing feet and their targeted contact surfaces. Given a configuration  $q$ , the guard condition and its derivatives can be computed using forward kinematics and its Jacobian.

For the covariance propagation equation (9), we assume:

$$\Gamma_i := \begin{bmatrix} O_{(6+n_J) \times 6} \\ I_{6 \times 6} \end{bmatrix}, \quad W_i := \sigma I_{6 \times 6}. \quad (19)$$

<sup>1</sup>Website: [https://www.sony.com/en/SonyInfo/research/technologies/new\\_mobility](https://www.sony.com/en/SonyInfo/research/technologies/new_mobility)

<sup>2</sup>Video: <https://youtu.be/sorw7o73ydc>

<sup>3</sup>Sony Interactive Entertainment, "EVAL-03", Press: <https://www.sony.com/en/SonyInfo/News/Press/202210/22-037E/>

The parameter  $\sigma$  is set to  $1.0 \times 10^{-6}$  for Tachyon 3 and  $1.0 \times 10^{-8}$  for EVAL-03, to reflect the disparate scales and dynamics of the robots.

The constraints and cost function design for Tachyon 3's NMPC adhere to our preceding work [23]: we enforced joint position, velocity, and torque limits, friction cone, and foot placement constraints as inequality constraints. Furthermore, we incorporated collision avoidance constraints between the feet of Tachyon 3 and the environment, which was modeled by a finite number of cuboids.

The NMPC formulation for EVAL-03 largely follows [21], [22] while a surface contact of a foot is modeled by four point contacts at the vertices. We imposed joint position, friction cone, and self-collision avoidance constraints between the left and right legs as inequality constraints.

### B. Comparison in Covariance Prediction

Before conducting simulation and hardware experiments, we assessed how the proposed covariance propagation (11)–(13) predicts future state covariance under contact uncertainties within the stochastic/robust NMPC framework. We compared the following three methods for covariance propagation with the above NMPC settings of Tachyon 3:

- (a) The proposed a priori and a posteriori covariance update (11)–(13).
- (b) The proposed a priori covariance update (11).
- (c) Covariance update using only the dynamics, i.e., the reset map, in the form of (9).

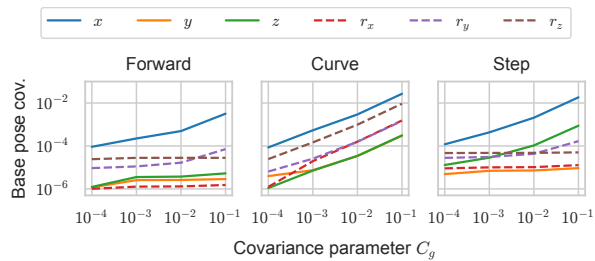
Specifically, case (c) employs the covariance update between contact events as:

$$P_{j+1} = \nabla_x R_{(m,m+1)} P_j \nabla_x R_{(m,m+1)}^T + \Gamma_j W_j \Gamma_j^T, \quad (20)$$

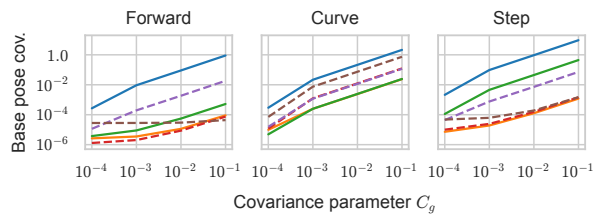
where  $\Gamma_j$  and  $W_j$  are in the same form as (19). We evaluated the state covariance predicted at the terminal time of the horizon (set to 2.5 s) after solving an offline trajectory optimization problem with different parameters for covariance prediction in each algorithm. In cases (a) and (b), we varied the covariance parameter of the guard conditions, i.e.,  $C_g$ . In case (c), we varied the covariance parameter  $W_j$  in (20) (i.e.,  $\sigma$  in (19) for  $i = j \in \mathcal{J}$ ). We investigated three types of trotting<sup>3</sup> motions in Tachyon 3: 1) moving forward, 2) moving forward with a curve, and 3) moving forward while ascending a step.

Fig. 3 illustrates the predicted base pose covariances using the three different methods. The state covariance is a  $(12+n_J) \times (12+n_J)$  matrix, and the plots show the diagonal elements corresponding to the base pose dimensions. The comparison between methods (a) and (b) reveals that relying solely on the a priori covariance update (11) leads to an over-growth of the state covariance. This can lead to excessively large back-off terms (15) and (16), which in turn leads to a conservative solution or makes the numerical optimization challenging to solve. Conversely, the results from method (a) demonstrate that the combined use of a priori and a posteriori

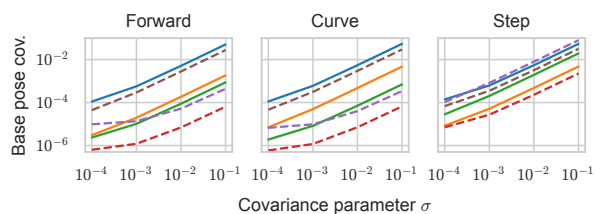
<sup>3</sup>Trot of Tachyon 3 is defined as a walking pattern alternating [LF, RF, MR] feet contact and [MF, LR, RR] feet contact [23].



(a) A priori and a posteriori covariance updates (11)–(13)



(b) A priori covariance update (11)



(c) Dynamics-based covariance update (20)

Fig. 3: Comparison of state covariance propagation methods. Plots  $x$ ,  $y$ , and  $z$  denote the covariances of the base position expressed in the world coordinate. Plots  $r_x$ ,  $r_y$ , and  $r_z$  represent the covariances of the base rotation around  $x$ ,  $y$ , and  $z$  axes expressed in the base local coordinate, respectively.

covariance updates effectively avoids the overgrowth of state covariance.

When contrasting methods (a) and (c), it is evident that the covariance update relying solely on dynamics (20) leads to a consistent augmentation in covariance across all directions as the covariance parameter  $W_i$  is increased. On the other hand, the covariance prediction from method (a) is motion-dependent. In the case of forward movement, an increase in covariance is predominantly observed in the  $x$  direction (the direction of motion), while covariances in other directions, such as  $y$  and  $z$ , remain relatively contained, even as  $C_g$  is increased. For curvilinear motion, the covariances in directions other than  $x$ , particularly around the  $r_z$  axis (the direction of the curve), are more pronounced than in the forward motion scenario. In the scenario of ascending a step, the covariance in the  $z$  direction (the direction of ascent) is more significant compared to the forward motion case. These results suggest that the proposed method can accurately predict future state covariance in accordance with the anticipated motion (i.e., state and control input trajectories), whereas the dynamics-based covariance update (20) does not offer such motion-specific predictive capabilities.

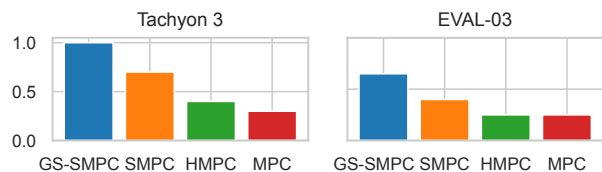


Fig. 4: Success rate of four MPC methods for randomized simulations on Tachyon 3 and EVAL-03.

### C. Simulation Comparison

To evaluate the efficacy of various MPC methodologies, we executed two sets of simulations under randomized environmental conditions: 1) Tachyon 3 ascending a four-step staircase with height estimation errors, and 2) EVAL-03 traversing a rough terrain area without environment perception. Both scenarios are depicted in Fig. 1. In both instances, we assessed the success rate of each method across 20 randomized environments. The stair geometry and height estimation errors (between  $-4\text{cm}$  and  $+4\text{cm}$ ) were randomized for Tachyon 3, while the rough terrain geometry was randomized for EVAL-03. The MPC methods compared were:

- GS-SMPC (Stochastic MPC with Guard Saltation Matrix)
- SMPC (Stochastic MPC)
- HMPC (MPC with heuristic constraint margins)
- MPC (MPC without constraint margins)

SMPC corresponds to the approach presented in a previous study [8]. In GS-SMPC and SMPC, we introduced chance constraints in the form of (17) for collision avoidance and joint torque limit constraints in the Tachyon 3 scenario, and for self-collision avoidance and friction cone (specifically the unilateral force) constraints in the EVAL-03 scenario, with satisfaction probabilities set between 0.8 and 0.95. For GS-SMPC, the covariance of the guard condition was chosen as  $C_g = 1.0 \times 10^{-3}$  for Tachyon 3 and  $C_g = 1.0 \times 10^{-4}$  for EVAL-03 to encompass contact height uncertainties. SMPC utilized (20) for covariance propagation between contact events, with  $W_j$  set to values that yielded a state covariance prediction magnitude similar to that of GS-SMPC. HMPC introduced hand-tuned heuristic margins into the constraints that were treated as chance constraints in GS-SMPC and SMPC. MPC did not include constant margins.

The horizon lengths were set to 1.5 s and 0.8 s in the Tachyon 3 and EVAL-03 cases, respectively. The discretization time step was set to 0.02 s in the both cases. In Tachyon 3 scenario, the controller proposed in our previous work using the control barrier function (CBF) [19] was used to track the solution of the MPC while particular CBFs regarding the collision avoidances were turned off to purely evaluate the capabilities of the MPC methods. In EVAL-03 scenario, a simple inverse-dynamics controller without any equality and inequality constraints was used to compute the joint torques from the MPC solution.

Fig. 4 shows the success rates of the four methods in the

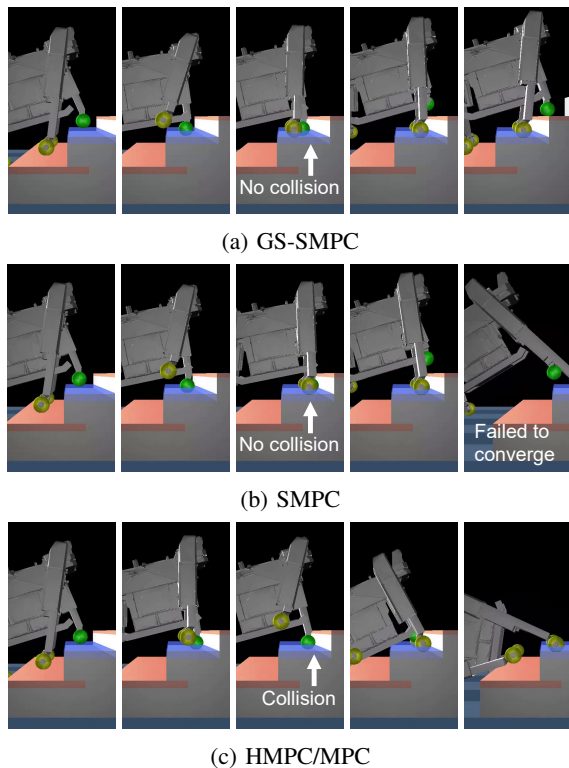


Fig. 5: Snapshots of the Tachyon 3 simulations ascending steps under terrain height uncertainties via (a) GS-SMPC, (b) SMPC, and (c) HMPC/MPC. The transparent blue boxes indicate the overestimate (e.g., +4 cm overestimate in the second step) and red ones indicate underestimate of terrain heights (e.g., -4 cm underestimate in the first step).

two benchmarks. The proposed GS-SMPC outperformed the other methods in both scenarios, demonstrating its robustness against contact uncertainties. Figs. 5–7 display typical behaviors of Tachyon 3 and EVAL-03 during the experiments. As illustrated in Figs. 5 and 6, GS-SMPC successfully generated trajectories that avoided collisions while preventing joint torque saturation despite the discrepancies between expected and actual terrain heights. While SMPC also managed to avoid collisions, it encountered ill-conditioned optimization problems (e.g., too small step size in Fig. 6) caused by overly large backoff terms, as also illustrated in Fig. 3. It should be noted that even though we attempted SMPC with wide range of  $\sigma$  in (19), but could not find successful one. HMPC and MPC were unable to accurately predict constraint margins, leading to intense collisions following the terrain height mispredictions. Fig. 7 also demonstrated effectiveness of GS-SMPC in preventing self-collisions while enhancing the unilateral force constraints under terrain height disturbances. The robot frequently experienced self-collisions between the left and right legs due to the contact disturbances when using the other MPC methods. However, as suggested by the success rate for EVAL-03 in Fig. 4, the proposed method still encountered challenges in preventing the robot from falling down, e.g., when a swing foot collided with bumps.

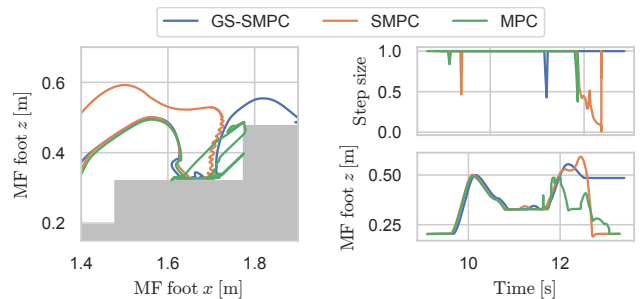


Fig. 6:  $x$ - $z$  trajectories of the MF foot (left) and step sizes of the optimization solver, i.e., convergence behavior (right) during the Tachyon 3 simulations ascending steps.

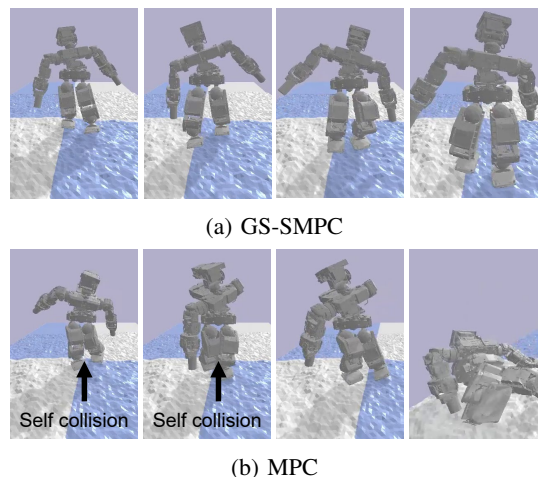


Fig. 7: EVAL-03 traversing over rough terrain via GS-SMPC (top row) and MPC (bottom row).

#### D. Hardware Experiments

To verify the practical applicability of the proposed method, we conducted hardware experiments on the perceptive locomotion of Tachyon 3. The software system, including the NMPC implementation, is detailed in [23]. The horizon length was set to 1.5 s and the discretization time step was set to 0.02 s. Although no artificial noise was introduced in the perception as in the simulations mentioned above, the real hardware inherently encompassed noise in the dynamics, contacts, and perception.

Fig. 8 displays snapshots and plots of Tachyon 3's perceptive locomotion using the proposed GS-SMPC. Tachyon 3 adeptly navigated over a step and managed to ascend and descend a series of two steps via GS-SMPC, which was implemented on the robot's on-board PC. The average computational time of GS-SMPC was 29 ms while that of the nominal NMPC (i.e., NMPC without covariance propagation) was 26 ms measured on the on-board PC (CPU: Intel(R) Core(TM) i7-8850 H CPU @ 2.606 GHz). Thanks to the efficient algorithm of the stochastic NMPC outlined in Section IV, our GS-SMPC showed only a slight increase (approximately 3 ms in this instance) in computational time compared to the nominal NMPC.

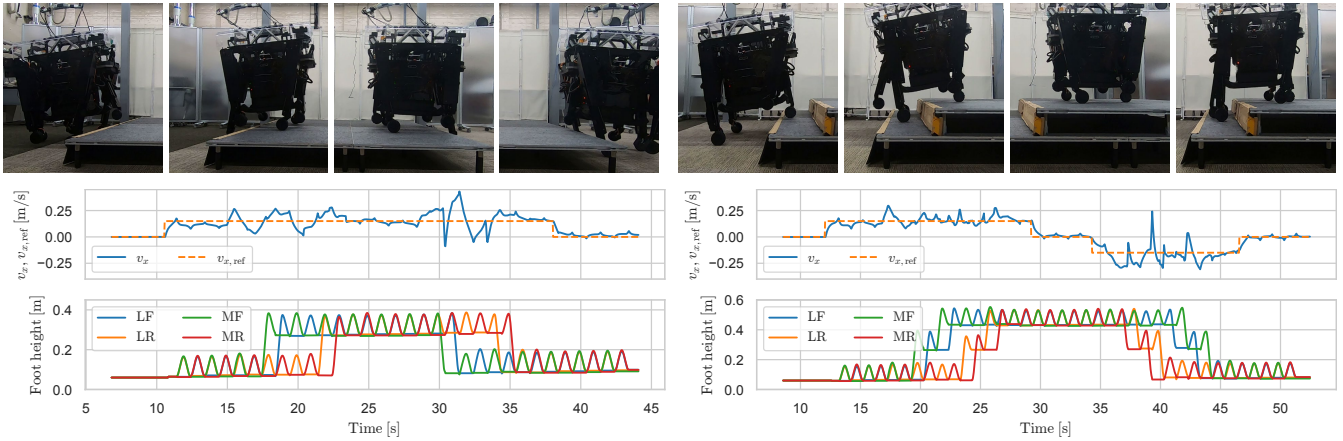


Fig. 8: Snapshots and plots of the perceptive locomotion of Tachyon 3 traversing a step (left) and ascending and descending two steps (right) using the proposed stochastic NMPC.  $v_x$  denotes the estimated base linear velocity expressed in the local coordinate frame and  $v_{x,ref}$  does its reference.

## VI. CONCLUSIONS

This paper introduced a stochastic/robust NMPC for robust model-based legged locomotion against contact uncertainties. We leveraged the saltation matrices and output-feedback MPC technique in forecasting the future state covariance, of which the performance had been shown in the numerical studies. Furthermore, the proposed method could be implemented on a on-board PC by utilizing the efficient zero-order stochastic/robust NMPC algorithm that fully leverages the Riccati recursion to obtain the feedback gains.

On the other hand, as indicated by the simulation study of the humanoid rough terrain walking, the proposed method may not be sufficient in completely preventing the robot from falling down. To address this limitation, it is crucial to explore other aspects of MPC, such as emergent contact re-planning, from the perspective of contact uncertainty.

## REFERENCES

- [1] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.
- [2] J. J. Payne, N. J. Kong, and A. M. Johnson, "The uncertainty aware Salted Kalman Filter: State estimation for hybrid systems with uncertain guards," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 8821–8828.
- [3] J. Zhang and T. Ohtsuka, "Output-feedback stochastic model predictive control of chance-constrained nonlinear systems," *IET Control Theory & Applications*, 2023.
- [4] A. Zanelli, J. Frey, F. Messerer, and M. Diehl, "Zero-order robust nonlinear model predictive control with ellipsoidal uncertainty sets," *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 50–57, 2021.
- [5] F. Messerer and M. Diehl, "An efficient algorithm for tube-based robust nonlinear optimal control with optimal linear feedback," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 6714–6721.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] N. J. Kong, C. Li, G. Council, and A. M. Johnson, "Hybrid iLQR model predictive control for contact implicit stabilization on legged robots," *IEEE Transactions on Robotics*, 2023.
- [10] M. Tucker, N. Csomay-Shanklin, and A. D. Ames, "Robust bipedal locomotion: Leveraging saltation matrices for gait optimization," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 12 218–12 225.
- [11] N. J. Kong, J. J. Payne, J. Zhu, and A. M. Johnson, "Saltation matrices: The essential tool for linearizing hybrid dynamical systems," *arXiv preprint arXiv:2306.06862*, 2023.
- [12] H. G. Bock and K.-J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," *IFAC Proceedings Volumes*, vol. 17, no. 2, pp. 1603–1608, 1984.
- [13] S. A. Burden, S. S. Sastry, D. E. Koditschek, and S. Revzen, "Event-selected vector field discontinuities yield piecewise-differentiable flows," *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 2, pp. 1227–1267, 2016.
- [14] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State estimation for legged robots-consistent fusion of leg kinematics and imu," *Robotics*, vol. 17, pp. 17–24, 2013.
- [15] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer, 1999.
- [16] C. Feller and C. Ebenbauer, "Relaxed logarithmic barrier function based model predictive control of linear systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1223–1238, 2016.
- [17] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, "Model predictive control: Theory, computation, and design," 2017.
- [18] M. Diehl, H. G. Bock, and J. P. Schlöder, "A real-time iteration scheme for nonlinear optimization in optimal feedback control," *SIAM Journal on control and optimization*, vol. 43, no. 5, pp. 1714–1736, 2005.
- [19] N. Takasugi, M. Kinoshita, Y. Kamikawa, R. Tsuzaki, A. Sakamoto, T. Kai, and Y. Kawanami, "Real-time perceptive motion control using control barrier functions with analytical smoothing for six-wheeled-telescopic-legged robot Tachyon 3," *arXiv preprint arXiv:2310.11792*, 2023.
- [20] M. Taylor, S. Bashkurov, J. F. Rico, I. Toriyama, N. Miyada, H. Yanagisawa, and K. Ishizuka, "Learning bipedal robot locomotion from human movement," *arXiv preprint arXiv:2105.12277*, 2021.
- [21] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, "A unified MPC framework for whole-body dynamic locomotion and manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4688–4695, 2021.
- [22] R. Grandia, F. Jenelten, S. Yang, F. Farshidian, and M. Hutter, "Perceptive locomotion through nonlinear model-predictive control," *IEEE Transactions on Robotics*, 2023.
- [23] S. Katayama, N. Takasugi, M. Kaneko, and M. Kinoshita, "Versatile telescopic-wheeled-legged locomotion of Tachyon 3 via full-centroidal nonlinear model predictive control," *arXiv preprint arXiv:2312.08668*, 2023.