Cascaded Model Predictive Control of Underactuated Bipedal Walking with Impact and Friction Considerations

Sait Sovukluk, Christian Ott, and M. Mert Ankaralı

DOI: 10.1109/Humanoids57100.2023.10375153

Cascaded Model Predictive Control of Underactuated Bipedal Walking with Impact and Friction Considerations

Sait Sovukluk[1,3], Christian Ott[1,2] and M. Mert Ankaralı[3,4]

Abstract— **This study demonstrates a cascaded model predictive control (C-MPC) method for input constrained control of underactuated planar bipedal walking with any predefined stabilizable trajectory. Our approach aims to increase the trajectory tracking performance of the system and produce realistic and applicable responses while respecting friction constraints and considering impact dynamics. Primarily focusing on zero dynamics with PD (ZD+PD) control, this proposed control method constitutes a second layer controller on top of the well-known trajectory tracking controllers for underactuated bipedal walking systems. We successfully implement this model-based controller and test against large modeling errors, noises, and disturbances, where the conventional ZD+PD control fails to maintain stability.**

I. INTRODUCTION

The complexity of control of bipedal walkers with point feet originates from their underactuated nature. The underactuation requires an indirect control of unactuated degree of freedom through high-bandwidth control of actuated joints. The actuated joints follow a specially generated trajectory such that the unactuated degree of freedom performs the desired behavior based on the system dynamics. Combining underactuation with high-dimensional nonlinear hybrid dynamics requires special treatment for their control. Due to the high dimensionality of the system and several gait characterization constraints, one usually needs to solve these trajectory generation optimization problems offline for one step. Some of these trajectory generation methods are Hybrid Zero Dynamics (HZD) [1], Human-Inspired Control [2], and direct trajectory optimization [3]. Online implementation of the predefined optimized trajectory employs a calculated feedback-linearization input with a local high-gain controller, i.e., zero dynamics with PD (ZD+PD) control.

Locomotion with a predefined trajectory on unknown stochastic terrains requires a combination of different gaits. Depending on surface and stability conditions, the next foot placement point may need to be adapted. The literature employs previously calculated gait libraries with dedicated switching methods to walk over randomly generated stepping stones [4], reject velocity disturbances efficiently [5], walk over an uneven terrain [6] and parameterize bipedal robot locomotion [7]. Even though optimization problems for trajectory generation consider various gait restrictions, such as step length, friction cone, stability conditions, and torque limitations, the online implementation employed in these studies uses a calculated feedback-linearization input with a local high-gain PD (ZD+PD) controller, which is enough to prove exponential stability in theory. Unless one tests the trajectories against every possible source of disturbances and uncertainties, these online implementation methods tend to exceed the input limitations of the actual system or violate friction constraints. Exceeding torque and friction limitations of the system causes discrepancies between the system model and the robot, errors in estimations, and instabilities or slow convergence rates. Additionally, excessive torque input usually causes a violation in the other gait restrictions, such as friction cone, which also causes instabilities.

Studies addressing the input torque limitation problem in the online implementation of walking gaits usually employ model-based controllers using simplified or linearized dynamics. Additionally, they do not consider friction limitations or impacts. The studies in [8], [9] solve this problem by employing MPC around linearized output dynamics, [10] employs Hybrid-Linear Inverted Pendulum (H-LIP) dynamics for MPC implementation, [11] employs a simplified model of the robot dynamics projected in swing foot space for their control, [12] uses linear centroidal dynamics. The problem with constant linear system assumptions is that the accuracy of these models is valid only for a small region of configuration variables. As the prediction horizon increases, accuracy decreases even more. Additionally, even input noises of the motors affect the correctness of linearized closed-loop restriction dynamics whose correction depends on a particular input. On unknown stochastic terrain, unavoidable sources of errors reduce the performance of these online implementation methods. The objective and approach of this study are similar to the study in [8]. This study varies from [8] in the way it handles the problem. First, we consider friction limitations and impact dynamics during prediction. Also, instead of employing linearized output dynamics to predict future errors, this study employs full-body closed-loop dynamics to estimate future states and calculate for errors. Additionally, in order to enhance the prediction accuracy, this study employs MPC in a cascaded form, such that the MPC takes effect only when the closed-loop dynamics with zero dynamics control starts to diverge from the desired trajectory, for example, due to modeling errors and disturbances, or exceeds the input limitations.

[1] Automation and Control Institute (ACIN), TU Wien, 1040 Vienna, Austria

[2] Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Wessling, Germany

[3] Department of Electrical and Electronics Engineering, Middle East Technical University (METU), Ankara, Turkey

[4] METU Robotics and AI Technologies Application and Research Center (METU-ROMER), Ankara, Turkey

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819358).

This study proposes a cascaded MPC implementation method with input torque and friction limitations on sampled whole-body dynamics as an online trajectory tracking controller. We also switch system dynamics in case of an impact during horizon estimation. Instead of employing a constant linearized system or output dynamics model, we sample closed-loop system dynamics with ZD+PD control at each time step and update model approximation in our controller structure. As the closed-loop system with this conventional controller starts to diverge from the desired trajectory due to disturbances and modeling errors, our cascaded MPC (C-MPC) implementation optimizes the input. It decreases trajectory tracking errors while considering friction limitations and impact. With this approach, we enhance the well-known weaknesses of ZD+PD control, such as its sensitivity against significant modeling errors, excessive input torque usage under disturbances, exceeding input limitations, and violating friction constraints. This control structure considerably enlarges the basin of attraction of the closed-loop system with ZD+PD control. We show the controller's ability against various aggressive disturbances where feedback linearization input with high-gain PD (ZD+PD) trajectory tracking controllers failed to maintain stability under input torque saturation. We also show that this model-based controller is robust to modeling errors and can navigate through unexpected inclinations, where the leg collision happens before and after the expected state (see supplemental video).

II. Background

This section summarizes the conventional Hybrid Zero Dynamics approach for trajectory generation and control for 5-link underactuated walkers (see Fig. 1). Studies in [1], [13] provide throughout definitions for this approach.

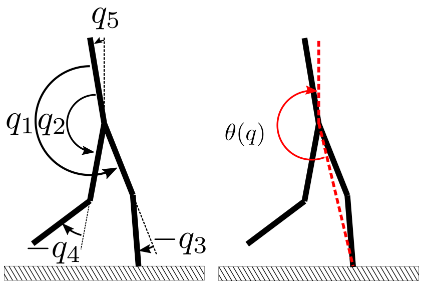

Fig. 1. Schematic of a 5-link bipedal walker with point feet, where q_{1-4}, q_5, and θ represent body angles, absolute angle, and internal clock, respectively. The internal clock is a strictly monotonic function, and a combination of configuration variables constitutes it.

A. System Dynamics

General state-space representation of nonlinear robotic stance dynamics with $n-number$ configuration variables $q \in \mathcal{Q}$, where $\mathcal{Q}$ is the configuration space and an $n-dimentional$ manifold, is expressed as:

$$\dot{x} = f(x) + g(x)u \tag{1}$$

where $x(t) = (q(t); \dot{q}(t)) \in T\mathcal{Q}$,

$$\begin{aligned} f(x) &= \begin{bmatrix} \dot{q} \\ D^{-1}(q)[-C(q,\dot{q})\dot{q} - G(q)] \end{bmatrix} \\ g(x) &= \begin{bmatrix} 0 \\ D^{-1}(q)B \end{bmatrix}. \end{aligned} \tag{2}$$

Matrices $D(q) = D(q)^\top \in \mathbb{R}^{n\times n} > 0$, $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$, $G(q) \in \mathbb{R}^n$ and $B \in \mathbb{R}^{n\times m}$ represent mass-inertia matrix, Coriolis matrix, gravity vector and input matrix for m number of actuation, respectively.

Walking is a combination of two consecutive events called stance and impact. Stance dynamics ($\dot{x}$) is a nonlinear whole-body Lagrangian system model. The impact (Δ) is a momentary event that happens when the swing leg collides with the ground, causes jumps in velocities, and triggers coordination changes, i.e., the swing leg becomes the stance leg, and the stance leg becomes the swing leg. A widely used impact model for kinematic chains with multiple contact points [14] can be implemented for the bipedal walkers with some assumptions [1]. Defining x^- and x^+ to represent states just before and just after the impact, the overall system dynamics leads to:

$$\Sigma : \begin{cases} \dot{x} = f(x) + g(x)u & x \notin \mathcal{S} \\ x^+ = \Delta(x^-) & x^- \in \mathcal{S} \end{cases}. \tag{3}$$

Switching surface $\mathcal{S}$ captures the moment when the tip of the swing leg hits the ground ahead of the stance leg:

$$\mathcal{S} := \{(q,\dot{q}) \in T\mathcal{Q} \mid p_2^v(q) = 0,\ p_2^h(q) > 0\}, \tag{4}$$

where $p_2^h : \mathcal{Q} \to \mathbb{R}$ and $p_2^v : \mathcal{Q} \to \mathbb{R}$ represent the coordinates of the tip of the swing leg in the horizontal and vertical axis, respectively.

B. Trajectory Generation Using HZD

Hybrid Zero Dynamics (HZD) is a well-known trajectory generation method for underactuated hybrid 5-link bipedal walkers [1], [13]. The method ensures that for a certain subset of the configuration variables $q \in \tilde{\mathcal{Q}} \subset \mathcal{Q}$, there exists a set of input such that the output:

$$y = h(q) := h_0(q) - h_d \circ \theta(q) \tag{5}$$

is zero, where $h_0(q)$ specifies $n-1$ independent quantities that are to be controlled and $h_d \circ \theta(q)$ specifies the desired evolution of these quantities as a function of a scalar quantity $\theta(q)$, which is called internal clock. The internal clock is a monotonic one-to-one function that slaves the desired states, and a combination of configuration variables constitutes it, $\theta(q) = cq$, where $c \in \mathbb{R}^{1\times n}$ (see Fig. 1). Using HZD, one can fit a degree M Bézier polynomial, $b_i : [0,1] \to \mathbb{R}$, for the desired state evolution $h_d \circ \theta(q)$. The desired state evolution mimics walking behavior and satisfies defined constraints, such as friction, input, and other physical parameter constraints, during the HZD optimization. The study in [13, Chapter 6] provides a more throughout explanation of obtaining a stabilizable periodic trajectory for such systems.

C. Local Control with Feedback Linearization

The second derivative of output definition (5) yields to:

$$\frac{d^2y}{dt^2} = L_f^2 h(q,\dot{q}) + L_g L_f h(q) u, \tag{6}$$

where $L_f^2 h$ and $L_g L_f h$ are second order Lie derivatives and $L_g L_f h$ is called decoupling matrix. For a subset of the configuration variables $q \in \tilde{\mathcal{Q}} \subset \mathcal{Q}$, where the decoupling matrix is invertible, selecting the input u to be:

$$u(x) = (L_g L_f h(q))^{-1}(v - L_f^2 h(q,\dot{q})), \tag{7}$$

yields to the following output dynamics:

$$\frac{d^2y}{dt^2} = v. \tag{8}$$

As a result, selecting v to be:

$$v = -K_D \dot{y} - K_P y, \tag{9}$$

results in exponentially stable output dynamics as long as K_D and K_P are $m \times m$ positive definite matrices. Throughout the study, the feedback linearization control input (7) is named zero dynamics with PD (ZD+PD) control.

III. Proposed Control Method

A. Cascaded MPC Formulation

To solve for an input, MPC requires a combination of present states and their consecutive finite-time future estimations in a discrete manner. The performance of this controller depends on how accurate the estimation is. As the system dynamics are nonlinear, an exact estimation of future states is not possible. We employ a cascaded structure to reduce estimation errors such that the MPC is placed on top of ZD+PD control. The motivation behind this approach is that ZD+PD control input can be calculated using present state information, and it already tries to converge to the desired trajectory. Besides its tendency to exceed the input limitations and violate friction constraints, when there is no limitation, this controller tracks the desired trajectory, under ideal conditions, quite well [13, Chapter 5]. Knowing the input that comes from ZD+PD control allows us to make a partial state prediction for the future more accurately than not knowing anything about future inputs of the system. This structure employs MPC only if the closed-loop dynamics with ZD+PD control diverge from the desired trajectory or exceed the input or friction limits. Otherwise, the total input from MPC is zero.

Using second-order discretization, the evolution of configuration variables $q(t)$ can be approximately represented as:

$$q_{k+1} \approx q_k + T\dot{q}_k + (T^2/2)\ddot{q}_k, \tag{10}$$

where $q_k = q(kT)$, $T \in \mathbb{R}$ is sampling time, and $k \in \mathbb{N}$ is time step. Consequently, the system model (1) yields to:

$$x_{k+1} \approx x_k + (f_k + g_k \tilde{u}_k)T, \quad k = 0, \dots, N-1 \ , \tag{11}$$

where $\tilde{u}$ is the input to be solved,

$$f_k = \begin{bmatrix} I & T/2 \\ 0 & I \end{bmatrix} f + g_k u, \quad g_k = \begin{bmatrix} D^{-1}BT/2 \\ D^{-1}B \end{bmatrix}, \tag{12}$$

and u comes from (7) and represents input from ZD+PD control. Using (11), the general representation of the system becomes as:

$$x_N = x_0 + \sum_{j=0}^{N-1} f_j T + \sum_{j=0}^{N-1} g_j \tilde{u}_j T. \tag{13}$$

For a predefined orbit, i.e., for a known desired state x^d, error dynamics is a basic manipulation of (13):

$$e_N = x_N - x_N^d. \tag{14}$$

As a result of (14), general error representation can be shown as:

$$\underbrace{\begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix}}_{E} = \underbrace{\left(\begin{bmatrix} x_0 \\ x_0 \\ x_0 \\ \vdots \\ x_0 \end{bmatrix} + \begin{bmatrix} 0 \\ f_0 \\ f_0 + f_1 \\ \vdots \\ \sum_{j=0}^{N-1} f_j \end{bmatrix} T - \begin{bmatrix} x_0^d \\ x_1^d \\ x_2^d \\ \vdots \\ x_N^d \end{bmatrix} \right)}_{F} + \underbrace{\begin{bmatrix} 0 & 0 & \dots & 0 \\ Tg_0 & 0 & \dots & 0 \\ Tg_0 & Tg_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ Tg_0 & Tg_1 & \dots & Tg_{N-1} \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} \tilde{u}_0 \\ \tilde{u}_1 \\ \tilde{u}_2 \\ \vdots \\ \tilde{u}_{N-1} \end{bmatrix}}_{U} \tag{15}$$

where $E \in \mathbb{R}^{2n(N+1)}$, $F \in \mathbb{R}^{2n(N+1)}$, $G \in \mathbb{R}^{2n(N+1) \times Nm}$, and lastly $U \in \mathbb{R}^{Nm}$. During calculating rows of partial prediction matrix F, in case of an impact, i.e., $x^- \in \mathcal{S}$, one should switch the system dynamics using (3) such that trajectory estimation continues with the next step inside the horizon. The impact detection depends not on fixed time but on partially predicted states' evolution. In (15), a linear dependency is assumed between the additional MPC input and the system dynamics. More comments on this assumption will follow. The general error representation (15) enables us to write a cost function for error in a quadratic form,

$$\begin{aligned} J &= e_N^\top Q_f e_N + \sum_{k=0}^{N-1} (e_k^\top Q e_k + \tilde{u}_k^\top R \tilde{u}_k) \\ &= E^\top \underbrace{\begin{bmatrix} Q & & 0 \\ & \ddots & \\ 0 & & Q_f \end{bmatrix}}_{\underline{Q}} E + U^\top \underbrace{\begin{bmatrix} R & & 0 \\ & \ddots & \\ 0 & & R \end{bmatrix}}_{\underline{R}} U \end{aligned} \tag{16}$$

in terms of input, where $N \in \mathbb{Z}^+$, $Q = Q^\top \in \mathbb{R}^{2n \times 2n} \geq 0$, $Q_f = Q_f^\top \in \mathbb{R}^{2n \times 2n} \geq 0$, $R = R^\top \in \mathbb{R}^{m \times m} > 0$ represent horizon, error weight, final error weight and input weight, respectively. Substituting (15) into the general representation of the cost function (16) yields the cost function in terms of input:

$$J(U) = U^\top \underbrace{(G^\top \underline{Q} G + \underline{R})}_{M} U + 2 \underbrace{F^\top \underline{Q} G}_{\alpha^\top} U + \underbrace{F^\top \underline{Q} F}_{\beta}. \tag{17}$$

B. Friction Constraint

Calculation of the tangential and normal component of contact force $F_c = [F_c^T; F_c^N]$ requires unpinned $(n+2) - DoF$ model of the system. Assume $J_c \in \mathbb{R}^{2 \times (n+2)}$ represents contact Jacobian for the unpinned model and $(\cdot)_e$ subscript represent the unpinned system model. Combination of the task acceleration of the contact point $\dot{J}_c\dot{q}_e + J_c\ddot{q}_e = 0$ with the unpinned system dynamics yields the contact force:

$$F_c = (J_c D_e^{-1} J_c^\top)^{-1}(J_c D_e^{-1}(-BU_t + C_e\dot{q}_e + G_e) - \dot{J}_c\dot{q}_e). \tag{18}$$

Since the MPC is a second layer controller, the resulting input also includes the input of ZD+PD control (see Fig. 2). As a result, the total input U_t that should be supplied to the system becomes:

$$U_t = (u + U). \tag{19}$$

In order to imply friction limitations in input limits, we substitute (19) into (18) and separate the effect of two controllers on contact forces. With this approach, MPC will act as a correction layer again, and if ZD+PD control violates the friction cone, MPC will correct it by changing the input profile. Labeling the fixed portion of the contact force that comes from system dynamics and ZD+PD control effects as F_f, the identical F_c representation becomes as

$$F_c = F_f - \underbrace{(J_c D_e^{-1} J_c^\top)^{-1} J_c D_e^{-1} B}_{S_{MPC}} U. \tag{20}$$

Limiting friction to be $\left|F_c^T / F_c^N\right| \leq \mu$, one can write input constraint for friction limitations as

$$\frac{\left|F_f^T - S_{MPC}^T U\right|}{\left|F_f^N - S_{MPC}^N U\right|} \leq \mu \tag{21}$$

where $(\cdot)^T$ and $(\cdot)^N$ represent first and second rows, respectively.

C. Optimization Problem

The resultant optimization problem, with modified input constraints considering the input u that comes from ZD+PD control (7), takes the following form:

$$\min_U \{U^\top M U + 2\alpha^\top U\} \quad \text{s.t.} \begin{cases} x_{k+1} = x_k + (f_k + g_k \tilde{u}_k)T \\ U_{min} - u \leq U \leq U_{max} - u \\ \text{friction constraint (21)} \end{cases} \tag{22}$$

Since the MPC is the second layer controller, the resulting input optimization should be combined with the input of ZD+PD control. As a result, the total input U_t that should be supplied to the system becomes

$$U_t = (u + U). \tag{23}$$

Block diagram representation of the proposed control method is shown in Fig. 2. The control system first uses sampled state of the walker as an initial condition and simulates the closed-loop system with ZD+PD control up to horizon N. Then, the controller constructs a cost function using resultant partial predicted errors. Since the prediction is made with some part of the input, i.e., only using input from ZD+PD control, the partial prediction term is used here. The MPC considers the amount of input that comes from the ZD+PD controller and finds an additional input that minimizes the overall error of the closed-loop system while considering impact dynamics and limiting total input and friction constant. Finally, the resultant total input is discretely fed into the system via a zero-order hold.

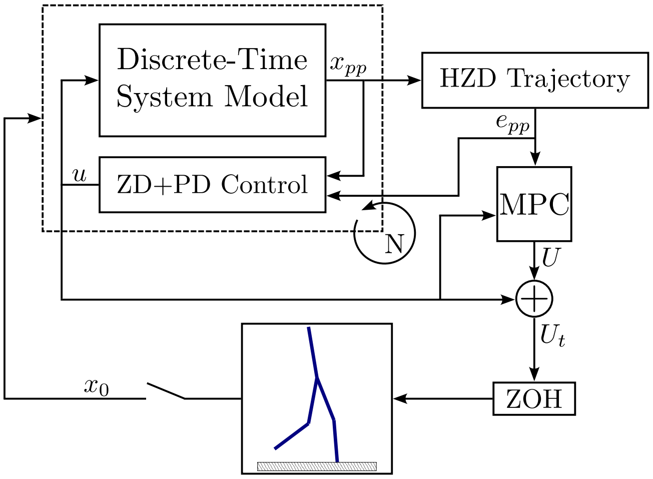

Fig. 2. Block diagram representation of the proposed control model. In the figure x_{pp} and e_{pp} represent state and error partial prediction, respectively.

D. Implementation and Comments

The biggest assumption in this study is fixing the partial prediction matrix in (15). It is assumed that the additional input from MPC does not affect the system's dynamics and is formulated independently as input U is implemented via fixed matrices at each time step. This assumption holds when the error, hence the additional input, is not too big. Otherwise, the effect of the additional input on the system dynamics may not be negligible, and the accuracy of the system model decreases. On the other hand, since the additional MPC input is included only for regulation purposes, it appears only in case of a violation or disturbance, and the system dynamics is mostly driven by the feedback linearization input. In the following sections, we show that the input from MPC is zero when there is no disturbance or violation. In case of an aggressive disturbance, we again show that input from MPC is considerably less than the feedback linearization input, and it vanishes quickly as the error decreases. These results conclude that our cascaded MPC formulation allows us to assume linearity between the system dynamics and additional regulatory inputs.

For each time step, the required operations to apply the controller can be listed as:

- Employing (7) inside f_k, using (11) and (12), for $\tilde{u} = 0$ and switching the system dynamics in case of an impact where $x^+ = \Delta(x^-)$, simulate ahead up to horizon N and store f_k and g_k matrices,
- Using (15), (16) and (17), generate G and F, then obtain M, $\alpha^\top$ and β,
- Using (21) calculate the additional input constraints for friction limitations,
- Using (22), solve for input and combine the solution with ZD+PD control using (23).

E. Controller Parameter Selection

ZD+PD control requires eight controller parameter selections (9). We chose the parameters by observing the eigenvalues and eigenvectors of Poincaré return map estimations, which are calculated numerically around the predefined periodic orbit. We employ a basic search algorithm to select the parameters that seeks small eigenvalue combinations in magnitude around the predetermined trajectory. Implementation details of this search algorithm can be found in [15, Chapter 3.6.1]. The resultant parameter sets are $K_P = diag(60, 90, 90, 50)$ and $K_D = diag(10, 20, 20, 10)$.

MPC control, on the other hand, requires twelve parameters to choose for error and input weight matrices, as shown in (16). After setting the input cost to be an identity matrix, i.e., $R = I_{4\times4}$, preserving the same ratios between parameters as in K_P and K_D matrices, we scale these values by their discretization rates (see (11) and (12)) such that $Q = diag(2K_P/T^2, K_D/T)$. Additionally, to increase the importance of future states, we scale $\underline{Q}$ by $diag(I_{2m\times2m}, \ldots, N_{2m\times2m})$ such that in case of an aggressive disturbance, the controller behaves flexibly in terms of errors in near future to converge nicely back to the trajectory in further future. This approach also helps the system adjust for future impacts, which is the primary source of instability in case of errors.

Horizon selection is another control parameter that affects both performance and solution time. Thanks to the cascaded structure of the overall control system and since the feedback linearization input mostly drives the system dynamics, sampling the system dynamics only with the feedback linearization input is accurate enough to predict multiple steps. Starting from $N = 2000$, we decreased the horizon until the solution time was small and performance was well enough. By observation, we selected $N = 100$ mostly to match with the control frequency. Still, also we observed that there were not any drastic changes in control performance against the disturbances we applied to the system.

IV. Simulation Results

In order to test the proposed control method on a 5-link planar underactuated bipedal walking, we utilize a simulation environment. While obtaining the system dynamics, we used RABBIT's dimension and inertia information [16] as a basis with $5Nms$ and $10Nms$ damping at the knee and hip joints, respectively. Then we followed [13, Chapter 6.6.2.1] to obtain a periodic orbit using HZD. Table I collects the resultant parameters of fitted trajectory into degree $M = 6$ Bézier polynomials, $h_{d,i}(\theta, \alpha)$, for initial condition $x_0^- = [2.92,\ 3.54,\ -0.15,\ -0.36,\ 0.015,\ -0.21,\ -0.16,\ -0.25,\ 0.23,\ -0.85]^\top$. We illustrate the posture of the walker around the optimized trajectory in Fig. 3. Hip and knee torques are limited to $75Nm$ and $50Nm$ inside the simulation environment. Any input torque higher than these in magnitude saturates, i.e., stays constant at the limit. Finally, considering the system's nonlinearity, the sampling time is $T = 0.001$. In order to present the tendency of ZD+PD control to exceed the input torque limitations, we first simulate the system with this controller without any input saturation. In Fig. 4, we compare input behaviors of the ZD+PD and C-MPC controllers when a forward force of $40N$ from the torso is applied during the first step. The figure shows that the unsaturated ZD+PD control uses much more input than the proposed control method, even with the fine-tuned parameters. The system with the ZD+PD controller failed to maintain stability under saturation limits. Any force higher than $25N$ causes the system with ZD+PD control to fall due to saturation.

TABLE I

Bézier polynomial parameters for fitted desired trajectory

i	α_0	α_1	α_2	α_3	α_4	α_5	α_6
1	3.54	3.66	3.39	3.28	3.02	2.93	2.92
2	2.92	2.94	3.12	3.73	3.62	3.55	3.54
3	−0.36	−0.60	−0.28	−0.18	−0.03	−0.13	−0.15
4	−0.15	−0.18	−0.51	−0.81	−0.48	−0.38	−0.36

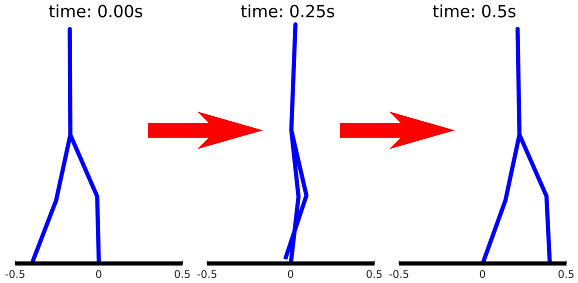

Fig. 3. The posture of the walker throughout the generated periodic trajectory with 0.4 meters step length and 0.5 seconds step time.

Accordingly, Fig. 5 illustrates deviations of both closed-loop systems from the desired trajectory when there is $25N$ forward force from the torso of the walker throughout the second step with saturation. Even though both controllers reject the disturbance, since the C-MPC takes account of input saturation, the deviation from the desired trajectory is much smaller than that of the closed-loop system with only ZD+PD control. Additionally, since the ZD+PD control is a continuous-time controller, when implemented in discrete time, the closed-loop system with this controller shows a remaining error (see top row of Fig. 5). The proposed model predictive controller compensates for this remaining error. As a result, no remaining error is observed with this controller (see bottom row of Fig. 5).

Finally, Fig. 6 illustrates ZD+PD and MPC input portions of C-MPC control, again when a forward force of $25N$ is applied to the torso throughout the second step. The figure shows that when there is no disturbance at the beginning, the input portion of MPC is almost zero. Under disturbance, MPC takes effect to limit the total input and prevents large deviation from the desired trajectory (see Fig. 5). After convergence, the input that comes from MPC again reduces.

Even though HZD trajectory generation considers friction limitations, it is highly likely to exceed these limitations in case of disturbances. Two C-MPC simulations we present in Fig. 7 show that in case of a disturbance, the friction

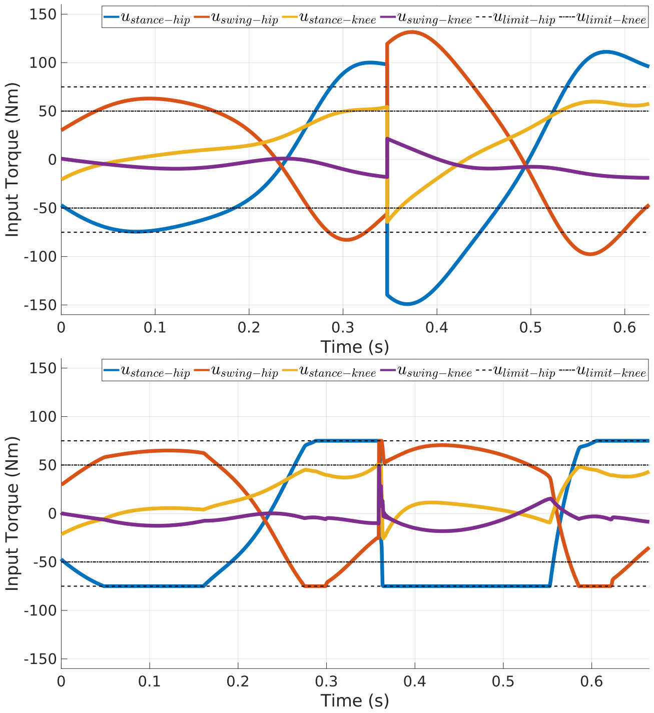

Fig. 4. Two-step response of the unsaturated ZD+PD controller (top) and the proposed controller (bottom) against a forward force of 40N to the torso throughout the first step of the robot.

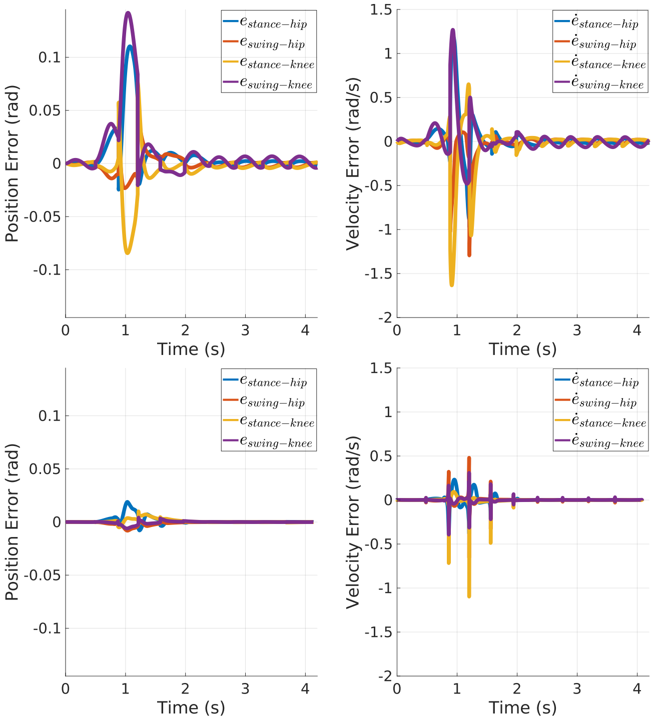

Fig. 5. Error evolution of the systems with saturated ZD+PD control (top) and C-MPC (bottom) against a forward force of $25N$ to the torso throughout the second step with input saturation.

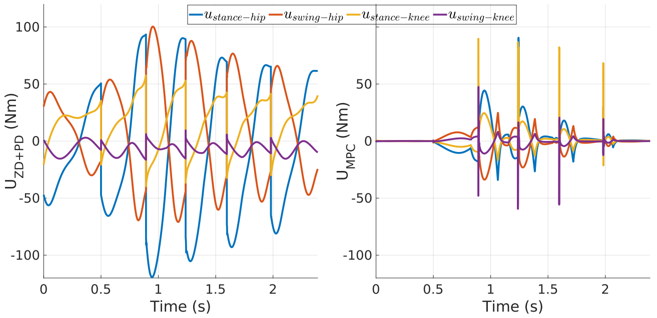

Fig. 6. ZD+PD and MPC input portion of C-MPC control when a forward force of $25N$ is applied to the torso throughout the second step. These inputs are summarized to the system (23).

exceeds the limitations when not limited during formulation. On the other hand, as long as there is a possible solution, the constraint we introduce inside the formulation (21) prevents the controller from exceeding its maximum value.

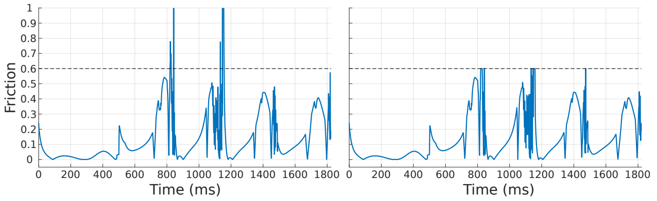

Fig. 7. Friction response of the system without (left) and with (right) friction limitation in C-MPC formulation against a forward force of 50N to the torso throughout the second step of the robot. The vertical axis is limited to one.

V. Performance of the Controllers

In order to comment on controllers' performances, this chapter compares the basin of attraction of each controller along with their closed-loop Poincaré return map eigenvalue estimations.

A. Basin of Attraction Analysis

To show the amount of disturbance each of these controllers can reject, we calculate the basin of attraction of the closed-loop systems with C-MPC and ZD+PD controllers for torso position and a disturbance force applied horizontally from the torso throughout the first step. The torso is selected because it is the heaviest, the longest, and the unactuated joint of the walker. During the analysis, we disturb q_5 and introduce disturbance force F_d throughout the first step. If the walker does not fall, the simulation is terminated when $||x^{-*} - x^{-}|| < \epsilon$ where x^{-*} is the periodic state just before the impact. The input torque of the ZD+PD controller is also limited to $75Nm$ at the hips and $50Nm$ at the knees. Any input torque exceeds this limit, saturates, i.e., stays constant at the limit. Fig. 8 shows the basin of attraction for both controllers. The figure shows that C-MPC considerably enlarges the basin of attraction of ZD+PD control when there is saturation. Since the high-speed forward disturbances require high input torques, the C-MPC expands the basin of attraction from the top, considering input saturation. As shown in previous discussions, one should note that, even though they all seem stable in some regions, each closed-loop system with different controllers handles disturbances differently (for example, see Fig.5). Input profiles and posture evolution under disturbance are distinct for each method.

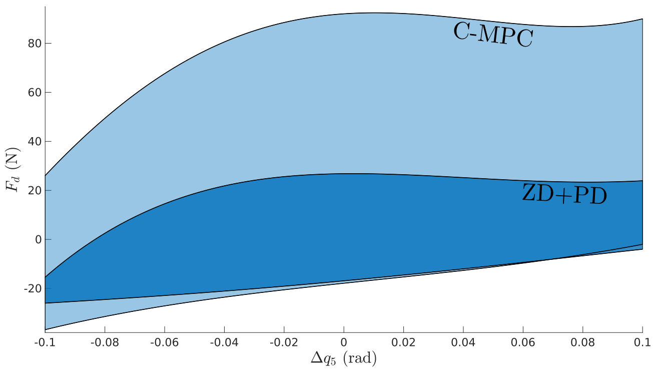

Fig. 8. Basin of attraction of the closed-loop systems with ZD+PD control and C-MPC with respect to the amount of disturbance applied to the initial position of the torso and a constant horizontal force again from the torso throughout the first step. Positive forces indicate forward disturbance.

B. *Poincaré Return Map Analysis*

Gait stability can approximately be determined by analyzing the eigenvalues of Poincaré return map of the system, which is linearized about a fixed point [17]–[20]. Poincaré return map transforms the problem of finding periodic orbit into finding a fixed point of the map, which is a discrete-time nonlinear system. If the Poincaré section is selected to be the switching surface $\mathcal{S}$, then the fixed point x^* lies within $\mathcal{S} \cap \mathcal{Z}$ and the resultant Poincaré map definition becomes $P : \mathcal{S} \cap \mathcal{Z} \to \mathcal{S} \cap \mathcal{Z}$. For $P(x_k) = x_{k+1}$ periodicity is achieved only if $P(x^-) = x^-$. A periodic stable fixed point satisfies that, in its neighborhood, all eigenvalues of the Jacobian matrix

$$\mathbf{DP} = \frac{\partial f_{cl}}{\partial x_k}(x^*) \tag{24}$$

lie within a unit circle ($|\lambda| < 1$) [21], where f_{cl} indicates closed-loop system dynamics. An inspiring basic explanation of Poincaré return map analysis for such systems can be found in [13, Chapter B.3]. The numerical implementation details we employed for this study are discussed in detail in [15, Chapter 6.2].

The maximum magnitude eigenvalues of the undisturbed closed-loop systems with ZD+PD control and C-MPC are given in the first column of Table II, respectively. Since the cascaded MPC is applied on top of the ZD+PD control, the MPC part of the controller has no effect under ideal conditions. As a result, the maximum magnitude eigenvalue estimations for the undisturbed case are identical.

Generally, model-based controllers are vulnerable to modeling errors. In order to check the robustness of the controllers against modeling errors, we multiply the mass of the torso by two inside the simulator. The maximum magnitude eigenvalues of the disturbed closed-loop systems with ZD+PD control and C-MPC are given in the second column of Table II, respectively. Since the ZD+PD controller mainly relies on direct torque computation using the system dynamics (see (7)), any error introduced in the system model reduces its robustness. We observe that the ZD+PD controller cannot maintain stability, and the walker falls forward after one step. The result is the same even if we disable the input saturation. On the other hand, since the C-MPC determines input by minimizing the cost function (16) and considers future state evolution, it handles modeling errors better and compensates for the weaknesses of ZD+PD control. As a result, the resultant closed-loop system with C-MPC shows stable behavior, with only a slight increase in the maximum magnitude eigenvalue estimation.

TABLE II
MAXIMUM MAGNITUDE EIGENVALUES OF THE DISTURBED AND UNDISTURBED CLOSED-LOOP SYSTEMS.

	Undisturbed	Doubled Torso Mass
ZD+PD Control	0.75	unstable
Cascaded MPC	0.75	0.78

VI. CONCLUSION

This study covers a cascaded model predictive control (C-MPC) method for input-constrained control of underactuated bipedal walking with any predefined trajectory. Unlike known approaches, while constructing for the horizon of MPC, we employ a full-order system model and implement one more controller at the lower level to enhance future state prediction accuracy. Additionally, this structure enables us to account for friction limitations and hybrid dynamics inside the horizon. Therefore, as a second layer controller, the MPC compensates for the weaknesses of the ZD+PD control, such as its sensitivity to modeling errors and exceeding amount of input usage. We showed that C-MPC outperforms the classical ZD+PD controller in terms of the size of the basin of attraction, forward force disturbance rejection, modeling error rejection, remaining error in discrete time implementation, and input torque usage. Even though, in the cases where systems with both controllers performed stable behaviors under disturbances, we showed that deviation from the desired trajectory is much less for the system that employs C-MPC (see Fig. 5).

REFERENCES

[1] E. Westervelt, J. Grizzle, and D. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 42–56, 2003.

[2] A. D. Ames, "Human-inspired control of bipedal walking robots," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1115–1130, 2014.

[3] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.

[4] Q. Nguyen, A. Agrawal, W. Martin, H. Geyer, and K. Sreenath, "Dynamic bipedal locomotion over stochastic discrete terrain," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1537–1553, 2018.

[5] R. Hartley, X. Da, and J. W. Grizzle, “Stabilization of 3d underactuated biped robots: Using posture adjustment and gait libraries to reject velocity disturbances,” in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, 2017, pp. 1262–1269.

[6] M. Fevre, H. Lin, and J. P. Schmiedeler, “Stability and gait switching of underactuated biped walkers,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 2279–2285.

[7] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, “Reinforcement learning for robust parameterized locomotion control of bipedal robots,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 2811–2817.

[8] M. J. Powell, E. A. Cousineau, and A. D. Ames, “Model predictive control of underactuated bipedal robotic walking,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 5121–5126.

[9] E. Cousineau and A. D. Ames, “Realizing underactuated bipedal walking with torque controllers via the ideal model resolved motion method,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 5747–5753.

[10] X. Xiong, J. Reher, and A. D. Ames, “Global position control on underactuated bipedal robots: Step-to-step dynamics approximation for step planning,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 2825–2831.

[11] E. Daneshmand, M. Khadiv, F. Grimminger, and L. Righetti, “Variable horizon mpc with swing foot dynamics for bipedal walking control,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2349–2356, 2021.

[12] G. Gibson, O. Dosunmu-Ogunbi, Y. Gong, and J. Grizzle, “Terrain-aware foot placement for bipedal locomotion combining model predictive control, virtual constraints, and the alip,” *arXiv preprint arXiv:2109.14862*, 2021.

[13] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2018.

[14] Y. Hurmuzlu and D. B. Marghitu, “Rigid body collisions of planar kinematic chains with multiple contact points,” *The international journal of robotics research*, vol. 13, no. 1, pp. 82–92, 1994.

[15] S. Sovukluk, “Dynamic modeling and control of underactuated planar bipedal walking,” Master’s thesis, Middle East Technical University, 2022.

[16] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. Westervelt, C. C. De Wit, and J. Grizzle, “Rabbit: A testbed for advanced control theory,” *IEEE Control Systems Magazine*, vol. 23, no. 5, pp. 57–79, 2003.

[17] M. M. Ankarali and U. Saranli, “Stride-to-stride energy regulation for robust self-stability of a torque-actuated dissipative spring-mass hopper,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 20, no. 3, p. 033121, 2010.

[18] J. SCHMITT, “A simple stabilizing control for sagittal plane locomotion,” *Journal of computational and nonlinear dynamics*, vol. 1, no. 4, pp. 348–357, 2006.

[19] S. G. Carver, N. J. Cowan, and J. M. Guckenheimer, “Lateral stability of the spring-mass hopper suggests a two-step control strategy for running,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 19, no. 2, p. 026106, 2009.

[20] H. Geyer, A. Seyfarth, and R. Blickhan, “Compliant leg behaviour explains basic dynamics of walking and running,” *Proceedings of the Royal Society B: Biological Sciences*, vol. 273, no. 1603, pp. 2861–2867, 2006.

[21] S. H. Strogatz, *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. CRC press, 2018.