

Bipedal Walking Control using Variable Horizon MPC*

Elham Daneshmand^{1,2}, Majid Khadiv¹, Felix Grimminger¹, Ludovic Righetti^{1,3}

Abstract—In this paper, we present a novel two-level variable Horizon Model Predictive Control (VH-MPC) framework for bipedal locomotion. In this framework, the higher level computes the landing location and timing (horizon length) of the swing foot to stabilize the unstable part of the center of mass (CoM) dynamics, using feedback from the CoM state. The lower level takes into account the swing foot dynamics and generates dynamically consistent trajectories for landing at the desired time as close as possible to the desired location. To do that, we use a simplified model of the robot dynamics projected in swing foot space that takes into account joint torque constraints as well as the friction cone constraints of the stance foot. We show the effectiveness of our proposed control framework by implementing robust walking patterns on our torque-controlled and open-source biped robot, Bolt. We report extensive simulations and real robot experiments in the presence of various disturbances and uncertainties.

I. INTRODUCTION

Humanoid robots should be able to walk robustly on different terrains in the presence of various uncertainties. Hence, the main goal of a walking controller is to find an optimal set of contact schedule and feasible contact forces to robustly achieve a desired task. However, since the system is hybrid, nonlinear and highly constrained, solving the problem holistically is extremely hard [1], [2]. That is why most approaches based on optimal control use simplified dynamic models and solve multi-stage optimization to enable real-time computations. The main paradigm to break down the problem is to first decide the optimal contact sequence [3], [4], [5] and then optimize over the contact forces [6], [7] sometimes together with step location and timing adaptation [8].

For bipedal walking on regular surfaces, there is no ambiguity in contact sequence, i.e. the role of the stance and swing foot switches at each walking phase. Therefore, a walking controller should decide where and when to step and how to control highly constrained interaction forces to generate a desired walking behaviour and reject disturbances. Using the linear inverted pendulum model (LIPM), a walking controller can be written as a linear quadratic program that can be solved quickly [9]. Therefore, Linear Model predictive

control (MPC) has become a powerful tool for controlling bipedal walking [10], [11], [12], [13]. The optimal control problem (OCP) solved at each control cycle in these approaches focuses on the center of mass (CoM) dynamics and investigates how to stabilize it using force modulation, for predefined or adaptive step locations and timings. They all consider some proxy constraints to guide roughly the swing foot to touch the ground at the desired time and location. However, in order to make sure that this is the case, one needs to generate a feasible swing foot trajectory that establishes contact at the desired time and location, and be consistent with the CoM trajectory and planned step location and time.

Establishing contact is equivalent to imposing a state dependent switching constraint at a certain time, i.e. the swing foot should touch the ground at that time. This can be formalized using a variable horizon MPC (VH-MPC) framework [14], [15] where the main goal is to establish contact at a certain time (the horizon length which is a decision variable). In the VH-MPC framework, a terminal constraint (and cost) is considered at the end of the horizon, but rather than having a moving horizon with a fixed duration, the horizon length changes such that the switching constraint at a certain time remains a terminal constraint. In the special case where the time of the terminal constraint is fixed, this framework becomes similar to a shrinking horizon MPC [16], [17].

We argue that VH-MPC is a suitable framework to control walking of legged robots for two reasons. First, using VH-MPC, we can stabilize the CoM dynamics by only looking at the end of the current step (switching surface/manifold), optimizing both step location and timing, without a need to consider several steps (switches) in the horizon [18]. Second, the swing foot should land on the ground at a designed time and this is a final-value problem after which the swing foot becomes stance with different control objectives (modulate force to control the CoM). It is important to note that there is no reason in general to track a swing foot trajectory, the only thing that matters is landing at a certain time at a desired location. That is why VH-MPC *with terminal constraints* is a suitable approach to control the swing foot motion.

In this paper, we propose a two-level VH-MPC framework, where the high-level MPC adapts the next step location and timing to stabilize the divergent component of motion (DCM) of the CoM dynamics, and the low-level MPC takes into account the swing foot dynamics to land as close as possible to the planned location at the desired time. For the swing foot dynamics, we project the dynamics of the robot to the swing foot space. Since this dynamic model is nonlinear,

*This work was supported by the Max-Planck Institute for Intelligent Systems' Grassroots program (M10338), New York University, the European Union's Horizon 2020 research and innovation program (grant agreement 780684 and European Research Council's grant 637935) and the National Science Foundation (CMMI-1825993).

¹Max Planck Institute for Intelligent Systems, Tuebingen, Germany. firstrname.lastname@tuebingen.mpg.de

²Amirkabir University of Technology, Tehran, Iran. elham.daneshmand@aut.ac.ir

³Tandon School of Engineering, New York University, Brooklyn, USA. Email ludovic.righetti@nyu.edu

we use a linearized version of it. We also take into account the stance foot friction cone and joint torque constraints to find bounds on the available force that can be applied by the swing force. Using the swing foot dynamics and constraint on the available force, we write down a VH-MPC problem that finds a dynamically consistent trajectory for the swing foot to land at the desired position at the desired time given by the high-level MPC. Note that in the swing foot MPC problem, the horizon is given at each control cycle by the high-level MPC and is not part of the decision variables.

A similar approach to ours has been adopted for controlling robotic systems with contact in [19], where they showed successful ball-bouncing task. Considering the switching manifold to be the end of the horizon, they formulated a MPC problem (named first-exit) up to some set of terminal states, and applied a final cost at the terminal states equal to the differential cost-to-go for the infinite-horizon problem. However, they proposed some domain-specific heuristics to find the terminal cost. Our approach has fundamental differences with respect to [19], i.e. 1) We propose a set of viability-based terminal cost and constraint, where the effects of constraints after the terminal state are also taken into account 2) We use a two-stage constrained MPC problem with viability guarantees, compared to [19] that used unconstrained iLQG 3) we use a highly-underactuated biped robot, and demonstrate an extensive set of real-world experiments with various disturbances.

The main contributions of the paper are as follows:

- we propose a two-level VH-MPC framework that takes into account both the CoM and swing foot dynamics to control bipedal walking.
- We compare our proposed swing foot trajectory generation to polynomial based approach used in the literature.
- We demonstrate, for the first time, an MPC walking controller with both step location and timing adaptation on a real biped robot.
- We present walking experiments on uneven, soft, and slippery surfaces using our two-level VH-MPC on a robot without actuated ankles.

The paper is structured as follows: Section II briefly summarizes the high-level MPC problem that finds optimal step location and timing. In Section III, we present the procedure of projecting the robot dynamics to the swing foot and formulating its corresponding MPC problem. In Section IV, we summarize the whole control pipeline we use in the paper. In Section V we present an extensive set of simulation and experimental results. Finally, Section VI concludes the findings.

II. FOOT LOCATION AND TIMING ADJUSTMENT

Using the CoM and DCM as states, the LIPM dynamics can be written as [20]

$$\dot{c} = \omega_0(\xi - c) \quad (1a)$$

$$\dot{\xi} = \omega_0(\xi - u_0) \quad (1b)$$

where $c \in \mathbb{R}^2$ is the CoM horizontal position (CoM height has a fixed value z_0), and $\xi = (c + \dot{c}/\omega_0) \in \mathbb{R}^2$ is the 2-D

DCM. $u_0 \in \mathbb{R}^2$ is the CoP position, and in case of point contact feet, u_0 is identical to the contact point. ω_0 is the natural frequency of the pendulum ($\omega_0 = \sqrt{g/z_0}$, where g is the gravity constant). Equation (1) explicitly separates the stable and unstable parts of the LIPM dynamics, where the CoM converges to the DCM (1a) and the DCM is pushed away by the CoP (1b).

To stabilize the DCM dynamics, we proposed in [13], [18] to adapt the next step location and timing of the swing foot based on the DCM measurement ξ_{mea} using

$$\begin{aligned} \text{minimize}_{u_T, \tau, b_T} \quad & \alpha_1 \|u_T - u_0 - \begin{bmatrix} l_{nom} \\ w_{nom} \end{bmatrix}\|^2 + \alpha_2 |\tau - \tau_{nom}|^2 \\ & + \alpha_3 \|b_T - \begin{bmatrix} b_{x,nom} \\ b_{y,nom} \end{bmatrix}\|^2 \end{aligned} \quad (2a)$$

$$\text{s.t.} \quad \begin{bmatrix} l_{min} \\ w_{min} \end{bmatrix} \leq u_T - u_0 \leq \begin{bmatrix} l_{max} \\ w_{max} \end{bmatrix} \quad (2b)$$

$$e^{\omega_0 T_{min}} \leq \tau \leq e^{\omega_0 T_{max}} \quad (2c)$$

$$u_T + b_T = (\xi_{mea} - u_0)e^{-\omega_0 t} \tau + u_0 \quad (2d)$$

$$\tau \geq \tau_0 e^{\omega_0 t} \quad (2e)$$

$$\begin{bmatrix} b_{x,min} \\ b_{y,max,out} \end{bmatrix} \leq b_T \leq \begin{bmatrix} b_{x,max} \\ b_{y,max,in} \end{bmatrix} \quad (2f)$$

where $b_T = \xi_T - u_T$ is the DCM offset, $\tau = e^{\omega_0 T}$ is an exponential transformation of the step time T . u_0 and u_T are the current and next step locations, while l and w are the step length and width, respectively. $\tau_0 = e^{\omega_0 T_0}$, where T_0 is a minimum time required for the swing foot to touch the ground, given its current state (cf. Section III-D for computing τ_0). Subscript *nom* stands for the nominal value and *in* and *out* are the inward and outward lateral directions [18].

The three cost terms in (2a) try to bring the gait variables to their nominal values which correspond to a desired walking velocity. Constraints (2b), (2c) are box constraints on the location and (exponential of) time of the step. Equation (2d) is the DCM dynamics which is linear as a function of next step location and (exponential of) time and DCM offset. (2e) is a new constraint we define in this paper to prevent the planner to ask for an instantaneous step in the presence of uncertainties. Finally, (2f) is the viability constraint on the DCM offset which is implemented in practice as a soft constraint to make sure that the program always find a feasible solution, even if the system is not viable [18].

The last cost term and constraint on the DCM offset have crucial role in our walking controller. The former incentivizes the DCM offset toward a desired offset that corresponds to a desired walking velocity, while the latter ensures viability of the gait, i.e. there exist at least a set of feasible steps into the future by which the robot can prevent a fall. Solving (2) at each control cycle using the current measurement of the DCM yields the next step location and timing that should be realized by the swing foot trajectory. Since the step time is a decision variable, the horizon can be potentially adapted to preserve viability. Hence, (2) as the high-level MPC problem

of our framework is a VH-MPC.

III. PROJECTED SWING FOOT DYNAMICS

In this section, we examine the structure of the robot dynamics projected to the swing foot space. The main goal is to find a dynamic model with minimum simplification based on which we adapt the swing foot trajectories in real-time.

The dynamics of a floating-base system can be written as

$$M(\mathbf{q})\dot{\mathbf{v}} + \mathbf{h}(\mathbf{q}, \mathbf{v}) = B\boldsymbol{\tau} + J_c^T \boldsymbol{\lambda} \quad (3)$$

where $M \in \mathbb{R}^{(n+6) \times (n+6)}$ is the robot mass matrix, $\mathbf{q} \in SE(3) \times \mathbb{R}^n$ denotes the configuration space, $\mathbf{v} \in \mathbb{R}^{n+6}$ encodes the vector of generalized velocities (or more precisely quasi-velocities [21]), $\mathbf{h} \in \mathbb{R}^{n+6}$ is a concatenation of nonlinear terms including centrifugal, Coriolis and gravitational effects. $B \in \mathbb{R}^{(n+6) \times n}$ is a selection matrix that separates the actuated and unactuated Degrees of Freedom (DoFs), $\boldsymbol{\tau} \in \mathbb{R}^n$ is the vector of actuating torques, $J_c \in \mathbb{R}^{3m \times (n+6)}$ is the Jacobian of m end-effectors in contact, and finally $\boldsymbol{\lambda} \in \mathbb{R}^{3m}$ is the vector of contact forces (here we assume point-contact end-effectors).

For bipedal walking, we assume one of the robot's feet is in stationary contact with the ground. With this assumption, we can write down the constraint-consistent projected dynamics of the robot to the swing foot as [22]

$$\underbrace{\Lambda_c \ddot{\mathbf{x}}}_{\text{Inertia}} + \underbrace{\Lambda_c (JM_c^{-1} P \mathbf{h} - (\dot{\mathbf{J}} + JM_c^{-1} \dot{P}) \mathbf{v})}_{\text{nonlinear terms}} = \mathbf{f} \quad (4)$$

where P is the orthogonal projection operator $I - J_c^\dagger J_c$ (where \dagger stands for the Moore-Penrose inverse) which is a mapping to the nullspace of the contact constraint and $\dot{P} = J_c^\dagger \dot{J}_c$. The constraint inertia matrix is denoted by $M_c = PM + I - P$ and $\Lambda_c = (JM_c^{-1} P J^T)^{-1}$ is the apparent mass at the end-effector. In this equation, \mathbf{f} is the available force at the swing foot as a consequence of available joint torques and contact forces on the stance foot and it can be derived as

$$\mathbf{f} = \Lambda_c J M_c^{-1} P B \boldsymbol{\tau} \quad (5)$$

Given the friction cone constraints of the stance foot and joint torque limits, the available force to move the swing foot is constrained. In the sequel we find a conservative approximation of these constraints and a simplified dynamics of the swing foot that can be used to generate dynamically consistent swing foot trajectories.

A. Simplified swing foot dynamics

The swing foot dynamics (4) is highly nonlinear as a function of the robot configuration. As a result, using this dynamic model in MPC leads to a nonlinear and non-convex optimization problem that needs several iterations to be solved and it might get stuck in undesired local minima. However, we need to be able to regenerate the swing foot trajectory quickly based on the updated landing location and time from the high-level DCM planner. Hence, we resort to a simplified linear dynamics of the foot in the reachable area of the swing foot.

To be able to compare the effects of different terms in (4), we used the approach in [18] and generated different swing foot trajectories using polynomials in simulation for different walking velocities. We exerted random disturbances during these motions (we reset the simulation if the robot falls down) and used a whole body controller (WBC) to find joint trajectories for these motions. Using the state trajectories at each time, we compute the effects of different terms. In Fig. 1 we show the contribution of the inertia and nonlinear terms to the swing foot force for 50 different swing phase (different landing location and time). Note that we did all the simulations on the robot we study in this paper, Bolt. As we can see in Fig. 1, inertia is the main effect in swing foot dynamics while the nonlinear terms remain relatively constant. In an effort to simplify the swing foot dynamics, we approximate all the nonlinear terms with a constant force \mathbf{h}_c .

Now, we inspect further the swing foot mass matrix structure and plot its variation as a function of the robot configuration, in the range of swing foot motion that we are interested in. As the mass matrix is symmetric, we only need to examine the diagonal and upper(or lower)-triangular part of the matrix. In Fig. 2, we plotted the components of the swing foot mass matrix, for the same set of motion we used to compare inertial and nonlinear effects. As we can see in this figure, the projected mass matrix components do not change significantly during one step, hence we propose to use the following linear dynamics to be used inside a linear MPC problem for the swing foot.

$$\mathbf{f} = \Lambda \ddot{\mathbf{x}} + \mathbf{h}_c \quad (6)$$

where the components of Λ and \mathbf{h}_c are constant.

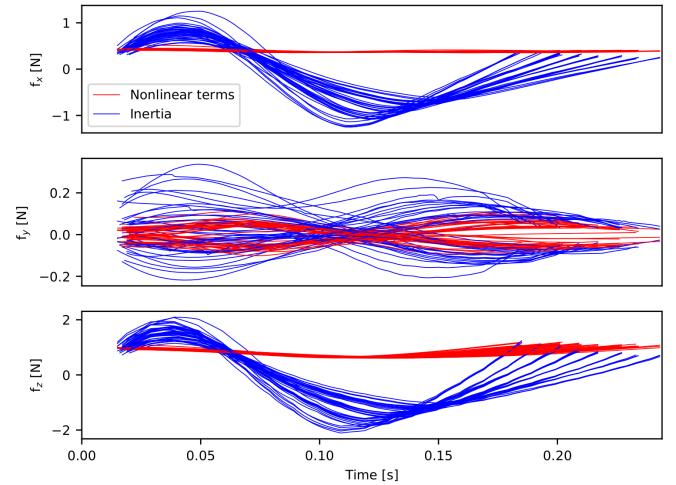


Fig. 1. inertial vs nonlinear effects at the swing foot for a set of 50 different landing location and time

B. constraints on the swing foot available force

In this section, we aim at finding bounds on the available swing foot forces, given the joint torque limits and the stance foot friction cone constraints. The idea is to use the set

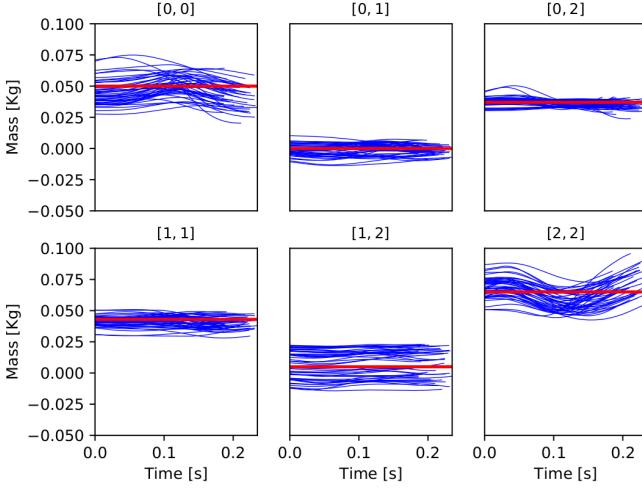


Fig. 2. Mass matrix elements for 50 different landing location and time; the mean value of each component is specified by red.

of different configurations we considered in the previous section and construct the following quadratic program (QP) to find the boundaries of the forces that can be applied by the swing foot without violating any of the constraints at each configuration

$$\underset{\boldsymbol{\tau}}{\text{minimize}} \quad J \triangleq \mathbf{s}^T \mathbf{f} = \mathbf{s}^T \Lambda_c J M_c^{-1} P B \boldsymbol{\tau} \quad (7a)$$

$$\text{s.t.} \quad \rho_z + \eta_z \geq 0 \quad (7b)$$

$$\mu(\rho_z + \eta_z) \geq \sqrt{(\rho_x + \eta_x)^2 + (\rho_y + \eta_y)^2} \quad (7c)$$

$$\tau_{\min} \leq \boldsymbol{\tau} \leq \tau_{\max} \quad (7d)$$

In this equation, $\rho = -(J_c^T)^{\dagger}(I - P)((I - MM_c^{-1}P)\mathbf{h} + MM_c^{-1}\dot{P}\mathbf{v})$ and $\eta = -(J_c^T)^{\dagger}(I - P)(I - MM_c^{-1}P)B\boldsymbol{\tau}$ [23]. $\mathbf{s} \in \mathbb{R}^3$ in (7a) is a selection vector whose one of elements is either +1 or -1, and the rest are zero. By iterating over all six possibilities, we can compute the maximum and minimum available force in each direction, for a given configuration. The constraints of the problem are unilaterality of contact (7b), friction cone constraint (7c) of the stance foot, and the joint torque limits (7d).

We solved (7) for the same set of trajectories we used in Section III-A. Figure 3 shows the time history of the maximum and minimum of the available swing foot force in each direction. We also found a conservative bound (red lines) on the forces to be used in the swing foot MPC problem. The conservative bounds on the maximum and minimum forces will be used inside an MPC problem in the next subsection to generate swing foot trajectories. Note that for solving (7) we set the friction coefficient to $\mu = 0.5$, and the minimum and maximum joint torques to -2 N.m and 2 N.m which are robot-dependent.

C. Proposed foot trajectory generator

In this section, we formulate the swing foot reactive trajectory generation problem as a VH-MPC. The main goal of this MPC is to bring the swing foot on the ground at the

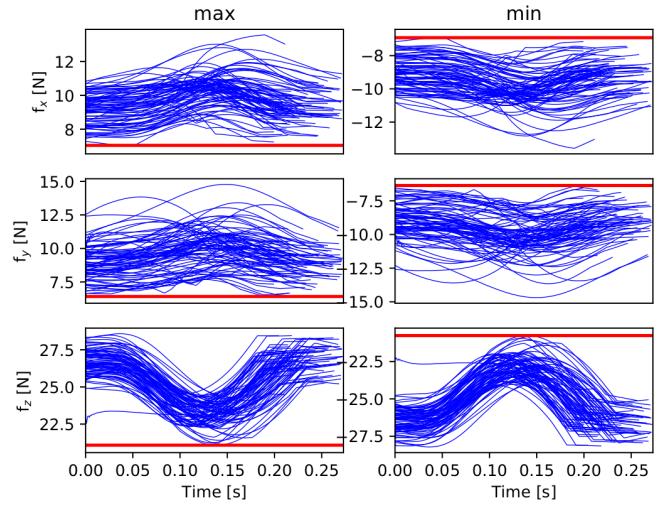


Fig. 3. Maximum and minimum available force at the swing foot for 50 different landing location and time; a conservative upper/lower bound is specified by red.

desired time computed by (2) and to be as close as possible to the desired foot location.

We use the simplified swing foot dynamics in (6) and the conservative approximation of the force boundaries computed based on (7). We write then the following VH-MPC to find optimal force applied to the swing foot to land on the ground at a desired time and as close as possible to the desired location

$$\underset{\mathbf{f}_i}{\text{minimize}} \quad \sum_{i=1}^N \alpha_1 \|\mathbf{f}_i\|^2 + \alpha_2 \|\mathbf{x}_N - \mathbf{x}_f\|^2 + \dots$$

$$\alpha_3 \|\dot{\mathbf{x}}_N - \dot{\mathbf{x}}_f\|^2 + \alpha_4 \|z(\frac{T}{2}) - z_{mid}\|^2 \quad (8a)$$

$$\text{s.t.} \quad \mathbf{x}_i = A^i \mathbf{x}_0 + \sum_{i=1}^N A^{N-i} B^i ((\mathbf{f}_i - \mathbf{h}_c) \Lambda_c^{-1})^T \quad (8b)$$

$$\mathbf{f}_{\min} \leq \mathbf{f}_i \leq \mathbf{f}_{\max}, \quad \forall i = 1, \dots, N. \quad (8c)$$

$$z_{\min} \leq z_i \leq z_{\max}, \quad \forall i = 1, \dots, N. \quad (8d)$$

$$z_f = 0, \quad \dot{z}_f = 0 \quad (8e)$$

where A and B are obtained, writing (6) in standard discrete state space form. \mathbf{f}_i is the applied force at the swing foot at i 'th node in the horizon, $\mathbf{x} = [x, y, z]^T$ and $\dot{\mathbf{x}} = [\dot{x}, \dot{y}, \dot{z}]^T$ are the swing foot position and velocity and subscript f stands for the end of the current step (horizon). z_{mid} is the desired step height. Note that in this program, only the terminal state of the swing foot in vertical direction are constraint which ensures landing at the *desired time*.

D. The minimum landing time

Depending on the current state of the swing foot, a minimum time can be computed that ensures landing the foot on the ground, given the constraints on the available force. This minimum time is used in (2e) to make sure that (2) does not ask for a contact time that is not possible to be realized by the swing foot in (8). We compute this minimum

time using the following program

$$\underset{f_i, i}{\text{minimize}} \quad i \quad (9a)$$

$$\text{s.t. } \mathbf{x}_i = A^i \mathbf{x}_0 + \sum_{i=1}^N A^{N-i} B^i ((\mathbf{f}_i - \mathbf{h}_c) \Lambda_c^{-1})^T \quad (9b)$$

$$\mathbf{f}_{min} \leq \mathbf{f}_i \leq \mathbf{f}_{max}, \quad \forall i = 1, \dots, N. \quad (9c)$$

$$z_f = 0, \dot{z}_f = 0 \quad (9d)$$

Equation (9) is a very simple mixed-integer program (i is an integer variable) that can be solved very fast. In our experiments, we used a simple approach to solve this program; we start from zero and increase i and solve (9) until the program becomes feasible. In our experiments, such a brute-force approach to solve (9) took in worst case around $1 \mu s$.

IV. FULL CONTROL PIPELINE

In this section, we summarize the full control pipeline we use in this paper. The main parts of our control pipeline are two VH-MPC problems (2) and (8) that reactively update the next contact location and time, and accordingly the swing foot trajectory. The generated trajectories then are fed into a WBC to compute the joint torques. We used in this paper the task-space impedance controller we proposed in [24] as a WBC. The full control pipeline is summarized in Algorithm 1.

Algorithm 1: Our proposed control framework

Input: Walking velocity;
initialize nominal gait values and their boundaries;
while not interrupted by the user **do**
 | Use (9) to find the minimum stepping time T_0 ;
 | Use (2) to update the step location and time;
 | Use (8) to generate the swing foot trajectory;
 | Use WBC [24] to compute joint torques;
end

After specifying the desired walking velocity and nominal gait parameter (step length, width, time, ...), the first step of the algorithm uses (9) to compute the minimum time required for the swing foot to land on the ground. Then, (2) is executed to find the optimal step location and timing of the step, given the current measurement of the DCM. This stage is formulated as a VH-MPC where the MPC horizon (step timing) is a decision variable. The resulting step location and timing are then used to generate a swing foot trajectory using the swing foot MPC (8). The goal in the swing foot MPC is to realize landing of the foot at the time that is planned by the first stage, as close as possible to the desired position. In this stage we also have a VH-MPC, but the horizon is not decided by the MPC itself and it is set by the first stage MPC. Finally, the resulting trajectories are passed to the WBC to compute joint torques.

Remark 1: The main reason behind splitting the problem into two MPC problems is that one holistic MPC that finds

both the swing foot trajectory and optimal step location and timing would result in either a mixed-integer program or a non-convex problem. A mixed-integer program is not suitable for real-time applications because of its combinatorial complexity. A non-convex optimization needs several iterations to converge and may get stuck in a poor local minimum. We prefer our two-level MPC despite the fact that it is sub-optimal, because 1) it can be solved in real-time inside MPC 2) it always finds a feasible solution using the minimum time (9) as a constraint in (2).

V. RESULTS

In this section, we first present a comparison between our proposed approach for swing foot trajectory generation to a polynomial-based trajectory generation approach. Then, we show an extensive set of real robot experiments demonstrating the capability of our framework in generating robust walking motions in the presence of various disturbances and uncertainties. For all the experiments, we ran the two-level MPC in Algorithm 1 at 100 Hz and the WBC at 1 kHz.

A. Swing foot trajectories comparison

1) *Case study:* To compare qualitatively the performance of our proposed MPC-based swing foot trajectory generation (8) to a standard polynomial-based one [18], we consider a push recovery scenario in simulation. We did a simulation where the robot steps in place and is pushed laterally by an impulse of 10 N.s, at $t = 0.6$ sec.

All components of the control pipeline are exactly the same, except the swing foot trajectory generation part. For the polynomial-based swing foot trajectory generation, we parameterize the trajectories in horizontal directions using minimum-jerk (5th order) polynomials and find the unique coefficients that connect the current position, velocity and acceleration of the swing foot to the end position with zero velocity and acceleration. For the vertical direction, we construct a QP with inequality constraints (the foot height should remain in a certain bound) to find the coefficients of the polynomial. In the case that the corresponding problem gets infeasible, we use the solution of the previous iteration and shift it by one sample time. Compared to the heuristics used in [18] that we would not update the gait values at the instances close to the end of the step, here we use this simple but more systematic and practical strategy of shifting the previous solutions and this was enough to always have a feasible solution in our experiments.

Figure 4 compares the resulting swing foot tracking performance of the two approaches. As we can see, after the push is exerted at $t = 0.6$ sec, the next step location is updated using (2). Since the polynomial-based swing foot trajectory generation (left figure) gets infeasible at this time, the solution in the previous iteration is used. This causes a discrepancy between the desire landing point asked by (2) and the end point of the trajectory. This is enough for the robot to get unstable and fall down. However, using our novel MPC-based approach (8), the swing foot is adapted after the

disturbance as much as the constraints allow and the robot is able to recover from the disturbance.

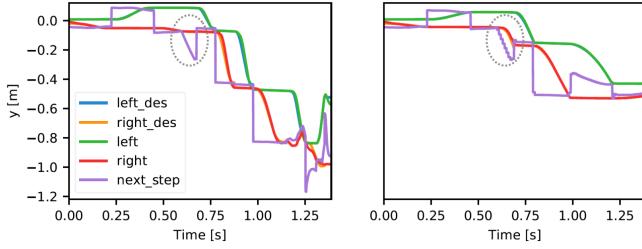


Fig. 4. Comparison between swing foot trajectories in the y direction generated using polynomials (left) and proposed MPC (8) (right). The robot is pushed laterally at $t = 0.6$ sec by an approximate impulse of 10 N.s. As it is specified by the dashed circle, after the push the desired next step location is adjusted. The polynomial-based trajectory generation method is not able to find a feasible solution to adapt the trajectory, hence uses the solution from previous iterations and cannot recover from the disturbance. However, our proposed approach based on MPC can adapt the swing foot trajectory consistent with the dynamics and constraints and enable recovery from the push.

2) Systematic comparison: To compare systematically the two approaches, we conducted 450 walking steps simulation with different pushes sampled inside the range of ± 1.5 N.s for the horizontal directions and ± 0.75 N.s in the vertical direction. Note that this range is small compared to the disturbance we exerted in the previous subsection (10 N.s). In case of large disturbances, the robot might lose its balance and this would make it difficult to compare the landing location and time of the swing foot between two approaches. We already showed in the previous subsection that in the case of a large disturbance, our approach clearly outperforms the polynomial-based trajectory generation approach.

As we can see in Fig. 5, in both x and y directions, the error in landing location of our approach is less than the polynomial-based trajectory generation. Note that error in landing location can not only cause a degradation in velocity tracking (performance), but it may also jeopardize stability. More important than landing location error, landing time error of the swing foot in our approach is significantly lower than polynomial-based approach (see Fig. 6). Landing time error has a very important role on realization of the gait, i.e. premature landing can cause high impact forces, while late landing can cause high acceleration of the foot once the gait phase changes, given that the controller is not event-based.

B. Real robot experiment

We implemented our proposed framework in Algorithm 1 on Bolt, our new open-source biped robot. Bolt has been designed and fabricated using the 3D-printing technology and proprioceptive actuator concept that gives transparency in the drive system for torque-control proposed in [24]. The robot is capable of very dynamic movements and fast stepping, which renders the swing foot dynamics non negligible. Each leg of Bolt has 3 active DoFs (2 in hip and one in knee) together with a passive ankle joint. We considered 5 different scenarios for our experiments that we will explain briefly in the following. Note that we use the exact same controller

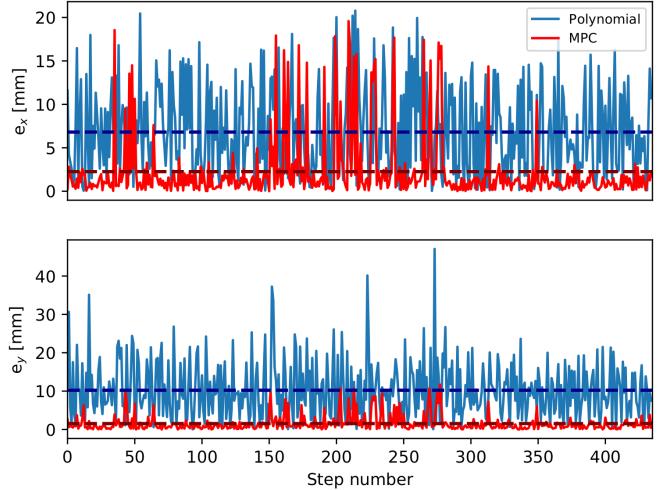


Fig. 5. Landing location error for 450 walking steps simulation with random pushes sampled inside the range of ± 1.5 N.s for the horizontal directions and ± 0.75 N.s for the vertical direction. The mean values for each approach are shown by dashed lines with the same colour.

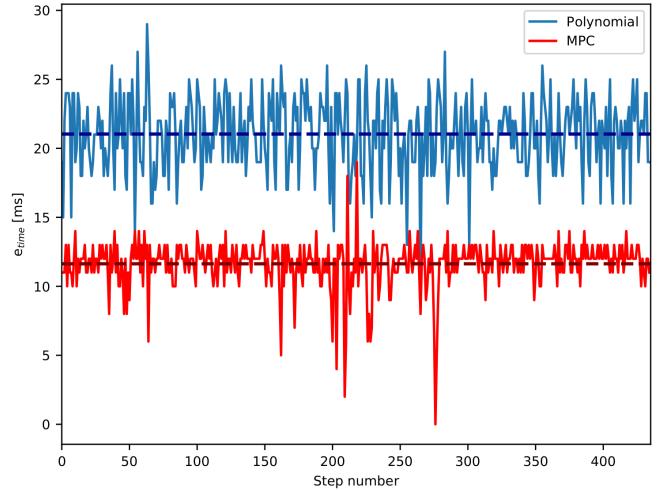


Fig. 6. Landing time error for 450 walking steps simulation with random pushes sampled inside the range of ± 1.5 N.s for the horizontal directions and ± 0.75 N.s for the vertical direction. The mean values for each approach are shown by dashed lines with the same colour.

parameters for all these experiments. The snapshots of the experiments are shown in Fig. 7. The short accompanying video illustrates the experiments and the full version of the video can be found here.

1) Forward/Backward walking: The robot is commanded to walk forward and backward on a flat surface with a desired velocity. Through this test we showed that our feedback control based on simplified models is robust enough to stabilize walking motion for a long time. As it can be seen in the video, the robot is able to walk forward and backward for 10 minutes without any problem.

2) Walking on uneven surfaces: We commanded stepping in place (zero desired velocity) on an uneven surface where the surface height is varied up to 4 cm. The walking controller does not have any notion of the height variations;



Fig. 7. Snapshots of the experiments, from left to right: 1) forward walking, 2) uneven surface, 3) soft surface, 4) push recovery, 5) slippage recovery.

however, thanks to the adaptation of the step location and time and compliance due to torque-control, the robot can successfully step on surfaces with unknown heights.

3) Walking on soft surfaces: Bolt performs in-place stepping on a soft surface (sponge, see the video). This surface is specially challenging, as the stance foot does not remain fixed but oscillates due to the surface compliance. These oscillations would magnify shaking of the robot due to the structural flexibility. Although the robot would start shaking at some instances of the motion, the controller was able to damp them and the robot could successfully step on this surface without any instability.

4) Push recovery: In this scenario, we exert external forces to the robot's pelvis in different direction, at different times of a step and in different directions. Thanks to the fast update of the control loop, the robot quickly reacts to the pushes and adapts both step location and timing to recover. We performed push recovery scenarios both in sagittal and lateral directions.

5) Slippage recovery: In the last scenario, the robot performs stepping on a surface that can slide on the ground. We disturbed this surface in different directions randomly during stepping, and the robot again was able to adapt the landing location and time of the swing foot to preserve its balance.

VI. CONCLUSIONS

In this paper, we presented a two-level variable horizon MPC to control bipedal walking. In the high level, given the current state of the CoM, the landing location and time of the swing foot is decided. In the lower level, the swing foot trajectory is adapted such that the swing foot lands on the ground at the desired time, as close as possible to the desired location. We integrated this two-level VH-MPC with an impedance-based WBC to compute torque commands to the robot. We conducted an extensive set of simulations and experiments through which we demonstrated the capabilities and robustness of our proposed control framework.

In future work, we plan to generalize our proposed approach to more general motions, i.e. 3D walking and running. Using the statistical information in the discrepancy between the nominal and real swing foot dynamics, we would like to extend our formulation to robust and stochastic MPC frameworks [25]. Another interesting extension of the current work is alternating between the two levels of the MPC such that they come to a consensus in terms of contact location

and timing at each control cycle. Finally, our approach with contact switch as a terminal constraint can be easily generalized to an event-based controller where the switch in the gait phase is triggered at the detection of swing foot contact.

REFERENCES

- [1] 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.
- [2] S. Lengagne, J. Vaillant, E. Yoshida, and A. Kheddar, “Generation of whole-body optimal dynamic multi-contact motions,” *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1104–1119, 2013.
- [3] R. Deits and R. Tedrake, “Footstep planning on uneven terrain with mixed-integer convex optimization,” in *2014 IEEE-RAS international conference on humanoid robots*. IEEE, 2014, pp. 279–286.
- [4] S. Tonneau, A. Del Prete, J. Pettré, C. Park, D. Manocha, and N. Mansard, “An efficient acyclic contact planner for multiped robots,” *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 586–601, 2018.
- [5] Y.-C. Lin, B. Ponton, L. Righetti, and D. Berenson, “Efficient humanoid contact planning using learned centroidal dynamics prediction,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5280–5286.
- [6] H. Dai and R. Tedrake, “Planning robust walking motion on uneven terrain via convex optimization,” in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*. IEEE, 2016, pp. 579–586.
- [7] J. Carpenter and N. Mansard, “Multicontact locomotion of legged robots,” *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1441–1460, 2018.
- [8] B. Ponton, M. Khadiv, A. Meduri, and L. Righetti, “Efficient multi-contact pattern generation with sequential convex approximations of the centroidal dynamics,” <https://arxiv.org/abs/2010.01215>, 2020.
- [9] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, “Biped walking pattern generation by using preview control of zero-moment point,” in *Robotics and Automation (ICRA), IEEE International Conference on*. IEEE, 2003, pp. 1620–1626.
- [10] P. Wieber, “Trajectory free linear model predictive control for stable walking in the presence of strong perturbations,” in *2006 6th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2006, pp. 137–142.
- [11] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, “Online walking motion generation with automatic footstep placement,” *Advanced Robotics*, vol. 24, no. 5-6, pp. 719–737, 2010.
- [12] S. Feng, X. Xinjilefu, C. G. Atkeson, and J. Kim, “Robust dynamic walking using online foot step optimization,” in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 5373–5378.
- [13] M. Khadiv, A. Herzog, S. A. A. Moosavian, and L. Righetti, “Step timing adjustment: A step toward generating robust gaits,” in *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*. IEEE, 2016, pp. 35–42.
- [14] R. C. Shekhar and J. M. Maciejowski, “Robust variable horizon mpc with move blocking,” *Systems & Control Letters*, vol. 61, no. 4, pp. 587–594, 2012.
- [15] G. Mirzaeva and G. C. Goodwin, “Harmonic suppression and delay compensation for inverters via variable horizon nonlinear model predictive control,” *International Journal of Control*, vol. 88, no. 7, pp. 1400–1409, 2015.

- [16] S. S. Farahani, R. Majumdar, V. S. Prabhu, and S. Soudjani, “Shrinking horizon model predictive control with signal temporal logic constraints under stochastic disturbances,” *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3324–3331, 2018.
- [17] H. Farooqi, L. Fagiano, P. Colaneri, and D. Barlini, “Shrinking horizon parametrized predictive control with application to energy-efficient train operation,” *Automatica*, vol. 112, p. 108635, 2020.
- [18] M. Khadiv, A. Herzog, S. A. A. Moosavian, and L. Righetti, “Walking control based on step timing adaptation,” *IEEE Transactions on Robotics*, 2020.
- [19] P. Kulchenko and E. Todorov, “First-exit model predictive control of fast discontinuous dynamics: Application to ball bouncing,” in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 2144–2151.
- [20] J. Englsberger, C. Ott, and A. Albu-Schaffer, “Three-dimensional bipedal walking control based on divergent component of motion,” *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 355–368, 2015.
- [21] H. Baruh, *Analytical dynamics*. WCB/McGraw-Hill Boston, 1999.
- [22] M. Mistry and L. Righetti, “Operational space control of constrained and underactuated systems,” *Robotics: Science and systems VII*, pp. 225–232, 2012.
- [23] G. Xin, W. Wolfslag, H.-C. Lin, C. Tiseo, and M. Mistry, “An optimization-based locomotion controller for quadruped robots leveraging cartesian impedance control,” *Frontiers in Robotics and AI*, vol. 7, p. 48, 2020.
- [24] F. Grimminger, A. Meduri, M. Khadiv, J. Viereck, M. Wüthrich, M. Naveau, V. Berenz, S. Heim, F. Widmaier, T. Flayols *et al.*, “An open torque-controlled modular robot architecture for legged locomotion research,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3650–3657, 2020.
- [25] A. Gazar, M. Khadiv, A. Del Prete, and L. Righetti, “Stochastic and robust mpc for bipedal locomotion: A comparative study on robustness and performance,” *arXiv preprint arXiv:2005.07555*, 2020.