Full-Body Optimal Control Toward Versatile and Agile Behaviors in a Humanoid Robot

Koji Ishihara, Takeshi D. Itoh, and Jun Morimoto

***Abstract*—In this letter, we develop an optimal control framework that takes the full-body dynamics of a humanoid robot into account. Employing full-body dynamics has been explored in, especially, an online optimal control approach known as model predictive control (MPC). However, whole-body motions cannot be updated in a short period of time due to MPC's large computational burden. Thus, MPC has generally been evaluated with a physical humanoid robot in a limited range of tasks where high-speed motion executions are unnecessary. To cope with this problem, our multi-timescale control framework drives whole-body motions with a computationally efficient hierarchical MPC. Meanwhile, a biologically inspired controller maintains the robot's posture for a very short control period. We evaluated our framework in skating tasks with simulated and real lower-body humanoids that have rollers on the feet. Our simulated robot generated various agile motions such as jumping over a bump and flipping down from a cliff in real time. Our real lower-body humanoid also successfully generated a movement down a slope.**

***Index Terms*—Optimization and Optimal Control, Humanoid Robots, Motion Control.**

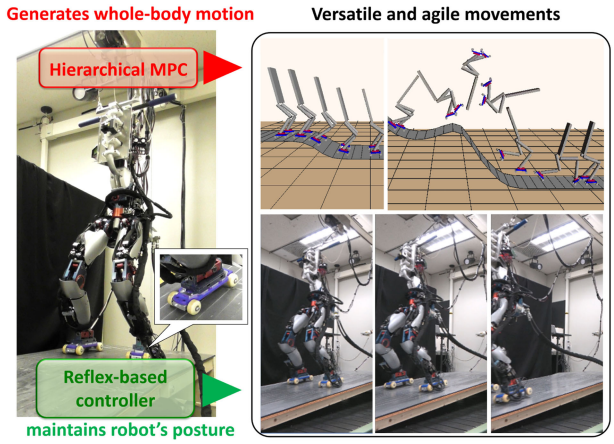


Fig. 1. Schematic diagram of our control framework: Hierarchical MPC with reflex-based controller successfully generated versatile motions in real-time.

I. Introduction

ALTHOUGH humanoid robots have been promoted for working in such real environments as hazardous situations instead of humans, they remain difficult to generate human-like behaviors, especially in terms of versatility and agility [1].

We can partially attribute their lack of versatility and agility to their control approaches that approximate a humanoid robot as a highly reduced model like an inverted pendulum [2], [3]. Such a model can only be applied to a limited range of tasks since generable movements are significantly restricted [4]. An inverted pendulum represents only a slow dynamical component of a biped robot [5].

Manuscript received June 7, 2019; accepted September 24, 2019. Date of publication October 11, 2019; date of current version November 20, 2019. This letter was recommended for publication by Associate Editor Y. Guo and Editor D. Song upon evaluation of the reviewers' comments. This work was supported in part by JSPS KAKENHI under Grant JP16H06565, Grant JP19K21540, Grant JP18H06472, and Grant MIC-SCOPE (182107105), and in part by NEDO, JST-Mirai Program under Grant JPMJMI18B8, Japan. *(Corresponding author: Koji Ishihara.)*

K. Ishihara and J. Morimoto are with the Department of Brain Robot Interface, ATR Computational Neuroscience Laboratories, Kyoto 619-0288, Japan (e-mail: ishihara-k@atr.jp; xmorimo@atr.jp).

T. D. Itoh is with the Department of Brain Robot Interface, ATR Computational Neuroscience Laboratories, Kyoto 619-0288, Japan and also with the Graduate School of Information Science, Nara Institute of Science and Technology, Nara 630-0192, Japan (e-mail: takeshi.itoh@atr.jp).

This letter has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/LRA.2019.2947001

We thus aim to develop a control framework that takes full-body dynamics into account. In previous studies, an online optimal control approach known as model predictive control (MPC) generated *simulated* robot motions under full-body dynamics [4]. MPC is, however, computationally intensive because a large optimization problem needs to be solved at each time-step when full-body dynamics are used as constraints [6]. Thus, a *real* humanoid robot has to be controlled for a long control period due to large computational time, and MPC has been applied to limited tasks in which fast movements are unnecessary [7].

To cope with this problem, we propose a novel multi-timescale control framework in this letter. A schematic diagram of our proposed framework is depicted in Fig. 1. A two-layered hierarchical MPC (HMPC) with two different time scales generates whole-body motions in a much shorter computational time than conventional MPC approaches. A biologically inspired reflex-based controller maintains the robot's posture with an even smaller time scale.

In this study, skating tasks are employed to evaluate our proposed framework. Since we are interested in generating wide variety of human-like agile motions, skating is an illustrative example where such motions are generated by efficiently utilizing potential and kinetic energies. Therefore, we used a lower-body humanoid model that wore roller skates. We constructed ground surfaces that were extracted from a standard skating-rink design in a simulated environment and built a slope for our actual lower-body humanoid.

We show that a variety of agile movements like jumping over a bump and flipping down from a cliff of our simulated robot were generated in real time. Furthermore, our real humanoid successfully generated a movement for going down a slope as an initial attempt to show control performance of our proposed method in the real environment towards the agile skating motion generation.

II. Related Works

Although there are previous studies working on the humanoid skating motion [8], [9], they utilized a task-specific low-dimensional approximated model. For instance, an inverted pendulum model was considered by assuming that the rate of upper-body angular momentum of a robot is zero. This assumption restricts generable movements. In other words, jumping over a bump or flipping down from a cliff motions cannot be generated with that restriction.

MPC approaches under full-body dynamics have been deemed unsuitable for real-time control of a high-dimensional practical system. They have been evaluated on a low-dimensional system such as a hovercraft [10], a ballbot and a hexacopter [11]. Their degrees of freedoms (DoFs) were up to six. With such small DoFs, calculation periods of MPC could be less than 5 ms. On the other hand, DoFs of the original humanoid model tends to be more than double. Thus, solving MPC problems under the full-body dynamics has only been evaluated in simulated environment [4], [6]. One exception was the study that applied MPC to a real humanoid robot [7]. However, fast movements were not demonstrated because they implemented MPC with the relatively large control period of 50 ms. Since the typical control period of less than 5 ms cannot be achieved with MPC, a restricted model of an inverted pendulum has been used for generating balance or biped locomotion [12], [13]. As a matter of practice, offline calculation tends to be used to solve an optimal full-body control problem for real environment [14]. For whole-body quadruped robot control, fast MPC calculation method with parallel computation and an optimal model switching approach were proposed [15], [16].

In this study, we develop a novel multi-timescale control framework for a humanoid robot. We formulate small-sized MPC problems using the full-body dynamics, which are solved in a hierarchical manner at a higher level. In a lower level, whole-body plans are executed while maintaining the robot's posture with a fast-rate reflex-based controller. The ways of adopting a hierarchical structure to MPC have been explored to ease the computational burden of MPC [17], [18]. In our previous preliminary study, we investigated the usefulness of a hierarchical structure for MPC calculation particularly on a full-body humanoid model [19]. In this letter, we propose a more computationally efficient HMPC approach; eigenvalue analysis is not required to extract the hierarchical structure from the original humanoid model. The reflex-based controller is newly adopted because the computational cost is very low and reflex-based systems have been successfully used to generate real robot movements [20], [21].

III. Model Predictive Control

A. A Standard MPC Problem

In MPC at each control period, a finite optimal control problem is solved to find an optimal control sequence that minimizes the accumulated cost over a finite future horizon. The accumulated cost is:

$$J(\mathbf{x}_k, \mathbf{U}_k) = \sum_{t=k}^{k+N-2} \ell(\mathbf{x}_t, \mathbf{u}_t) + \ell_T(\mathbf{x}_{k+N-1}), \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{u} \in \mathbb{R}^m$ respectively denote the state vectors and the control inputs and $\mathbf{U}_k \equiv \{\mathbf{u}_k, \mathbf{u}_{k+1}, \ldots, \mathbf{u}_{k+N-2}\}$ is the control sequence. $\ell(\mathbf{x}, \mathbf{u})$ is the immediate cost, and $\ell_T(\mathbf{x}) = \ell(\mathbf{x}, \mathbf{0})$ is the terminal cost. The finite-horizon optimal control problem is formulated as

$$\begin{aligned} &\min_{\mathbf{U}_k} J(\mathbf{x}_k, \mathbf{U}_k) \\ &\text{s.t. } \mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \end{aligned} \tag{2}$$

where the dynamics model is used as a constraint. In this letter, we use a full-body dynamics model (described in Sec. IV-A). Δt denotes a step size (a control period).

By solving the above optimal control problem, an optimal control sequence $\mathbf{U}_k^\star$ can be obtained. Simultaneously, an optimal state sequence $\mathbf{X}_k^\star \equiv \{\mathbf{x}_k^\star, \mathbf{x}_{k+1}^\star, \ldots, \mathbf{x}_{k+N-1}^\star\}$ is also gained. However, the sequences cannot be analytically derived for nonlinear systems in general. Instead, to derive a locally optimal controller, we use a Differential Dynamic Programming (DDP) [22], [23] approach called iterative Linear Quadratic Regulator (iLQR) [6], [24] in this letter.

Once the optimal control sequence $\mathbf{U}_k^\star$ is derived, the first few elements of the sequence are applied to the robot. After a new state is measured at subsequent time-step $k+1$, optimal control sequence $\mathbf{U}_{k+1}^\star$ is derived and the robot is controlled in the same manner.

B. A Hierarchical MPC Design

For a high-dimensional system such as a humanoid robot, the optimal control problem (2) is required to be solved within a very short time period while computing the large number of input variables. To reduce such large computational burden, we proposed a HMPC design [19]. Computational burden of MPC with a gradient-based optimization algorithm is related to prediction length N and dimension of state space n [25]. Therefore, we formulated two smaller optimal control problems for either of them. They are sequentially solved as an upper- and a lower-layer MPC.

The upper-layer MPC derives a whole system behavior coarsely; the same optimal control problem of (2) except a coarse-time-step ($\Delta t_c > \Delta t$) is solved to derive optimal control and state sequence for the coarse optimization $\tilde{\mathbf{U}}_k^\star \equiv \{\tilde{\mathbf{u}}_k^\star, \tilde{\mathbf{u}}_{k+1}^\star, \ldots, \tilde{\mathbf{u}}_{k+N_c-2}^\star\}$ and $\tilde{\mathbf{X}}_k^\star \equiv \{\tilde{\mathbf{x}}_k^\star, \tilde{\mathbf{x}}_{k+1}^\star, \ldots, \tilde{\mathbf{x}}_{k+N_c-1}^\star\}$. N_c is a horizon for the upper-layer optimization ($N_c \Delta t_c = N \Delta t$). The upper-layer optimization problem is, thus, smaller than the original one in terms of the prediction horizon ($N_c < N$).

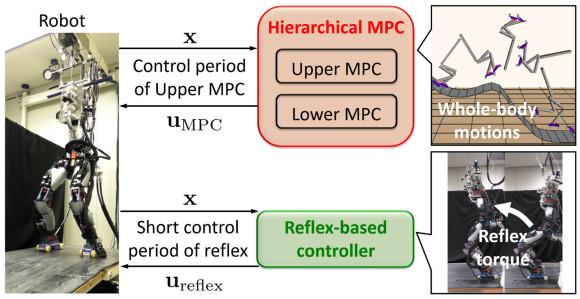


Fig. 2. Control diagram: Whole-body motions are generated with hierarchical MPC while reflex controller maintain robot's posture in very short period.

To generate a fast movement, the coarse input sequence derived in the upper-layer optimization need to be further optimized with a fine-time resolution, but the modification must be less time-consuming. Let define vectors of slow and fast state variable as $\mathbf{y} \in \mathbb{R}^{n_s}$ and $\mathbf{z} \in \mathbb{R}^{n_f}$ respectively, and slow and fast control variables as $\mathbf{u}^{\mathbf{y}} \in \mathbb{R}^{m_s}$ and $\mathbf{u}^{\mathbf{z}} \in \mathbb{R}^{m_f}$, where $n = n_s + n_f$ and $m = m_s + m_f$. The lower-layer MPC is formulated as an optimal control problem to refine only the fast control sequence $\mathbf{U}^{\mathbf{z}}_k \equiv \{\mathbf{u}^{\mathbf{z}}_k, \mathbf{u}^{\mathbf{z}}_{k+1}, \ldots, \mathbf{u}^{\mathbf{z}}_{k+N_f-2}\}$ by minimizing a total cost for the fast variables

$$J^f(\mathbf{z}_k, \mathbf{U}^{\mathbf{z}}_k, \tilde{\mathbf{X}}^{\star}_k, \tilde{\mathbf{U}}^{\star}_k) = \sum_{t=k}^{k+N_f-2} \ell^f(\mathbf{z}_t, \mathbf{u}^{\mathbf{z}}_t, \tilde{\mathbf{x}}^{\star}_t, \tilde{\mathbf{u}}^{\star}_t) + \ell^f_T(\mathbf{z}_{k+N_f-1}, \tilde{\mathbf{x}}^{\star}_{k+N_f-1}), \quad (3)$$

under a fast sub-system with shorter time-step Δt_f rather than coarse-time-step Δt_c:

$$\min_{\mathbf{U}^{\mathbf{z}}_k} J_f(\mathbf{z}_k, \mathbf{U}^{\mathbf{z}}_k, \tilde{\mathbf{X}}^{\star}_k, \tilde{\mathbf{U}}^{\star}_k)$$
$$\text{s.t. } \mathbf{z}_{k+1} = \mathbf{z}_k + \Delta t_f \mathbf{f}_f(\mathbf{z}_k, \mathbf{u}^{\mathbf{z}}_k), \quad (4)$$

where the horizon for the fine optimization is defined as N_f. The detailed process to extract a fast sub-system of a humanoid robot is described in Section IV-B. Horizon N_f is set to $\Delta t_f N_f < \Delta t_c N_c$ because the controller for the fast sub-system modifies the robot's short-term effect. Thus, the lower-layer optimization problem can be smaller than the original one in terms of both state dimension ($n_f < n$) and prediction horizon ($N_f < N$).

Using the optimal inputs of the upper-layer for the slow variables $\tilde{\mathbf{u}}^{\mathbf{y}^\star}$ and the lower-layer $\mathbf{u}^{\mathbf{z}^\star}$, the control input is composed of them: $\mathbf{u}^{\star}_{\mathrm{MPC}} = [\tilde{\mathbf{u}}^{\mathbf{y}^\star\top}, \mathbf{u}^{\mathbf{z}^\star\top}]^\top$.

IV. Proposed Control Framework

In our proposed control framework, HMPC derives input commands under full-body dynamics for whole-body motion generation. A reflex-based controller behaves in a finer time scale than that of MPC to maintain the robot's posture. Fig. 2 shows the schematic control diagram of our proposed framework. We proposed an optimization process by deriving fast dynamics of a humanoid robot, which is used in the lower-layer MPC in our hierarchical approach. We, first, describe the full-body dynamics, and then we derive the fast sub-system from the full-body model. The details of the reflex-based controller are also presented in this section.

A. Full-Body Dynamics Model

The full-body dynamics of the robot's movements are generally given by the following equations of motion:

$$\mathbf{M}(\mathbf{q})\dot{\mathbf{v}} = \mathbf{c}(\mathbf{q}, \mathbf{v}) + J_C(\mathbf{q})^\top \mathbf{f}_C + J_E(\mathbf{q})^\top \mathbf{f}_E + \mathbf{u}_{\mathrm{MPC}} + \mathbf{u}_{\mathrm{reflex}}, \quad (5)$$

where $\mathbf{M}$ is the inertia matrix and $\mathbf{c}$ is the vector of such external forces as gravity, coriolis and viscous friction. J_C and J_E stand for Jacobian matrices of active contacts and equality constraints. Contact force is represented as $\mathbf{f}_C$. $\mathbf{f}_E$ is force caused by equality constraints. $\mathbf{q}$ is the generalized positions (base link positions, attitude, and joint angles), $\mathbf{v}$ is their velocities. State of the state-space model consists of the following positions and velocities: $\mathbf{x} = [\mathbf{q}^\top, \mathbf{v}^\top]^\top$. $\mathbf{u}_{\mathrm{MPC}}$ is the applied control to be decided by MPC. $\mathbf{u}_{\mathrm{reflex}}$ is reflex torque which is computed by the reflex-based controller (described in Sec. IV-C). A smooth contact model [6] are used to compute the contact forces $\mathbf{f}_C$ (see [6] for more details).

B. Fast Dynamics Extraction and Hierarchical Optimization

To apply HMPC described in Sec. III-B, we must extract the fast dynamics from the full-body dynamics. Based on the idea of singular perturbation technique for deriving fast dynamics from an original system [26], we proposed a method to extract fast dynamics from a humanoid model. The singularly perturbed system is represented in standard form as bellow [18]:

$$\dot{\mathbf{y}} = \mathbf{g}(\mathbf{y}, \mathbf{z}, \mathbf{u}^{\mathbf{y}})$$
$$\varepsilon\dot{\mathbf{z}} = \mathbf{h}(\mathbf{y}, \mathbf{z}, \mathbf{u}^{\mathbf{z}}), \quad (6)$$

where the dynamics for slow $\mathbf{y}$ and fast state $\mathbf{z}$ are explicitly separated with a small positive parameter ε. In this study, we use this form to extract fast dynamics which can be used with less computational burden in MPC calculation.

To transform the full-body model of (5) into the form of (6), we decomposed the inertia matrix only relative to base link (hip and torso) and the other:

$$\mathbf{M} = \begin{bmatrix} \mathbf{H}_{\mathrm{base}} & O \\ O & O \end{bmatrix} + \begin{bmatrix} O & \mathbf{H}^{\mathrm{base}}_{\mathrm{limb}} \\ \mathbf{H}^{\mathrm{base}\top}_{\mathrm{limb}} & \mathbf{H}_{\mathrm{limb}} \end{bmatrix} = \mathbf{M}_{\mathrm{base}} + \mathbf{M}_{\mathrm{other}}. \quad (7)$$

By considering that inertia $\mathbf{M}_{\mathrm{other}}$ is a perturbation matrix of $\mathbf{M}_{\mathrm{base}}$, the equations of motion in (5) become

$$(\mathbf{M}_{\mathrm{base}} + \varepsilon\mathbf{Q})\dot{\mathbf{v}} = \mathbf{s}, \quad (8)$$

where $\mathbf{Q} = \mathbf{M}_{\mathrm{other}}/\varepsilon$, ε is a positive small parameter defined as $||\mathbf{M}_{\mathrm{other}}||/||\mathbf{M}_{\mathrm{base}}||$ and the vector $\mathbf{s}$ is defined as $(\mathbf{s} = \mathbf{c} + J_C^\top \mathbf{f}_C + J_E^\top \mathbf{f}_E + \mathbf{u}_{\mathrm{MPC}} + \mathbf{u}_{\mathrm{reflex}})$. Let velocities $\mathbf{v} = [\mathbf{v}^\top_{\mathrm{base}}, \mathbf{v}^\top_{\mathrm{limb}}]^\top$ and $\mathbf{s} = [\mathbf{s}^\top_{\mathrm{base}}, \mathbf{s}^\top_{\mathrm{limb}}]^\top$, we obtain the following full-body dynamics written in the form of (6) by multiplying a matrix $(\mathbf{I} - \varepsilon\mathbf{S})$ on both sides of (8) and ignoring a squared term of ε:

$$\begin{bmatrix} \mathbf{H}_{\mathrm{base}} & \mathbf{Q}_{12} \\ O & \mathbf{Q}_{22} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}}_{\mathrm{base}} \\ \varepsilon\dot{\mathbf{v}}_{\mathrm{limb}} \end{bmatrix} = \begin{bmatrix} \mathbf{s}_{\mathrm{base}} \\ \mathbf{s}_{\mathrm{limb}} - \varepsilon\mathbf{Q}_{21}\mathbf{H}^{-1}_{\mathrm{base}}\mathbf{s}_{\mathrm{base}} \end{bmatrix}, \quad (9)$$

where $\mathbf{I}$ is an identity matrix, the matrix $\mathbf{Q}$ is partitioned into four blocks and $\mathbf{S}$ is given as:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q_{11}} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix}, \ \mathbf{S} = \begin{bmatrix} O & O \\ \mathbf{Q}_{21}\mathbf{H}_{\text{base}}^{-1} & O, \end{bmatrix}. \quad (10)$$

The slow variables $\mathbf{y}$ and fast variables $\mathbf{z}$ in (6) are denoted as $[\mathbf{y}^\top, \mathbf{z}^\top]^\top = [\mathbf{v}_{\text{base}}^\top, \mathbf{v}_{\text{limb}}^\top]^\top$.

Now a fast sub-system is extracted from (9). We introduce time-scale $\varepsilon\tau = t$, and the slow and fast variables are

$$\dot{\mathbf{v}}_{\text{base}} = \frac{d\mathbf{v}_{\text{base}}}{dt} = \frac{1}{\varepsilon}\frac{d\mathbf{v}_{\text{base}}}{d\tau}, \ \varepsilon\dot{\mathbf{v}}_{\text{limb}} = \varepsilon\frac{d\mathbf{v}_{\text{limb}}}{dt} = \frac{d\mathbf{v}_{\text{limb}}}{d\tau}. \quad (11)$$

Let $\mathbf{v}'_{\text{base}} = d\mathbf{v}_{\text{base}}/d\tau$ and $\mathbf{v}'_{\text{limb}} = d\mathbf{v}_{\text{limb}}/d\tau$, (9) is rewritten with the new tame-scale:

$$\begin{bmatrix} \mathbf{H}_{\text{base}} & \varepsilon\mathbf{Q}_{12} \\ O & \mathbf{Q}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{v}'_{\text{base}} \\ \mathbf{v}'_{\text{limb}} \end{bmatrix} = \begin{bmatrix} \varepsilon\mathbf{s}_{\text{base}} \\ \mathbf{s}_{\text{limb}} - \varepsilon\mathbf{Q}_{21}\mathbf{H}_{\text{base}}^{-1}\mathbf{s}_{\text{base}} \end{bmatrix}. \quad (12)$$

By setting $\varepsilon \to 0$, we obtain a fast subsystem that is described in new time-scale τ. The fast subsystem takes the following form when we restore to original time-scale t:

$$\begin{bmatrix} \mathbf{H}_{\text{base}} & O \\ O & \mathbf{H}_{\text{limb}} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}}_{\text{base}} \\ \dot{\mathbf{v}}_{\text{limb}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{s}_{\text{limb}} \end{bmatrix}, \quad (13)$$

which describes the dynamics of the fast states for limb movements under fixed slow variables for the base link. Note that the smooth contact model [6] is also considered in the fast subsystem for the lower-layer MPC. The contact force included in $\mathbf{s}_{\text{limb}}$ is computed so that the fast subsystem does not penetrate to the ground surface.

From (13), a discretized system can be obtained:

$$\begin{bmatrix} \mathbf{v}_{\text{base},\mathbf{k+1}} \\ \mathbf{v}_{\text{limb},k+1} \\ \mathbf{q}_{\text{base},k+1} \\ \mathbf{q}_{\text{limb},k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{\text{base},\mathbf{k}} \\ \mathbf{v}_{\text{limb},k} + \Delta t_f \mathbf{H}_{\text{limb}}^{-1}\mathbf{s}_{\text{limb}} \\ \mathbf{q}_{\text{base},k} + \Delta t_f \mathbf{v}_{\text{base},k+1} \\ \mathbf{q}_{\text{limb},k} + \Delta t_f \mathbf{v}_{\text{limb},k+1} \end{bmatrix}, \quad (14)$$

where a semi-implicit Euler method is used for the time discretization. The semi-implicit refers to the fact that the next step velocity is used for the position integration. The time evolution of the states $\mathbf{v}_{\text{base}}$ and $\mathbf{q}_{\text{base}}$ do not depend on the control inputs included in $\mathbf{s}$. Therefore, in the lower-layer optimization of HMPC, only the reduced-order subsystem for $\mathbf{q}_{\text{limb}}$ and $\mathbf{v}_{\text{limb}}$ was considered. We used the subsystem to modify the results of the upper-layer optimization in HMPC. In other words, coarse whole-body motions were derived under the full-body dynamics in the upper-layer, and then the motions of limbs which moves fast were further optimized with a fine-time resolution in the lower-layer optimization. Note that (13) is an approximation that ignores the inertial coupling that limbs have on the base. In practice, limb movements do have coupled effects onto the motion of the base. We include these couplings in our upper-layer optimization by using the full-body dynamics model, but ignore them for our fast model. The effectiveness of this approach is demonstrated in the results.

In our previous work [19], the optimization problem was not small in terms of state dimension n_f. A fast sub-system was derived with eigenvalue analysis of the inertia matrix $\mathbf{M}$ based on [5], whose computational complexity was related to the original state dimension n [27]. In this letter, we avoid to use the eigenvalue algorithm by assuming $\mathbf{M}_{\text{other}}$ as a perturbation matrix of $\mathbf{M}_{\text{base}}$ for efficiently computing the fast sub-system.

C. A Reflex-Based Controller

In this study, a reflex-based controller inspired by biped control with reflexive mechanisms [20], [21] is designed to maintain the robot's posture with much finer time resolution than that of the lower-layer MPC $\Delta t_{\text{reflex}} < \Delta t_f$. Since the behavior of the fine-rate controller is also taken into account during planning with upper- and lower- MPC in both dynamics models (5) and (14) (the reflex torque is included in $\mathbf{s}_{\text{limb}}$), the output torque $\mathbf{u}_{\text{reflex}}$ must be computed with low computational burden. Thus, computational simplicity of the reflex based controller is desirable to combine with MPC.

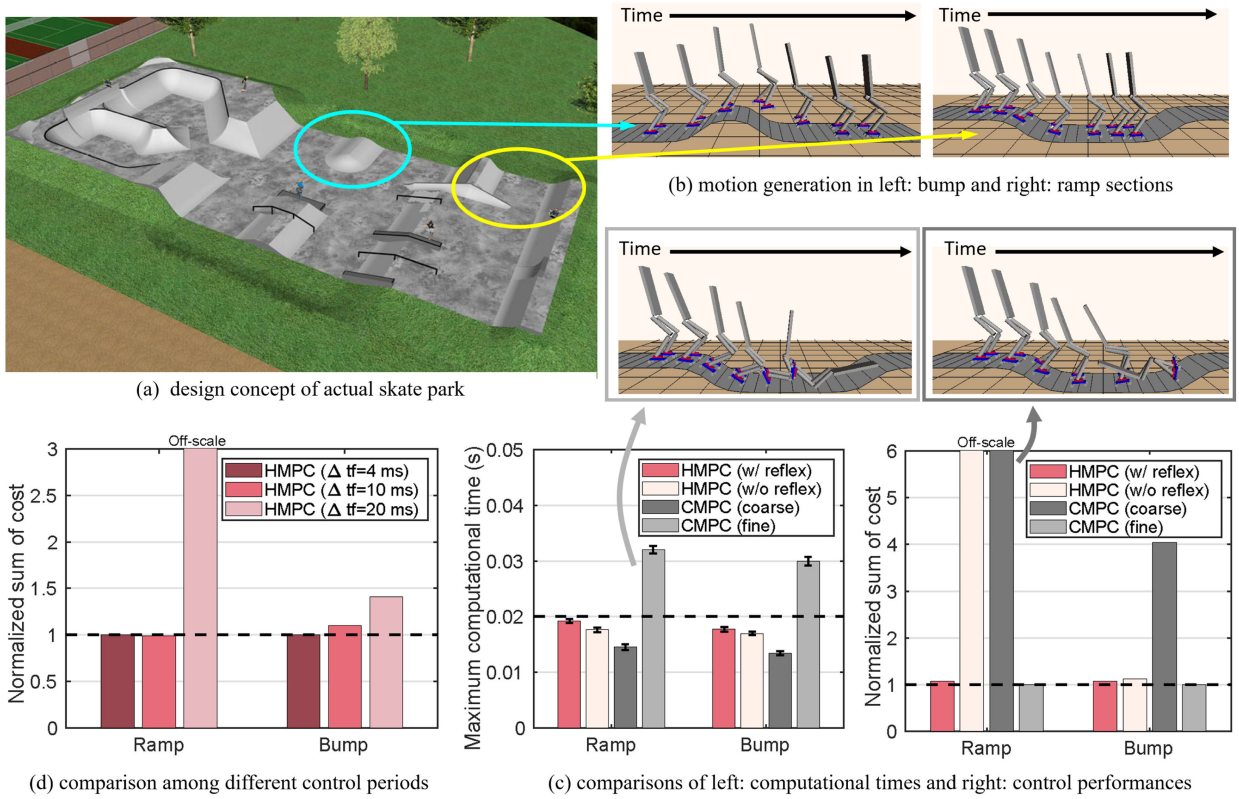
The reflex torque $\mathbf{u}_{\text{reflex}}$ in (5) consists of an output torque τ_{reflex} of the reflex model in each joint, which is determined by the following computation. Let define a sigmoid function $\text{sig}(x, \alpha, \theta) = (1 + e^{\alpha(\theta - x)})^{-1}$, the output of a flexor sensor neuron a_{SF} is modeled as $\text{sig}(q, \alpha_S, \theta_S)$ where α_S is a positive constant. We used $\alpha_S = 5$. For an extensor neuron, the output a_{SE} is computed with the same model except the constant $\alpha_S = -5$. θ_S is the threshold of the neuron. We defined it as the initial joint angle of the robot so that the reflex is activated when the difference between the initial and current joint angles q become large. The output of flexor and extensor motor neurons $r_{F,E}$ are $\text{sig}(W(a_{SE,F} - a_{SF,E}), \alpha_r, \theta_r)$. We set the constant and threshold to $\alpha_r = 5$ and $\theta_r = 5$. W represents the connection strength, where $W = 20$. Finally, the output torque in each joint is determined by $\tau_{\text{reflex}} = (K + D\dot{q})(r_F - r_E)$ where K and D are the output gain. We set them for each joint to $(K^{FE}_{\text{Hip}}, K^{FE}_{\text{Knee}}, K^{FE}_{\text{Ankle}}) = (20.0, 8.0, 20.0)$ and $(D^{FE}_{\text{Hip}}, D^{FE}_{\text{Knee}}, D^{FE}_{\text{Ankle}}) = (10.0, 4.0, 10.0)$. The reflex mechanism for abduction and adduction movements are designed with same sensor and motor neuron models. The gains for each joint are $(K^{AbAd}_{\text{Hip}}, K^{AbAd}_{\text{Ankle}}) = (25.0, 15.0)$ and $(D^{AbAd}_{\text{Hip}}, D^{AbAd}_{\text{Ankle}}) = (25.0, 15.0)$.

V. The Skating Task

In this study, we consider skating tasks on various ground surfaces (see also Fig. 3). We defined the task goal as to maintain the robot's balance, the upper-layer MPC generates whole-body motions which satisfy the goal. Suppose the initial state of the robot remains stable, we specified the task goal with the weighted sum of the squares of cost functions:

$$\ell = w_{\text{CoM}}\ell_{\text{CoM}} + w_{\text{att}}\ell_{\text{att}} + w_{\text{ang}}\ell_{\text{ang}} + w_{\text{input}}\ell_{\text{input}}. \quad (15)$$

The first term ℓ_{CoM} forced the robot to regulate the deviation between CoM and the mean position of ankle links to the initial one. The second term ℓ_{att} was the quadratic penalties on the difference between the current and initial base link attitude. The deviation of the joint angles from initial one is penalized in the third term ℓ_{ang}. The control-cost ℓ_{input} was the quadratic penalties on the control inputs. The weights are set as $(w_{\text{CoM}}, w_{\text{att}}, w_{\text{ang}}, w_{\text{input}}) = (8000.0, 7000.0, 200.0, 0.001)$. We used same costs and weights



(a) design concept of actual skate park

(b) motion generation in left: bump and right: ramp sections

(d) comparison among different control periods

(c) comparisons of left: computational times and right: control performances

Fig. 3. Real-time skating movement generation in simulated environment extracted from (a) actual skate park design [30]: Skating motions are successfully generated with our control framework as depicted in (b) left: bump and right: ramp sections. (c) Comparison of left: maximum computational times and right: control performances of proposed and conventional MPCs. Generated motions with conventional MPCs in ramp section are also depicted. (d) Comparison of control performances in different control periods of fast dynamics.

in both simulation and real robot experiments but we found that sudden huge changes in control inputs induced undesired oscillations on the physical robot. Thus, a term is additionally included in the task goal for the real robot experiments, which penalized the deviation of the inputs from the optimal ones obtained previous time step. Suppose an optimal input sequence was derived at time $k-1$ as $\mathbf{U}^-_{k-1} = \{\mathbf{u}^-_{k-1}, \mathbf{u}^-_k, \cdots\}$, the cost at time k is defined as

$$\ell^{\text{diff}}_k = w^{\text{diff}}_{\text{input}} \|\mathbf{u}_k - \mathbf{u}^-_k\|_2. \tag{16}$$

The weight $w^{\text{diff}}_{\text{input}}$ is set as 0.1.

Let define the derived coarse trajectories for the state $(\tilde{\mathbf{q}}^\star_{\text{base}}, \tilde{\mathbf{q}}^\star_{\text{limb}}, \tilde{\mathbf{v}}^\star_{\text{base}}, \tilde{\mathbf{v}}^\star_{\text{limb}})$ and input $(\tilde{\mathbf{u}}^{\mathbf{y}^\star}, \tilde{\mathbf{u}}^{\mathbf{z}^\star})$, we defined the task goal for the lower-layer optimization as to realize the coarse sequences in fine time-resolution,

$$\begin{aligned}\ell_{\text{low}} = \; & w^{\text{low}}_{\text{pos}} \|\mathbf{q}_{\text{limb}} - \tilde{\mathbf{q}}^\star_{\text{limb}}\|_2 + w^{\text{low}}_{\text{vel}} \|\mathbf{v}_{\text{limb}} - \tilde{\mathbf{v}}^\star_{\text{limb}}\|_2 \\ & + w^{\text{low}}_{\text{input}} \|\mathbf{u}^{\mathbf{z}} - \tilde{\mathbf{u}}^{\mathbf{z}^\star}\|_2.\end{aligned} \tag{17}$$

The weights are $(w^{\text{low}}_{\text{pos}}, w^{\text{low}}_{\text{vel}}, w^{\text{low}}_{\text{input}}) = (10.0, 0.1, 1.0)$.

VI. Simulation and Experimental Setups

A. Simulation Settings

The simulation experiments aim, first, to show how efficient our HMPC (w/ reflex) without significantly degrading the control performance is compared with the conventional MPC (CMPC). We call an MPC that solves the original problem of (2) CMPC (fine), and an MPC solving the same control problem except coarse-time-step CMPC (coarse). The reflex-based controller is also active in both CMPCs and it updates the input once every $\Delta t_{\text{reflex}} = 2$ ms. We, then, perform an ablation study[1] where we compare the results of our HMPC (w/ reflex) with HMPC (w/o reflex) and the reflex-based controller only. We summarized all settings of the methods in Table I. Finally, to evaluate the effect of the lower-layer MPC, we compared our control frameworks with different control periods of the fast dynamics: $\Delta t_f = 4$ and 20 ms ($N_f = 38$ and 8, respectively). We determined the prediction length of about $N\Delta t = 300$ ms so

TABLE I
Settings of Methods: HMPC and CMPC Denote Hierarchical and Conventional MPC, Respectively

Method	Control period [ms]	Horizon
HMPC (w/ reflex)	$\Delta t_c = 20$, $\Delta t_f = 10$	$N_c = 15$, $N_f = 15$
HMPC (w/o reflex)	$\Delta t_c = 20$, $\Delta t_f = 10$	$N_c = 15$, $N_f = 15$
CMPC (coarse)	$\Delta t = 20$	$N = 15$
CMPC (fine)	$\Delta t = 10$	$N = 30$

[1] An ablation study examines which components in a framework contribute to an overall improvement by comparing how lower performances are achieved if each component is removed from the framework.

that the robot can predict enough long-term future state trajectories because a short horizon results in myopic behaviors. For real-time motion generation, an optimization at each time-step must be finished within each control period. We carried out five simulations and averaged the series of maximum computational times.

In this study, we used a 16-DoF lower-body humanoid model with a torso link and five links in each leg. The robot's fast DoF is 10 ($n_f = 20$ dimensions). We were aware that the torque sensors for abduction-adduction joints of the real humanoid robot could not be calibrated well, and servo control for them with MPC outputs was difficult. We, thus, solved optimization problem of the upper-layer MPC under constraints that the velocities for abduction-adduction movements kept zero. The constraint forces for such equality constraints are computed as $\mathbf{f}_E$ in (5). The control inputs for abduction-adduction movements are computed with a high gain PID controller to keep the initial angles.

Based on a design of an actual skate park as shown in Fig. 3(a), we constructed a bump and a ramp section in a simulated environment. Moreover, we designed two cliff environments to generate versatile and agile whole-body motions with our proposed framework. Since we set the time-step of the simulator to 2 ms, modeling error occurs. All experiments were performed in a C language programming environment by an Intel Core i7-4790 CPU, 3.60 GHz computer. Derivatives of dynamics and costs in iLQR were computed in parallel with eight computer threads using Open Multi-Processing (OpenMP) [28].

B. Experimental Settings

In the real robot experiments, we evaluate our control framework: HMPC (w/ reflex) with a real robot on a flat surface and on a slope. The experiments were performed on a humanoid robot CB-i [29] which had linear hydraulic actuators with force feedback on every joint. Each hydraulic actuator was controlled with the PID controller described in [13] to track the optimal state and input derived in HMPC and the reflex torque at 1 kHz. We compared our framework with the setting of HMPC (w/o reflex). We empirically demonstrate that the reflex-based controller improves the task performance on the flat surface by properly maintaining the robot's posture. Furthermore, we also show that our real humanoid robot successfully generates a movement down the slope. We show the average maximum computational times of three experimental trials.

VII. Results

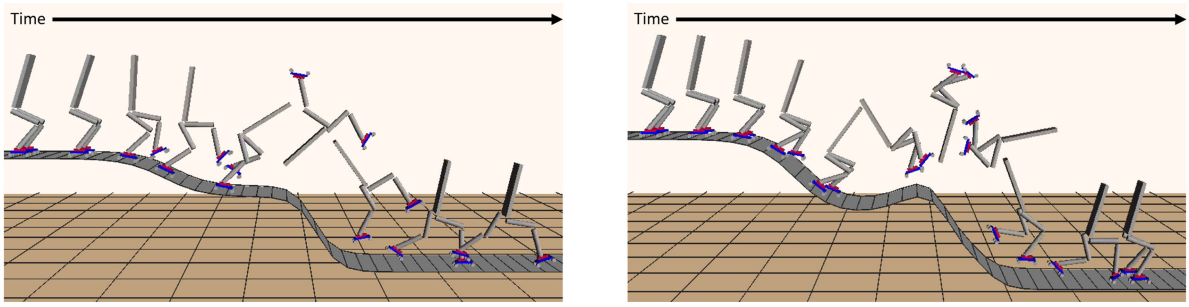
A. Control Performances in the Simulation Environment

We, first, compared our HMPC (w/ reflex) with CMPC (fine) and CMPC (coarse). In the bump section, we gave an initial forward velocity of the robot's base link 6.0 m/s. During going up the bump, both robot's feet were apart from the ground due to the large initial speed. The robot was able to land on the ground, and kept moving to the end of the section as shown in the Fig. 3(b). To skate in the ramp section, we set the initial forward velocity to 1.0 m/s. The robot successfully moves down one side of the ramp without falling over as also shown in Fig. 3(b). Both motions were generated with HMPC (w/ reflex). Fig. 3(c) shows the mean maximum computational time required for computing an optimal control sequence in one time-step. Thanks to our efficient HMPC, no computational times exceeded the control period (dashed line in Fig. 3(c) left). On the other hand, all of the maximum computational times of CMPC (fine) are about three times longer than its control period of 10 ms. If we truncated the computation at the control period for real-time control, few optimizations of CMPC (fine) converged. Fig. 3(c) also shows the accumulated cost during each simulated duration as the control performances of each method. Since the optimizations were not truncated in this case, the total costs of CMPC (fine) became lower. We normalized all the costs with the total costs of CMPC (fine). The control performances of CMPC (coarse) are the worst in all situations. CMPC (coarse) could not sufficiently take changes of states into account. Thus, the robot did not generate stable skating motions with both CMPC (fine) and CMPC (coarse) as shown in Fig. 3(c). By developing our HMPC, motion generation under full-body dynamics were achieved in real-time.

We performed an ablation study to clarify both components of our multi-timescale framework (HMPC and the reflex-based controller) contributed to the overall improvement. It was expressed in Fig. 3(c); the control performances of HMPC (w/o reflex) were inferior to HMPC (w/ reflex). The control period of HMPC is much longer than the required one for robot control (typically, less than 5 ms). On the other hand, we confirmed that the robot fell over in both sections if we used only the reflex-based controller without HMPC. The robot could not ensure long-term balance without HMPC. By taking advantages of HMPC and the fast-rate controller, our multi-timescale framework successfully generated the whole-body motions. It is also worth noting that the computational times between HMPC (w/ and w/o reflex) were slightly increased. Thus, the reflex-based controller is a good candidate to combine with MPC.

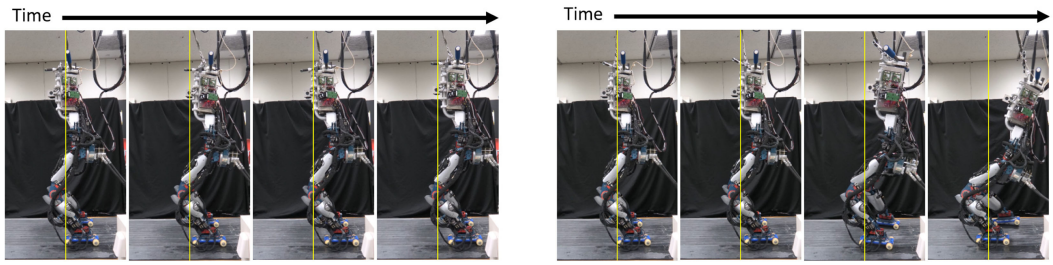
Fig. 3(d) illustrates that the better control performances were achieved if the extracted subsystem was used with the shorter control periods in the lower-layer MPC. HMPC ($\Delta t_f = 10$ ms) corresponds to HMPC (w/ reflex), but we normalized all control performances with the results of HMPC ($\Delta t_f = 4$ ms). We found that if the control period was set as it same with the upper-layer MPC: 20 ms, the lower-layer MPC could not improve the control performance as shown in Fig. 3(d) left. Thus, we consider that our extracted subsystem describes the fast dynamics of the humanoid robot. In our multi-timescale control framework, we used $\Delta t_f = 10$ ms because the maximum computational times of HMPC ($\Delta t_f = 4$ ms) exceeded 20 ms (bump: 23.9 ms and ramp: 25.6 ms) due to the large number of the horizon: $N_f = 38$.

Finally, we confirmed that our multi-timescale control framework could be applied to a more general context of motion generation. Fig. 4 shows the generated motions for a front-flip and back-flip, where the robot motions are constrained in the sagittal plane. It is necessary for the robot to generate fast motions to execute the flip movements. The robot successfully generated both motions with our proposed framework. Since



(a) front flip (b) back flip

Fig. 4. Real-time motion generation with our hierarchical MPC for (a) front and (b) back flip motions: Our hierarchical MPC could be successfully applied to dynamic tasks in which fast movements are necessary.



(a) generated motion on the flat floor (b) generated motion without the reflex-based controller

Fig. 5. Real robot control performances on flat surface: (a) with our proposed framework and (b) without the reflex-based controller (only hierarchical MPC). Yellow vertical lines placed at horizontal location with initial robot's toe positions.

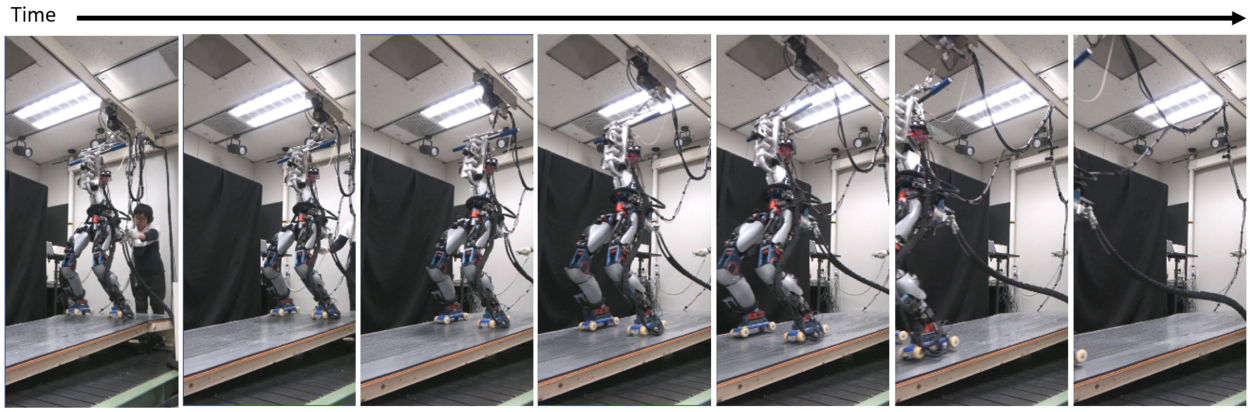


Fig. 6. Generated humanoid movement on slope by using our proposed control framework: To initiate sliding, robot was pushed at beginning of experiment. Our robot successfully kept balance during such disturbance and moved down slope.

the averaged maximum computational times were 19.5 ms for the front-flip and 19.8 ms for the back-flip, real-time motion generation was realized. For this experiments, we added two terms in the task goals described in Sec. V. They forced the robot to follow a time-dependent desired attitude trajectory of the base link and penalized the velocities of the base link to desired values.

B. Generated Real Robot Movements

Fig. 5(a) shows a generated standing motion on the flat floor with our proposed framework. Fig. 5(b) demonstrates motions generated without the reflex-based controller, only HMPC. It becomes difficult to stand stably when a robot wears roller skates because the robot requires a motion which keeps the heavy torso upright but does not slide the wheels. If the legs start sliding, the robot must respond rapidly to change its posture, otherwise it results in a fall. Since the robot can quickly react to maintain its posture due to the reflex-based controller in our proposed framework, the robot did not fall over compared with the case where the reflex-based controller was not activated as shown in Fig. 5(b). The averaged maximum computational time was 22.5 ms, which is slightly over the control period (20 ms). However, since only 1.3% (6.3/500) of the computations exceeded the limit, the overruns only slightly influenced the optimization result.

Fig. 6 represents a series of motions for sliding down a ramp generated with our proposed framework. At the beginning of the

experiment, we pushed the robot to let it start to go down the ramp. During such disturbances, the robot successfully maintained its balance. After initiating the sliding down, the robot did not fall over. In this experiment, the averaged maximum computational time was 21.6 ms, but only 0.7% (3.3/500) of the computations exceeded the control period. Note that the robot easily falls over if we only use the reflex-based controller without HMPC. The reflex-based controller is useful only for keeping a posture in short time scale not for maintaining balance for long period. The above results demonstrate the our control framework can be used in the real system.

VIII. Conclusions

Although humanoid robots are expected to work in various situations instead of humans, their versatility and agility are significantly low. They are crucial issues to be solved before humanoid robots can engage in various tasks in the real environment. To cope with this problem, we developed a multi-timescale control framework using optimal control under full-body dynamics of a humanoid robot in this study.

Since one of the biggest challenges to realize such full-body optimal control is its computational burden, we proposed an efficient MPC controller with the hierarchical architecture for the humanoid model in this letter. Moreover, a biologically-inspired reflex controller is required to control the robot in short timescale. We demonstrated that variety of agile skating motions can be generated on different ground surfaces. Furthermore, the real robot successfully moved down the slope with our proposed control framework.

To extract low-dimensional dynamics for fast computation of optimal control, centroidal dynamics could be used [31]. This approach yet considers simpler dynamics than our proposed method while taking full kinematics into account. Therefore, our hierarchical dynamics representation would be closer to full dynamics but requires heavier computation in optimization process than centroidal dynamics. As a future study, we will further investigate design of the fast-rate controller. For example, a local feedback controller using the gains from iLQR would provide a unified control strategy with MPC. Future work will apply our control framework to generate wider variety of motion in the physical robot.

References

[1] E. Krotkov *et al.*, "The Darpa robotics challenge finals: Results and perspectives," *J. Field Robot.*, vol. 34, no. 2, pp. 229–240, 2017.

[2] S. Kuindersma *et al.*, "Optimization-based locomotion planning, estimation, and control design for the Atlas humanoid robot," *Auton. Robot.*, vol. 40, no. 3, pp. 429–455, 2016.

[3] S. Feng, X. Xinjilefu, C. G. Atkeson, and J. Kim, "Optimization based controller design and implementation for the atlas robot in the Darpa robotics challenge finals," in *Proc. IEEE-RAS 15th Int. Conf. Humanoid Robots.*, 2015, pp. 1028–1035.

[4] T. Erez, K. Lowrey, Y. Tassa, V. Kumar, S. Kolev, and E. Todorov, "An integrated system for real-time model predictive control of humanoid robots," in *Proc. IEEE-RAS 13th Int. Conf. Humanoid Robots.*, 2013, pp. 292–299.

[5] F. Miyazaki and S. Arimoto, "A control theoretic study on dynamical biped locomotion," *J. Dyn. Syst., Meas., Control*, vol. 102, no. 4, pp. 233–239, 1980.

[6] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 4906–4913.

[7] J. Koenemann *et al.*, "Whole-body model-predictive control applied to the HRP-2 humanoid," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 3346–3351.

[8] S.-H. Jo, J.-U. Chu, and Y.-J. Lee, "Motion planning for biped robot with quad roller skates," in *Proc. IEEE Int. Conf. Control, Autom. Syst.*, 2008, pp. 1173–1177.

[9] N. Takasugi *et al.*, "Extended three-dimensional walking and skating motion generation for multiple noncoplanar contacts with anisotropic friction: Application to walk and skateboard and roller skate," *IEEE Robot. Autom. Lett.*, vol. 4, no. 1, pp. 9–16, Jan. 2018.

[10] H. Seguchi and T. Ohtsuka, "Nonlinear receding horizon control of an underactuated hovercraft," *Int. J. Robust Nonlinear Control*, vol. 13, no. 3/4, pp. 381–398, 2003.

[11] M. Neunert *et al.*, "Fast nonlinear model predictive control for unified trajectory optimization and tracking," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 1398–1404.

[12] S. Kajita *et al.*, "Biped walking pattern generation by using preview control of zero-moment point," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, vol. 2. pp. 1620–1626.

[13] B. J. Stephens and C. G. Atkeson, "Push recovery by stepping for humanoid robots with force controlled joints," in *Proc. IEEE-RAS 10th Int. Conf. Humanoid Robots*, 2010, pp. 52–59.

[14] R. Budhiraja, J. Carpentier, C. Mastalli, and N. Mansard, "Differential dynamic programming for multi-phase rigid contact dynamics," in *Proc. IEEE-RAS 18th Int. Conf. Humanoid Robots*, 2018, pp. 1–9.

[15] F. Farshidian, E. Jelavic, A. Satapathy, M. Giftthaler, and J. Buchli, "Real-time motion planning of legged robots: A model predictive control approach," in *Proc. IEEE-RAS 17th Int. Conf. Humanoid Robot.*, 2017, pp. 577–584.

[16] F. Farshidian, M. Neunert, A. W. Winkler, G. Rey, and J. Buchli, "An efficient optimal planning and control framework for quadrupedal locomotion," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 93–100.

[17] R. Scattolini, "Architectures for distributed and hierarchical model predictive control—A review," *J. Process Control*, vol. 19, no. 5, pp. 723–731, 2009.

[18] X. Chen, M. Heidarinejad, J. Liu, and P. D. Christofides, "Composite fast-slow MPC design for nonlinear singularly perturbed systems," *AIChE J.*, vol. 58, no. 6, pp. 1802–1811, 2012.

[19] K. Ishihara and J. Morimoto, "Real-time model predictive control with two-step optimization based on singularly perturbed system," in *Proc. IEEE-RAS 15th Int. Conf. Humanoid Robots*, 2015, pp. 173–180.

[20] P. Manoonpong, T. Geng, T. Kulvicius, B. Porr, and F. Wörgötter, "Adaptive, fast walking in a biped robot under neuronal control and learning," *PLoS Comput. Biol.*, vol. 3, no. 7, 2007, Art. no. e134.

[21] C. Goldbeck, L. Kaul, N. Vahrenkamp, F. Worgotter, T. Asfour, and J.-M. Braun, "Two ways of walking: Contrasting a reflexive neuro-controller and a LIP-based ZMP-controller on the humanoid robot ARMAR-4," in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots*, 2016, pp. 966–972.

[22] D. H. Jacobson, "New second-order and first-order algorithms for determining optimal control: A differential dynamic programming approach," *J. Optim. Theory Appl.*, vol. 2, no. 6, pp. 411–440, 1968.

[23] J. Morimoto and C. G. Atkeson, "Minimax differential dynamic programming: An application to robust biped walking," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 1563–1570.

[24] E. Todorov and W. Li, "A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *Proc. IEEE Amer. Control Conf.*, 2005, pp. 300–306.

[25] T. Erez, "Optimal control for autonomous motor behavior," Ph.D. dissertation, Washington Univ., St. Louis, MO, USA, 2011.

[26] P. Kokotovic, H. K. Khalil, and J. O'reilly, *Singular Perturbation Methods in Control: Analysis and Design*. Philadelphia, PA, USA: SIAM, 1999, vol. 25.

[27] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Numerical recipes in C," Cambridge, U.K., Cambridge Univ. Press, vol. 1, p. 3, 1988.

[28] OpenMP ARB, "Openmp," 1997. [Online]. Avaialble: http://www.openmp.org/

[29] G. Cheng *et al.*, "CB: A humanoid research platform for exploring neuroscience," *Adv. Robot.*, vol. 21, no. 10, pp. 1097–1114, 2007.

[30] Dreamland Skateparks, LLC, "Design + concept," 1990. [Online]. Avaialble: https://www.dreamlandskateparks.com/design-concept/

[31] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *Proc. IEEE-RAS 14th Int. Conf. Humanoid Robots*, 2014, pp. 295–302.