Key Pose-Based Dynamic Humanoid Motion Generation Using Parallel Multi-Fidelity Model Predictive Control

Yuichi Tazaki[1]

***Abstract*—This paper proposes a method that realizes dynamic motion of humanoid robots from reference key poses. The proposed control method runs two types of model predictive controllers with different fidelity and time scale in parallel; one performs long-horizon prediction by making use of a closed-form solution of the centroidal dynamics, and the other performs short-horizon prediction based on the whole-body dynamics. In dynamical simulation of 32-DoF humanoid robot, the controller was able to perform challenging motions including toe contact and jumps over unlevel surfaces in real time computation speed without any offline optimization.**

***Index Terms*—Humanoid Robots, Trajectory Optimization, Model Predictive Control**

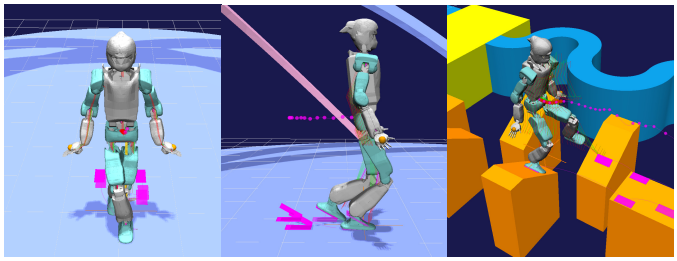

Fig. 1. Screenshot images of generated motion. From left to right : Box Step, Moonwalk, and Stepping Stones.

I. Introduction

Humanoid robots can perform a variety of dynamic motions by utilizing their large degrees of freedom. However, their high dimensional nature and inherently unstable floating-base dynamics, coupled with the discontinuity of contact makes motion generation and tracking control extremely challenging. To tackle high complexity, hierarchical strategy consisting of reference motion generation, motion retargeting, and tracking control has been a widely accepted approach.

Reference motion can be acquired by a variety of means such as motion capture, key poses, and generative models. In any case, raw reference motion is often incompatible with the kinematics and/or dynamics of the robot. Therefore, motion retargeting (a.k.a. dynamics filter) is required to convert reference motion to physically plausible motion that can be executed by a robot. Existing model-based dynamics retargeting techniques [1, 2] are based on classical walking pattern generation of biped locomotion, and are limited to simple motions in which at least one foot makes flat contact with the floor. Trajectory optimization based on whole-body models can handle wider variety of motions. Because of its high computation cost, however, optimization is often performed offline.

In order to realize motion on a real humanoid robot, one also needs to construct a feedback control loop for tracking retargeted (dynamically filtered) motion data under model uncertainty and disturbances. Trajectory tracking control is necessary even in simulations because there are differences in models and algorithms used inside dynamical simulation and those used for trajectory generation. Whole-body model predictive control has attracted strong research interest because it can handle the dynamics and kinematic limits of the robot as well as contact-related constraints without simplification. Some recent studies reported its hardware implementation to humanoid robots [3][4]. Numerical solution methods of MPC has polynomial complexity in terms of the dimension of the state and control input. For this reason, application of MPC to humanoid robots poses greater challenge than to other legged robots such as quadrupeds because the former generally has much greater degrees-of-freedom than the latter. For speeding up MPC, parallelization based on bi-level optimization [5, 6] and subdividision of the prediction horizon [7] have been studied. However, decomposing the overall problem generally deteriorates convergence rate and therefore requires more iterations. Another approach is to vary the fidelity of prediction models over the prediction horizon. In [8, 9], a whole-body model with high time resolution was used in a leading part of the prediction horizon while a reduced-order model with coarse stepping was used in the later part. This method has so far been applied to quadrupeds only, and extension to various humanoid motions is an unexplored topic.

Recently, deep reinforcement learning techniques have shown impressive performance in both motion retargeting [10] and tracking [11]. However, this approach generally requires massive computation resource for offline training. Moreover, it requires uniform and finely sampled reference motion, and application to non-uniform and coarse key poses has not been investigated so far.

Manuscript received: May 15, 2025; Revised: September 9, 2025; Accepted September 30, 2025.

This paper was recommended for publication by Editor Abderrahmane Kheddar upon evaluation of the Associate Editor and Reviewers ' comments. This work was supported by JSPS KAKENHI Grant Number 24K03017.

[1]Yuichi Tazaki is with Graduate School of Engineering, Department of Mechanical Engineering, Kobe University tazaki@mech.kobe-u.ac.jp

Digital Object Identifier (DOI): see top of this page.

This paper proposes a trajectory generation and tracking technique for dynamic motion specified by a series of key poses. It consists of two model predictive controllers that run at different model fidelity and time scale: Centroidal (CD-)MPC and Whole-body (WB-)MPC. A main difference from [9] is that we use non-uniform time stepping in CD-MPC, whereas in [9], uniform low-resolution time stepping was used for SRB (single rigid body) MPC. Our formulation based on a close-form solution of the centroidal dynamics can accept nonuniform step durations, and therefore much more suited to key frame-based motion specification. Another benefit of our method is that the update rate of CD-MPC and WB-MPC can also be set differently. There is also a difference in the formulation of WB-MPC; while the joint coordinate rigid body dynamics was used in [9], the centroidal dynamics + full kinematics formulation is used in this study. The developed technique is tested with a number of challenging key pose sequences that include jumping and toe contact. In dynamical simulation, a closed-loop system consisting of the proposed controller and a 32-DoF humanoid robot model was constructed, and dynamic motion was successfully realized as shown in screenshot images (Fig. 1). Base on an earlier work reported in [12], this paper makes a number of significant extensions as summarized below.

- A new contact description which enables compact and uniform description of flat and non-flat contacts on unlevel surfaces is proposed.
- The value function computed in CD-MPC is used to construct a terminal cost of WB-MPC to effectively extend its prediction horizon.
- Quantitative evaluation of disturbance rejection performance is conducted. Moreover, the closed-loop performance under different settings of terminal cost and prediction horizon length is compared.

The remaining part of this paper is organized as follows. In Section II, the proposed trajectory generation and tracking method is described in detail. Followed by an overview of the control system architecture (II-A), the mathematical description of CD-MPC (II-C) and WB-MPC (II-D) are given. Furthermore, parallel execution of multiple MPC threads (II-E) and terminal cost design (II-F) are discussed. In Section III, simulation results are shown. Concluding remarks are given in Section IV.

II. Control System Design

A. Control System Architecture

The overall block diagram and information flow of the control system is shown in Fig. 2. The control system consists of a hierarchy of components which run at different frequencies: CD-MPC, WB-MPC, and linear feedback. Both trajectory optimization problems are computed by the iterative linear quadratic regulator (iLQR) algorithm [13]. It is a variant of differential dynamic programming (DDP) and shows super-linear convergence to a local optimal solution thanks to Gauss-Newton approximation [14]. The CD-MPC thread receives a reference trajectory computed by interpolation of key poses as inputs and outputs optimized centroidal states together with

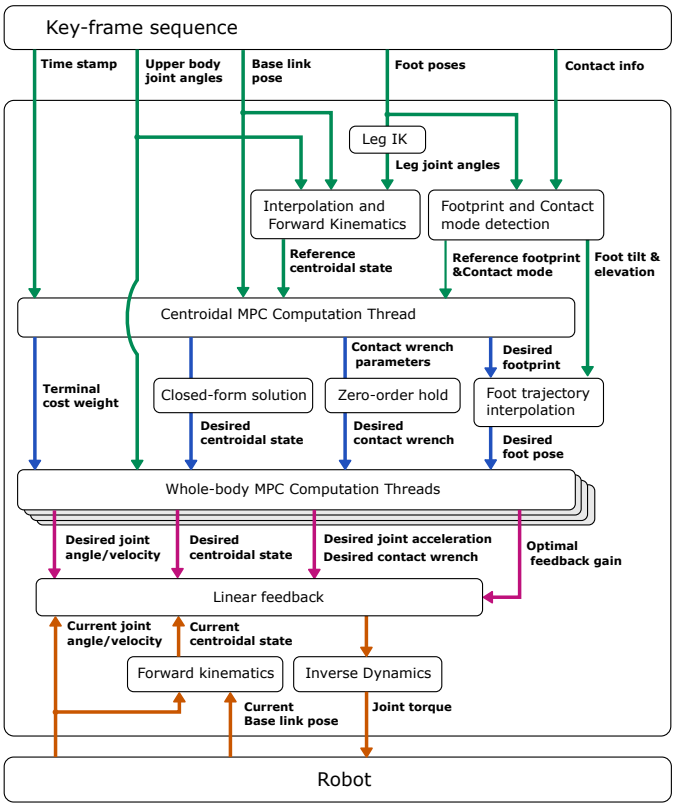

Fig. 2. Control system architecture

modified contact locations and timing. The update cycle of CD-MPC is synchronized with the irregular spacing of key frames, which normally ranges between 0.05 and 0.3 seconds. The WB-MPC thread group receives the output of CD-MPC and computes optimized whole-body motion at a medium frequency of 250Hz. At the lowest level, linear feedback control with the optimal feedback gain runs at a high frequency of 1kHz. Desired joint acceleration and contact wrench output are converted to desired joint torque by means of inverse dynamics. Full state of the robot is retrieved from the simulator and fed back to WB-MPC and CD-MPC. In [12], CD-MPC was used without state feedback, meaning that it worked as a pure dynamics filter. In this work, state feedback of CD-MPC is enabled to realize footstep adjustment in response to disturbances, as demonstrated in Section III.

B. Contact Description and Relationship with Key Poses

Reference motion is input to the controller as a series of key poses. Each key frame consists of a time stamp, a base link pose, upper body joint angles, and left and right foot poses. It also includes for each foot a boolean value indicating whether the foot is in contact or not, and a unit vector indicating the normal direction of the associated contact surface. In addition, each foot of the robot has a rectangular contact geometry whose edges are parallel to the x and y axes of the local coordinate frame of the foot.

In order to handle different contact modes on non-level surfaces in a unified and compact way, each foot pose is

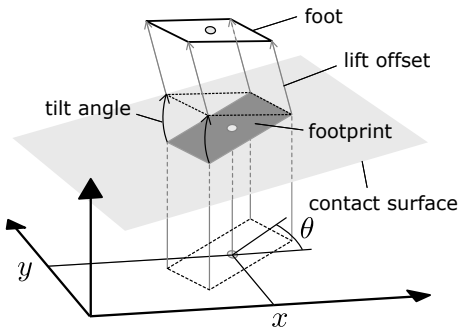

Fig. 3. Relationship between contact plane, footprint, and foot pose

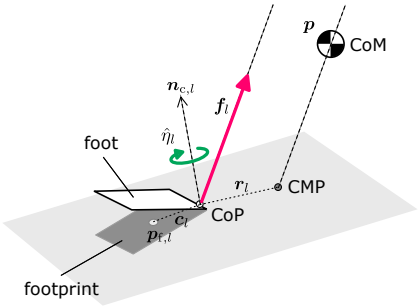

Fig. 4. Stiffness-based parameterization of contact wrench

expressed in terms of footprint pose, tilt angle, and lift offset, as illustrated in Fig. 3. Similar idea has been used for footstep planning on uneven terrain [15]; we extend this idea to express poses in line contact and floating state. Given a foot pose and a contact normal, the corresponding footprint pose, tilt angle, and lift offset can be calculated. For a foot not in contact, the z direction of the global cooridnate frame is used as the normal direction.

Contact states between key frames are determined by the following rules, where possible types of contact state are *float*, *line contact*, and *surface contact*.

- If the foot is off-contact in either key frame, then the contact state is *float*.
- If the foot is in contact with the same contact surface in both key frames, and:
  - if the tilt angle is zero in both key frames, then the contact state is *surface contact*.
  - if the tilt angle is zero in one key frame and non-zero in the other, or if the tilt angle is non-zero in both key frame with a common contact edge, then the contact mode is *line contact*.

If neither of the above conditions hold, then the input key frames are considered as invalid.

The overall processing flow of key frames is the following. Input key frames initially do not have velocity information. So in the first step, reference velocity is computed from finite difference of adjacent key poses. Next, for each key frame, footprint, tilt angle, and lift offset and their time derivatives are calculated. Moreover, the contact states between each adjacent key frames are indentified based on the rules described above, and reference centroidal states (CoM position, velocity, and angular momentum) are calculated based on the kinematics and mass distribution of the robot model. Note that only 2D parameters of footprints (i.e., x, y, and yaw angle) are passed to the CD-MPC while other values (tilt angle and lift offset) are retained. This is because the 2D information of footprints is sufficient for expressing the contact complementarity condition regardless of whether the contact mode is surface contact or line contact. It also contributes to reducing the state and input dimensions of the CD-MPC and hence its computation cost. Continuous-time foot movement is obtained by interpolation and given to the WB-MPC as references. Upon interpolation, the footprint pose, tilt angle, and lift offset and their time derivatives are transformed back to the pose and velocity of the foot, and conventional cubic interpolation is applied to the position and Euler angles.

C. Centroidal MPC

The centroidal dynamics equation [16] is expressed as follows.

$$\ddot{\boldsymbol{p}} = \frac{1}{m}\sum_l \boldsymbol{f}_l, \quad \dot{\boldsymbol{L}} = \sum_l \left[\boldsymbol{\eta}_l + (\boldsymbol{p}_{\mathrm{f},l} - \boldsymbol{p}) \times \boldsymbol{f}_l\right] \tag{1}$$

Here, m is the total mass, $\boldsymbol{p}$ is the center-of-mass (CoM), and $\boldsymbol{L}$ is the total angular momentum with respect to the CoM. Moreover, $\boldsymbol{p}_{\mathrm{f},l}$ is the position of the l-th foot and $\boldsymbol{f}_l$ and $\boldsymbol{\eta}_l$ are the linear and rotational components of the contact wrench applied to it. Further more, the relationship between the total angular momentum and the whole-body configuration of the robot is expressed as follows.

$$\boldsymbol{L} = I(\boldsymbol{\theta})\boldsymbol{\omega} + \boldsymbol{q}\hat{\boldsymbol{L}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}). \tag{2}$$

Here, $\boldsymbol{q}$ is a unit quaternion expressing the rotation of the base link and $\boldsymbol{\omega}$ is the angular velocity of the base link. Moreover I is the composite rigid-body inertia with respect to the CoM, which is dependent on the joint angle vector $\boldsymbol{\theta}$. Furthermore, $\hat{\boldsymbol{L}}$ is the *local* angular momentum of the system; namely, it expresses the total angular momentum expressed of the movement of the links expressed in the local coordinate frame of the base link.

The stiffness-based parametrization [17] is used, in which contact wrench is expressed by a tuple $\lambda \geq 0$, $\hat{\boldsymbol{c}} \in \mathbb{R}^2$, $\hat{\boldsymbol{r}} \in \mathbb{R}^2$, $\hat{\eta} \in \mathbb{R}$. Here, λ is the stiffness and it expresses the strength of the contact force. Moreover, $\hat{\boldsymbol{c}}$ is the offset from the center of the footprint to the center-of-pressure (CoP) and $\hat{\boldsymbol{r}}$ is the offset from the CoP to the centroidal moment pivot (CMP), both consisting of x and y components in the contact coordinate frame. These points determine the point of application and the direction of the contact force as illusrated in Fig. 4. Furthermore, $\hat{\eta}$ is the normalized torsional moment. It is *normalized* in the sense that multiplying it with $m\lambda_l^2$ gives moment with respect to the z-axis of the contact frame. With these parameters, the contact wrench is transcribed as follows.

$$\begin{aligned} \boldsymbol{f}_l &= m\lambda_l^2 \left(\boldsymbol{p} - (\boldsymbol{p}_{\mathrm{f},l} + \boldsymbol{c}_l + \boldsymbol{r}_l)\right) \\ \boldsymbol{\eta}_l &= \boldsymbol{c}_l \times \boldsymbol{f}_l + m\lambda_l^2 (\boldsymbol{n}_{\mathrm{c},l}\, \hat{\eta}_l) \end{aligned} \tag{3}$$

Here, $\boldsymbol{c}_l = \boldsymbol{q}_{\mathrm{c},l} \begin{bmatrix} \hat{\boldsymbol{c}}_l \\ 0 \end{bmatrix}$ and $\boldsymbol{r}_l = \boldsymbol{q}_{\mathrm{c},l} \begin{bmatrix} \hat{\boldsymbol{r}}_l \\ 0 \end{bmatrix}$ are 3D offsets of the CoP and the CMP. Moreover, $\boldsymbol{q}_{\mathrm{c},l}$ and $\boldsymbol{n}_{\mathrm{c},l}$ denote the orientation and the normal direction of the contact surface associated with the l-th foot, respectively. By substituting (3) into (1), we obtain the following equations.

$$\ddot{\boldsymbol{p}} = \bar{\lambda}^2(\boldsymbol{p} - (\bar{\boldsymbol{p}} + \bar{\boldsymbol{r}})), \quad \dot{\boldsymbol{L}} = m(\ddot{\boldsymbol{p}} \times \bar{\boldsymbol{r}} + \bar{\boldsymbol{\eta}}) \tag{4}$$

where

$$\begin{aligned} &\bar{\lambda} = \sqrt{\textstyle\sum_l \lambda_l^2 + \epsilon^2}, \ \ \bar{\boldsymbol{p}} = \frac{\sum_l \lambda_l^2(\boldsymbol{p}_{\mathrm{f},l} + \boldsymbol{c}_l) + \boldsymbol{g}}{\bar{\lambda}^2}, \ \ \bar{\boldsymbol{r}} = \frac{\sum_l \lambda_l^2 \boldsymbol{r}_l}{\bar{\lambda}^2}, \\ &\bar{\boldsymbol{\eta}} = \bar{\lambda}^2 \bar{\boldsymbol{p}} \times \bar{\boldsymbol{r}} + \textstyle\sum_l \lambda_l^2(\boldsymbol{n}_{\mathrm{c},l}\, \hat{\eta}_l - (\boldsymbol{p}_{\mathrm{f},l} + \boldsymbol{c}_l) \times \boldsymbol{r}_l). \end{aligned} \tag{5}$$

Here, $\epsilon \geq 0$ is a small constant inserted to avoid dividing by zero in the case of a flight-phase (i.e., $\lambda_l = 0 \quad \forall l$). If flight phases need not be considered, then ϵ can be set as 0 and (4) holds precisely. Otherwise the equalities hold approximately. Nevertheless, practically ϵ can be set very small (e.g., 10^{-6}) and error results from it is negligible. By applying the zero-order hold to the wrench parameters, the centroidal dynamics can be analytically integrated over a time interval between each consecutive key frames to obtain a discrete-time prediction model for $\boldsymbol{p}$, $\boldsymbol{v}$, and $\boldsymbol{L}$ (See [17] for detailed derivation). Here, the composite inertia I and the local angular momentum $\hat{\boldsymbol{L}}$ are calculated from interpolated key-pose sequence and provided to CD-MPC as references. The state and control input of CD-MPC are defined as follows.

$$\begin{aligned} \boldsymbol{x}_k^{\mathrm{cd}} &= \left[\boldsymbol{p}_k, \boldsymbol{v}_k, \boldsymbol{q}_k, \boldsymbol{L}_k, t_k, \{x_{l,k}, y_{l,k}, \theta_{l,k}\}\right], \\ \boldsymbol{u}_k^{\mathrm{cd}} &= \left[\tau_k, \{\dot{x}_{l,k}, \dot{y}_{l,k}, \dot{\theta}_{l,k}, \lambda_{l,k}, \hat{\boldsymbol{c}}_{l,k}, \hat{\boldsymbol{r}}_{l,k}, \hat{\eta}_{l,k}\}\right] \end{aligned} \tag{6}$$

Note that each time step corresponds to a single time interval between consecutive key frames. The tuple $\{x_{l,k}, y_{l,k}, \theta_{l,k}\}$ denotes the 2D information (x and y coordinates and yaw angle) of the l-th footprint in the k-th key frame. The variable t_k denotes the time instant of the beginning of the k-th key frame.

The cost function is defined as a weighted sum of quadratic error terms:

$$\begin{aligned} J^{\mathrm{cd}} &= \|\boldsymbol{x}_{N^{\mathrm{cd}}}^{\mathrm{cd}} - \boldsymbol{x}_{N^{\mathrm{cd}}}^{\mathrm{cd,ref}}\|_{W^{\mathrm{cd,x}}}^2 \\ &+ \sum_{k=0}^{N^{\mathrm{cd}}-1} \left[\|\boldsymbol{x}_k^{\mathrm{cd}} - \boldsymbol{x}_k^{\mathrm{cd,ref}}\|_{W^{\mathrm{cd,x}}}^2 + \|\boldsymbol{u}_k^{\mathrm{cd}} - \boldsymbol{u}_k^{\mathrm{cd,ref}}\|_{W^{\mathrm{cd,u}}}^2\right] \end{aligned} \tag{7}$$

where N^{cd} is the number of prediction steps (counted by number of key frames), $\boldsymbol{x}_k^{\mathrm{cd,ref}}$ and $\boldsymbol{u}_k^{\mathrm{cd,ref}}$ are desired values, and $W^{\mathrm{cd,x}}$ and $W^{\mathrm{cd,u}}$ are diagonal weight matrices. The desired values of the centroidal state and foot poses are calculated from key poses. The desired value of λ_l is dependent on the contact state of the l-th contact during the k-th frame. If it is in contact, the desired stiffness is set as a value needed for statically supporting the weight of the robot, and otherwise it is set as zero. The desired value of other wrench parameters are always set as zero. Contact complementarity constraint can be expressed simply as follows.

$$\begin{aligned} &\lambda_{l,k} \geq 0, \ \dot{x}_{l,k} = \dot{y}_{l,k} = \dot{\theta}_{l,k} = 0 \quad \text{if } l\text{-th foot is in contact} \\ &\lambda_{l,k} = 0 \qquad\qquad\qquad\qquad\quad \text{otherwise} \end{aligned} \tag{8}$$

Note that this condition restricts the velocity of the footprint but not that of the foot itself. In the case of toe contact, for example, the foot can tilt with respect to the contact edge while the footprint remains fixed. Moreover, box constraints are imposed on the CoP offset $\hat{\boldsymbol{c}}_l$ and the torsional moment $\hat{\eta}_l$. Their bounds must be set according to the contact mode. Furthermore, the static friction constraint is expressed as follows (the subscript k is omitted).

$$\begin{aligned} &\|\boldsymbol{n}_{\mathrm{c},l} \times (\boldsymbol{n}_{\mathrm{c},l} \times (\boldsymbol{p} - (\boldsymbol{p}_{\mathrm{f},l} + \boldsymbol{c}_l + \boldsymbol{r}_l)))\| \\ &\quad \leq \mu\, \boldsymbol{n}_{\mathrm{c},l}^{\mathsf{T}}(\boldsymbol{p} - (\boldsymbol{p}_{\mathrm{f},l} + \boldsymbol{c}_l + \boldsymbol{r}_l)) \end{aligned} \tag{9}$$

where μ is the static friction coefficient. Thanks to the conic nature of friction constraint, the stiffness λ_l can be omitted from the inequality. These inequality constraints are transformed into logarithmic barrier functions and integrated into the cost function.

One limitation of CD-MPC is that it cannot take kinematic limits of legs into account. In this study, key poses were designed carefully so that the foot poses will stay inside the movable range even after optimization of CD-MPC.

D. Whole-body MPC

Whole-body MPC is based on the centroidal dynamics + full kinematics formulation proposed in [18] with minor modifications. The state and control input of WB-MPC are defined as follows.

$$\begin{aligned} \boldsymbol{x}_\kappa^{\mathrm{wb}} &= \left[\boldsymbol{p}_\kappa, \boldsymbol{v}_\kappa, \boldsymbol{q}_\kappa, \boldsymbol{L}_\kappa, \boldsymbol{\theta}_\kappa, \dot{\boldsymbol{\theta}}_\kappa\right], \\ \boldsymbol{u}_\kappa^{\mathrm{wb}} &= \left[\ddot{\boldsymbol{\theta}}_\kappa, \{\boldsymbol{f}_{l,\kappa}, \boldsymbol{\eta}_{l,\kappa}\}\right] \end{aligned} \tag{10}$$

One main difference from [18] is that the state variable includes the position and velocity of the CoM instead of those of the base link. This formulation makes WB-MPC more compatible with CD-MPC and enables coordination between them in terms of reference values and terminal cost with less effort of variable transformation. It also requires little complication of internal computation because base link position can be computed easily from the CoM position and joint angles by means of forward kinematics. A discrete-time prediction model for WB-MPC is derived by applying the Euler method with a fixed time step Δt_{wb}. The angular velocity $\boldsymbol{\omega}$ needed for updating $\boldsymbol{q}$ is given by $\boldsymbol{\omega} = I^{-1}(\boldsymbol{L} - q\hat{\boldsymbol{L}})$.

The cost function of WB-MPC is defined as follows:

$$\begin{aligned} J^{\mathrm{wb}} &= V(t_{\mathrm{f}}, f_{\mathrm{wb2cd}}(\boldsymbol{x}_{N^{\mathrm{wb}}}^{\mathrm{wb}})) \\ &\sum_{\kappa=0}^{N^{\mathrm{wb}}-1} \left[\|\boldsymbol{x}_\kappa^{\mathrm{wb}} - \boldsymbol{x}_\kappa^{\mathrm{wb,ref}}\|_{W^{\mathrm{wb,x}}}^2 + \|\boldsymbol{u}_\kappa^{\mathrm{wb}} - \boldsymbol{u}_\kappa^{\mathrm{wb,ref}}\|_{W^{\mathrm{wb,u}}}^2\right] \end{aligned} \tag{11}$$

where $V(\cdot)$ is the terminal cost (see II-F for details), N^{wb} is the number of prediction steps, $\boldsymbol{x}_k^{\mathrm{wb,ref}}$ and $\boldsymbol{u}_k^{\mathrm{wb,ref}}$ are desired values, and $W^{\mathrm{wb,x}}$ and $W^{\mathrm{wb,u}}$ are diagonal weight matrices. The desired values of the centroidal states are obtained by substituting the output of CD-MPC into the closed-form solution of the centroidal dynamics. The desired values of the foot poses and velocities are obtained by interpolating the foot poses output by CD-MPC. Desired contact wrenches are calculated by substituting the output of CD-MPC to

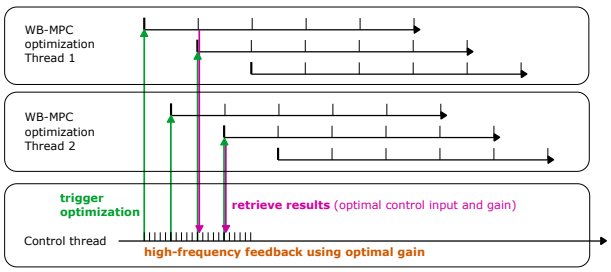

Fig. 5. Illustration of a group of multiple optimization threads. In this figure, $\Delta t_{\mathrm{delay}} = 10\Delta t_{\mathrm{ctrl}}$ and $n_{\mathrm{p}} = 2$ which means $\Delta t_{\mathrm{update}} = \Delta t_{\mathrm{delay}}/2 = 5\Delta t_{\mathrm{ctrl}}$.

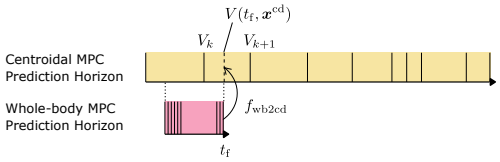

Fig. 6. Illustration of terminal cost setting.

(3). Desired joint angle and velocity are directly taken from interpolated key poses, and desired joint acceleration is set as zero. Inequality constraints expressing the friction and moment limit are imposed on contact wrenches. Moreover, range limits on joint angles and velocities are imposed. In particular, joint limits on the knee joints prevent them from fully stretching with a small margin to avoid numerical instability caused by kinematic singularity.

E. Parallel Execution of Whole-body MPC Threads

Because of high dimensionality of state and control input, each optimization cycle of WB-MPC takes several milliseconds even if warm start is used. This severely limits the update frequency of WB-MPC. To improve the update rate, multiple computation threads are executed and optimization is triggered in shifted timing. As illustrated in Fig. 5, n_{p} optimization threads are executed in parallel. Consider that the optimization computation of the i-th thread is triggered at time t. This optimization will be completed by $t + \Delta t_{\mathrm{delay}}$. For simplicity we assume that $\Delta t_{\mathrm{delay}}$ is equal to Δt_{wb}, the time step of the prediction model of WB-MPC, although this assumption is generally not necessary. The $(i+1)$-th thread will be triggered at $t+\Delta t_{\mathrm{update}}$, where $\Delta t_{\mathrm{update}} = \Delta t_{\mathrm{delay}}/n_{\mathrm{p}}$. In this manner, the update period is shortened to $\Delta t_{\mathrm{update}}$ while the latency is still determined by $\Delta t_{\mathrm{delay}}$. Because of the limitation of the number of available CPU cores, it is still difficult to reduce the update period down to the control period Δt_{ctrl}. We therefore combine this technique with linear feedback control with the optimal feedback gain produced by the DDP algorithm [19]. Linear feedback is performed at every control cycle, while the feedback gain is updated every $\Delta t_{\mathrm{update}}/\Delta t_{\mathrm{ctrl}}$ cycles.

F. Terminal Cost Design

Is has been well known in the literature that careful terminal cost design is crucial for ensuring the closed-loop stability of MPC. One popular method is to use the value function of an unconstrained LQR problem as the terminal cost [20]. In the case of dynamic whole-body motion, however, it is difficult to require that the robot is always in a steady state at the end of the prediction horizon. Instead, we propose to use the value function of CD-MPC as the terminal cost of WB-MPC. We consider that it has essentially the same effect as the method proposed in [9], where the fidelity of the model is switched in a single prediction horizon.

Consider that the terminal time t_{f} of the prediction horizon of WB-MPC is included in the k-th prediction time interval of CD-MPC; that is, $t_k \leq t_{\mathrm{f}} \leq t_{k+1}$ holds. As a result of DDP computation, the value function of CD-MPC at t_k and t_{k+1}, V_k and V_{k+1}, respectively, are available. Based on linear interpolation, the value function at t_{f} together with its gradient and Hessian is approximately given by

$$\begin{aligned} V(t_{\mathrm{f}}, \boldsymbol{x}^{\mathrm{cd}}) &= (1-\alpha)V_k(\boldsymbol{x}^{\mathrm{cd}}) + \alpha V_{k+1}(\boldsymbol{x}^{\mathrm{cd}}), \\ V_x &= (1-\alpha)V_{x,k} + \alpha V_{x,k+1}, \\ V_{xx} &= (1-\alpha)V_{xx,k} + \alpha V_{xx,k+1} \end{aligned} \tag{12}$$

where $\alpha = \frac{t_{\mathrm{f}} - t_k}{t_{k+1} - t_k}$. The terminal cost for WB-MPC is given by $V(t_{\mathrm{f}}, f_{\mathrm{wb2cd}}(\boldsymbol{x}^{\mathrm{wb}}))$, where f_{wb2cd} is a mapping from $\boldsymbol{x}^{\mathrm{wb}}$ to $\boldsymbol{x}^{\mathrm{cd}}$. More precisely, the centroidal states in $\boldsymbol{x}^{\mathrm{wb}}$ are directly mapped to those in $\boldsymbol{x}^{\mathrm{cd}}$. The joint angle and velocity variables in $\boldsymbol{x}^{\mathrm{wb}}$ can be mapped to the foot poses and velocities in $\boldsymbol{x}^{\mathrm{cd}}$ by forward kinematics. The time variable in $\boldsymbol{x}^{\mathrm{cd}}$, for which no corresponding variable exists in $\boldsymbol{x}^{\mathrm{wb}}$, is simply set as zero. In actual DDP computation of WB-MPC, the gradient and Hessian of the terminal cost are needed. They are given by $J_{\mathrm{wb2cd}}^{\mathsf{T}} V_x$ and $J_{\mathrm{wb2cd}}^{\mathsf{T}} V_{xx} J_{\mathrm{wb2cd}}$, where J_{wb2cd} is the Jacobian matrix of f_{wb2cd}.

G. Parameter Setting

Typical parameter setting of MPC is summarized in Tables I(a)-(c). Decision variables are normalized based on their physical dimensions so that their values range within the same order of magnitude. The weights shown in Tables I(b)(c) are applied to normalized variables. In this study, good combination of weights was searched by manual tuning. Starting from a typical setting shown in the tables, additional adjustment could be made according to the needs. For example, lowering the weights of time, duration, and foot position of the CD-MPC results in more aggressive step timing and position adaptation. Moreover, lowering the weights of the upper body joints in WB-MPC encourages use of upper body movement for balance compensation. The weights on leg joint angles and velocities are set as zero to avoid interference with desired foot pose and velocity.

III. Simulation Study

A. Setup

A model of RHP Friends (Kawasaki Heavy Industry, [21, 22]) was used for both MPC and dynamical simulation. The model has 32 DoF (4 DoF in torso and neck, 8 DoF in each arm, 6 DoF in each leg). The kinematic and inertial parameters were matched to the real robot. Note that geometries

TABLE I
TYPICAL PARAMETER AND WEIGHT SETTING

	CD-MPC	WB-MPC
prediction steps	30	10
time step	1 key frame	20ms
update period	1 key frame	4ms
number of iterations	10	2
number of threads	1	5
state dimension	19	76
input dimension	19	50

(a) MPC parameters

CoM position	1
CoM velocity	1
Base rotation	1
Angular momentum	1
Time	10
Foot position	1
Foot rotation	1
Duration	10
Foot velocity	1
Foot angular velocity	1
Foot stiffness	10
Foot CoP offset	10
Foot CMP offset	10
Foot torsional moment	10

(b) CD-MPC weights

CoM position	5
CoM velocity	5
Base rotation	5
Angular momentum	5
Foot position	5
Foot rotation	5
Foot velocity	5
Foot angular velocity	5
Foot force	200
Foot moment	200
Leg joint angle	0
Leg joint velocity	0
Other joint angle	1
Other joint velocity	1
Joint acceleration	1

(c) WB-MPC weights

resembling the real robot are used for visualization only. For dynamical simulation, a box geometry with dimensions of $[-0.1, +0.15] \times [-0.05, +0.05] \times [0.0, 0.03]$ was attached to each foot to compute contact with the ground, while self-collision was ignored.

In the following subsections, simulation results of three example motions, Box Step, Moonwalk, and Stepping Stones, are presented. Key frames and contact phases of these motions are shown in Fig. 7. Choreonoid [23] was used for both dance motion editing and dynamical simulation [1]. Body Motion Controller (a built-in dynamics filter of Choreonoid) was not used and raw key frame sequence was directly input to the controller. The AIST simulator item was used for physics simulation. Joint torque commands computed by the controller are directly input to the simulator. To obtain high accuracy results, the time step of physics simulation was set as 0.1ms, whereas the control cycle was set as 1ms. In this setup, the computation cost of simulation itself was too high to run in real time. With 1ms time step, simulation including MPC computation could run in real time on a laptop computer and motions still could be performed successfully. The trajectory optimization algorithm for centroidal and whole-body dynamics as well as the model predictive controller with interfaces to Choreonoid were implemented in C++ [2].

[1] Minor modification was applied to handle unlevel contact plane in key-pose editing.

[2]The code is publicly available on github: https://github.com/ytazz/dymp_mpc.

B. Results

Screenshots of dynamical simulation are shown in Fig. 1. See the attached video for better visualization of simulation.

Box Step: One cycle of box step consists of four foot steps placed in a square pattern. In the second step of each cycle, the left and the right legs are crossed. The edited key frames consist of two cycles of box-step.

First, robustness of the proposed controller against disturbances was evaluated. At time 2.5, an impulsive disturbance in the forward direction was applied to the base link of the robot. The magnitude of disturbances is expressed in equivalent DCM shift. Figure 8 shows the response of the robot to various magnitude of disturbance. Two cases with different degrees of feedback were compared: full state feedback in which all states were directly obtained from the simulator, and partial state feedback in which only base link rotation, joint angles, and joint velocities were provided and CoM position and velocity were estimated based on the assumption that the support foot is in the desired position. For both cases, the robot could withstand up to 0.2m DCM shift although the latter case was more perturbed and needed longer time to settle. For small disturbances, the robot mainly used its upper body movement to recover balance. For greater disturbances, step adaptation was also used.

Next, different settings of prediction horizon length N of WB-MPC and presence of centroidal terminal cost were compared. The result is summarized in Fig. 9. The plot shows errors between the CoM position and the angular velocity of the reference trajectory generated by the CD-MPC and those computed in dynamical simulation. In this experiment, CD-MPC simply worked as a reference trajectory generator with state feedback turned off. Therefore, the plot shows the tracking performance of WB-MPC in different configurations. The height of each bar indicates the average over time while the error bar indicates the maximum error. When N is large enough (e.g., 20), there is virtually no difference between w/ and w/o terminal cost. Without terminal cost, error grows gradually as N is decreased. When N was smaller than 8, the robot fell down in the middle of motion. With terminal cost, on the other hand, errors do not increase notably until N decreases to 8. Moreover, the robot could complete the motion without falling down even when N was as small as 4. This indicates that terminal cost is in fact useful for realizing stable close-loop performance with small N. However, with small N the movement of the robot became jerky as indicated by large maximum error in the plot. It is concluded that setting N to 8 to 10 with terminal cost strikes a good balance between tracking performance and computation cost. As a side note, it was possible to realize Box Stepping with WB-MPC alone. However, it required $N \geq 50$ to realize stable motion.

Moonwalk: In moonwalk, the swing foot is slid backward while the support foot makes toe contact with the ground. In the later part of each swing phase, the heel of the swing foot is lifted up while the heel of the support foot is lowered. The overall movement generates an illusion that the robot glides backward even though it is making forward-walking motion. Real sliding movement involving dynamical friction force cannot be expressed in our centroidal dynamics model.

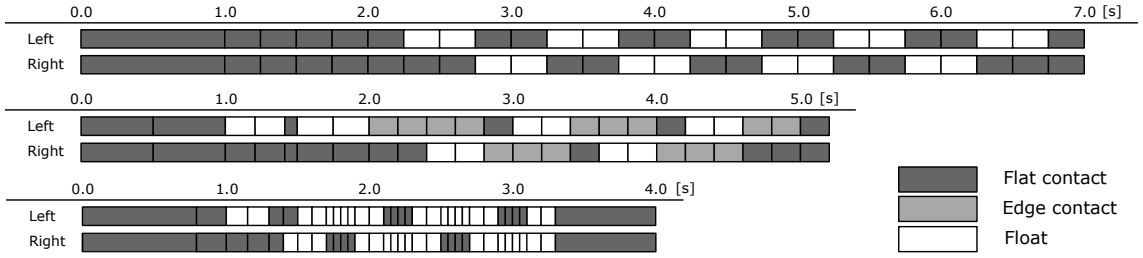

Fig. 7. Key frames and contact phases. From top to bottom: BoxStep, MoonWalk, and SteppingStone. Screenshots of selected key frames are shown. Contact modes are indicated by color: dark gray (flat contact), light gray (toe contact), and white (float).

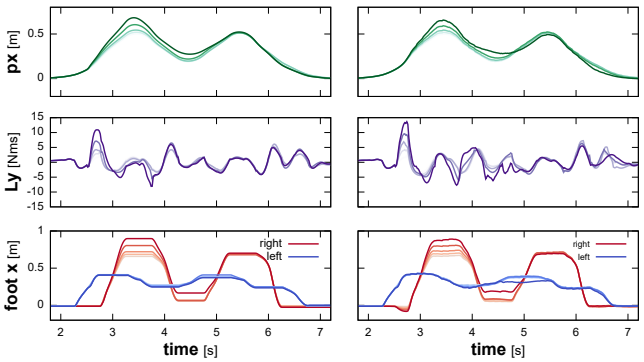

Fig. 8. Response to impulsive disturbance. Full state feedback (left) and partial state feedback + state estimation (right). From top to bottom, CoM position in x direction, angular momentum in y (pitch) direction, and foot positions in x direction. The disturbance is applied at time 2.5s as depicted by a dashed line. Magnitude of the disturbance ranges between 0.0 and 0.2 DCM shift[m]. Lines with darker colors depict responses to stronger disturbances.

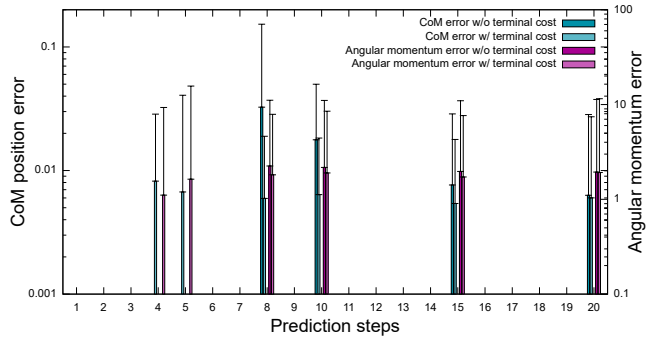

Fig. 9. Comparison of closed loop tracking performance.

Instead, sliding-like movement was realized by lifting the swing foot very slightly from the ground. The static friction coefficient was set as 1.0. We can observe in the plot of my in Fig. 10 that ground reaction moment needed for maintaining toe-contact of the support foot is generated by CD-MPC and tracked well by WB-MPC.

Stepping Stones: In this example, the robot hops over non-level stepping stones. As shown in the screenshot image, stepping stones are placed in the left and the right alternately and they are sloped 30 degrees towards the center line. The friction coefficient was set as 2.0. In each single-support-phase (SSP) during hopping, the direction and magnitude of CoM acceleration needs to be carefully controlled while maintaining

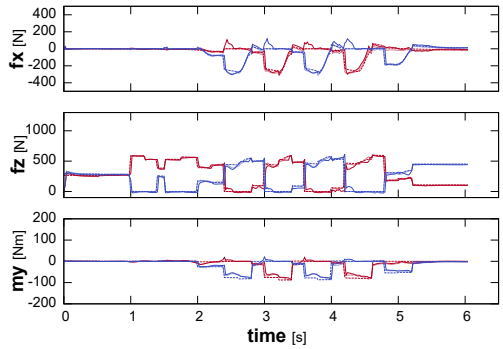

Fig. 10. Computed trajectories of Moonwalk. Lines indicate the output of CD-MPC (dashed), WB-MPC (dotted), and simulation response (solid). From top to bottom: contact force (x and z components, right foot (red), left foot(blue)) and contact moment (y component, right foot (red), left foot(blue)).

TABLE II
COMPUTATION TIME

	ave [ms]		max [ms]	
	PC1	PC2	PC1	PC2
CD-MPC (10 iterations)	20.692	19.790	22.989	22.907
WB-MPC (2 iterations)	8.241	8.581	9.437	13.249
WB-MPC (3 iterations)	13.647	13.057	15.398	15.456
WB-MPC (4 iterations)	19.717	16.090	22.050	17.892

the variation of angular momentum to minimum. To provide sufficient degrees of freedom of contact wrench to the planner, each single-support-phase (SSP) with a period of 0.2s was equally subdivided into a number of sub-phases. In Fig. 11, simulation results of 2 and 6 sub-phases per SSP are shown. With greater number of subdivisions, the vertical movement of the CoM as well as the peak magnitude of contact force could be reduced. This result indicates that high enough time resolution is needed for some critical time intervals to ensure good quality of generated trajectories. It was also found important to impose torsional moment limit to avoid twisting of the foot and maintain stable contact. Thanks to whole-body MPC, upper body movement that reduces torsional moment was automatically generated.

C. Computation Time

The average and maximum computation time of MPC threads measured on two different processors (PC1(Intel Core i7-1280P), PC2(AMD Ryzen 9 5950X)) are summarized in

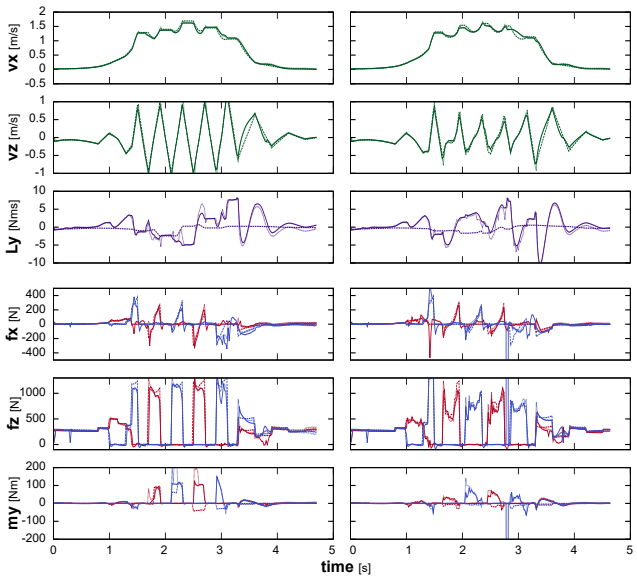

Fig. 11. Computed trajectories of Stepping Stones. Lines indicate the output of CD-MPC (dashed), WB-MPC (dotted), and simulation response (solid). From top to bottom: CoM velocity (x and z components), angular momentum (y component), contact force (x and z components, right foot (red), left foot(blue)), and contact moment (y component, right foot (red), left foot(blue)).

Table II. The computation time of CD-MPC was negligible considering its low update rate (one update per key frame). The computation time of WB-MPC could be made smaller than the specified optimization period (20ms) by limiting the number of iterations to 3.

IV. Conclusion

This paper proposed a parallel multi-fidelity model predictive control framework that can generate dynamic whole-body motion of humanoid robots from a series of key poses. Combining the proposed framework with a contact planner may be an effective solution for reducing the manual effort of key pose editing. In the current design, CD-MPC uses moment of inertia and local angular momentum calculated from key poses. This point may require improvement in order to apply the method to motions involving large rotation such as parcour, somersault, and cart-wheel. Moreover, quantitative comparison with other multi-fidelity MPC and learning-based approaches should be conducted. Lastly, major technical challenges toward sim2real would be to integrate whole-body state estimation into the control system loop and to respect joint velocity and torque limits.

References

[1] S. Nakaoka, S. Kajita, and K. Yokoi. Intuitive and flexible user interface for creating whole body motions of biped humanoid robots. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1675–1682, 2010.

[2] G. Bin Hammam, D. E. Orin, and B. Dariush. Whole-body humanoid control from upper-body task specifications. In *2010 IEEE International Conference on Robotics and Automation*, pages 3398–3405, 2010.

[3] E. Dantec, R. Budhiraja, A. Roig, et al. Whole body model predictive control with a memory of motion: experiments on a torque-controlled talos. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8202–8208, 2021.

[4] C. Khazoom, S. Hong, M. Chignoli, et al. Tailoring solution accuracy for fast whole-body model predictive control of legged robots. *IEEE Robotics and Automation Letters*, 9(12):11074–11081, 2024.

[5] M. Kögel and R. Findeisen. Parallel solution of model predictive control using the alternating direction multiplier method. *IFAC Proceedings Volumes*, 45(17):369–374, 2012. 4th IFAC Conference on Nonlinear Model Predictive Control.

[6] L. Amatucci, G. Turrisi, A. Bratta, et al. Accelerating model predictive control for legged robots through distributed optimization. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12734–12741, 2024.

[7] F. Farshidian, E. Jelavic, A. Satapathy, et al. Real-time motion planning of legged robots: a model predictive control approach. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 577–584, 2017.

[8] H. Li, R. J. Frei, and P. M. Wensing. Model hierarchy predictive control of robotic systems. *IEEE Robotics and Automation Letters*, 6(2):3373–3380, 2021.

[9] H. Li and P. M. Wensing. Cafe-Mpc: a cascaded-fidelity model predictive control framework with tuning-free whole-body control. *IEEE Transactions on Robotics*, 41:837–856, 2025.

[10] Y. Yan, E. V. Mascaro, and D. Lee. Imitationnet: unsupervised human-to-robot motion retargeting via shared latent space. In *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, pages 1–8, 2023.

[11] Z. Luo, J. Cao, W. Alexander, et al. Perpetual humanoid control for real-time simulated avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10895–10904, Oct. 2023.

[12] Y. Tazaki. Humanoid dance simulation using hybrid model predictive control. In *2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids)*, pages 1004–1010, 2024.

[13] W. Li and E. Todorov. Iterative linear quadratic regulator design for nonlinear biological movements systems. In *ICINCO2004*, 2004.

[14] C. Mastalli, W. Merkt, J. Marti-Saumell, et al. A feasibility-driven approach to control-limited DDP. *Autonomous Robots*, 46(8):985–1005, 2022.

[15] R. J. Griffin, G. Wiedebach, S. McCrory, et al. Footstep planning for autonomous walking over rough terrain. In *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, pages 9–16, 2019.

[16] D. E. Orin, A. Goswami, and S. H. Lee. Centroidal dynamics of a humanoid robot. *Autonomous Robots*, 35(2-3):161–176, 2013.

[17] Y. Tazaki. Trajectory generation for legged robots based on a closed-form solution of centroidal dynamics. *IEEE Robotics and Automation Letters*, 9(11):9239–9246, 2024.

[18] J. Sleiman, F. Farshidian, M. Minniti, et al. A unified mpc framework for whole-body dynamic locomotion and manipulation. *IEEE Robotics and Automation Letters*, 6(3):4688–4695, 2021.

[19] R. Grandia, F. Farshidian, R. Ranftl, et al. Feedback mpc for torque-controlled legged robots. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4730–4737, 2019.

[20] P. Scokaert and J. Rawlings. Constrained linear quadratic regulation. *IEEE Transactions on Automatic Control*, 43(8):1163–1169, 1998.

[21] Y. Kakiuchi, M. Kamon, N. Shimomura, et al. Development of life-sized humanoid robot platform with robustness for falling down, long time working and error occurrence. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 689–696, 2017.

[22] M. Benallegue, G. Lorthioir, A. Dallard, et al. Humanoid robot rhp friends: seamless combination of autonomous and teleoperated tasks in a nursing context. *IEEE Robotics & Automation Magazine*, 32(1):79–90, 2025.

[23] S. Nakaoka. Choreonoid: extensible virtual robot environment built on an integrated gui framework. In *2012 IEEE/SICE International Symposium on System Integration (SII)*, pages 79–85, 2012.