2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Foot Stepping Algorithm of Humanoids with Double Support Time Adjustment based on Capture Point Control*

Myeong-Ju Kim[1], Daegyu Lim[1], Gyeongjae Park[1] and Jaeheung Park[1,2]

Abstract— **Recently, foot stepping strategies of humanoid robots have been actively developed for robust balancing of humanoids against disturbances. In this paper, a novel stepping algorithm adjusting double support phase (DSP) time is proposed. First, the stepping algorithm is proposed based on a model predictive control (MPC) framework for capture point (CP) control and footstep adjustment. Next, when the remaining step time is not enough to adjust the footstep, the DSP scaling method brings the next swing phase forward by reducing the DSP time, which enables the robot to maintain the balance robustly. The robust balance control performance of the proposed method is validated through simulations and experiments when the robot is walking in the presence of external pushes. A more stable balancing performance is realized compared to state-of-the-art stepping controllers.**

I. Introduction

Humanoids have the potential that they can be used directly in human-centered workplaces without environmental modifications. One of the most necessary technical requirements for humanoids to be practically utilized is a robust balance control performance against external perturbations.

At the beginning of humanoid walking research, a simplified model, linear inverted pendulum model (LIPM), was introduced [1], and walking pattern generation methods using the LIPM have been proposed [2], [3]. However, when disturbances are applied, it is difficult to maintain the balance of the robot only with the pre-designed walking pattern. One of the representative studies to address this problem is the zero moment point (ZMP) feedback control, which controls the ZMP to be inside the convex hull of the supporting foot preventing the robot from falling [4]–[9].

As another approach to overcome disturbance during walking, capture point (CP) feedback control methods have been proposed [10]–[16]. The CP has a characteristic that the position of the CoM converges to the CP [17], [18]. Based on this characteristic, CP is used as a control variable to stabilize the CoM of the robot. However, the balance control performance based on pre-designed footsteps, and walking patterns is limited by the contact force control performance.

Recently, many foot stepping strategies that adjust the step location, and step timing rather than using pre-determined footsteps have been proposed to overcome strong disturbances. Wieber et al. [19] proposed a model predictive control (MPC) scheme that generates a stable CoM trajectory changing the reference ZMP tracking problem in [5] to the CoM jerk minimization problem with ZMP inequality constraints, which enables smooth CoM, ZMP trajectory even when the predesigned footprint is suddenly changed. Expanding the work of [19], Herdt et al. [20] proposed a step location adjustment algorithm to track the reference CoM velocity and ZMP. In [21], the desired ZMP is controlled using the CoM motion, and when the ZMP error is large, the step location is adjusted to cope with the large disturbance. Studies using nonlinear MPC were also conducted. Romualdi et al. [22] proposed a walking algorithm that adjusts the contact wrenches and the contact location based on the centroidal dynamics and nonlinear MPC framework to respond to disturbances.

In addition to the MPC framework which considers every future state of the robot and future constraints, the foot stepping algorithm which considers only the next footstep states by propagating the current state of the robot to the end of the current swing phase based on LIPM has been proposed. Khadiv et al. [23], [24] proposed an online QP optimization method that adjusts the location and the timing of the next step to control the predicted CP at the end of the step. Jeong et al. [25] controlled the CP end-of-step by including the step location and timing as optimization variables as well as the ankle torque, and hip angle in the QP problem.

The previous stepping algorithms showed robust balance control performance in the presence of large disturbances by the adaptive step adjustment. However, these studies did not validate the balance performance when the remaining time of the single support phase (SSP) is not sufficient to adjust step location and timing. Also, the adjustment of the double support phase (DSP) time, which occupies a fixed time duration in the walking phase, was not considered before.

Therefore, a robust balancing control framework based on the CP-MPC and stepping controller is proposed in this work. The main contributions of this study are as follows: i) An MPC framework is proposed that introduces additional MPC variables, footstep displacement, to adjust the step location by expanding the previous CP controller based on the MPC framework [11]. ii) The DSP scaler is introduced to improve balancing performance for disturbances that occur when the remaining time of SSP is not enough to adjust step location and timing. To the best of our knowledge, this is the first study dealing with stepping strategies by adjusting the DSP time when the remaining step time is not sufficient for step adjustment. iii) The proposed method is evaluated through extensive simulations and real robot experiments using our humanoid robot, TOCABI, in the presence of disturbances.

*This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1A2C3005914).

[1]All authors with the Department of Intelligence and Information, Seoul National University, Republic of Korea. Contact: park73@snu.ac.kr

[2]Jaeheung Park is also with the Advanced Institutes of Convergence Technology, Republic of Korea and with ASRI, RICS, Seoul National University, Republic of Korea. He is the corresponding author of this paper.

II. Fundamentals

A. Linear Inverted Pendulum Model

The LIPM was proposed as a linearized abstract model to simplify the complex dynamics of humanoids. In LIPM, the total mass of the robot is concentrated in the CoM, and the centroidal angular momentum around CoM is not considered. The vertical motion of the CoM is constrained to be constant. The relationship between CoM and ZMP is provided,

$$\ddot{c}_x = \omega^2(c_x - p_x), \tag{1}$$

where c_x and p_x denote the positions of CoM and ZMP in the x-direction, respectively. $\omega = \sqrt{g/c_z}$ is the natural frequency, g is the gravitational acceleration, and c_z is the CoM height from the ground. In the LIPM dynamics, the x- and y-direction can be dealt with independently. Therefore, in this study, the derivation of equations will be treated only in the x-direction. Note that positive x-axis is forward direction, and positive y-axis is left direction for the robot, respectively.

B. Capture Point dynamics based on the LIPM

The CP was introduced in [17], [18], and the CP–ZMP dynamics based on LIPM is defined as below,

$$\dot{\xi}_x = \omega(\xi_x - p_x), \tag{2}$$

where ξ_x denotes the CP in the x-direction. Here, if ZMP, p_x, is assumed to be constant in each walking step, the step duration is T, and the time elapsed after the start of the swing phase is t, the behavior of CP [10], [25] can be defined as follows.

$$\xi_{x,T} = (\xi_x - p_x)e^{\omega(T-t)} + p_x. \tag{3}$$

The dynamics of (2) is used to derive the MPC framework in Section IV, and (3) is used for the stepping controller in Section V.

III. Walking Control framework

In this section, the proposed walking control framework is explained. Fig. 1 represents a schematic diagram of the overall framework. The goal of the framework is to control the CP, $\boldsymbol{\xi}$, to overcome disturbances. First, in order to track the reference CP trajectories, $\boldsymbol{\Xi}^{ref} \in \mathbb{R}^{N\times 2}$, obtained from the walking pattern generator, the CP-MPC controller calculates two MPC variables; $\mathbf{P} = [\mathbf{P}_x \mathbf{P}_y] \in \mathbb{R}^{N\times 2}$ for ZMP control and $\Delta\mathbf{F} = [\Delta\mathbf{F}_x \Delta\mathbf{F}_y]$ for step location adjustment. Note that N represents the time horizon for CP-MPC controller. The ZMP controller calculates the position and orientation, $\Delta\mathbf{e}_{L,R}$, of both feet based on the admittance control [9] to achieve the first desired ZMP, $\mathbf{P}_0$. In the stepping controller, the step location, $\mathbf{F}^{des}$, and the step time including SSP and DSP, $\{T^{opt}_{ssp}, T^{opt}_{dsp}\}$, are optimized using the three components; pre-planned footstep, $\mathbf{F}^{ref}$, MPC variables, $\Delta\mathbf{F}$ and CP, $\boldsymbol{\xi}$. The optimized step time, $\{T^{opt}_{ssp}, T^{opt}_{dsp}\}$, is used for the walking pattern generator and foot trajectory generator in real-time. The optimized step location, $\mathbf{F}^{des}$, is used to generate the foot trajectory. The footstep planner outputs the foothold locally based on the target velocity of the robot. Lastly, the desired joint angles, $\mathbf{q}^{des}$, are obtained through the inverse kinematics to realize the desired CoM position, $\mathbf{c}^{des}$, and the desired position, $\{\mathbf{e}^{des}_{L,P}, \mathbf{e}^{des}_{R,P}\}$, and orientation, $\{\mathbf{e}^{des}_{L,O}, \mathbf{e}^{des}_{R,O}\}$, of both feet, $\mathbf{e}^{des}_{L,R}$. The desired joint angles are controlled using the joint PD control with gravity torque compensation.

IV. Linear Model Predictive Control for Capture Point Tracking

The linear MPC framework for CP trajectory tracking is developed based on the two previous MPC frameworks [11], [21]. According to [11], the CP-ZMP dynamics in (2) is used as a prediction model of MPC. Equation (2) can be discretized with piecewise constant ZMP, $p_{x,k}$, and sampling time, T_s, as bellow.

$$\xi_{x,k+1} = A\xi_{x,k} + Bp_{x,k}, \tag{4}$$

where $A = e^{\omega T_s}$ and $B = 1 - e^{\omega T_s}$. Also, in [11], the cost function is formulated as

$$l_k(\xi_{x,k}, p_{x,k}) = w_{\xi,k}(\xi_{x,k} - \xi^{ref}_{x,k})^2 + w_{p,k}(p_{x,k} - p_{x,k-1})^2 \tag{5}$$

with a ZMP constraint, $\underline{b}_{p,k} \le p_{x,k} \le \bar{b}_{p,k}$, to ensure that the ZMP is in the support polygon. $w_{\xi,k}$, and $w_{p,k}$ are positive weighting gains for each cost term at each time step k. However, this formulation predicts future CP trajectory based on the pre-defined footsteps, and the ZMP control input is restricted by the fixed footsteps and hardware limitations such as foot size and joint torque limits. Then, the constrained ZMP input makes the robot vulnerable to large disturbance.

Therefore, the additional step adjustment variable, $\Delta\mathbf{F}_x$, is added to the MPC of [11] for the relaxation of the ZMP constraints. This approach is similar to the work of [21], but their MPC is formulated for the ZMP tracking, not for the CP tracking. Finally, the proposed CP-MPC can be formulated as follows,

$$\begin{aligned} \min_{\mathbf{P}_x, \Delta\mathbf{F}_x} \quad & \sum_{k=0}^{N-1} l_k(\xi_{x,k}, p_{x,k}) + \|\Delta\mathbf{F}_x\|^2_{\mathbf{w}_f} \\ \text{s. t.} \quad & \underline{\mathbf{b}}_p \le \mathbf{A}_p \begin{bmatrix} \mathbf{P}_x \\ \Delta\mathbf{F}_x \end{bmatrix} \le \bar{\mathbf{b}}_p \\ & \underline{\mathbf{b}}_f \le \Delta\mathbf{F}_x \le \bar{\mathbf{b}}_f. \end{aligned} \tag{6}$$

with $\mathbf{A}_p = \begin{bmatrix} \mathbf{I}_N & -\mathbf{S}_f \end{bmatrix}$. The cost function consists of the stage cost terms expressed in (5), and the regulation term of $\Delta\mathbf{F}_x$ is added with the positive diagonal weighting matrix $\mathbf{w}_f$. Note that the highest weight is assigned to the last state $w_{\xi,N-1}$ for convergence of CP control. The variable $\mathbf{P}_x \in \mathbb{R}^N$ denote the sequence of ZMP inputs along with the time horizon N in the x-direction, and $\Delta\mathbf{F}_x = [\Delta F_{x,1}\ \Delta F_{x,2}\ \cdots\ \Delta F_{x,m}]^T \in \mathbb{R}^m$ denotes the vector composed of the additional displacement from the pre-planned footsteps in x-direction where m is the number of pre-planned footsteps within the time horizon N. The matrix $\mathbf{I}_N \in \mathbb{R}^{N\times N}$ is the N size identity matrix, and the matrix $\mathbf{S}_f \in \mathbb{R}^{N\times m}$ is the selection matrix composed of 0 and 1 to activate $\Delta\mathbf{F}_x$ at the time step after the current SSP. The vectors $\bar{\mathbf{b}}_p$ and $\underline{\mathbf{b}}_p$ represent the upper and lower bounds of the ZMP constraint determined by the support-foot geometry. The vectors $\bar{\mathbf{b}}_f$ and $\underline{\mathbf{b}}_f$ represent the upper and lower bounds of the kinematic constraints to avoid the singularity of legs

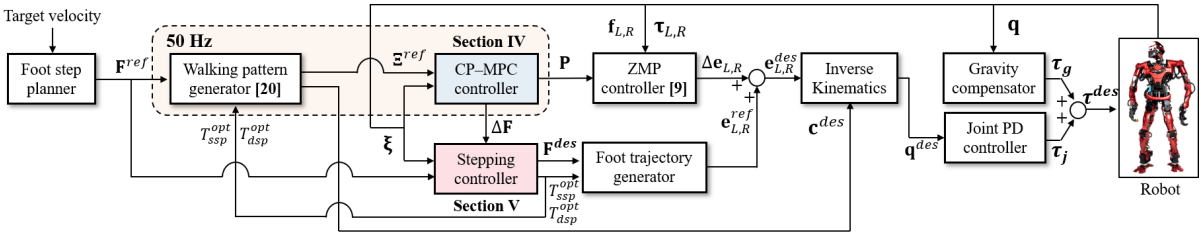

Fig. 1. Overall control framework of humanoid locomotion for capture point tracking including the proposed methods (CP-MPC and stepping controller)

and collision between the legs. In (6), when $\mathbf{P}_x$ is limited by the ZMP constraint and cannot generate an additional ZMP input for CP control, $\Delta\mathbf{F}_x$ is generated to increase the control performance of the CP by changing the future footstep position and relaxing the ZMP constraint. Note that the MPC variable for footstep adjustment, $\Delta\mathbf{F}_x$, is used for the stepping controller in Section V-A.

V. Stepping Controller

A. Online Optimization of Step Location and Step Timing

This section introduces a QP-based online foot stepping controller. The goal of this controller is to optimize the step location of the next foot and the step timing using the current CP, $\boldsymbol{\xi} = [\xi_x\ \xi_y]^T$, and the elapsed time, t, after the start of the swing phase, considering LIPM-based CP end-of-step dynamics of (3). This QP formulation was proposed by Khadiv et al. [23], [24] to control the CP end-of-step by adjusting the stepping position and time.

$$
\begin{aligned}
\min_{\mathbf{F}^{opt},\tau_{ssp}^{opt},\mathbf{b}} \quad & \frac{w_F}{2}\|\mathbf{F}^{opt}-\mathbf{F}_{nom}\|^2+\frac{w_\tau}{2}(\tau_{ssp}^{opt}-\tau_{nom})^2 \\
& +\frac{w_b}{2}\|\mathbf{b}-\mathbf{b}_{nom}\|^2 \\
\text{s. t.} \quad & \underline{\mathbf{F}}^{opt}\le\mathbf{F}^{opt}\le\bar{\mathbf{F}}^{opt} \\
& \underline{\tau}_{ssp}^{opt}\le\tau_{ssp}^{opt}\le\bar{\tau}_{ssp}^{opt} \\
& \underline{\mathbf{b}}\le\mathbf{b}\le\bar{\mathbf{b}} \\
& \mathbf{F}^{opt}+\mathbf{b}=(\boldsymbol{\xi}-\mathbf{F}_0)e^{-\omega t}\tau_{ssp}^{opt}+\mathbf{F}_0
\end{aligned}
\tag{7}
$$

Here, $\mathbf{F}^{opt}=[F_x^{opt}\ F_y^{opt}]^T$, $\tau_{ssp}^{opt}=e^{\omega T_{ssp}^{opt}}$, and $\mathbf{b}=[b_x\ b_y]^T$ are optimization variables, and indicate next step position, step time term, and CP offset, respectively. Note that the variable, τ_{ssp}^{opt}, is introduced to make the equality constraint linear in (7). Also, $\mathbf{F}_{nom}$, $\tau_{nom}=e^{\omega T_{nom}}$ and $\mathbf{b}_{nom}$ are nominal values for each optimization variable. The variable, $\mathbf{F}_0$, is the position of the current supporting foot. In the optimization problem (7), the optimal solution is obtained while error terms between the nominal values $\{\mathbf{F}_{nom},\tau_{nom},\mathbf{b}_{nom}\}$ and optimization variables $\{\mathbf{F}^{opt},\tau_{ssp}^{opt},\mathbf{b}\}$ are penalized by the weighting parameters w_F, w_τ and w_b. The step location of the next step, step time, and CP offset terms are constrained by upper and lower bound respectively, and the CP end-of-step dynamics in (3) is implemented as an equality constraint.

In [24], since the CoM trajectory is not defined, the reference CoM velocity tracking and walking stability are guaranteed by the accurate CP offset control of the stepping method. Accordingly, the highest weight is assigned to the CP offset error term with a weight of w_F:w_τ:w_b = 1:5:1000, and by assigning the lowest weight to the step location error term, the most error is allowed for the stepping position compared to other error terms.

In this work, the work by Khadiv et al. [24] is adopted as a baseline stepping algorithm. But some details are adjusted to our walking control framework. The nominal step location value is the sum of the pre-planned next step location and the MPC variable in Section. IV, $\mathbf{F}_{nom}=\mathbf{F}_1^{ref}+\Delta\mathbf{F}_1$. Also, nominal CP offset, $\mathbf{b}_{nom}$, is determined as a difference between the predicted CP at the end of SSP, $\boldsymbol{\xi}_T$, in the MPC of Section. IV and the predicted footstep, $\mathbf{F}_{nom}$. The optimized step location, $\mathbf{F}^{opt}$, is assigned with 500 times higher than the weight on the optimized step time as w_F:w_τ:w_b = 500:1:1000 because the footstep is optimized earlier in CP-MPC through $\Delta\mathbf{F}_1$, so the stepping controller complies with the output of the CP-MPC, and mainly optimizes the stepping time.

B. Adjustment of Double Support Phase Time

The ability to adjust the step location and step timing is only guaranteed when sufficient step time remains for the stepping controller. On the other hand, when the remaining step time is short, adjusting the step location and step timing in a short time generates fast swing foot motion and causes a large impact between the swing foot and the ground. To prevent the large impact in this situation, the step location and step timing are usually fixed without further adjustment based on a specific time margin [21], [24]. This means that it is difficult to respond to disturbances using stepping control that occur when the remaining step time is short. Therefore, it is necessary to begin the next stepping quickly to maintain balance rather than using the constant DSP time. However, this problem is not handled in the previous studies [25]–[28].

To solve this problem, we propose a DSP time adjustment algorithm (DSP scaler) that can quickly transit to the next step reducing the DSP time. This helps the robot to maintain balance by advancing the start of the next step adjustment. Fig. 2 represents the main idea of the DSP scaling method. In Fig. 2, it is assumed that the external force is applied to the robot from back to forward when the remaining time in the current swing phase is short. In this situation, in order to prevent the impact between the swing foot and the ground, the step time and the landing position of the swing foot are not changed during the last margin time of SSP, $[T_{fix}^{opt},T_{ssp}^{opt}]$ (pink period). However, even after the step location is fixed (red), the optimized step location (yellow) is still calculated by (7) during remaining step time.

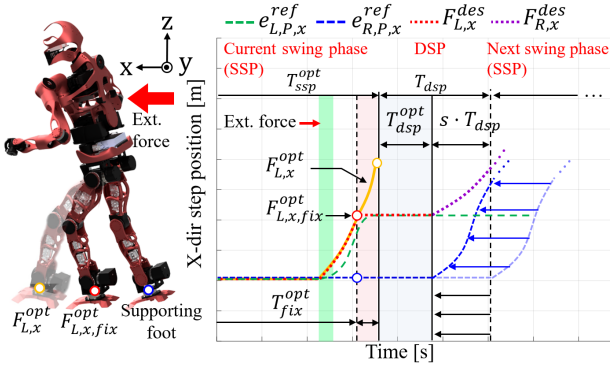

Fig. 2. Behavior example of the DSP scaler when the remaining time of SSP is insufficient to adjust the step.

Taking this situation into account, the proposed DSP scaler is designed to reduce the DSP time and advance the next swing phase fast according to the additionally generated swing foot location over the fixed step location, $\mathbf{F}^{opt} - \mathbf{F}^{opt}_{fix}$.

$$\dot{\mathbf{S}} = [\dot{S}_x\ \dot{S}_y]^T = \mathbf{K}_s \begin{bmatrix} |F_x{}^{opt} - F^{opt}_{x,fix}| \\ |F_y{}^{opt} - F^{opt}_{y,fix}| \end{bmatrix} - \mathbf{D}_s\mathbf{S}, \tag{8}$$

$$s = \|\mathbf{S}\|^2, \quad T^{opt}_{dsp} = T_{dsp}(1-s), \quad 0 \le s < 1,$$

where s is a scale factor, $\mathbf{K}_s \in \mathbb{R}^{2\times2}$, is a positive and proportional gain matrix, and $\mathbf{D}_s \in \mathbb{R}^{2\times2}$ is a positive diagonal matrix to converge $\mathbf{S}$ to zero. In (8), the error between the optimized, $\mathbf{F}^{opt}$, and fixed step location, $\mathbf{F}^{opt}_{fix}$, is calculated only during the last margin time $[T^{opt}_{fix}, T^{opt}_{ssp}]$, and the variable, $\mathbf{S}$, is reduced to zero by $\mathbf{D}_s$ in (8). The scaled DSP time at the end of the current swing phase is reflected as the following DSP time, T^{opt}_{dsp}.

VI. Results of Simulations and Experiments

A. System Overview

In this section, the system overview of simulations and experiments is provided. The humanoid robot, TOCABI, used in the simulation [29] and experiment has a total of 33 DOFs (both arms:16, both legs:12, waist:3 and neck:2), and the height and weight are approximately 1.8 m and 100 kg, respectively. All actuators consist of BLDC motors and harmonic gears. An IMU is located in the pelvis, and a FTS is mounted to each foot. The control frequency is 2 kHz, and the walking pattern generator [20] and the CP-MPC controller in Sec. IV are operated at 50 Hz through parallel thread due to high computational load. The results of the simulations and experiments are also available in the supplementary video.

B. Walking Simulation in the Presence of Disturbance

The robustness of the proposed method is demonstrated in the presence of external force through comparison simulations. In all simulations, a total of 14 in-place walking steps are planned with 0.9 s of the step duration including 0.6 s nominal SSP, T_{ssp}, and 0.3 s nominal DSP, T_{dsp}. In addition, to prevent a sudden change of the swing foot near the end of the swing phase, the footstep and step timing adjustments from the stepping control are fixed 0.1 s before the end of SSP.

1) Comparison Simulations for the Validation of the Proposed CP-MPC: Comparative simulations were conducted to validate the robustness of the proposed CP-MPC without DSP scaler against external forces. Three types of algorithms are implemented for comparative simulations (See Fig. 3(b)). Method 2) is a synthesis framework where previous CP-MPC [11] and stepping controller [24] are solved independently. The previous CP-MPC [11] is implemented to show the effectiveness of the footstep adjustment variable in the proposed CP-MPC. Method 3) is a synthesis framework in which the instantaneous CP control [15] and the stepping controller in [24] operate independently. This method is implemented to compare the balance performance of the CP controller based on the MPC scheme and the instantaneous CP control. Method 4) is a MPC framework for ZMP control [21], and the stepping controller [24] operates based on the footstep adjustment variable from the MPC similar to our approach in Section V-A. Method 4) was implemented to compare two similar MPC frameworks (ZMP control in [21] and the proposed CP control method) with the same stepping controller [24]. The weighting parameters of the stepping controller in 2) and 3) cases are the same as in the [24] (w_F:w_τ:w_b = 1:5:1000), and the weighting parameters of the stepping controller in case 4) are the same as ours (w_F:w_τ:w_b = 500:1:1000).

Before comparison with the other methods, a simulation was conducted to study how the robot recovers from external forces using the proposed method without the DSP scaler. In this simulation, a step external force of 225 N in the negative x-direction (from forward-to-back) is applied during 0.2 s to the pelvis of the robot at 5.9 s while walking in place (5th SSP with a left supporting leg, 70% of swing phase left). The upper graph in Fig. 3(a) shows the x-direction of reference ZMP, p^{ref}_x, CP-MPC variables in (6) (desired ZMP, p_x, and $\Delta F_{x,1}$), and the CP error, $\xi_{err,x}$. When the external force is applied to the robot, the desired ZMP is generated by the CP-MPC controller to reduce the increased CP error. In the case of this simulation, the desired ZMP was not enough to control the CP error within the ZMP constraint due to the large disturbance. Accordingly, $\Delta F_{x,1}$ is generated which changes the upcoming landing foot position and relaxes the future ZMP constraints to control the CP error further. The lower graph in Fig. 3(a) shows the reference foot trajectories and the optimized step time generated by the stepping controller in (7). The optimized step location, F^{des}_x, is calculated by the stepping controller, and the reference foot trajectories are interpolated using the cubic polynomial function from the initial swing foot position to F^{des}_x. The stepping controller also optimizes step timing, T^{opt}_{ssp}, online to respond to external forces. In conclusion, the robot stabilizes the CP against external force through ZMP control and two backward steps.

While changing the direction of an external force by 30 deg, the maximum magnitude of the external force that the robot can withstand according to the remaining step time is analyzed for four different methods including the proposed method. As in the first simulation, an external force is applied to the pelvis of the robot at the 5th SSP (left foot support) while walking in place. In Fig. 3(b), the magnitude of the external force that the

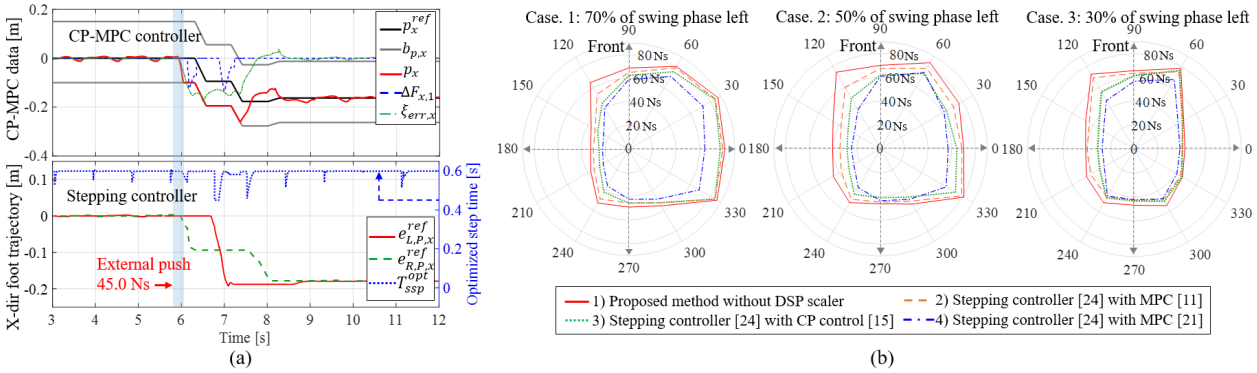

Fig. 3. (a) Output data of the proposed method without DSP scaler when 70% of swing phase left. (b) Comparisons of the maximum endurable impulse momentum between different controllers using the disturbance polygon according to the remaining step time.

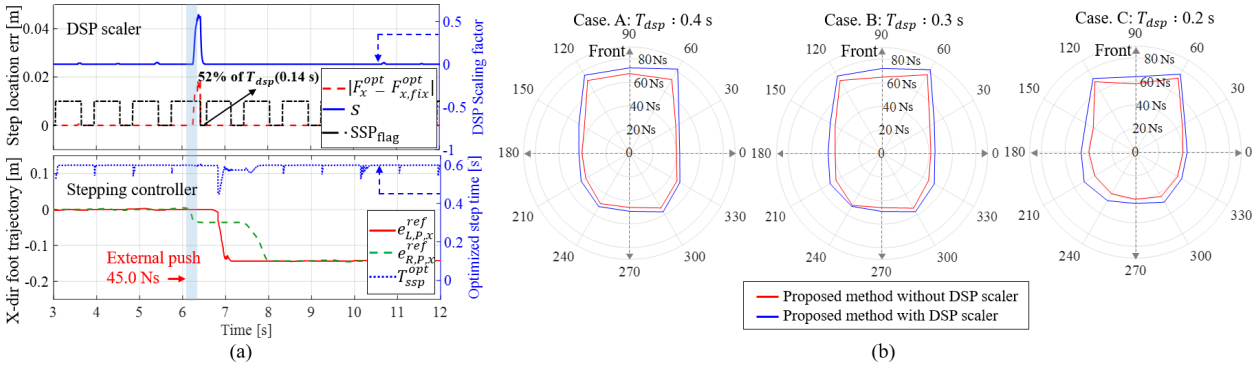

Fig. 4. (a) Output data of the stepping controller with DSP scaler ($T_{dsp} = 0.3s$) when 30% of swing phase left. (b) Maximum endurable impulse momentum analysis using the disturbance polygon for the usage of the DSP scaler according to the nominal DSP time.

robot can withstand depending on the direction is expressed in the form of a polygon. We will refer to this polygon as a disturbance polygon (DP) in this paper. Note that the direction of the external force vector starts from the robot and goes outward, and force in 90 deg indicates the push from back to front (+x axis). Fig. 3(b) is classified into three cases according to the remaining step time.

As shown in Fig. 3(b), compared to the other three types of the implemented control methods, the proposed method shows the largest area of DP in all three cases according to the remaining step time. In particular, compared to the method 2) (the second largest DP area), the proposed method has a larger DP area by 10.9, 17.2, and 12.5 %, respectively, for each case. It is conjectured that the proposed method outperforms the other methods because the proposed CP-MPC optimizes the CP tracking problem using both the support foot (**P**) and the swing foot (Δ**F**) in advance and uses the optimized footstep in the stepping controller jointly. On the other hand, method 2) solves the CP-MPC and stepping controller independently. Method 4) is also designed similarly but the objective of MPC is the ZMP tracking which is different with the control objective of the stepping controller (CP end-of-step). Thus, compared to method 4), the DP area of the proposed method has larger by 55.8, 67.7, and 52.7% for each case. Additionally, as the remaining step time decreases, the ability to control step location and timing decreases. In particular, in the direction of 0, 30, and 330 deg, Case.1 shows a more robust performance compared to Case.2 and 3. Because, in Case.1, the robot has more time to move the swing foot further than the other cases.

2) Comparison Simulations for the Validation of DSP Scaler: From the results of the simulation according to the remaining step time, it is observed that when the remaining step time is insufficient, the performance to withstand external force through step adjustment deteriorates. In this simulation, therefore, it is analyzed whether the robot's ability to withstand external force is improved according to the usage of the DSP scaler.

As same with the previous simulation, a step external force of 225 N is applied to the robot in the negative x-direction for 0.2 s, but at approximately 6.2 s (4th SSP, 30% of the swing phase left). The upper graph in Fig. 4(a) represents the control action of the DSP scaler. In Fig 4(a), when the robot receives an external force, even though the optimized step location is fixed to $F_{x,fix}^{opt}$, F_x^{opt} is generated during the remaining of the current swing phase and $|F_x^{opt} - F_{x,fix}^{opt}|$ of about 2 cm occurs. The scale factor, s, is generated by the step location error, and the next DSP time is reduced by s and fixed at the end of SSP. Note that due to s, the DSP time is reduced to 0.14 s, which is 52% less than the T_{dsp} (nominal value: 0.3 s) of the previous steps. The SSP flag indicates the support phase of the robot, which is positive for the SSP and zero for DSP. Through the

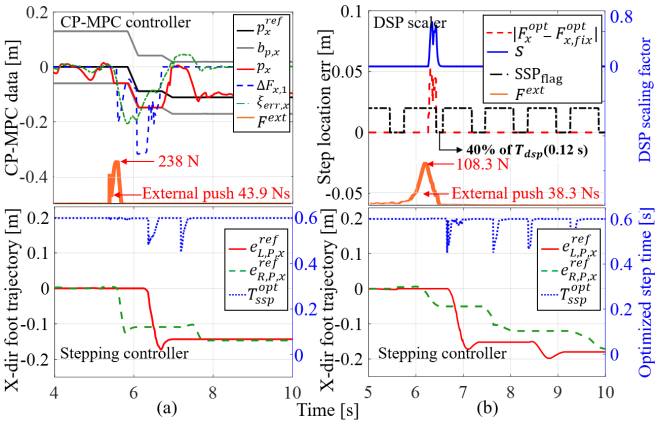

Fig. 5. Experimental results (a) CP-MPC outputs, foot step location, and step time when the remaining step time is sufficient (about 80% left). (b) DSP scaler outputs, foot step location, and step time when the remaining step time is insufficient (about 25% left).

SSP flag, it can be seen that the DSP duration around 6.4 s is reduced compared to the DSP of other steps. The lower graph of Fig. 4(a) shows the reference foot trajectories and the optimized step time. Because the remaining step time is not enough, the right foot cannot step farther than in the case of Fig. 3(a). In this case, the DSP scaler advanced the next step fast, and the left foot quickly adjusted the step, allowing the robot to maintain its balance. On the other hand, the robot could not maintained the balance when the DSP scaler is not used.

The previous simulation results verified that the DSP scaler can increase the balancing performance against the external forces in a situation where the remaining step time is short. However, it may be interpreted that the shorter the DSP time, the more robust the robot becomes, and the shorter nominal DSP time could show better robustness against disturbances. Therefore, the robustness to external forces while changing the nominal DSP time is analyzed. In Fig. 4(b), without the DSP scaler, when the nominal DSP time is 0.2 s, the area of the DP is 6.1% smaller than the one with 0.3 s nominal DSP time, and when the nominal DSP time is 0.1 s, the walking itself becomes unstable. Therefore, a short nominal DSP time does not always improve the balance control performance, and these results show that selecting a proper DSP time is necessary for stable walking. Additionally, when 30% of step time left, the comparisons with or without the DSP scaler are conducted while changing the nominal DSP time. In Fig. 4(b), with the DSP scaler, the area of the DP increased by 13.8%, 10.1%, and 8.4% in Case. A, B, and C, respectively, compared to the cases without the DSP scaler.

C. Walking Experiments

A total of two experiments are conducted to analyze the robustness of the proposed method using a real robot, TOCABI, in the presence of disturbance. In the first experiment, the performance of the stepping algorithm is demonstrated by applying external pushes to the robot when the remaining step time for step adjustment is sufficient. In the second experiment, the performance of the DSP scaler is verified when the step time for step adjustment is insufficient. External pushes are applied in the nagative x direction to the pelvis of the robot using a tool, and the magnitude of the impact is measured using a FTS attached to the end of the tool. As in the simulation, the step duration is 0.9 s, including a nominal SSP of 0.6 s and a nominal DSP of 0.3 s.

Fig. 5(a) represents the result of the first experiment. As shown in the upper graph in Fig. 5(a), an external push of 43.9 Ns is applied to the robot at approximately 5.4 s. The CP error is over -0.2 m by the external push, and the desired ZMP, p_x, is generated and reaches the lower bound of the ZMP constraint, $\underline{b}_p$, to control the CP error. In addition, $\Delta F_{x,1}$ is generated for swing foot adjustment to overcome the limitation of ZMP control by relaxing the ZMP constraint. The stepping controller optimizes the step location and timing based on the generated foot adjustment, $\Delta F_{x,1}$, and the current CP. The optimized time and swing foot trajectories are shown in the lower graph of Fig. 5(a). Through two back-step motion, the robot overcomes the external push and maintains its balance. However, when the stepping method was not used, the robot fell with a smaller impulse (26.6 Ns).

Fig. 5(b) shows the results of the second experiment. Unlike the first experiment, when the remaining step time is not enough to adjust the swing foot (approximately 25% of the swing phase left), an external push of 38.3 Ns in the negative x-direction is applied to the robot. After the push, the step adjustment is fixed at 0.1 s before the end of the SSP, and F_x^{opt} is additionally generated up to 5.2 cm compared to the fixed value, $F_{x,fix}^{opt}$, by the continuous disturbance. The scale factor, s, is generated approximately to be 0.6 by the step location error, and the DSP time is reduced by 60% compared to the nominal DSP (0.3 s). Similar to the SSP flag in Fig. 4(a), it can be seen that the DSP at 6.5 s is reduced compared to the DSP of other steps. Consequently, even though the first step is limited to $F_{x,fix}^{opt}$ by the time margin, the robot is able to maintain its balance by quickly advancing the second step. On the other hand, when the DSP scaler is not used, the robot could not maintain its balance through stepping even for a smaller external push (31.85 Ns).

VII. Conclusion

In this study, a novel foot stepping algorithm adjusting DSP time is proposed to improve the robustness of walking against external perturbations. First, an MPC-based CP controller is proposed that generates not only the desired ZMP for the supporting leg but also the desired footstep location for the swing leg. The foot stepping control is operated based on the footstep location obtained from the proposed MPC. Furthermore, to overcome the disturbance applied when the remaining step time is not sufficient to adjust the footstep, a DSP-adjusting approach is proposed to bring the next footstep forward by reducing the DSP time.

Through extensive simulations and real-robot experiments, the robustness and feasibility of the proposed method are validated using our full-size humanoid robot, TOCABI, in the presence of external pushes.

REFERENCES

[1] S. Kajita and K. Tani, "Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*. IEEE Computer Society, 1991, pp. 1405–1406.

[2] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 1. IEEE, 2001, pp. 239–246.

[3] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by a simple three-dimensional inverted pendulum model," *Advanced Robotics*, vol. 17, no. 2, pp. 131–147, 2003.

[4] S. Nakaura, M. Sampei *et al.*, "Balance control analysis of humanoid robot based on zmp feedback control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3. IEEE, 2002, pp. 2437–2442.

[5] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 2. IEEE, 2003, pp. 1620–1626.

[6] J.-Y. Kim, I.-W. Park, and J.-H. Oh, "Experimental realization of dynamic walking of the biped humanoid robot khr-2 using zero moment point feedback and inertial measurement," *Advanced Robotics*, vol. 20, no. 6, pp. 707–736, 2006.

[7] Y. Choi, D. Kim, Y. Oh, and B.-J. You, "Posture/walking control for humanoid robot based on kinematic resolution of com jacobian with embedded motion," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1285–1293, 2007.

[8] J.-Y. Kim, I.-W. Park, and J.-H. Oh, "Walking control algorithm of biped humanoid robot on uneven and inclined floor," *Journal of Intelligent and Robotic Systems*, vol. 48, no. 4, pp. 457–484, 2007.

[9] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, and K. Yokoi, "Biped walking stabilization based on linear inverted pendulum tracking," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 4489–4496.

[10] J. Englsberger, C. Ott, M. A. Roa, A. Albu-Schäffer, and G. Hirzinger, "Bipedal walking control based on capture point dynamics," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 4420–4427.

[11] M. Krause, J. Englsberger, P.-B. Wieber, and C. Ott, "Stabilization of the capture point dynamics for bipedal walking based on model predictive control," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 165–171, 2012.

[12] M. Morisawa, S. Kajita, F. Kanehiro, K. Kaneko, K. Miura, and K. Yokoi, "Balance control based on capture point error compensation for biped walking on uneven terrain," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. IEEE, 2012, pp. 734–740.

[13] J. Englsberger and C. Ott, "Integration of vertical com motion and angular momentum in an extended capture point tracking controller for bipedal walking," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*. IEEE, 2012, pp. 183–189.

[14] M. Shafiee-Ashtiani, A. Yousefi-Koma, R. Mirjalili, H. Maleki, and M. Karimi, "Push recovery of a position-controlled humanoid robot based on capture point feedback control," in *2017 5th RSI International Conference on Robotics and Mechatronics (ICRoM)*. IEEE, 2017, pp. 126–131.

[15] H.-M. Joe and J.-H. Oh, "A robust balance-control framework for the terrain-blind bipedal walking of a humanoid robot on unknown and uneven terrain," *Sensors*, vol. 19, no. 19, p. 4194, 2019.

[16] M.-J. Kim, D. Lim, G. Park, and J. Park, "Humanoid balance control using centroidal angular momentum based on hierarchical quadratic programming," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 6753–6760.

[17] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *2006 6th IEEE-RAS international conference on humanoid robots*. IEEE, 2006, pp. 200–207.

[18] A. L. Hof, "The 'extrapolated center of mass' concept suggests a simple control of balance in walking," *Human movement science*, vol. 27, no. 1, pp. 112–125, 2008.

[19] P.-B. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2006, pp. 137–142.

[20] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online walking motion generation with automatic footstep placement," *Advanced Robotics*, vol. 24, no. 5-6, pp. 719–737, 2010.

[21] H.-M. Joe and J.-H. Oh, "Balance recovery through model predictive control based on capture point dynamics for biped walking robot," *Robotics and Autonomous Systems*, vol. 105, pp. 1–10, 2018.

[22] G. Romualdi, S. Dafarra, G. L'Erario, I. Sorrentino, S. Traversaro, and D. Pucci, "Online non-linear centroidal mpc for humanoid robot locomotion with step adjustment," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 10 412–10 419.

[23] M. Khadiv, A. Herzog, S. A. A. Moosavian, and L. Righetti, "Step timing adjustment: A step toward generating robust gaits," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 35–42.

[24] ——, "Walking control based on step timing adaptation," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 629–643, 2020.

[25] H. Jeong, I. Lee, J. Oh, K. K. Lee, and J.-H. Oh, "A robust walking controller based on online optimization of ankle, hip, and stepping strategies," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1367–1386, 2019.

[26] F. Nazemi, A. Yousefi-Koma, M. Khadiv *et al.*, "A reactive and efficient walking pattern generator for robust bipedal locomotion," in *2017 5th RSI International Conference on Robotics and Mechatronics (ICRoM)*. IEEE, 2017, pp. 364–369.

[27] R. J. Griffin, G. Wiedebach, S. Bertrand, A. Leonessa, and J. Pratt, "Walking stabilization using step timing and location adjustment on the humanoid robot, atlas," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 667–673.

[28] H. Jeong, I. Lee, O. Sim, K. Lee, and J.-H. Oh, "A robust walking controller optimizing step position and step time that exploit advantages of footed robot," *Robotics and Autonomous Systems*, vol. 113, pp. 10–22, 2019.

[29] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.