Footstep and Timing Adaptation for Humanoid Robots Utilizing Pre-computation of Capture Regions

Yuichi Tazaki†

***Abstract*— This study proposes a real-time footstep and timing adaptation mechanism for humanoid robots that can be integrated into a conventional walking pattern generator and increase the robustness of walking against disturbances. In order to meet the strict real-time constraint of humanoid robot control, the proposed method computes viable capture basins in the design phase. This pre-computed data can be used at run-time to modify the foot placement, the timing of landing, and the center-of-mass movement in response to applied disturbances with small computation cost. The performance of the proposed method is evaluated in simulation experiments.**

I. Introduction

Humanoid robots are expected to assist and replace human labor by utilizing its morphological resemblance to human being. Bipedal locomotion, however, has inherent risk of falling. When a robot falls, it causes damage to not only the robot itself but also to the operational area and humans in the vicinity. Fall prevention is therefore one of the fundamental required functionalities of humanoids robots.

When large disturbance is applied while the robot is standing still, it needs to take one or more steps to recover its balance. Similarly, when disturbance is applied while walking, the robot needs to adjust the placement and timing of future footsteps to maintain stability. In either case, since the movement speed of the swing foot and allowable stepping region are both limited, adjusting not only the single immediate step but multiple future steps may be needed in order to tolerate greater disturbances, especially when walking at high speed. Some early studies attempted to improve the robustness of walking by replanning the walking pattern at high frequency. For example, in [1], the walking pattern was updated at 20[ms] update cycle, reflecting the current state of the robot. During the last decade, capturability analysis has served as a theoretical basis for stepping adaptation of bipedal robots ([2]-[5]). While earlier studies focused on push recovery only, it has been shown in [6] and [7] that capturability information can be utilized for implementing a footstep adaptation mechanism that can be used in combination with a conventional walking pattern generator. In order to maximize the fall-avoiding performance of the robot, it is important to compute precise capture regions reflecting the kinematics and dynamics of the robot. A pioneering work by Koolen et al. made a number of strong simplifications such as constant step duration and circular landing region [4]. Recent studies have made various attempts to remove these restrictions and to fully utilize the degrees of freedom of the centroidal dynamics ([6]-[9]). Simultaneous adjustment of footstep location and timing were realized in [6] and [7], but it was realized by imposing stronger restrictions than the general N-step capturability; namely, manual specification of the DCM offset ([6]), and requiring 0-step capturability after landing ([7]). In [10], the authors proposed a method capable of footstep and timing adjustment based on N-step capturability. The basic idea was to perform expensive computation of N-step capture basins in the design phase and use it in the real-time controller.

In this paper, we propose a footstep and timing adjustment method for humanoid robots that makes use of precomputed capture basins. One of the major improvements from the previous method is the consideration of turning movement. Moreover, to cope with the increased dimensionality of the dynamical model, a novel and efficient algorithm for capture basin computation is proposed, which dramatically reduces the offline computation time. Furthermore, the proposed step adaptation controller is combined with a conventional walking pattern generator and its performance is tested in dynamical simulation of a full degrees-of-freedom humanoid robot model.

The rest of this paper is organized as follows. In Section II, mathematical models of the center-of-mass dynamics and the swing foot movement are formulated. In Section III, the proposed algorithm for capture basin computation is presented. In Section IV, a footstep and step duration adjustment method based on precomputed capture basins is presented. In Section V, simulation results using a full kino-dynamic model of a humanoid robot are shown. Concluding remarks are given in Section VI.

II. Low-dimensional Dynamical Model for Capturability Analysis

A. State Space Model

In this section, a dynamical system model of center-of-mass movement used for capturability analysis is formulated. The dynamics of the center-of-mass (CoM) is govern by the Linear Inverted Pendulum Mode (LIPM) [12][12] as shown below, under the restriction that the total angular momentum around the CoM is zero.

$$\ddot{\boldsymbol{p}}_{\rm com} = \frac{1}{T^2}(\boldsymbol{p}_{\rm com} - \boldsymbol{p}_{\rm cop}) \tag{1}$$

Here, $\boldsymbol{p}_{\rm com} \in \mathbb{R}^2$ is the position of the CoM, $\boldsymbol{p}_{\rm cop} \in \mathbb{R}^2$ is the position of the center of pressure (CoP). Moreover, T is a constant defined as $T = \sqrt{h/g}$, where h is the

†The author is with Faculty of Engineering, Department of Mechanical Engineering, University of Kobe, 1-1 Rokkodai, Nada-ku, Kobe, Japan. tazaki@mech.kobe-u.ac.jp

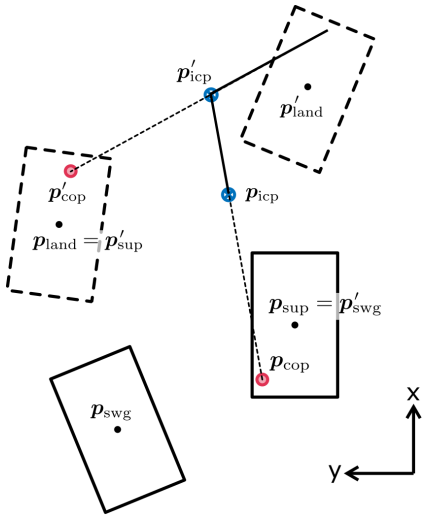


Fig. 1. Relationship of variables expressed in the global coordinate.

height of the CoM from the ground and g is the acceleration of gravity. The Instantaneous Capture Point (ICP, a.k.a. divergent component of motion, DCM) is defined as $\boldsymbol{p}_{\mathrm{icp}} = \boldsymbol{p}_{\mathrm{com}} + T\dot{\boldsymbol{p}}_{\mathrm{com}}$. By using the ICP, the second-order ODE (1) can be equivalently expressed as a pair of first-order ODEs as shown below.

$$\dot{\boldsymbol{p}}_{\mathrm{icp}} = \frac{1}{T}(\boldsymbol{p}_{\mathrm{icp}} - \boldsymbol{p}_{\mathrm{cop}}), \tag{2a}$$

$$\dot{\boldsymbol{p}}_{\mathrm{com}} = -\frac{1}{T}(\boldsymbol{p}_{\mathrm{com}} - \boldsymbol{p}_{\mathrm{icp}}) \tag{2b}$$

If the CoP is fixed to $\boldsymbol{p}_{\mathrm{cop}}$ during the interval $[t, t+\tau]$, from (2a) we obtain

$$\boldsymbol{p}_{\mathrm{icp}}(t+\tau) = (\boldsymbol{p}_{\mathrm{icp}}(t) - \boldsymbol{p}_{\mathrm{cop}})e^{\frac{\tau}{T}} + \boldsymbol{p}_{\mathrm{cop}} \tag{3}$$

Next, let us discuss the change of states in a single walking step. The configuration of each foot is expressed in $\mathrm{SE}(2)$. A configuration $\boldsymbol{q} \in \mathrm{SE}(2)$ consists of the position $\boldsymbol{p} \in \mathbb{R}^2$ and the rotation angle $\theta \in [-\pi, \pi]$. Let $\boldsymbol{q}_{\mathrm{sup}}$ be the configuration of the support foot, and $\boldsymbol{q}_{\mathrm{swg}}$ and $\boldsymbol{q}_{\mathrm{land}}$ be the configuration of the swing foot at lift-off and landing, respectively. Moreover, let $\boldsymbol{p}_{\mathrm{cop}} \in \mathbb{R}^2$ be the position of the CoP during the step, and $\boldsymbol{p}_{\mathrm{icp}}$ be the position of the ICP at the beginning of the step. Furthermore, let τ be the step duration. Using these symbols, the update of variables in one walking step can be expressed as follows.

$$\begin{aligned} \boldsymbol{q}'_{\mathrm{sup}} &= \boldsymbol{q}_{\mathrm{land}} \\ \boldsymbol{q}'_{\mathrm{swg}} &= \boldsymbol{q}_{\mathrm{sup}} \\ \boldsymbol{p}'_{\mathrm{icp}} &= (\boldsymbol{p}_{\mathrm{icp}} - \boldsymbol{p}_{\mathrm{cop}})e^{\frac{\tau}{T}} + \boldsymbol{p}_{\mathrm{cop}} \end{aligned} \tag{4}$$

This relationship is illustrated in Fig. 1. Note that in each walking step, the ICP moves away from the CoP exponentially on a straight line.

One effective way to reduce the dimensionality of capturability analysis is to express the variables in the local coordinate of the support foot. Let us define a coordinate frame whose origin is attached to the center of the support foot. Here, the x axis of this coordinate frame points to the forward direction of the support foot, whereas the y axis is orthogonal to the x axis and points to the direction of the other foot (i.e., the swing foot). Given a support foot configuration $(\boldsymbol{p}_{\mathrm{sup}}, \theta_{\mathrm{sup}})$ and a symbol δ indicating the current support foot (1 for right support, -1 for left support), an arbitrary point $\hat{\boldsymbol{p}}$ expressed in the local coordinate can be transformed into the global coordinate $\boldsymbol{p}$ as shown below:

$$\begin{aligned} \boldsymbol{p} &= R(\theta_{\mathrm{sup}}) \begin{bmatrix} 1 & 0 \\ 0 & \delta \end{bmatrix} \hat{\boldsymbol{p}} + \boldsymbol{p}_{\mathrm{sup}}, \\ \theta &= \delta\hat{\theta} + \theta_{\mathrm{sup}} \end{aligned}$$

where $R(\theta)$ is the rotation matrix in $\mathbb{R}^2$ with the rotation angle θ. In the following discussion, a variable expressed in the support foot coordinate is denoted by a symbol with the hat; for example, $\hat{\boldsymbol{p}}_{\mathrm{cop}}$.

By expressing the initial and landing configurations of the swing foot and the ICP in the support foot coordinate, the dynamics (4) can be rewritten as follows.

$$\hat{\boldsymbol{p}}'_{\mathrm{swg}} = -SR(\hat{\theta}_{\mathrm{land}})^{\mathsf{T}}\hat{\boldsymbol{p}}_{\mathrm{land}} \tag{5}$$

$$\hat{\theta}'_{\mathrm{swg}} = \hat{\theta}_{\mathrm{land}} \tag{6}$$

$$\begin{aligned} \hat{\boldsymbol{p}}'_{\mathrm{icp}} = SR(\hat{\theta}_{\mathrm{land}})^{\mathsf{T}}(e^{\frac{\tau}{T}}(\hat{\boldsymbol{p}}_{\mathrm{icp}} - \hat{\boldsymbol{p}}_{\mathrm{cop}}) \\ + \hat{\boldsymbol{p}}_{\mathrm{cop}} - \hat{\boldsymbol{p}}_{\mathrm{land}}) \end{aligned} \tag{7}$$

Here, $S = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right]$. For later use, let us define the state variable as follows.

$$\boldsymbol{x} = \begin{bmatrix} \hat{\boldsymbol{q}}_{\mathrm{swg}} \\ \hat{\boldsymbol{p}}_{\mathrm{icp}} \end{bmatrix} \tag{8}$$

Apart from the dynamical relationship (5)-(7), a number of constraints must be satisfied. First, the CoP must be included in the feasible region:

$$\hat{\boldsymbol{p}}_{\mathrm{cop}} \in \mathcal{R}_{\mathrm{cop}} \tag{9}$$

where $\mathcal{R}_{\mathrm{cop}}$ is a rectangular region that inner-approximates the contact surface of the support foot:

$$\mathcal{R}_{\mathrm{cop}} = \{\hat{\boldsymbol{p}}_{\mathrm{cop}} \,|\, \hat{\boldsymbol{p}}_{\mathrm{cop,min}} \le \hat{\boldsymbol{p}}_{\mathrm{cop}} \le \hat{\boldsymbol{p}}_{\mathrm{cop,max}}\} \tag{10}$$

Here, inequality between vectors is evaluated componentwise. Second, the landing configuration must be included in the feasible swing foot region:

$$\hat{\boldsymbol{q}}_{\mathrm{land}} \in \mathcal{R}_{\mathrm{swg}} \tag{11}$$

The feasible swing foot region can be specified by the designer based on the kinematics of the robot, but it must satisfy the following condition:

$$\hat{\boldsymbol{q}}_{\mathrm{land}} \in \mathcal{R}_{\mathrm{swg}} \Leftrightarrow \hat{\boldsymbol{q}}'_{\mathrm{swg}} \in \mathcal{R}_{\mathrm{swg}} \tag{12}$$

where $\hat{\boldsymbol{q}}'_{\mathrm{swg}}$ is given by (5) and (6). This condition requires that if a landing configuration is inside the feasible region, then the initial swing foot configuration of the next step should also be inside the feasible landing region. Third, the step duration should not be smaller than the minimum duration determined from the initial and landing configurations of the swing foot; namely,

$$\tau \ge \tau_{\min}(\hat{\boldsymbol{q}}_{\mathrm{swg}}, \hat{\boldsymbol{q}}_{\mathrm{land}}) \tag{13}$$

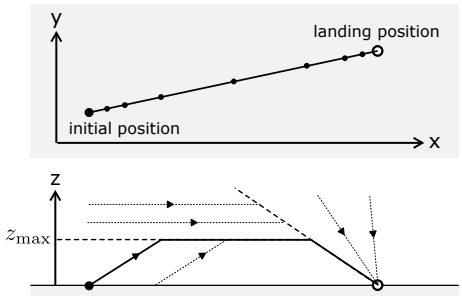


Fig. 2. Horizontal and vertical movement of the swing foot

The definition of $\tau_{\min}$ is discussed in detail in the next subsection.

The CoP constraint (9) can be equivalently transformed into a constraint on a tuple $(\boldsymbol{x}, \tau, \boldsymbol{x}')$. From (5) and (7), we get

$$\hat{\boldsymbol{p}}_{\rm icp} = (1 - e^{-\frac{\tau}{T}})\hat{\boldsymbol{p}}_{\rm cop} + e^{-\frac{\tau}{T}}\boldsymbol{\mu} \tag{14}$$

where $\boldsymbol{\mu} = R(\hat{\theta}'_{\rm swg})S(\hat{\boldsymbol{p}}'_{\rm icp} - \hat{\boldsymbol{p}}'_{\rm swg})$. Note that $e^{-\frac{\tau}{T}}$ is always within the range $[0, 1]$. The above equation therefore shows that the ICP is given by the linear interpolation between the CoP and $\boldsymbol{\mu}$ with the interpolation ratio $e^{-\frac{\tau}{T}}$. From this relationship, the condition $\hat{\boldsymbol{p}}_{\rm cop} \in \mathcal{R}_{\rm cop}$ can be expressed as a condition on ICP as follows.

$$\hat{\boldsymbol{p}}_{\rm icp} \in \mathcal{R}_{\rm icp}(\boldsymbol{\mu}, \tau) = (1 - e^{-\frac{\tau}{T}})\mathcal{R}_{\rm cop} + e^{-\frac{\tau}{T}}\boldsymbol{\mu} \tag{15}$$

We call a tuple $(\boldsymbol{x}, \tau, \boldsymbol{x}')$ satisfying $\hat{\boldsymbol{q}}_{\rm swg} \in \mathcal{R}_{\rm swg}$, $\hat{\boldsymbol{q}}'_{\rm swg} \in \mathcal{R}_{\rm swg}$, (13), and (15) a feasible transition. Moreover, we denote the set of all feasible transitions by $\mathcal{F}$.

B. Swing foot movement model

Capturability analysis requires a simple yet realistic model of swing foot movement. A swing foot model is used for two purposes: to estimate the amount of time needed for moving the swing foot to the next landing configuration (i.e., the step duration), and to determine a continuous-time trajectory that connects the current configuration of the swing foot and the landing configuration. The constant duration model [4] is not very practical because step duration for small steps are overestimated while step duration for large steps is underestimated. The constant speed model was proposed by the authors [10]. This model also tends to underestimate the step duration because it does not take into account acceleration after lift-off and deceleration before landing. In this paper, we propose an improved model in which the horizontal and rotational movements of the swing foot is expressed by cubic splines. By assuming that the initial and terminal velocities are both zero, the cubic spline trajectory takes the maximum velocity at the middle. From the condition that this maximum velocity does not exceed the specified limit, $v_{\max}$ for the horizontal velocity and $\omega_{\max}$ for the rotational velocity, the total travel time is derived as follows.

$$\begin{aligned}&\tau_{\min}(\hat{\boldsymbol{q}}_{\rm swg}, \hat{\boldsymbol{q}}_{\rm land}) \\ &\quad = \frac{3}{2}\max\left(\frac{\|\hat{\boldsymbol{p}}_{\rm swg} - \hat{\boldsymbol{p}}_{\rm land}\|}{v_{\max}}, \frac{|\hat{\theta}_{\rm swg} - \hat{\theta}_{\rm land}|}{\omega_{\max}}\right)\end{aligned} \tag{16}$$

The only difference from the constant speed model is the presence of the constant $\frac{3}{2}$. Moreover, the $\max$ operator comes from the consideration that the horizontal and the rotational movement of the swing foot can be executed simultaneously.

The vertical movement of the swing foot is defined as illustrated in the lower half of Fig. 2. When the swing foot is inside the conic region above the landing point, then it descends towards the landing position. If it is outside the cone and its altitude is below $z_{\max}$, then it ascends to $z_{\max}$. Otherwise, it maintains the current altitude.

Note that $\hat{\boldsymbol{q}}_{\rm land}$ and $\hat{\boldsymbol{q}}'_{\rm swg}$ have one-to-one relationship (5)(6). Based on this, we also use the notation $\tau_{\min}(\hat{\boldsymbol{q}}_{\rm swg}, \hat{\boldsymbol{q}}'_{\rm swg})$ for convenience.

III. Capturability Analysis Method Based on Discretization of States

A. Definition of Viable Capture Basin

The set of initial states for which the ICP can be captured in N steps is called the N-step viable capture basin. Formally, this set is defined recursively as explained below. First, the 0-step viable capture basin, denoted by $\mathcal{P}_0$, is defined as the set of states for which the ICP is inside the support region.

$$\mathcal{P}_0 = \mathcal{R}_{\rm swg} \times \mathcal{R}_{\rm cop} \tag{17}$$

The viable capture basin for $N = 1$ and beyond are defined recursively. Namely, a state $\boldsymbol{x}$ is an element of $\mathcal{P}_N$ if it is not an element of $\mathcal{P}_k$ $(k = 0, 1, \ldots, N-1)$ and there exists a feasible transition that moves it to $\mathcal{P}_{N-1}$.

$$\begin{aligned}\mathcal{P}_N = \Bigg\{ \boldsymbol{x} \;\Bigg|\; &\boldsymbol{x} \notin \bigcup_{k=0}^{N-1} \mathcal{P}_k, \\ &\exists \tau, \boldsymbol{x}' \text{ s.t. } (\boldsymbol{x}, \tau, \boldsymbol{x}') \in \mathcal{F}, \boldsymbol{x}' \in \mathcal{P}_{N-1} \Bigg\}\end{aligned} \tag{18}$$

Computing capture basins based on the definition above is quite impractical, because it requires enumeration over the set $\mathcal{F}$, which is 11 dimensional in our formulation. The following alternative but equivalent definition is more suited to fast computation of capture basins.

$$\begin{aligned}\mathcal{P}_N = \Bigg\{ \boldsymbol{x} = \begin{bmatrix}\hat{\boldsymbol{q}}_{\rm swg} \\ \hat{\boldsymbol{p}}_{\rm icp}\end{bmatrix} \;\Bigg|\; &\boldsymbol{x} \notin \bigcup_{k=0}^{N-1} \mathcal{P}_k, \\ &\exists \begin{bmatrix}\tau \\ \hat{\boldsymbol{q}}'_{\rm swg}\end{bmatrix} \in \mathcal{X}(\hat{\boldsymbol{p}}_{\rm icp}) \text{ s.t. } \tau_{\min}(\hat{\boldsymbol{q}}_{\rm swg}, \hat{\boldsymbol{q}}'_{\rm swg}) \le \tau \Bigg\}\end{aligned} \tag{19}$$

where $\mathcal{X}_N(\hat{\boldsymbol{p}}_{\rm icp})$ is a set defined as follows.

$$\begin{aligned}\mathcal{X}_N(\hat{\boldsymbol{p}}_{\rm icp}) = \Bigg\{ \begin{bmatrix}\tau \\ \hat{\boldsymbol{q}}'_{\rm swg}\end{bmatrix} \;\Bigg|\; &\exists \hat{\boldsymbol{p}}'_{\rm icp} \text{ s.t. } \begin{bmatrix}\hat{\boldsymbol{q}}'_{\rm swg} \\ \hat{\boldsymbol{p}}'_{\rm icp}\end{bmatrix} \in \mathcal{P}_{N-1}, \\ &\hat{\boldsymbol{p}}_{\rm icp} \in \mathcal{R}_{\rm icp}(\boldsymbol{\mu}(\hat{\boldsymbol{q}}'_{\rm swg}, \hat{\boldsymbol{p}}'_{\rm icp}), \tau) \Bigg\}\end{aligned} \tag{20}$$

Algorithm 1 Capture basin computation

1: $\mathcal{P}_0 \leftarrow [\mathcal{R}_{\rm swg}] \times [\mathcal{R}_{\rm cop}]$
2: $N \leftarrow 1$
3: **while** $\mathcal{P}_{N-1} \neq \varnothing$ **do**
4: $\quad \mathcal{P}_N \leftarrow \varnothing$
5: $\quad$ **for** $(\boldsymbol{q}'_{\rm swg}, \boldsymbol{p}'_{\rm icp}) \in \mathcal{P}_{N-1}$ **do**
6: $\quad\quad \boldsymbol{\mu} \leftarrow \boldsymbol{\mu}(\boldsymbol{q}'_{\rm swg}, \boldsymbol{p}'_{\rm icp})$
7: $\quad\quad$ **for** $\boldsymbol{p}_{\rm icp}$ **do**
8: $\quad\quad\quad \tau \leftarrow \textsc{DurationBound}(\boldsymbol{\mu}, \boldsymbol{p}_{\rm icp})$
9: $\quad\quad\quad \mathcal{X}_N(\boldsymbol{p}_{\rm icp}) \leftarrow \mathcal{X}_N(\boldsymbol{p}_{\rm icp}) \cup (\tau, \hat{\boldsymbol{q}}'_{\rm swg})$
10: $\quad\quad$ **end for**
11: $\quad$ **end for**
12: $\quad$ **for** $\boldsymbol{p}_{\rm icp}$ **do**
13: $\quad\quad \mathcal{R}_{\rm swg} \leftarrow \{\hat{\boldsymbol{q}}_{\rm swg} \,|\, \exists(\tau, \hat{\boldsymbol{q}}'_{\rm swg}) \in \mathcal{X}_N(\boldsymbol{p}_{\rm icp})\}$
14: $\quad\quad \mathcal{P}_N \leftarrow \mathcal{P}_N \cup ((\mathcal{R}_{\rm swg} \times \boldsymbol{p}_{\rm icp}) \backslash \mathcal{P}_{N-1})$
15: $\quad$ **end for**
16: $\quad N \leftarrow N + 1$
17: **end while**

The equivalence of these two alternative definitions of capture basins is fairly straightforward, and therefore its proof is omitted. Based on this alternative definition, one can compute capture basins by enumerating over much smaller dimensional space. A concrete algorithm is presented in the next subsection.

B. Computation of Viable Capture Basin based on the Discretization of States

There are a number of factors that make analytical calculation of viable capture basins a difficult task. First, the rotational transformations in (5) and (7), the exponential relationship between the ICP and the step duration in (7), and the relationship between the minimum step duration and the swing foot configurations in (16) are all nonlinear. Second, the feasible swing foot region $\mathcal{R}_{\rm swg}$ is non-convex in general. In this paper, we propose a method for approximate computation of viable capture basins that is based on the discretization of space. Let us introduce a set of grid points uniformly distributed over a bounded rectangular region expressed as $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \times [\theta_{\min}, \theta_{\max}]$. For a region $\mathcal{R} \in {\rm SE}(2)$, the set of grid points included in this region is denoted by $[\mathcal{R}]$. The pseudocode of an algorithm that computes capture basins based on (19) and (20) is shown in Algorithm 1. The N-step capture basins are computed recursively, starting from $N = 0$. If $\mathcal{P}_N$ results to be the empty set, the algorithm terminates. For each N, the computation of $\mathcal{P}_N$ consists of two phases. In the first phase (Lines 5-11), $\mathcal{X}_N(\hat{\boldsymbol{p}}_{\rm icp})$ is computed for each $\hat{\boldsymbol{p}}_{\rm icp}$ based on (20). The function DURATIONBOUND returns the maximum value of τ that satisfies $\hat{\boldsymbol{p}}_{\rm icp} \in \mathcal{R}_{\rm icp}(\boldsymbol{\mu}, \tau)$. In the second phase (Lines 12-15), $\mathcal{P}_N$ is computed based on (19).

IV. STEP ADAPTATION THAT UTILIZES PRE-COMPUTED CAPTURE BASINS

This section describes a filter that modifies the landing configuration, step duration, and the ICP at landing based on precomputed capture basins. The filtering procedure is executed at a constant interval (e.g., 5[ms]), and at every cycle, a series of steps shown in Fig. 3 is executed. Inputs to the filter are: the current state $\boldsymbol{x}(t) = (\hat{\boldsymbol{q}}_{\rm swg}(t), \hat{\boldsymbol{p}}_{\rm icp}(t))$, the reference step duration τ, and the target state $\boldsymbol{x}' = (\hat{\boldsymbol{q}}'_{\rm swg}, \hat{\boldsymbol{p}}'_{\rm icp})$ In the first step, the filter checks if the target state is reachable from the current state without timing adjustment; that is, $(\boldsymbol{x}(t), \tau, \boldsymbol{x}') \in \mathcal{F}$. If this condition is not met, in the second step, the filter solves the following minimization problem to find a modified step duration $\tilde{\tau}$.

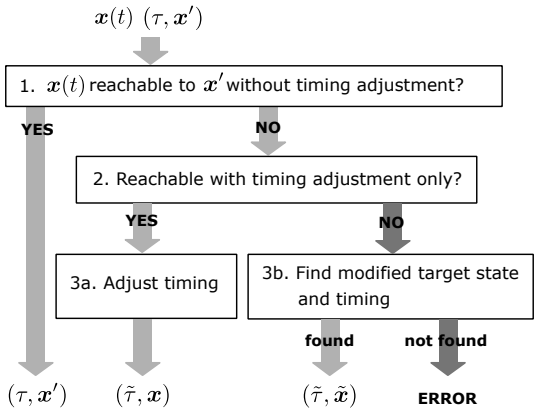


Fig. 3. Processing flow of foothold and timing adjustment filter.

$$\begin{aligned} &\text{minimize} && |\tilde{\tau} - \tau| && (21)\\ &\text{subject to} && (\boldsymbol{x}(t), \tau', \boldsymbol{x}') \in \mathcal{F} && (22) \end{aligned}$$

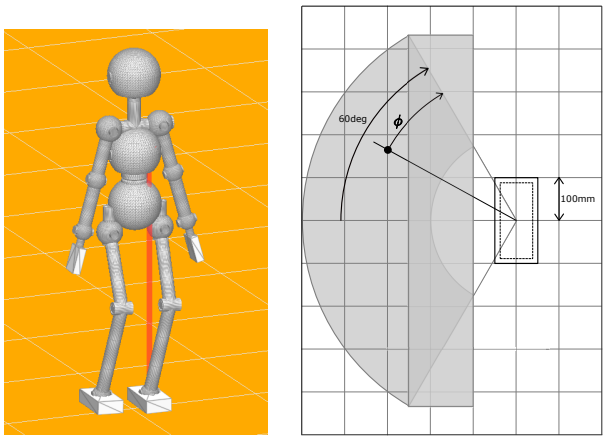
Note that up to the second step, the precomputed information is not used. If the second step yields no solution, then in the third step, the filter attempts to find a combination of a modified target state and a modified step duration that is reachable from the current state.

$$\begin{aligned} &\text{find} && N, \tilde{\tau}, \tilde{\boldsymbol{x}}' \in \mathcal{P}_N\\ &\text{minimize} && J(\tilde{\boldsymbol{x}}', \tilde{\tau})\\ &\text{subject to} && (\boldsymbol{x}(t), \tilde{\tau}, \tilde{\boldsymbol{x}}') \in \mathcal{F} \end{aligned}$$

Namely, among all states in the N-step capture basins, it finds a modified target state $\tilde{\boldsymbol{x}}'$ that has a feasible transition from the current state and minimizes the cost function J defined as follows.

$$\begin{aligned} &J(\tilde{\boldsymbol{x}}', \tilde{\tau}') = \\ &w_{\rm swg} \|\tilde{\hat{\boldsymbol{q}}}'_{\rm swg} - \hat{\boldsymbol{q}}'_{\rm swg}\| + w_{\rm icp} \|\tilde{\hat{\boldsymbol{p}}}'_{\rm icp} - \hat{\boldsymbol{p}}'_{\rm icp}\| + w_\tau |\tilde{\tau} - \tau| \end{aligned}$$

The objective of this cost function is to find a modified target state that is close to the reference target state as much as possible. Here, the weights $w_{\rm swg}$, $w_{\rm icp}$, and w_τ are used for specifying the priority of the landing configuration, the ICP after landing, and the timing of landing, respectively. If the above optimization problem has no solution, this implies that falling is inevitable, so the filter outputs error.



(a) (b)

Fig. 4. (a) Simulation model of robot, (b) Swing foot region

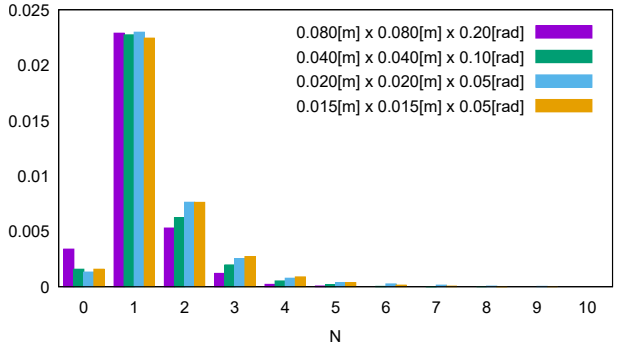
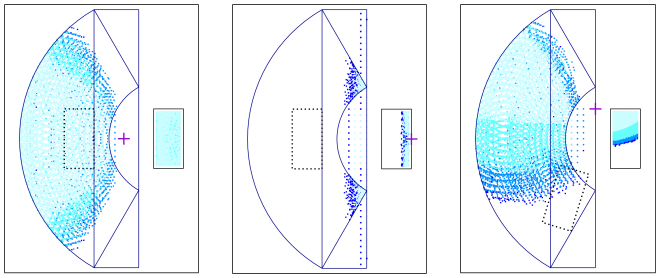


Fig. 5. Volume of capture basins in different resolutions

V. Simulation Experiments

A. Capturability Analysis Results

This section shows the results of capture basin computation using the proposed method. The method was implemented into a single-threaded program written in C++ and run on a computer with AMD Ryzen 9 5950 processor. The dynamical model for capturability analysis is based on a simulation model of a humanoid robot shown in Fig. 4(a). The robot has approximately the size of a young human; its total mass is 44[kg] and the height of the center-of-mass is 0.8[m]. The dimensions of the foot is 0.2×0.1[m]. The maximum horizontal and rotational speed of the swing foot are set as $v_{\max} = 2.0$[m/s] and $w_{\max} = 2.0$[rad/s], respectively. The feasible swing foot region and the feasible CoP region are shown in Fig. 4(b). The swing foot region is given by the union of a fan-like region and a rectangular region. If the swing foot position is inside the fan-like region and the angular margin (see the figure) is ϕ, the rotation angle of the foot may take any value within the range $\pm\phi$. If it is chosen inside the rectangle, on the other hand, the rotation angle must be 0. This specification of the swing foot region satisfies the condition (12). The ranges of the grid are set as $[x_{\min}, x_{\max}] = [-0.5, 0.5]$[m], $[y_{\min}, y_{\max}] = [-0.1, 0.5]$[m], and $[\theta_{\min}, \theta_{\max}] = [-0.5, 0.5]$[rad].



(a) $\hat{\boldsymbol{p}}_{\mathrm{swg}} = [0, 0.3, 0]^{\mathsf{T}}$, $\hat{\boldsymbol{p}}_{\mathrm{icp}} = [0, 0.15]^{\mathsf{T}}$ (b) $\hat{\boldsymbol{p}}_{\mathrm{swg}} = [0, 0.3, 0]^{\mathsf{T}}$, $\hat{\boldsymbol{p}}_{\mathrm{icp}} = [0, -0.05]^{\mathsf{T}}$ (c) $\hat{\boldsymbol{p}}_{\mathrm{swg}} = [-0.2, 0.2, -0.3]^{\mathsf{T}}$, $\hat{\boldsymbol{p}}_{\mathrm{icp}} = [0.1, 0.1]^{\mathsf{T}}$

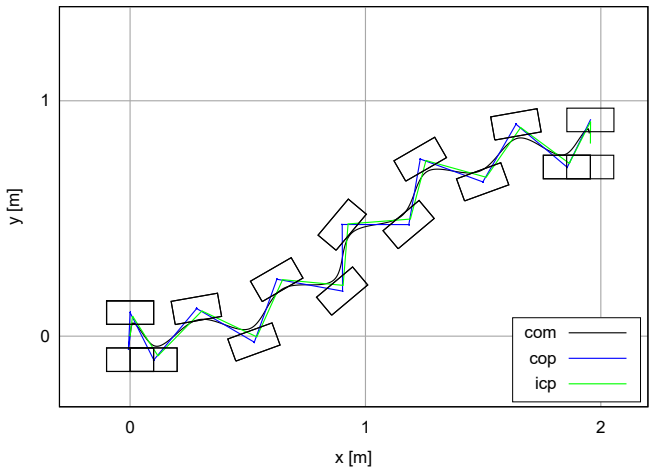
Fig. 6. Visualization of capture regions

Capture basins are computed with different combinations of grid resolutions and the results are compared. Figure 5 shows the volume of computed capture basins. Here, the volume of a capture basin is calculated by counting the number of discrete grid points and multiplying it by the volume of the neighborhood of a single grid point. Common to all resolution settings, the volume of $\mathcal{P}_1$ is the largest, while the size of $\mathcal{P}_N$ with $N \geq 2$ decreases gradually. Moreover, no notable change in volume is observed beyond 0.02[m] $\times$ 0.02[m] $\times$ 0.05[rad]; we therefore chose this resolution setting for the simulation experiments reported in the next subsection. For this resolution setting, the total data size is approximately 50[MB], which means storing the result into a mass storage device as well as loading it into memory can be done with little storage cost. Moreover, the computation time was approximately 111[s].

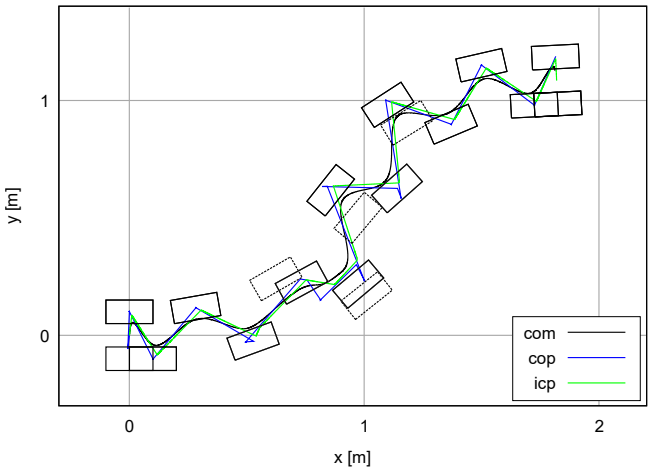
The visualization of the computed capture basins is shown in Figs. 6(a)-(d). Direct visualization of capture basins is difficult because of its high dimensionality. Instead, for a given state $\boldsymbol{x}$, all transitions satisfying $(\boldsymbol{x}, \tau, \boldsymbol{x}') \in \mathcal{F}$, $\boldsymbol{x}' \in \mathcal{P}_N$ for some N are enumerated, and their landing positions and CoPs are plotted on the x-y plane. The color of plotted points indicate the value of N. It can be observed in the figures that distribution of capturable landing positions and CoPs varies greatly depending on the value of $\boldsymbol{x}$. For example, in Fig. 6(b), the ICP is located in the opposite side of the swing foot. In this situation, the feasible CoPs are distributed close to the ICP for minimizing the deviation of the ICP from the support foot, and the feasible landing positions are also distributed close to the ICP.

B. Evalulation of Disturbance Response during Walking

This section shows the results of dynamical walking simulation experiments. Multi-body simulation using a full DoF model of the robot shown in Fig. 4(a) is executed on Choreonoid [14] with AIST simulator plugin. The time resolution of simulation is set as 1[ms]. Reference foot placements consisting of 12 steps are specified manually; The first and the last steps have the stride length of 0.1[m], while each of the ten other steps have 0.2[m] stride and 10[deg] turning angle (five left turns and five right turns). The reference step duration of each step is 0.7[s]. The reference



Case (a) No disturbance



Case (b) 15[Ns] impulse in the x direction at $t = 3.0$[s], 15[Ns] impulse in the y direction at $t = 4.0$[s]

Fig. 7. Simulation results

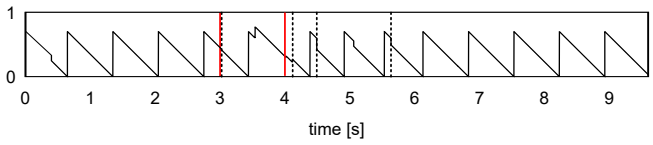


Fig. 8. History of time-to-landing

ICP trajectory is generated by using the method proposed in [13]. The low-level control loop is executed in 1[ms] update period. In the low-level control, desired foot configurations and desired CoM position are converted to desired joint angles by inverse kinematics computation and provided to the PD controllers attached to the actuated joints. In addition, ground reaction force control for reference CoP tracking is also performed. The update cycle of the adjustment filter is set as 5[ms].

In the first experiment, walking simulation is conducted without applying any external disturbance. The footsteps and the centroidal trajectories of the simulation result is shown in Fig. 7(a). During this simulation, the filter performed no timing adjustment or step adjustment. This confirms that the filter does not make any unnecessary adjustment.

In the second experiment, two impulsive disturbances with the magnitude of 15[Ns] are applied to the base link of the robot, the first impulse is applied at $t = 3$[s] in the x direction, and the second one is applied at $t = 4$[s] in the y direction. A single impulse amounts to approximately 0.34[m/s] change of velocity, or 0.1[m] change of ICP. The footsteps and the trajectories are shown in Fig. 7(b). As we can see from this figure, the robot is able to recover from perturbation caused by the two impulsive disturbances. Moreover, Fig. 8 shows the history of time-to-landing (TTL). Several discontinuous changes of TTL observed in this figure indicate that timing adjustment is performed at these instants. The instants of disturbance injection are marked by red lines, whereas the instants of step adjustment are marked by dotted lines. After the first disturbance, a single step adjustment is performed, while after the second disturbance, three consecutive step adjustments are performed. In Fig. 7(b), reference landing configurations before the step adjustment are drawn with dashed lines. More detailed movement of the robot in these simulation experiments can be seen in the attached video.

VI. Conclusion

This paper presented a footstep and timing adjustment controller that makes use of precomputed capture basins. Simulation experiments using a full DoF model of a humanoid robot confirmed that the proposed method is capable of generating fall-avoiding movements involving multiple steps. Our ongoing work includes testing the proposed controller on a real humanoid robot. Enriching the repertoire of fall-avoiding movements, such as utilizing center-of-mass height variation and angular momentum is also an interesting direction.

Acknowledgement

This research was supported by Nagamori Foundation research grant.

References

[1] K. Nishiwaki, S. Kagami: “Online Walking Control System for Humanoids with Short Cycle Pattern Generation”, The International Journal of Robotics Research, Vol.28, No.6, pp.729-742, 2009.

[2] B. Stephens: “Humanoid Push Recovery”, 2007 IEEE-RAS 7th International Conference on Humanoid Robots, pp. 589-595, 2007.

[3] J. Pratt, J. Carff, S. Drakunov and A. Goswami: “Capture Point: A Step Toward Humanoid Push Recovery”, 2006 IEEE-RAS 6th International Conference on Humanoid Robots, pp. 200-207, 2006.

[4] T. Koolen, T. de Boer, J. Rebula, A. Goswami and J. Pratt: “Capturability-based Analysis and Control of Legged Locomotion, Part 1: Theory and Application to Three Simple Gait Models”, The International Journal of Robotics Research, Vol.31, Issue 9, pp.1094-1113, 2012.

[5] J. Pratt, T. Koolen, T. De Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson and P. Neuhaus: “Capturability-based Analysis and Control of Legged Locomotion, Part 2: Application to M2V2, a Lower-body Humanoid”, The International Journal of Robotics Research, Vol.31, Issue 10, pp.1117-1133, 2012.

[6] M. Khadiv, A. Herzog, S. A. A. Moosavian and L. Righetti, "Step Timing Adjustment: A Step Toward Generating Robust Gaits", IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), pp.35-42, 2016.
[7] T. Yamamoto, T. Sugihara: "Foot-Guided Control of a Biped Robot through ZMP Manipulation", Advanced Robotics, Vol.34, No.21-22, pp.1472-1489, 2020.
[8] G. Park, J. H. Kim, J. Jo, Y. Oh: "Lyapunov-based Approach to Reactive Step Generation for Push Recovery of Biped Robots via Hybrid Tracking Control of DCM", IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3504-3509, 2020.
[9] S. Caron, B. Mallein: "Balance Control Using Both ZMP and COM Height Variations: A Convex Boundedness Approach", IEEE International Conference on Robotics and Automation, pp. 1779-1784, 2018.
[10] G. Kim, H. Kuribayashi, Y. Tazaki and Y. Yokokohji: "Omni-Directional Fall Avoidance of Bipedal Robots with Variable Stride Length and Step Duration", 2018 IEEE-RAS 18th International Conference on Humanoid Robots, pp. 718-724, 2018.
[11] 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", 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 239-246, 2001.
[12] 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", 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4489-4496, 2010.
[13] J. Englsberger, O. Christian and A. Alin: "Three-dimensional Bipedal Walking Control Using Divergent Component of Motion" IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013 .
[14] S. Nakaoka: "Choreonoid: Extensible virtual robot environment built on an integrated GUI framework", IEEE/SICE International Symposium on System Integration (SII), pp. 79-85, 2012.