2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Lyapunov-based Approach to Reactive Step Generation for Push Recovery of Biped Robots via Hybrid Tracking Control of DCM

Gyunghoon Park[1], Jung Hoon Kim[2], Joonhee Jo[1,3], and Yonghwan Oh[1]

Abstract— **This paper addresses reactive generation of step time and location of biped robots for balance recovery against a severe push. Key idea is to reformulate the balance recovery problem into a tracking problem for "hybrid" inverted pendulum model of the biped, where taking a new step implicitly yields a discrete jump of the tracking error. This interpretation offers a Lyapunov-based approach to reactive step generation, which is possibly more intuitive and easier to analyze than large-scaled or nonlinear optimization-based approaches. With the continuous error dynamics for the divergent component of motion (DCM), our strategy for step generation is to decrease the "post-step" Lyapunov level for DCM error at each walking cycle, until it eventually becomes smaller than a threshold so that no more footstep needs to be adjusted. We show that implementation of this idea while obeying physical constraints can be done by employing a hybrid tracking controller (together with a reference model) as our reactive step generator, consisting of a simple DCM-based continuous controller and a small-sized quadratic programming-based discrete controller. The validity of the proposed scheme is verified by simulation results.**

I. Introduction

We have witnessed a vast of advances in the field of biped robots over past decades, especially on their abilities to operate stably against uncertain environments. In particular, a tremendous amount of attentions have been paid to recover the biped's balance against external push by generating appropriate step time and location on-line, studied under the name of "reactive step generation".

Initial research efforts have focused on adjusting the step location, with the step duration fixed *a priori*. Interpreting dynamical motion of center of mass (CoM) via an inverted pendulum model (IPM), the authors of [1] proposed the so-called "capture point" (or its equivalent definition "divergent component of motion (DCM)" presented later in [3]) as a key concept of the balance recovery in a spatial sense, which can be viewed as an ideal location of next footstep to stop. The advance in computational power in turn has led to the emergence of optimization-based approaches to online step generation, mainly in two directions. One distinctive stream of research has employed the model predictive control (MPC) as a step generator [4]–[6], where the step location is set as a control input and then the best among possible candidates is derived in the sense of cost minimization. On the other hand, while the conventional MPC framework embeds the stacked system dynamics into an optimization problem, another approach has been proposed in [7], [8] by putting the analytic solution of the DCM into the formulation of an optimization problem instead of its dynamics. More recently, these two philosophies have been extended to a more general problem where the step time remains not pre-determined, encouraged by a rule-of-thumb that step time adjustment would significantly improve robustness of the biped [9], [10]. Yet as introducing step time as an additional variable to be chosen results in complexity of the optimization problem, these optimization-based approaches often suffer from heavy computational loads or cannot guarantee the feasibility.

At this point, two important questions arise: (a) in urgent situations (e.g., in the presence of a strong push), is it really important to compute the "optimal" step time and location at the expense of heavy computation?; moreover, (b) how can we make sure whether no more footstep is needed for regaining the balance, especially in a closed-loop point of view? As yet another attempt to answer these questions, in this paper we present a Lyapunov-based approach to reactive step generation for balance recovery. Our key idea is to recast the balance recovery problem as a tracking problem for the "hybrid" IPM in the hybrid systems framework [11], where taking a step implicitly yields a discrete jump of the tracking error between the IPM and its nominal counterpart. This allows us to represent a condition of where and when to step for balance recovery as a Lyapunov stability criterion for the DCM tracking error: that is, the balance recovery is achieved by regulating a Lyapunov function of the error dynamics to zero. As an illustration of the idea, we propose a reactive step generator (RSG) that consists of three components. The first is a hybrid reference model that mimics the nominal behavior of the hybrid IPM. Then a DCM-based tracking controller is constructed to stabilize the continuous DCM error dynamics in the sense of Lyapunov (where the zero-moment point and the rate-of-change of angular momentum are used as continuous control inputs). Finally, we employ the solution of a small-sized quadratic programming (QP) as a discrete control input, which finds the best step location for minimizing the "post-footstep" level of the Lyapunov function and runs only when increased amount of the Lyapunov function during the current walking cycle reaches a threshold (by which the step time is directly determined). We believe that, compared with nonlinear or large-scaled optimization approaches, our Lyapunov-based approach could relax computational burdens as well as bring a clear way of understanding the underlying

*This work was supported by the KIST Institutional Program (Project No. 2E30280)

[1]G. Park, J. Jo, and Y. Oh are with Center for Intelligent and Interactive Robotics, Korea Institute of Science and Technology, Republic of Korea {gyunghoon.p,jhjo,oyh}@kist.re.kr

[2]J. H. Kim is with Department of Electrical Engineering, Pohang University of Science and Technology (POSTECH), Republic of Korea junghoonkim@postech.ac.kr

[3] J. Jo is also with Department of HCI & Robotics, University of Science and Technology (UST), Daejeon, 305-350, Republic of Korea

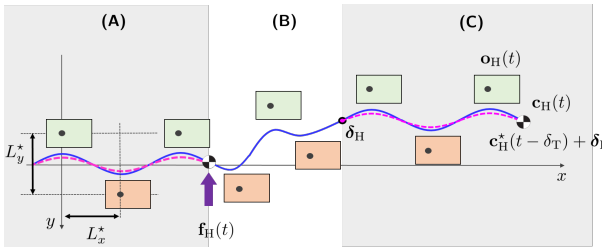


Fig. 1. Graphical interpretation of balance recovery problem. Detailed explanations on each phase in the figure are: (A) Before the robot is pushed, the actual CoM $\mathbf{c}_{\mathrm{H}}(t)$ (blue solid) is kept close to the nominal one $\mathbf{c}^{\star}_{\mathrm{H}}(t)$ (orange dashed); (B) In the transient period caused by a push, $\mathbf{c}_{\mathrm{H}}(t)$ is forced to move far away from its nominal trajectory; (C) After recovering the balance, the robot's CoM $\mathbf{c}_{\mathrm{H}}(t)$ tracks $\mathbf{c}^{\star}_{\mathrm{H}}(t-\delta_{\mathrm{T}})+\boldsymbol{\delta}_{\mathrm{H}}$ again.

rationale behind DCM-based balance recovery. The present work is an extension of the authors' previous works [12], [13] in which a conceptual sketch of the basic idea was drawn in the sagittal plane. We here extend the core into general 2D walking scenarios, for which the Lyapunov-based argument and QP formalism are brought together into the picture.

II. Formulation of Balance Recovery Problem

Consider a linear inverted pendulum model (LIPM) with a constant height c_z of the center of mass (CoM) and the rate-of-change of angular momentum around the CoM:

$$\ddot{\mathbf{c}}_{\mathrm{H}} = \frac{g}{c_z}(\mathbf{c}_{\mathrm{H}} - \mathbf{p}_{\mathrm{H}}) + \frac{1}{mc_z}\begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix}\dot{\mathbf{h}}_{\mathrm{H}} + \frac{1}{m}\mathbf{f}_{\mathrm{H}} \tag{1}$$

where $\mathbf{c}_{\mathrm{H}}$ and $\mathbf{p}_{\mathrm{H}}$ are the CoM and the zero-moment point (ZMP), respectively, and $\dot{\mathbf{h}}_{\mathrm{H}}$ is the rate-of-change of angular momentum.[1] Total mass of the robot and the gravitational constant are denoted by m and g, respectively. The pushing force $\mathbf{f}_{\mathrm{H}}$ is assumed to be an impulsive signal with an impulse $\mathbf{i}_{\mathrm{H}}$ such that $\mathbf{f}_{\mathrm{H}}(t) = 0$ for all $t \notin [t_{\mathrm{f}}, t_{\mathrm{f}} + \Delta_{\mathrm{f}}]$ and $\int_{t_{\mathrm{f}}}^{t_{\mathrm{f}}+\Delta_{\mathrm{f}}} \mathbf{f}_{\mathrm{H}}(s)\mathrm{d}s = \mathbf{i}_{\mathrm{H}}$ in which t_{f} and $\Delta_{\mathrm{f}} \geq 0$ denote time moment and (possibly small) duration of push, respectively. In planning phase we assume that the biped has massless legs and finite-sized feet.

For future use some notations are introduced below. One walking cycle is composed of a double support phase (DSP) and a single support phase (SSP), with nominal time periods $T^{\star}_{\mathrm{DSP}}$ and $T^{\star}_{\mathrm{SSP}}$ for each. Also let $T^{\star} := T^{\star}_{\mathrm{DSP}} + T^{\star}_{\mathrm{SSP}}$. The j-th "step time" s_j represents the time moment at which the j-th walking cycle begins. On the other hand, the "step location" $\mathbf{o}_{\mathrm{H}}(t)$ is a right-continuous and piece-wise constant function that indicates the horizontal position of the ankle of the current stance foot.

In view of the LIPM (1), balance recovery is an ability to make the robot's CoM $\mathbf{c}_{\mathrm{H}}$ and footsteps $\mathbf{o}_{\mathrm{H}}$ track their (disturbance-free) nominal counterparts $\mathbf{c}^{\star}_{\mathrm{H}}$ and $\mathbf{o}^{\star}_{\mathrm{H}}$ after a severe push is applied, without significant loss of balance in transient. In other words, it is expected that, with a small $\epsilon > 0$,

$$\left\|\mathbf{c}_{\mathrm{H}}(t) - \left(\mathbf{c}^{\star}_{\mathrm{H}}(t-\delta_{\mathrm{T}}) + \boldsymbol{\delta}_{\mathrm{H}}\right)\right\| < \epsilon, \tag{2a}$$

$$\mathbf{o}_{\mathrm{H}}(t) = \mathbf{o}^{\star}_{\mathrm{H}}(t-\delta_{\mathrm{T}}) + \boldsymbol{\delta}_{\mathrm{H}} \tag{2b}$$

holds for each SSP $t \in [s_j + T^{\star}_{\mathrm{DSP}}, s_{j+1})$, $j \geq N$, with sufficiently large N and time and space shifts δ_{T} and $\boldsymbol{\delta}_{\mathrm{H}}$. (See also Fig. 1(b) for a graphical interpretation of (2).) The nominal trajectory $(\mathbf{c}^{\star}_{\mathrm{H}}, \mathbf{o}^{\star}_{\mathrm{H}})$ under consideration is assumed to be generated by a nominal LIPM

$$\ddot{\mathbf{c}}^{\star}_{\mathrm{H}} = \frac{g}{c_z}(\mathbf{c}^{\star}_{\mathrm{H}} - \mathbf{o}^{\star}_{\mathrm{H}}) \tag{3}$$

with nominal values of stride length $L^{\star}_x$, foot width $L^{\star}_y$, and step duration $T^{\star} = T^{\star}_{\mathrm{DSP}} + T^{\star}_{\mathrm{SSP}}$.

In this work we are interested in simultaneous reactive generation of step information $(s_j, \mathbf{o}_{\mathrm{H}})$ as well as $(\mathbf{p}_{\mathrm{H}}, \dot{\mathbf{h}}_{\mathrm{H}})$, to recover the balance of the robot in the sense of achieving (2). (Throughout this paper, a generating dynamics of such $(s_j, \mathbf{o}_{\mathrm{H}}, \mathbf{p}_{\mathrm{H}}, \dot{\mathbf{h}}_{\mathrm{H}})$ is called a "reactive step generator (RSG)".) The main difficulties in resolving this problem are: (a) there is no priori knowledge on $(\delta_{\mathrm{T}}, \boldsymbol{\delta}_{\mathrm{H}})$ (so that it is not possible to directly utilize $\mathbf{c}^{\star}_{\mathrm{H}}(t-\delta_{\mathrm{T}}) + \delta_{\mathrm{H}}$ as a reference for $\mathbf{c}_{\mathrm{H}}(t)$); and (b) how the balance recovery is possibly done by taking a footstep remains unclear in (2). We will see below that these issues can be efficiently dealt with as the hybrid systems framework comes into the picture.

III. Introduction to Hybrid LIPM

This section is devoted to represent the LIPM (1) in the hybrid systems framework [11]. A hybrid system is a dynamical system that admits continuous flow and discrete jump of state variable [11], having a general expression[2]

$$\text{Flow dynamics:} \quad \dot{\boldsymbol{\chi}} = F(\boldsymbol{\chi}, \mathbf{u}_{\mathrm{c}}, t), \quad \forall(\boldsymbol{\chi}, \boldsymbol{\eta}) \notin \mathcal{D}, \tag{4a}$$

$$\text{Jump dynamics:} \quad \boldsymbol{\chi}^{+} = G(\boldsymbol{\chi}, \mathbf{u}_{\mathrm{d}}, t), \quad \forall(\boldsymbol{\chi}, \boldsymbol{\eta}) \in \mathcal{D} \tag{4b}$$

where $\boldsymbol{\chi}$ is the state, $\mathbf{u}_{\mathrm{c}}$ and $\mathbf{u}_{\mathrm{d}}$ are continuous and discrete inputs, $\boldsymbol{\eta} \in \mathbb{R}^{\nu}$ is an external signal (or a controller state), F and G are flow and jump maps, and $\mathcal{D}$ is a jump set.

Underlying reason behind the conversion of the LIPM is to merge the step time s_j and location $\mathbf{o}_{\mathrm{H}}$ "implicitly" into a system model, as some components of (4). This actually can be done by focusing more on the relative distance between the CoM and the footstep at each step time $t = s_j$. Indeed, by continuity of $\mathbf{c}_{\mathrm{H}}(t)$ on t, the difference $\mathbf{c}_{\mathrm{H}} - \mathbf{o}_{\mathrm{H}}$ experiences an instantaneous jump as $\mathbf{c}_{\mathrm{H}}(s_j) - \mathbf{o}_{\mathrm{H}}(s_j) = \mathbf{c}_{\mathrm{H}}(s_j^-) - \mathbf{o}_{\mathrm{H}}(s_j^-) - \left(\mathbf{o}_{\mathrm{H}}(s_j) - \mathbf{o}_{\mathrm{H}}(s_j^-)\right)$. More precisely, the hybrid systems framework offers a compact expression of this phenomenon, in a natural coordinate for $(\mathbf{c}_{\mathrm{H}}, \dot{\mathbf{c}}_{\mathrm{H}})$

$$\boldsymbol{\xi} = (\boldsymbol{\xi}_1, \boldsymbol{\xi}_2) := \left(\mathbf{c}_{\mathrm{H}} - \mathbf{o}_{\mathrm{H}}, \dot{\mathbf{c}}_{\mathrm{H}}\right), \tag{5}$$

as follows: $\boldsymbol{\xi}_1^{+} = \boldsymbol{\xi}_1 - \mathbf{u}_{\mathrm{d}}$, $\forall(\boldsymbol{\chi}, \boldsymbol{\eta}) \in \mathcal{D}$ where the discrete input $\mathbf{u}_{\mathrm{d}}$ is defined as

$$\mathbf{u}_{\mathrm{d}}(t) := \mathbf{o}_{\mathrm{H}}(t) - \mathbf{o}_{\mathrm{H}}(s_{j-1}), \quad \forall t \in [s_j, s_{j+1})$$

while the step time s_j is automatically determined as the time when $(\boldsymbol{\chi}, \boldsymbol{\eta})$ encounters $\mathcal{D}$. A similar computation brings a

[1] Hereinafter, the subscript H is used for a two-dimensional vector in the horizontal plane that is denoted by, for instance, $\mathbf{c}_{\mathrm{H}} = (c_x, c_y)$.

[2] Although the solution of a hybrid system (4) is formally expressed as $\boldsymbol{\chi}(t, j)$ to emphasize continuous and discrete time segments, for brevity we use $\boldsymbol{\chi}(t)$ instead of $\boldsymbol{\chi}(t, j)$ with no multiple jump at every t guaranteed by the proposed algorithm.

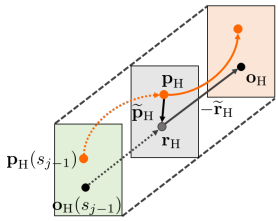


Fig. 2. Concept of moving window $\mathcal{W}(\mathbf{r}_\mathrm{H})$ (gray rectangle) for a DSP lying on the support region (dashed area) with respect to two feet (green and orange rectangles)

hybrid system expression of the LIPM (1) (called hybrid LIPM), which will be presented in Subsection III-B.

A. Simplified ZMP Constraint with Moving Window

As a prerequisite for introducing the hybrid LIPM, we adopt the concept of "moving window" for the ZMP $\mathbf{p}_\mathrm{H}$ (similarly in the related works such as [14]) to simplify the ZMP constraint. The moving window is a convex region on the ground in which the ZMP is desired to remain, defined with an auxiliary point $\mathbf{r}_\mathrm{H}(t)$ as $\mathcal{W}(\mathbf{r}_\mathrm{H})$, where for $\mathbf{x}_\mathrm{H} \in \mathbb{R}^2$,

$$\mathcal{W}(\mathbf{x}_\mathrm{H}) := \{\overline{\mathbf{x}}_\mathrm{H} \in \mathbb{R}^2 : \mathbf{w}_\mathrm{H}^\mathrm{m} \preceq \overline{\mathbf{x}}_\mathrm{H} - \mathbf{x}_\mathrm{H} \preceq \mathbf{w}_\mathrm{H}^\mathrm{M}\}, \tag{6}$$

the symbol $\mathbf{a} \preceq \mathbf{b}$ indicates component-wise inequalities, and $\mathbf{w}_\mathrm{H}^\mathrm{m} \preceq \mathbf{w}_\mathrm{H}^\mathrm{M}$ are selected such that $\mathcal{W}(\mathbf{o}_\mathrm{H})$ is included in the foothold. (We here let $w_y := w_y^\mathrm{M} = -w_y^\mathrm{m} > 0$, for y-directional symmetry of the foot.) As is seen in Fig. 2, the representative $\mathbf{r}_\mathrm{H}(t)$ of the moving window lies on the line between sequential footsteps $\mathbf{o}_\mathrm{H}(s_{j-1})$ and $\mathbf{o}_\mathrm{H}(s_j)$, and moves ahead and stays on $\mathbf{o}_\mathrm{H}(s_j)$ during the SSP.

With $\mathbf{r}_\mathrm{H}$, one can represent the ZMP as

$$\mathbf{p}_\mathrm{H} = \mathbf{o}_\mathrm{H} + \widetilde{\mathbf{p}}_\mathrm{H} + \widetilde{\mathbf{r}}_\mathrm{H} \tag{7}$$

where $\widetilde{\mathbf{p}}_\mathrm{H} := \mathbf{p}_\mathrm{H} - \mathbf{r}_\mathrm{H}$ and $\widetilde{\mathbf{r}}_\mathrm{H} := \mathbf{r}_\mathrm{H} - \mathbf{o}_\mathrm{H}$ (among which the latter is vanished in the SSP by definition, and its explicit form will be presented shortly). A sufficient condition for $\mathbf{p}_\mathrm{H}$ to remain in the supporting region $\mathrm{conv}\{\mathcal{W}(\mathbf{o}_\mathrm{H}(s_{j-1})), \mathcal{W}(\mathbf{o}_\mathrm{H}(s_j))\}$ is then given by

$$\mathbf{w}_\mathrm{H}^\mathrm{m} \preceq \widetilde{\mathbf{p}}_\mathrm{H} \preceq \mathbf{w}_\mathrm{H}^\mathrm{M}, \tag{8}$$

which is regarded as a simplified ZMP constraint.

B. Hybrid LIPM with State and Input Constraints

The hybrid LIPM has the form (4) with the state

$$\boldsymbol{\chi} := (\boldsymbol{\xi}, \sigma) \in \mathbb{R}^4 \times \mathbb{R}^1 \tag{9}$$

where the variable $\sigma : \mathbb{R} \to \{-1, 1\}$ indicates which foot is the stance foot: $\sigma = -1$ if the right foot is the stance foot, while $\sigma = 1$ otherwise. By letting the continuous input and an additional perturbation term as

$$\mathbf{u}_\mathrm{c} := \left(\widetilde{\mathbf{p}}_\mathrm{H}, \dot{\mathbf{h}}_\mathrm{H}\right), \quad \boldsymbol{\Delta}_\mathrm{c} := \left(\widetilde{\mathbf{r}}_\mathrm{H}, \mathbf{f}_\mathrm{H}\right), \tag{10}$$

respectively, and by applying (7) to (1), we obtain

$$F := \begin{bmatrix} \mathbf{A}\boldsymbol{\xi} + \mathbf{B}\mathbf{u}_\mathrm{c} + \mathbf{E}\boldsymbol{\Delta}_\mathrm{c} \\ 0 \end{bmatrix}, \quad G := \begin{bmatrix} \boldsymbol{\xi} + \mathbf{F}\mathbf{u}_\mathrm{d} \\ -\sigma \end{bmatrix} \tag{11}$$

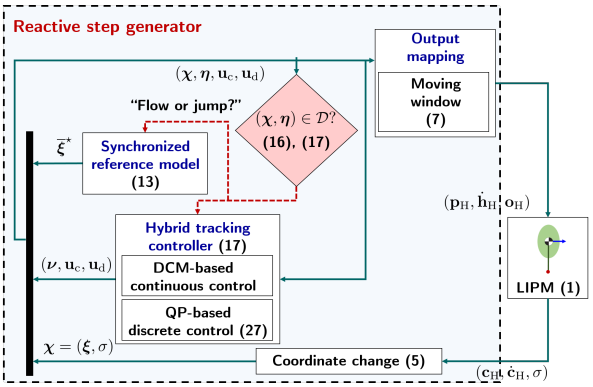


Fig. 3. Overall configuration of the proposed RSG (dashed block): The numbers indicate those of the equations in the main text.

and (hereinafter, $\mathbf{0}$ and $\mathbf{I}$ represent the zero and identity matrices with appropriate dimensions, respectively)

$$\mathbf{A} := \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\frac{g}{c_z}\mathbf{I} & \mathbf{0} \end{bmatrix}, \ \mathbf{B} := \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -\frac{g}{c_z}\mathbf{I} & \frac{1}{mc_z}\mathbf{H} \end{bmatrix}, \ \mathbf{H} := \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$
$$\mathbf{E} := \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -\frac{g}{c_z}\mathbf{I} & \frac{1}{m}\mathbf{I} \end{bmatrix}, \quad \mathbf{F} := \begin{bmatrix} -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

while selection of the jump set $\mathcal{D}$ is still upon the designer.

To keep the robot from losing the whole-body balance in the presence of physical limits of components, additional requirements should be dealt with in the step generation, including: (a) the (simplified) ZMP constraint (8) necessarily holds; (b) $\dot{\mathbf{h}}_\mathrm{H}$ and $\mathbf{u}_\mathrm{d} = \mathbf{o}_\mathrm{H}^+ - \mathbf{o}_\mathrm{H}$ need to be bounded; (c) overlapping two sequential footholds should be avoided; and (d) the y-directional position of $\mathbf{c}_\mathrm{H}$ remains between those of two end points of two feet. In our approach, all these are imposed as state and input constraints on (4) of the form

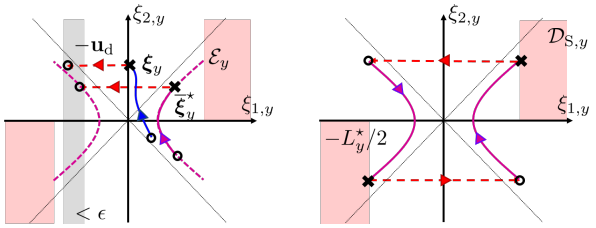
$$\mathbf{u}_\mathrm{c}^\mathrm{m} \preceq \mathbf{u}_\mathrm{c} \preceq \mathbf{u}_\mathrm{c}^\mathrm{M}, \quad \mathbf{u}_\mathrm{d}^\mathrm{m} \preceq \mathbf{u}_\mathrm{d} \preceq \mathbf{u}_\mathrm{d}^\mathrm{M}, \tag{12a}$$
$$\begin{bmatrix} 0 & \sigma \end{bmatrix} \mathbf{u}_\mathrm{d} + 2w_y \le 0, \quad \begin{bmatrix} 0 & -\sigma \end{bmatrix} \boldsymbol{\xi}_1 - w_y \le 0 \tag{12b}$$

where $\mathbf{u}_\mathrm{c}^\mathrm{m}$, $\mathbf{u}_\mathrm{c}^\mathrm{M}$, $\mathbf{u}_\mathrm{d}^\mathrm{m}$, and $\mathbf{u}_\mathrm{d}^\mathrm{M}$ are lower and upper bounds for the input variables, chosen in regards of robot's specification.

IV. Interpretation of Balance Recovery as Hybrid Tracking Problem

As highlighted at the end of Section II, closeness of two solutions $(\mathbf{c}_\mathrm{H}, \mathbf{o}_\mathrm{H})$ and $(\mathbf{c}_\mathrm{H}^\star, \mathbf{o}_\mathrm{H}^\star)$ is the essential in achieving the balance recovery while this is not straightforward since a lack of knowledge on the shifts $(\delta_\mathrm{T}, \boldsymbol{\delta}_\mathrm{H})$ in time and space. As a remedy for tackling this issue, we will recast the balance recovery problem as a "$(\delta_\mathrm{T}, \boldsymbol{\delta}_\mathrm{H})$-free" tracking problem for the hybrid LIPM (4). A particular emphasis here is placed on the time shift δ_T, noting that a naively-selected reference $\boldsymbol{\xi}_1^\star := \mathbf{c}_\mathrm{H}^\star - \mathbf{o}_\mathrm{H}^\star$ jumps at a pre-determined step time, whereas the actual state $\boldsymbol{\xi}_1$ will not so long as the step time needs to be adjusted. This motivates us to propose a "synchronized" reference model for $\boldsymbol{\xi}$, sharing the common condition for jump with the hybrid LIPM (4) (i.e., $(\boldsymbol{\chi}, \boldsymbol{\eta}) \in \mathcal{D}$).

For ease of explanation, it is noted in advance that the state $\boldsymbol{\eta}$ of the proposed RSG will take the form of $\boldsymbol{\eta} := (\vec{\boldsymbol{\xi}}^\star, \boldsymbol{\nu})$ where $\vec{\boldsymbol{\xi}}^\star$ and $\boldsymbol{\nu}$ stand for the state variables of



(a) Transient period (b) Steady-state period

Fig. 4. Phase portrait of hybrid LIPM (4) and synchronized reference model (13) in the $\boldsymbol{\xi}_y$-plane: (a) In transient caused by a push, the reference $\boldsymbol{\xi}_y^\star(t)$ (purple solid) jump at the same moment as $\boldsymbol{\xi}(t)$ (blue solid), while lying on the projection $\mathcal{E}_y$ (purple dashed) of $\mathcal{E}$ into $\boldsymbol{\xi}_y$-plane; (b) After the balance is recovered, the reference $\boldsymbol{\xi}^\star(t)$ behaves like the (time-shifted) ideal one $\boldsymbol{\xi}^\star(t-\delta_\mathrm{T})$, and does not jump until it encounters $\mathcal{D}_\mathrm{S}$.

the synchronized reference model (Subsection IV-A) and the hybrid tracking controller (Section V), respectively. Overall configuration of the proposed RSG is depicted in Fig. 3.

A. Synchronized Reference Model

The synchronized reference model of our interest is expected to preserve key properties of $\boldsymbol{\xi}^\star = (\mathbf{c}_\mathrm{H}^\star - \mathbf{o}_\mathrm{H}^\star, \dot{\mathbf{c}}_\mathrm{H}^\star)$ generated by the nominal LIPM (3), with its flow and jump conditions replaced by the same ones of (4) for synchronization. With this kept in mind, we design the synchronized reference model as a hybrid model

$$\dot{\overline{\boldsymbol{\xi}}}^\star = \mathbf{A}\overline{\boldsymbol{\xi}}^\star, \qquad \forall(\boldsymbol{\chi},\boldsymbol{\eta}) \notin \mathcal{D}, \tag{13a}$$

$$\overline{\boldsymbol{\xi}}^{\star+} = \overline{\boldsymbol{\xi}}^\star + \mathbf{F}\mathbf{u}_\mathrm{d}^\star, \qquad \forall(\boldsymbol{\chi},\boldsymbol{\eta}) \in \mathcal{D} \tag{13b}$$

with the nominal discrete input $\mathbf{u}_\mathrm{d}^\star(\overline{\boldsymbol{\xi}}^\star) := 2\overline{\boldsymbol{\xi}}_1^\star$, where the flow and jump dynamics are exactly the same as the hybrid systems expression of the $\boldsymbol{\xi}^\star$-dynamics. For future use, the jump set $\mathcal{D}$ is decomposed into

$$\mathcal{D} = \mathcal{D}_\mathrm{T} \bigcup \mathcal{D}_\mathrm{S} \tag{14}$$

where $\mathcal{D}_\mathrm{S} := \{(\boldsymbol{\chi},\boldsymbol{\eta}) : \mathrm{sign}(\overline{\xi}_{2,y}^\star)\overline{\xi}_{1,y}^\star \geq L_y^\star/2\}$ aims to, roughly speaking, mimicking the jump condition of $\boldsymbol{\xi}^\star$, and $\mathcal{D}_\mathrm{T}$ is used to adjust the step time during a transient caused by the external push. In addition, $\overline{\boldsymbol{\xi}}^\star(0)$ is chosen such that the "orbital energy" of (13) is conserved as the same amount as the nominal LIPM (3): that is, $E(\overline{\boldsymbol{\xi}}_x^\star(t)) = E(\boldsymbol{\xi}_x^\star(t)) = E_x^\star$ and $E(\overline{\boldsymbol{\xi}}_y^\star(t)) = E(\boldsymbol{\xi}_y^\star(t)) = E_y^\star$, $\forall t \geq 0$, in which $E(w,\dot{w}) := (1/2)\dot{w}^2 - g/(2c_z)w^2$ denotes the orbital energy and $E_x^\star$ and $E_y^\star$ are nominal levels in x- and y-directions.

B. From Balance Recovery to Tracking Problem

It is pointed out that introduction of the synchronized reference model (13) allows to represent the balance recovery problem in a simpler form. To sketch the idea, for now we assume that $(\boldsymbol{\chi}(t_\mathrm{S}),\boldsymbol{\eta}(t_\mathrm{S})) \in \mathcal{E}$ and

$$(\boldsymbol{\chi}(t),\boldsymbol{\eta}(t)) \notin \mathcal{D}_\mathrm{T}, \quad \forall t \geq t_\mathrm{S} \tag{15}$$

for some t_S and $\mathcal{E} := \{(\boldsymbol{\chi},\boldsymbol{\eta}) : E(\overline{\boldsymbol{\xi}}_x^\star) = E_x^\star,\ \text{and}\ E(\overline{\boldsymbol{\xi}}_y^\star) = E_y^\star\}$. Noting that the orbital energy of the $\boldsymbol{\xi}^\star$-dynamics is the same as that of the $\overline{\boldsymbol{\xi}}^\star$-dynamics, one finds a certain δ_T such that $\overline{\boldsymbol{\xi}}^\star(t) = \boldsymbol{\xi}^\star(t-\delta_\mathrm{T})$ for all $t \geq t_\mathrm{S}$. (See also Fig. 4.) Using the above equality and (2b) we have $\mathbf{c}_\mathrm{H}(t) - \big(\mathbf{c}_\mathrm{H}^\star(t-\delta_\mathrm{T}) + \boldsymbol{\delta}_\mathrm{H}\big) = \boldsymbol{\xi}_1(t) - \boldsymbol{\xi}_1^\star(t-\delta_\mathrm{T}) = \boldsymbol{\xi}_1(t) - \overline{\boldsymbol{\xi}}_1^\star(t)$. From this, it can be concluded (with additional computations) that the key equations (2) for the balance recovery are equivalent to "$(\delta_\mathrm{T},\boldsymbol{\delta}_\mathrm{H})$-free" statements

$$\|\boldsymbol{\xi}_1(t) - \overline{\boldsymbol{\xi}}_1^\star(t)\| < \epsilon,\ \forall t \geq t_\mathrm{S}, \quad \mathbf{u}_\mathrm{d} = \mathbf{u}_\mathrm{d}^\star(\overline{\boldsymbol{\xi}}_1^\star) = 2\overline{\boldsymbol{\xi}}_1^\star. \tag{16}$$

(For the page limit, the detailed derivation of (16) is omitted.) With the equivalence of (2) and (16) in mind, in the next section we will construct a hybrid tracking controller

$$\dot{\boldsymbol{\nu}} = F_\nu(\boldsymbol{\chi},\boldsymbol{\eta}), \quad \forall(\boldsymbol{\chi},\boldsymbol{\eta}) \notin \mathcal{D}, \tag{17a}$$

$$\boldsymbol{\nu}^+ = G_\nu(\boldsymbol{\chi},\boldsymbol{\eta}), \quad \forall(\boldsymbol{\chi},\boldsymbol{\eta}) \in \mathcal{D}, \tag{17b}$$

$$(\mathbf{u}_\mathrm{c},\mathbf{u}_\mathrm{d}) = H_\nu(\boldsymbol{\chi},\boldsymbol{\eta}) \tag{17c}$$

(as well as $\mathcal{D}_\mathrm{T}$ and $\widetilde{\mathbf{r}}_\mathrm{H}$), by which the tracking error $\mathbf{e} := \boldsymbol{\xi} - \overline{\boldsymbol{\xi}}^\star$ is regulated in the sense of Lyapunov as well as the constraints (12) and the steady-state requirements (15) and (16) are satisfied at the same time.

V. Reactive Step Generation for Balance Recovery via Hybrid Tracking Control of DCM

A. Main Idea

We begin by noting that the jump dynamics of $\boldsymbol{\xi}$ is uncontrollable with respect to $\mathbf{u}_\mathrm{d}$, which makes it fundamentally impossible to regulate $\mathbf{e}$ only by stepping (i.e., by $\mathbf{u}_\mathrm{d}$). From a control-theoretic perspective, an alternative way would be to regulate the "unstable" part of $\mathbf{e}$, defined by

$$\mathbf{e}_\mathrm{u} = \boldsymbol{\xi}_\mathrm{u} - \boldsymbol{\xi}_\mathrm{u}^\star := \big(\boldsymbol{\xi}_1 + \sqrt{(c_z/g)}\boldsymbol{\xi}_2\big) - \big(\overline{\boldsymbol{\xi}}_1^\star + \sqrt{(c_z/g)}\overline{\boldsymbol{\xi}}_2^\star\big), \tag{18}$$

under the belief that regulation of $\mathbf{e}_\mathrm{u}$ leads to the overall stability in the end. From the terminology used in [3], we name $\mathbf{e}_\mathrm{u}$ in (18) "divergent component of motion (DCM) error". Along the hybrid LIPM (4), one can easily derive the DCM error dynamics as

$$\dot{\mathbf{e}}_\mathrm{u} = \mathbf{A}_\mathrm{u}\mathbf{e}_\mathrm{u} + \mathbf{B}_\mathrm{u}\mathbf{u}_\mathrm{c} + \mathbf{E}_\mathrm{u}\boldsymbol{\Delta}_\mathrm{c}, \quad \forall(\boldsymbol{\chi},\boldsymbol{\eta}) \notin \mathcal{D} \tag{19a}$$

$$\mathbf{e}_\mathrm{u}^+ = \mathbf{e}_\mathrm{u} - \mathbf{u}_\mathrm{d} + 2\overline{\boldsymbol{\xi}}_1^\star, \quad \forall(\boldsymbol{\chi},\boldsymbol{\eta}) \in \mathcal{D} \tag{19b}$$

where $\mathbf{A}_\mathrm{u} := \sqrt{g/c_z}\mathbf{I}$, $\mathbf{B}_\mathrm{u} := [-\sqrt{g/c_z}\mathbf{I}, \sqrt{c_z/g}\mathbf{H}]$, and $\mathbf{E}_\mathrm{u} := [-\sqrt{g/c_z}\mathbf{I}, (1/m)\sqrt{g/c_z}\mathbf{I}]$. It is clear that unlike (4), both flow and jump dynamics of the DCM error system (19) are now controllable, which makes it possible to achieve the balance recovery via the DCM error regulation.

In our approach, a Lyapunov function, say $V_\mathrm{u}(\mathbf{e}_\mathrm{u})$, for the DCM error serves as the core of the hybrid DCM error regulation. A natural candidate for V_u could be derived from the (controllable) continuous DCM error dynamics (19a) as

$$V_\mathrm{u}(\mathbf{e}_\mathrm{u}) = \mathbf{e}_\mathrm{u}^\top \mathbf{P}_\mathrm{u}\mathbf{e}_\mathrm{u} \tag{20}$$

with a gain matrix $\mathbf{K}_\mathrm{u}$ and a positive definite and symmetric matrix $\mathbf{P}_\mathrm{u}$ such that $(\mathbf{A}_\mathrm{u}+\mathbf{B}_\mathrm{u}\mathbf{K}_\mathrm{u})^\top\mathbf{P}_\mathrm{u} + \mathbf{P}_\mathrm{u}(\mathbf{A}_\mathrm{u}+\mathbf{B}_\mathrm{u}\mathbf{K}_\mathrm{u}) < 0$ holds. For future use we choose a small $\gamma_\mathrm{v} > 0$ such that

$$V_\mathrm{u}(\mathbf{e}_\mathrm{u}) \leq \gamma_\mathrm{v} \quad \Rightarrow \quad \mathbf{u}_\mathrm{c}^\mathrm{m} \preceq \mathbf{K}_\mathrm{u}\mathbf{e}_\mathrm{u} \preceq \mathbf{u}_\mathrm{c}^\mathrm{M}, \tag{21}$$

which will be used as a threshold for V_u.

With V_u selected above, we introduce a Lyapunov-based criterion to determine when and where to step, which will

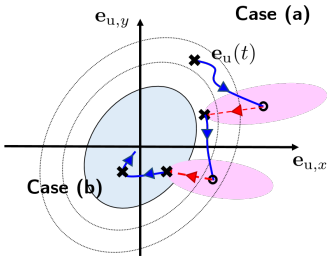


Fig. 5. Conceptual view of DCM error regulation via the ideal strategy presented in Subsection V-A: In Case (a), the Lyapunov level $V_{\mathrm{u}}(\mathbf{e}_{\mathrm{u}}(t))$ of the DCM error is possibly increased during the walking cycle, which is then compensated in the next footstep derived from the best among possible solutions of the QP (pink shadow); In Case (b), once $V_{\mathrm{u}}(\mathbf{e}_{\mathrm{u}})$ reaches the push-free region $\{V_{\mathrm{u}} : V_{\mathrm{u}} \leq \gamma_{\mathrm{v}}\}$ (blue shadow), it never escapes the region until another push is applied.

be realized by design of (17) in the next subsection. For simplicity of explanation, assume for now that the DSP is instantaneous (i.e., $T^{\star}_{\mathrm{DSP}} = 0$ and thus $\widetilde{\mathbf{r}}_{\mathrm{H}} \equiv 0$), and the continuous input $\mathbf{u}_{\mathrm{c}}$ takes the form of the DCM-based feedback control law

$$\mathbf{u}_{\mathrm{c}} = \bar{s}(\mathbf{K}_{\mathrm{u}}\mathbf{e}_{\mathrm{u}}) =: \mathbf{u}^{\star}_{\mathrm{c}} \tag{22}$$

with a component-wise saturation function $\bar{s} : \mathbb{R}^4 \to \mathbb{R}^4$ whose saturation levels are set by $\mathbf{u}^{\mathrm{m}}_{\mathrm{c}}$ and $\mathbf{u}^{\mathrm{M}}_{\mathrm{c}}$, respectively. (We note in advance that, in the next subsection, the control law (22) will be slightly modified to take nonzero perturbation $\widetilde{\mathbf{r}}_{\mathrm{H}}$ and continuity of $\mathbf{u}_{\mathrm{c}}$ on time into account.) Then after a push is applied, the continuous DCM error dynamics (19a) under the assumptions turns out to be an ideal form

$$\dot{\mathbf{e}}_{\mathrm{u}} = \mathbf{A}_{\mathrm{u}}\mathbf{e}_{\mathrm{u}} + \mathbf{B}_{\mathrm{u}}\bar{s}\big(\mathbf{K}_{\mathrm{u}}\mathbf{e}_{\mathrm{u}}\big), \quad \forall(\boldsymbol{\chi},\boldsymbol{\eta}) \notin \mathcal{D} \tag{23}$$

while the DCM error $\mathbf{e}_{\mathrm{u}}$ possibly moves away from the origin due to a strong push. For the ideal DCM dynamics (23) and (19b), our Lyapunov-based strategy for generating step time and location (or equivalently, selecting $\mathbf{u}_{\mathrm{d}}$ and $\mathcal{D}_{\mathrm{T}}$) is presented below, with a particular emphasis on the threshold γ_{v} (whose graphical explanation can be found in Fig. 5).

- **Case (a)** (if $V_{\mathrm{u}}(\mathbf{e}_{\mathrm{u}}) \geq \gamma_{\mathrm{v}}$): Since $\mathbf{u}_{\mathrm{c}}$ is limited, $V_{\mathrm{u}}(\mathbf{e}_{\mathrm{u}}(t))$ gets possibly increased during a walking cycle, which means that the DCM error regulation cannot be done without stepping properly. The key idea for selecting $\mathbf{u}_{\mathrm{d}} = \mathbf{o}^{+}_{\mathrm{H}} - \mathbf{o}_{\mathrm{H}}$ in this case is to compensate for the increased amount of the "post-step" Lyapunov level $V_{\mathrm{u}}(\mathbf{e}_{\mathrm{u}}(s^{+}_{j}))$ during the previous walking cycle as

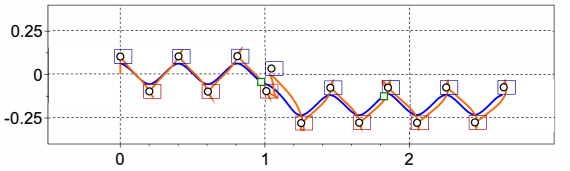
$$V_{\mathrm{u}}\big(\mathbf{e}_{\mathrm{u}}(s^{+}_{j})\big) - V_{\mathrm{u}}\big(\mathbf{e}_{\mathrm{u}}(s^{+}_{j-1})\big) \leq -\gamma_{\mathrm{v}}, \tag{24}$$

  by which the Lyapunov level $V_{\mathrm{u}}(\mathbf{e}_{\mathrm{u}})$ becomes eventually smaller than γ_{v} after finite number of footsteps. To derive the inequality (24), take $\mathbf{u}_{\mathrm{d}}$ satisfying that

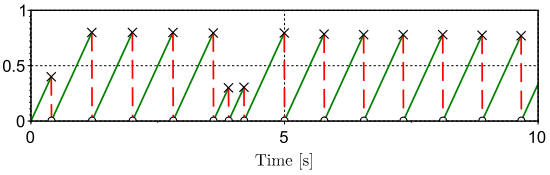
$$\begin{aligned} &V_{\mathrm{u}}(\mathbf{e}_{\mathrm{u}}(s_j)) - V_{\mathrm{u}}(\mathbf{e}_{\mathrm{u}}(s^{-}_{j})) \\ &= V_{\mathrm{u}}\big(\mathbf{e}_{\mathrm{u}}(s^{-}_{j}) - \mathbf{u}_{\mathrm{d}} + 2\overline{\boldsymbol{\xi}}^{\star}_{1}\big) - V_{\mathrm{u}}(\mathbf{e}_{\mathrm{u}}(s^{-}_{j})) \leq -\mu_1 \end{aligned} \tag{25}$$

  for a small positive constant μ_1, and choose the step time s_j as the moment when

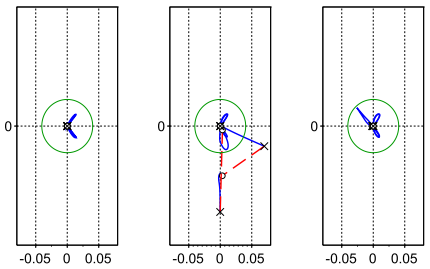
$$V_{\mathrm{u}}(\mathbf{e}_{\mathrm{u}}(s^{-}_{j})) - V_{\mathrm{u}}(\mathbf{e}_{\mathrm{u}}(s_{j-1})) = -\gamma_{\mathrm{v}} + \mu_1 \tag{26}$$



(a) CoM $\mathbf{c}_{\mathrm{H}}$ (blue solid), step location $\mathbf{o}_{\mathrm{H}}$ (black circle), ZMP $\mathbf{p}_{\mathrm{H}}$ (orange solid), and footholds (squares) in top view: The green square indicates the CoM position when a push is applied.



(b) The time $\tilde{s}(t) = t - s_{j\ 1}$ spent from the previous step



(c) Phase portrait of $\mathbf{e}_{\mathrm{u}}$ for $t \in [0, t_{\mathrm{f},1})$ (left), $t \in [t_{\mathrm{f},1}, t_{\mathrm{f},2})$ (middle), and $t \in [t_{\mathrm{f},2}, 10)$ (right): The green ellipsoid indicates $\{\mathbf{e}_{\mathrm{u}} : V_{\mathrm{u}}(\mathbf{e}_{\mathrm{u}}) = \gamma_{\mathrm{v}}\}$.

Fig. 6. Simulation results with the proposed RSG in the presence of two sequential pushes

  holds (so that (25) and (26) directly implies (24)).
- **Case (b)** (if $V_{\mathrm{u}}(\mathbf{e}_{\mathrm{u}}) < \gamma_{\mathrm{v}}$): In this case $\dot{V}_{\mathrm{u}} < 0$ by (21) and thus $V_{\mathrm{u}}(\mathbf{e}_{\mathrm{u}}(t)) < \gamma_{\mathrm{v}}$ during the current walking cycle. Thus for the Lyapunov stability of $\mathbf{e}_{\mathrm{u}}$ it is enough to ensure $\mathbf{e}^{+}_{\mathrm{u}} = \mathbf{e}_{\mathrm{u}}$ so that $V_{\mathrm{u}}(\mathbf{e}^{+}_{\mathrm{u}}) = V_{\mathrm{u}}(\mathbf{e}_{\mathrm{u}})$. This is in fact simply done by taking $\mathbf{u}_{\mathrm{d}}$ to satisfy the steady-state constraint (16), and by setting the step time s_j when $(\boldsymbol{\chi}(s_j), \boldsymbol{\eta}(s_j)) \in \mathcal{D}_{\mathrm{S}}$ and thus (15) holds.

We point out that when $\mathbf{u}_{\mathrm{d}}$ is chosen in Case (a), the constraints on $\mathbf{u}_{\mathrm{d}}$ and $\boldsymbol{\xi}$ in (12) (in addition to (25)) also need to be satisfied. This encourages us to consider the following constrained optimization problem with respect to $\mathbf{u}_{\mathrm{d}}$:

$$\begin{aligned} \underset{\mathbf{u}_{\mathrm{d}} \in \mathbb{R}^2}{\text{minimize}} \quad & V_{\mathrm{u}}\big(\mathbf{e}^{+}_{\mathrm{u}}\big) && \text{(27a)} \\ \text{subject to} \quad & \begin{bmatrix} 0 & \sigma^{+} \end{bmatrix} \mathbf{u}_{\mathrm{d}} + 2w_y \leq -\mu_2, && \text{(27b)} \\ & \begin{bmatrix} 0 & -\sigma^{+} \end{bmatrix} \boldsymbol{\xi}^{+}_{1} - w_y \leq -\mu_3, && \text{(27c)} \\ & \mathbf{u}^{\mathrm{m}}_{\mathrm{d}} \preceq \mathbf{u}_{\mathrm{d}} \preceq \mathbf{u}^{\mathrm{M}}_{\mathrm{d}} && \text{(27d)} \end{aligned}$$

where $\sigma^{+} = -\sigma$, $\mathbf{e}^{+}_{\mathrm{u}}$ and $\boldsymbol{\xi}^{+}$ are given in (19b) and (4), respectively, and $\mu_2 > 0$ and $\mu_3 > 0$ are some margins for the original constraints. One can readily see that (27) in fact takes the form of the quadratic programming (QP) with a relatively small size compared with other optimization-based frameworks. The solution of (27) is denoted by $\mathbf{u}^{\mathrm{QP}}_{\mathrm{d}}(\boldsymbol{\chi}, \boldsymbol{\eta})$, which will be used as the discrete input $\mathbf{u}_{\mathrm{d}}$ for Case (a). Note that once the DCM level $\|\mathbf{e}_{\mathrm{u}}\|$ is bounded, the QP (27) is always feasible with sufficiently small μ_2 and μ_3.

B. Hybrid Tracking Controller as a RSG

This subsection is devoted to presenting the detailed structure of (17), which realizes our stepping strategy presented above. Here some modifications on $\mathbf{u}_\mathrm{c}$ in (22) are made to ensure the continuity of $\mathbf{u}_\mathrm{c}$ on time (via low-pass filtering) and also to compensate the effect of the known perturbation $\widetilde{\mathbf{r}}_\mathrm{H}$ (that is non-zero in the DSP) on the DCM error dynamics. (It is noted that such a modification is not problematic so long as the time constant of the low-pass filter is sufficiently small, supported by the singular perturbation theory.)

Let the state of the hybrid tracking controller (17) be $\boldsymbol{\nu} := \big(\boldsymbol{\zeta}, \mathbf{o}_\mathrm{H}, \mathbf{r}_\mathrm{H}^\dagger, \widetilde{s}, V_\mathrm{u}^\dagger\big) \in \mathbb{R}^{10}$ where $\boldsymbol{\zeta} \in \mathbb{R}^4$ is the state of the low-pass filter for $\mathbf{u}_\mathrm{c}$, $\mathbf{r}_\mathrm{H}^\dagger \in \mathbb{R}^2$ and $V_\mathrm{u}^\dagger \in \mathbb{R}$ indicate $\mathbf{r}_\mathrm{H}(s_j^-)$ and $V_\mathrm{u}(s_j)$, respectively, and $\widetilde{s} \in \mathbb{R}$ is the time spent from the previous step time. The flow and jump maps are given by $F_\nu := \big(-(1/\theta)\boldsymbol{\zeta} + (1/\theta)\hat{\mathbf{u}}_\mathrm{c}^\star, \mathbf{0}, \mathbf{0}, 1, 0\big)$ and $G_\nu := \big(\boldsymbol{\zeta}, \mathbf{o}_\mathrm{H} + \mathbf{u}_\mathrm{d}, \mathbf{r}_\mathrm{H}, 0, V_\mathrm{u}\big(\mathbf{e}_\mathrm{u} - \mathbf{u}_\mathrm{d} + 2\overline{\boldsymbol{\xi}}_1^\star\big)\big)$, $\theta > 0$ is the time constant of the low-pass filter, and $\hat{\mathbf{u}}_\mathrm{c}^\star = \bar{s}\big(\mathbf{K}_\mathrm{u}\mathbf{e}_\mathrm{u} + \big(\mathbf{0}, mg\mathbf{H}^{-1}\widetilde{\mathbf{r}}_\mathrm{H}\big)\big)$ is a modified version of (22) in which the term $mg\mathbf{H}^{-1}\widetilde{\mathbf{r}}_\mathrm{H}$ is added to generate additional rate-of-change of angular momentum for compensating for the effect of $\widetilde{\mathbf{r}}_\mathrm{H}$ on the DCM error and thus for enhancing the DCM tracking performance. In addition, the output map H_ν of (17) is taken such that $H_\nu = (\boldsymbol{\zeta}, \mathbf{u}_\mathrm{d}^\mathrm{QP}(\boldsymbol{\chi}, \boldsymbol{\eta}))$, if $V_\mathrm{u}(\mathbf{e}_\mathrm{u}) \geq \gamma_\mathrm{v}$, and $H_\nu = (\boldsymbol{\zeta}, 2\overline{\boldsymbol{\xi}}_1^\star)$ otherwise. while we choose the explicit form of $\widetilde{\mathbf{r}}_\mathrm{H}$ as $\widetilde{\mathbf{r}}_\mathrm{H} := -(\mathbf{o}_\mathrm{H} - \mathbf{r}_\mathrm{H}^\dagger)\big((T_\mathrm{DSP}^\star - \widetilde{s})/T_\mathrm{DSP}^\star\big), \forall\ \widetilde{s} \in [0, T_\mathrm{DSP}^\star]$ and $\widetilde{\mathbf{r}}_\mathrm{H} = 0$ otherwise. The jump set $\mathcal{D}$ is selected as in (14), with $\mathcal{D}_\mathrm{S}$ below (14) and $\mathcal{D}_\mathrm{T} = \{(\boldsymbol{\chi}, \boldsymbol{\eta}) : \widetilde{s} \geq T_\mathrm{DSP}^\star + T_\mathrm{SP}\} \bigcap \big(\mathcal{D}_{\mathrm{T},1} \bigcup \mathcal{D}_{\mathrm{T},2}\big)$ where T_SP denotes the time required for landing of the swing foot, $\mathcal{D}_{\mathrm{T},1} := \big\{(\boldsymbol{\chi}, \boldsymbol{\eta}) : V_\mathrm{u}(\mathbf{e}_\mathrm{u}) \geq \gamma_\mathrm{v}$ and $V_\mathrm{u}(\mathbf{e}_\mathrm{u}) - V_\mathrm{u}^\dagger \geq -\gamma_\mathrm{v} + \mu_1\big\}$ and $\mathcal{D}_{\mathrm{T},2} := \big\{(\boldsymbol{\chi}, \boldsymbol{\eta}) : \begin{bmatrix}0 & -\sigma\end{bmatrix}\boldsymbol{\xi}_1 - w_y \geq 0\big\}$.

VI. Simulation Results

To verify the validity of the proposed scheme, we perform computer simulations on the biped robot "MAHRU-R" developed by KIST in Korea. Details on its specification and features can be found in previous works [12], [13] of the authors, which is omitted here due to the page limit. Simulations to be presented are done in the Scilab/Xcos environment [15], which is known as a free alternative for MATLAB/Simulink. We consider the x-directional walking scenario with the nominal values $L_x^\star = 0.2$ m, $L_y^\star = 0.2$ m, $T_\mathrm{DSP}^\star = 0.1$ s, and $T_\mathrm{SSP}^\star = 0.7$ s. The LQR scheme is adopted in selection of $\mathbf{P}_\mathrm{u}$ and $\mathbf{K}_\mathrm{u}$ (with the associated cost function properly taken), while other design parameters are chosen as $\theta = 0.05$, $\gamma_\mathrm{v} = 0.0015$, $\mu_1 = 0.003$, $\mu_2 = 0.05$, $\mu_3 = 0.0001$, and $T_\mathrm{SP} = 0.2$ s.

Fig. 6 shows simulation results for the case when two impulsive pushes are applied sequentially at $t_{\mathrm{f},1} = 3.8$ s and $t_{\mathrm{f},2} = 6.8$ s, with $(\mathbf{i}_\mathrm{H}, \Delta_\mathrm{f})$ set as $(\mathbf{i}_{\mathrm{H},1}, \Delta_{\mathrm{f},1}) = ((8, -6), 0.1)$ and $(\mathbf{i}_{\mathrm{H},2}, \Delta_{\mathrm{f},2}) = ((-6, 8), 0.2)$. It is seen from the top view in Fig. 6(a) that the biped robot with the proposed algorithm recovers its balance against the pushes, for which the step time and location are adjusted appropriately by the proposed RSG (Figs. 6(a) and 6(b)). The behavior of the overall system can also be explained with the contour of the DCM error and its Lyapunov function, in Fig. 6(c). The first push enforces the DCM error $\mathbf{e}_\mathrm{u}$ to leave the Lyapunov level set in a transient, and in turn the error becomes smaller enough again by the proposed RSG. It is important to point out that the proposed RSG does not modify the stepping plan for the second push, as the Lyapunov level V_u for the DCM error remains smaller than γ_v even after the push.

VII. Conclusion

In this paper, we presented a promising Lyapunov-based approach to reactive step generation for balance recovery against a severe push. The hybrid systems framework played the key role in reformulating the balance recovery as the regulation problem of the DCM error. In this point of view, we proposed a hybrid tracking controller as a RSG that generates step time and location, composed of a DCM-based continuous controller and a QP-based discrete controller.

References

[1] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots,* 2006, pp. 200–207.

[2] J. Englsberger, C. Ott, M. A. Roa, A. Albu-Schäffer, and G. Hirzinger, "Bipedal walking control based on capture point dynamics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 4420–4427.

[3] J. Englsberger, C. Ott, and A. Albu-Schäffer, "Three-dimensional bipedal walking control based on divergent component of motion," *IEEE Trans. Robot.*, vol. 31, no. 2, pp. 355–368, 2015.

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

[5] J. Ding, C. Zhou, S. Xin, X. Xiao, and N. Tsagarakis, "Nonlinear model predictive control for robust bipedal locomotion exploring CoM height and angular momentum changes," `arXiv:1902.06770`, 2019.

[6] R. J. Griffin and A. Leonessa, "Model predictive control for dynamic footstep adjustment using the divergent component of motion," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 1763–1768.

[7] T. Kamioka, H. Kaneko, T. Takenaka, and T. Yoshiike, "Simultaneous optimization of ZMP and footsteps based on the analytical solution of divergent component of motion," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 1763–1770.

[8] R. J. Griffin, A. Leonessa, and A. Asbeck, "Disturbance compensation and step optimization for push recovery," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 5385–5390.

[9] M. Khadiv, A. Herzog, S. Ali, A. Moosavian, and L. Righetti, "Step timing adjustment: A step toward generating robust gaits," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2016, pp. 35–42.

[10] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 667–673.

[11] R. Goedel, R. G. Sanfelice, and A. R. Teel, *Hybrid Dynamical Systems: Modeling Stability and Robustness*, Princeton University Press, 2012.

[12] G. Park, J. H. Kim, and Y. Oh, "Towards fully reactive multi-step generation for humanoids against instantaneous push: A case of walking in place in sagittal plane," in *Proc. Annu. Conf. IEEE Ind. Electron. Soc.*, 2019, pp. 577–582.

[13] G. Park and Y. Oh, "Preliminary study on role of finite-sized foot in push recovery of biped robot in sagittal plane via stabilization of divergent component of motion," to be presented at *IEEE Int. Conf. Emerging Tech. Factory Autom., 2020.*

[14] A. Zamparelli, N. Scianca, L. Lanari, and G. Oriolo, "Humanoid gait generation on uneven ground using intrinsically stable MPC," in *Proc. IFAC World Congress*, 2018, pp. 393–398.

[15] S. L. Campbell, J. P. Chancelier, and R. Nikoukhah, *Modeling and Simulation in SCILAB*, Springer, 2006.