2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

Simultaneous Optimization of ZMP and Footsteps based on the Analytical Solution of Divergent Component of Motion

Takumi Kamioka[1], Hiroyuki Kaneko[2], Toru Takenaka[2] and Takahide Yoshiike[1]

Abstract— **Real-time planning of footsteps has been a big challenge for bipedal research. A large number of methods have been proposed over the last several years. In addition, divergent component of motion (DCM) of linear inverted pendulum (LIP) has been applied to solve this problem thus attracting a great deal of attention. In this paper, we derive an analytical solution of DCM for an arbitrary input function and propose a novel quadratic programming (QP) problem for the simultaneous optimization of zero moment point (ZMP) and foot placements based on the analytical solution. To validate the method, we conducted a push recovery experiment on real hardware. The result of the experiment shows that our new algorithm realizes a hierarchical strategy for disturbance compensation.**

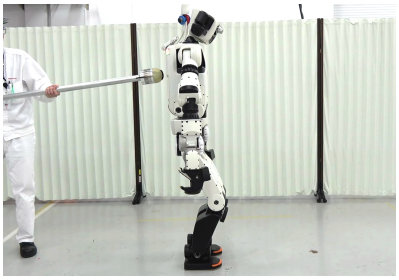

Fig. 1: Push recovery experiment.

I. Introduction

Humanoid robots are now expected to collaborate with humans and/or to work in their place in environments developed for them. In such situations, humanoid robots must be able to maintain their balance with respect to external disturbances.

A successive approach for bipedal locomotion has been a trajectory based approach which consists of a planner and a controller. The planner generates a reference which guarantees long-term stability and the controller tracks the reference via feedback control. However, the controller is unable to compensate for large disturbances because it only considers current errors.

Some methods for real-time planning have been proposed over the last decade. Nishiwaki et al. proposed real-time planning of the center of gravity (COG) and ZMP based on LIP [1], and Urata et al. extended it for footstep planning by a singular problem [2]. Morisawa et al. proposed a planning method for footsteps based on an analytical solution by polynomial ZMP trajectory [3]. Recently, some optimal control algorithms for footstep planning were formulated based on LIP [4], [5].

Recent advances in bipedal locomotion technologies have been based on an unstable part of a linearly divided state of the inverted pendulum. We named it DCM from an analysis of its dynamics [6] and proposed methods for real-time planning for both ZMP (for walking [6] and running [7]) and foot placements [8]. Equivalent values to DCM have been analyzed by Pratt et al. as the capture point (CP) which was derived by the supporting point to converge orbital energy [9], [10]. DCM was extended to three dimensions (3D) and a real-time planner and controller based on 3D DCM was proposed without direct planning of COG [11], [12]. Griffin et al. proposed a footstep optimization method based on 3D DCM [13]. Englsberger et al. derived an analytical solution of DCM for the reference of virtual repellent point trajectories which were defined by polynomial spline functions and applied it to the analytical adjustment of foot placements [14].

We also proposed a real-time planning algorithm which sequentially determined ZMP and foot placements [15]. The algorithm relies on an approximated gradient method, therefore, computational time must be longer to make larger modifications for foot placements. In this paper, we derive an analytical solution of DCM and propose a novel QP problem for the simultaneous optimization of ZMP and foot placements based on the analytical solution. The analytical solution of DCM was derived for a general input function, therefore, the solution may be widely applied to other problems. By solving the QP problem, planning can be done quickly. The algorithm was validated by a push recovery experiment on real hardware, as shown in Fig. 1.

This paper is organized as follows: In section II, we review the derivation of DCM and its characteristics. The analytical solution of DCM is given in section III. The QP problem of simultaneous optimization of ZMP and foot placements is described in Section IV. Section V provides a quick overview of our stabilizing controller. The result of the hardware experiment is shown in Section VI. Finally, we conclude the study and suggest future work in Section VII.

II. Linear Inverted Pendulum and DCM

In this section, we review the derivation and its characteristics of DCM described in [6]. The dynamics of LIP is represented by the state equation as

$$\frac{d}{dt}\boldsymbol{x}(t) = A\boldsymbol{x}(t) + Bp(t) \tag{1}$$

[1]T. Kamioka and T. Yoshiike are with the Honda Research Institute Japan Co., Ltd, 8-1 Honcho, Wako, Saitama, Japan. takumi.kamioka@jp.honda-ri.com
[2]H. Kaneko and T. Takenaka are with the Honda R&D Co., Ltd.

where,

$$\boldsymbol{x} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}, A = \begin{bmatrix} 0 & 1 \\ \omega^2 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ -\omega^2 \end{bmatrix}, \omega = \sqrt{\frac{g}{h}},$$

x, p, g and h denote the position of COG, the position of ZMP, the gravitational constant and the height of LIP, respectively. We analyzed the zero input response (i.e. $p = 0$) of the system (1). The matrix A has eigenvalues $\pm\omega$ and eigenvectors as follows:

$$\boldsymbol{v}_n = v \begin{bmatrix} 1 \\ -\omega \end{bmatrix}, \qquad \boldsymbol{v}_p = v \begin{bmatrix} 1 \\ \omega \end{bmatrix},$$

where $\boldsymbol{v}_n$ and $\boldsymbol{v}_p$ correspond to negative and positive eigenvalues, respectively, and v is an arbitrary scalar to determine its norm. A linear state transformation is obtained by matrix $V = \{\boldsymbol{v}_n, \boldsymbol{v}_p\}$ as follows:

$$\begin{bmatrix} \eta \\ \xi \end{bmatrix} = V^{-1} \begin{bmatrix} x \\ \dot{x} \end{bmatrix}. \tag{2}$$

By substituting the derivative of the (2) into (1), the dynamics can be diagonalized as

$$\frac{d}{dt} \begin{bmatrix} \eta \\ \xi \end{bmatrix} = V^{-1} A V \begin{bmatrix} \eta \\ \xi \end{bmatrix} = \begin{bmatrix} -\omega & 0 \\ 0 & \omega \end{bmatrix} \begin{bmatrix} \eta \\ \xi \end{bmatrix}. \tag{3}$$

The dynamics of η and ξ are independent, because matrix $V^{-1}AV$ is diagonal. This technique is generally applied and called *modal decomposition* [16].

The general solution of the differential equation (3) is

$$\eta(t) = C_1 e^{-\omega t}, \tag{4}$$
$$\xi(t) = C_2 e^{\omega t}, \tag{5}$$

where C_1 and C_2 are arbitrary constants. (4) and (5) show that η will converge and ξ will diverge by time evolution. Hence, we can eliminate η and only consider ξ to control the COG of LIP. Therefore, we defined the convergent component of motion (CCM) η and DCM ξ of LIP as components of diagonalized dynamics of state space corresponding to negative and positive eigenvalues, respectively. Witt proposed a limit cycle controller based on the same analysis [17]. Both characteristics: 1) the diagonal dynamics and 2) the convergent and the divergent properties are essential to reduce the dimension of the state. Englsberger and Ott showed equivalent results from the analysis of coupled dynamics of LIP: the dynamics between ZMP and DCM is unstable, and the dynamics between DCM and the COG is stable [18]. Moreover, it is expanded to 3D by Englsberger et al. [11] and Hopkins et al. [12].

It should be noted that CCM and DCM are generally dimensionless values, because the scalar parameter v is arbitrary. In [6], we chose $v = 0.5$ to identify dimensions of CCM and DCM as the dimension of the position of COG. DCM with $v = 0.5$ corresponds to (instantaneous) CP [9]. Further, Koolen et al. analyzed the dynamics of CP by the same diagonalization technique [19]. The rest of the formulation in this paper is applicable for arbitrary v without loss of generality, however, we show the case $v = 0.5$ for simplification.

III. The analytical solution of DCM

In this section, we show the analytical solution of DCM for both continuous and discrete input functions.

A. Continuous input

DCM with arbitrary input function $p(t)$ is represented as

$$\xi(t) = e^{\omega t} \left\{ \xi(0) - \omega \int_0^t e^{-\omega s} p(s) ds \right\}. \tag{6}$$

Here, we define the second term of (6) as input dependent DCM $\xi_p(t)$ which is solved by

$$\begin{aligned} \xi_p(t) &= -\omega \left\{ \int_0^\infty e^{-\omega s} p(s) ds - \int_0^\infty e^{-\omega(s+t)} p(s+t) ds \right\} \\ &= -\omega \left\{ \mathcal{L}[p(s)] - e^{-\omega t} \mathcal{L}[p(s+t)] \right\}, \end{aligned} \tag{7}$$

where $\mathcal{L}[.]$ denotes the Laplace transform. Equation (7) shows that the solution of DCM is generally calculated by the Laplace transform. If analytical Laplace transforms of both the input function and the time-delayed input function[1] exist, we can obtain a closed-form solution of DCM.

For example, the solution of DCM for the time polynomial ZMP $p(t) = \sum_{j=0}^{n} w_j t^j$ is represented by

$$\xi_p^{pol}(t) = \sum_{j=0}^{n} w_j \xi_{p,j}^{mon}(t) \tag{8}$$

where $\xi_{p,j}^{mon}(t)$ denotes DCM depending on the j-th degree monomial t^j. By using the binomial theorem $(s+t)^j = \sum_{l=0}^{j} \binom{j}{l} s^{j-l} t^l$ and the Laplace transform of monomial $\mathcal{L}[t^j]$ equal to $j!\omega^{-(j+1)}$, DCM is solved by

$$\xi_{p,j}^{mon}(t) = -\left\{ j!\omega^{-j} - e^{-\omega t} \sum_{l=0}^{j} {}^jP_{j-l} t^l \omega^{-(j-l)} \right\} \tag{9}$$

where ${}^iP_j$ denotes j-permutations of i. (8) and (9) show that DCM for continuous time polynomial input is calculated by the closed-form.

B. Discrete input

DCM depending on discrete inputs which are defined by zero-order hold with sampling period T becomes

$$\begin{aligned} \xi_p[k] &= -\omega \sum_{i=0}^{k-1} p[i] \int_{iT}^{(i+1)T} e^{-\omega s} ds \\ &= (e^{-\omega T} - 1) \sum_{i=0}^{k-1} e^{-i\omega T} p[i]. \end{aligned} \tag{10}$$

Let z be $e^{\omega T}$ and using the same manner of the continuous case, (10) is represented by

$$\xi_p[k] = (z^{-1} - 1) \left\{ \mathcal{Z}[p[i]] - z^{-k} \mathcal{Z}[p[i+k]] \right\} \tag{11}$$

where $\mathcal{Z}[.]$ denotes the Z-transform. Hence, DCM depending on discrete inputs is also solved by closed-form, if analytical Z-transforms of both the input function and the time-delayed input function exist.

[1]The Laplace transform of a time-delayed function is different from the time shifted Laplace transform which is calculated by $e^{-\omega t}\mathcal{L}[p(s)]$.

Again, for example of the time polynomial ZMP $p[k] = \sum_{j=0}^{n} w_j (kT)^j$, DCM is also calculated by the sum of products of the j-th parameter w_j and DCM depending on the j-th monomial ZMP which is represented by

$$\xi_{p,j}^{mon}[k] = (z^{-1} - 1)\left\{ \mathcal{Z}[i^j] - z^{-k} \sum_{l=0}^{j} \binom{j}{l} (kT)^{j-l} \mathcal{Z}[i^l] \right\}. \tag{12}$$

Equation (12) shows that DCM is analytically calculated only depending on the Z-transform of the monomial function. Unfortunately, compared to the continuous case, the Z-transform of the monomial $\mathcal{Z}[i^j]$ is not a closed-form function taking the degree j as a parameter [20], but we can implement it by preparing a function table for each degree.

IV. The QP problem for simultaneous optimization of ZMP and footsteps

A. Piecewise relative-time polynomial ZMP

Here, we define ZMP trajectory by the piecewise relative-time polynomial:

$$p(t) = \sum_{i=0}^{e} \sum_{j=0}^{n} w_{ij} t_i'^j(t), \tag{13}$$

where

$$t_i'(t) = \begin{cases} \frac{t - t_i}{T_i} & \text{if } t_i \le t \le t_{i+1} \\ 0 & \text{otherwise} \end{cases},$$
$$T_i = t_{i+1} - t_i,$$

where e, n, t_i and T_i denote the number of polynomial pieces, the degree of each polynomial[2], the start time and the duration of the i-th polynomial, respectively. The absolute-time polynomial function has been widely used for ZMP trajectory and contributes to obtain closed form solutions [3], [21]. In particular, we apply the relative-time polynomial which simplifies some calculations shown in the following subsections.

B. The quadratic programming problem for ZMP

We suppose that ZMP trajectory should satisfy the following requirements: 1) stability criterion, 2) supporting region constraint, 3) continuity constraint and 4) smoothness. For the first requirement, we proposed the *cyclic gait criterion* which was commonly applied to walking [6], running [8], hopping [15] and quadrupedal locomotion [22]. The second requirement is a necessary condition for bipedal locomotion. Connecting two adjacent trajectories is the third requirement. The 4th requirement helps hardware tracking performance.

[2]The different degree for each polynomial can be applied and may decrease computational cost. However we assigned a fixed number for all degrees of polynomials for simplicity.

1) Stability criterion: Our basic idea of the stability criterion has been a cyclic gait which means the same gaits connect sequentially [6]. In the cyclic gait criterion, footsteps and the desired ZMP trajectory in the cyclic gait which includes an arbitrary number of steps starting from the end of the target time t_{e+1} could sequentially continue without any modification. For further details of this concept, refer to [6], [15], [22].

The cyclic gait criterion gives us the equality constraint of DCM at time t_{e+1}. DCM depending on the piecewise polynomial ZMP is calculated by substituting (13) into (9) or (12) as

$$\begin{aligned} \xi_p^{pp}(t_{e+1}) &= -\omega \int_0^{t_{e+1}} e^{-\omega s} \sum_{i=0}^{e} \sum_{j=0}^{n} w_{ij} t'^j(s) ds \\ &= \sum_{i=0}^{e} \sum_{j=0}^{n} w_{ij} T_i^{-j} e^{-\omega t_i} \xi_{p,j}^{mon}(t_{i+1} - t_i). \end{aligned} \tag{14}$$

Equation (14) can be represented as the internal product $\boldsymbol{a}_{cyc}^T \boldsymbol{w}$, where $\boldsymbol{w} = \{w_{00}, \ldots, w_{ij}, \ldots, w_{en}\}$ is a parameter vector. Therefore, the equality constraint of the cyclic criterion comes into a linear equation:

$$\boldsymbol{a}_{cyc}^T \boldsymbol{w} = b_{cyc}, \tag{15}$$

where,

$$\boldsymbol{a}_{cyc} = \left[a_{00}, \ldots, a_{ij}, \ldots a_{en}\right]^T, \tag{16}$$
$$a_{ij} = e^{-\omega t_i} T_i^{-j} \xi_{p,j}^{mon}(t_{i+1} - t_i), \tag{17}$$
$$b_{cyc} = -\xi(0) + e^{-\omega t_{e+1}} \xi_{target}. \tag{18}$$

The target DCM ξ_{target} is calculated by the cyclic gait criterion shown in [15]. The cyclic gait criterion is one of the methods to represent a stability criterion. Other methods are applicable for example to determine the final DCM from a reference of virtual repellent points [14].

2) Supporting region constraint: ZMP must lie within the convex hull of the supporting region. A supporting region defined by foot placements can be represented as a set of linear inequalities:

$$\left[c_{sup}^x, \quad c_{sup}^y\right] (\boldsymbol{p}(t) - \boldsymbol{f}) < d_{sup}, \tag{19}$$

where c_{sup}^x, c_{sup}^y and d_{sup} are coefficients determined by vertices of the convex hull, and $\boldsymbol{p}$ and $\boldsymbol{f}$ are two dimensional vectors of the position of ZMP and the landing position of the swing foot, respectively. Equation (19) is also represented by the linear inequality of the parameter vector $\boldsymbol{w}_{xy} = \{\boldsymbol{w}_x^T, \boldsymbol{w}_y^T\}^T$:

$$\boldsymbol{a}_{sup}^T \boldsymbol{w}_{xy} < b_{sup}, \tag{20}$$

where

$$\boldsymbol{a}_{sup} = \left[c_{sup}^x \left[1, t', \ldots, t'^n\right], \quad c_{sup}^y \left[1, t', \ldots, t'^n\right]\right]$$
$$b_{sup} = d_{sup} + \boldsymbol{c}_{sup}^T \boldsymbol{f}.$$

The linear constraint (20) depends on time t, therefore, we approximate the constraint as a number of linear constraints for sampled time $t = [t_0, \ldots, t_k, \ldots, t_e]$. Kalakrishnan et al. proposed the same method for quadrapedal locomotion [23].

3) Continuity constraints: To continuously connect two different polynomials, we constrain polynomial edge points. The condition to coincide the end position of the $(i-1)$-th polynomial with the start position of the next (i)-th polynomial is obtained as a linear equality constraint:

$$\sum_{j=0}^{n} w_{i-1j} t_{i-1}'^{j}(t_i) = \sum_{j=0}^{n} w_{ij} t_i'^{j}(t_i)$$
$$\sum_{j=0}^{n} w_{i-1j} = w_{i0}$$
$$\begin{bmatrix} 1, \ldots, 1, -1, 0, \ldots, 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_{i-1} \\ \boldsymbol{w}_i \end{bmatrix} = 0. \tag{21}$$

Here, relative time $t'(t)$ simplifies this calculation because $t'_{i-1}(t_i) = 1, t'_i(t_i) = 0$. Velocity and acceleration are linearly constrained in a similar way:

$$\frac{1}{T_{i-1}} \sum_{j=1}^{n} w_{i-1j} j t_{i-1}'^{j-1}(t_i) = \frac{1}{T_i} \sum_{j=1}^{n} w_{ij} j t_i'^{j-1}(t_i)$$
$$\begin{bmatrix} 0, \frac{1}{T_{i-1}}, \ldots \frac{n}{T_{i-1}}, 0, -\frac{1}{T_i}, 0, \ldots, 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_{i-1} \\ \boldsymbol{w}_i \end{bmatrix} = 0, \tag{22}$$

$$\frac{1}{T_{i-1}^2} \sum_{j=2}^{n} w_{i-1j} j(j-1) t_{i-1}'^{j-2}(t_i)$$
$$= \frac{1}{T_i^2} \sum_{j=2}^{n} w_{ij} j(j-1) t_i'^{j-2}(t_i)$$
$$\begin{bmatrix} 0, 0, \frac{2}{T_{i-1}^2}, \ldots, \frac{n(n-1)}{T_{i-1}^2}, 0, 0, -\frac{2}{T_i^2}, 0, \ldots, 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_{i-1} \\ \boldsymbol{w}_i \end{bmatrix} = 0. \tag{23}$$

In addition to connecting two polynomials, ZMP edge points should coincide to at a desired position such as a foot placement. The desired position constraint is also represented by a linear equality:

$$\sum_{j=0}^{n} w_{ij} t_i'^{j}(t_{des}) = p_{des}$$
$$\begin{bmatrix} 1, t'(t_{des}), \ldots, t'^{n}(t_{des}) \end{bmatrix} \boldsymbol{w}_i = p_{des}, \tag{24}$$

where p_{des} is the desired position of ZMP and t_{des} is the desired time.

4) Smooth trajectory: A smooth trajectory can be obtained by minimizing the integral of squared acceleration. The acceleration of the piecewise relative-time polynomial is represented by an inner product:

$$\ddot{p}(t) = \frac{1}{T_i^2} \sum_{j=0}^{n} w_j j(j-1) t'^{j-2}$$
$$= \boldsymbol{w}^T \boldsymbol{h}(t).$$

Therefore, the integration of squared acceleration is obtained by a squared form of parameter vector:

$$\int_0^{t_{e+1}} \ddot{p}^2(t) dt = \boldsymbol{w}^T H \boldsymbol{w}, \tag{25}$$

where

$$H = \begin{bmatrix} \frac{1}{T_0^3}\bar{H} & & \boldsymbol{0} \\ & \frac{1}{T_i^3}\bar{H} & \\ \boldsymbol{0} & & \frac{1}{T_{e+1}^3}\bar{H} \end{bmatrix}.$$

The element of the r-th row and the c-th column of $\bar{H}$ is

$$\bar{H}_{r,c} = \begin{cases} 0 & \text{if } r < 2 \text{ or } c < 2 \\ \frac{r(r-1)c(c-1)}{r+c-3} & \text{otherwise} \end{cases}.$$

The relative-time formulation contributes to eliminate time dependency from (25). The absolute-time case for the 5-th degree polynomial which was shown by Kalakrishnan et al. includes power of times in the elements of matrix $\bar{H}$ [23].

5) The Optimization problem of ZMP: We define an optimization problem for LIP as

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & (25) \\ \text{s.t.} \quad & (15), (20), (21), (22), (23), (24). \end{aligned} \tag{26}$$

Equations (15) to (24) are linear functions and (25) is a quadratic function, therefore, the optimization problem (26) is represented by the linearly constrained QP problem as

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \boldsymbol{w}^T H \boldsymbol{w} \\ \text{s.t.} \quad & A_{eq} \boldsymbol{w} = \boldsymbol{b}_{eq} \\ & A_{in} \boldsymbol{w} < \boldsymbol{b}_{in}. \end{aligned} \tag{27}$$

By solving (27), we can determine a high-degree of polynomials. A higher degree makes ZMP trajectory more flexible but requires more computational cost. The degree of polynomial n should be determined by considering the trade-off. For the experiment shown in this paper, we chose each degree of the polynomial equal to 9 and the number of polynomial pieces for one step equal to 2 as a supporting and a swing phase.

C. The problem for foot placements

The QP problem represented by (27) may have no solution under large disturbances. The modification of foot placements is an option to avoid this problem. We parameterize the variation of the footstep position as

$$\Delta \boldsymbol{x}_{foot} = S \boldsymbol{w}_f, \tag{28}$$

where S and $\boldsymbol{w}_f$ are a selection matrix and a parameter vector for foot placements, respectively. Constraints (15), (20) and (24) must be redefined, because they depend on the modification of foot placements.

1) Cyclic criterion depending on foot placements: Even if the modified foot placement lands after the optimization problem t_{e+1}, the equality constraint (15) depends on the foot placement. For example in Fig. 2, the duration of the optimization problem is defined by $t_{e+1} = t_2$ and the latter trajectory is fixed to determine the cyclic gait. Modification of the landing position of the first footstep only changes the cyclic ZMP trajectory rather than the target ZMP trajectory

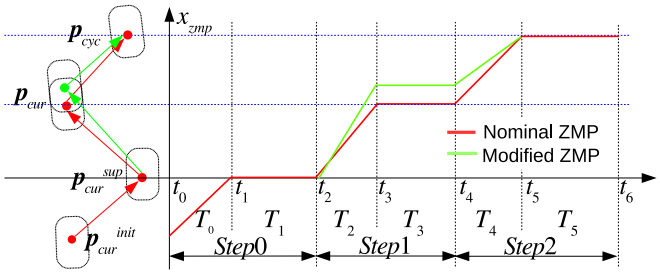

Fig. 2: The relationship between foot placements and ZMP trajectory.

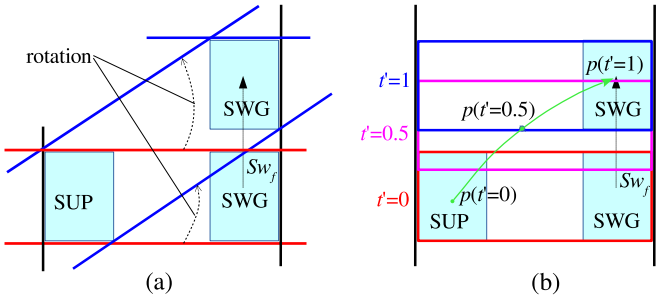

Fig. 3: The variation and the approximation of the double supporting region depending on the modification of foot placement. (a) Actual variation represented by rotation. (b) Linear approximation by parallel translation with time shift. The green arrow shows an example of the ZMP trajectory. The supporting region moves from red to blue according to time evolution.

depending on optimization variable $\boldsymbol{w}$. The modification changes the cyclic criterion (18) as

$$b_{cyc} = -\xi(0) + e^{-\omega t_{e+1}}(\xi_{target} + \xi_{\Delta foot}), \tag{29}$$

where $\xi_{\Delta foot}$ denotes DCM depending on the modification. As shown in Fig. 2, we assume that the ZMP trajectory in the cyclic duration forms the first degree polynomials, therefore, DCM is calculated by

$$\begin{aligned}\xi_{\Delta foot}(t) =& \left\{\xi_{p,1}^{mon}(T_2) + e^{-\omega t_3}\xi_{p,0}^{mon}(T_3)\right. \\ &\left. + e^{-\omega t_4}\left(\xi_{p,0}^{mon}(T_4) - \xi_{p,1}^{mon}(T_4)\right)\right\}\Delta x_{foot} \\ =& a_{cyc,f}\Delta x_{foot}\end{aligned} \tag{30}$$

Hence, by considering footstep modification, the cyclic criterion (15) becomes

$$\left[\boldsymbol{a}_{cyc}^T, \quad \boldsymbol{a}_{cyc,f}^T S\right]\begin{bmatrix}\boldsymbol{w} \\ \boldsymbol{w_f}\end{bmatrix} = b_{cyc,f}. \tag{31}$$

2) Supporting region condition depending on foot placements: The supporting region depends on the modification of foot placements for both the swing phase and the double supporting phase, if the landing time of a modified footstep is in the target duration to optimize. For the swing phase, the supporting region moves parallel to the modified footstep and is represented by deforming (20) as

$$\left[\boldsymbol{a}_{sup}^T, \quad \boldsymbol{a}_{sup}^T S\right]\begin{bmatrix}\boldsymbol{w}_{xy} \\ \boldsymbol{w}_f\end{bmatrix} < d_{sup}. \tag{32}$$

For the double supporting phase, the supporting region moves nonlinearly due to rotation. Fig. 3 (a) illustrates the variation of a double supporting region depending on the modification of foot placement. The modification rotates boundaries of the supporting region, so the variation is nonlinear. We approximate the non-linearity as parallel translation with time shift as shown in Fig. 3 (b). Equation (20) depends on time, so linear constraints can be linearly translated with time evolution as

$$\left[\boldsymbol{a}_{sup}^T, \quad -t'\boldsymbol{a}_{sup}^T S\right]\begin{bmatrix}\boldsymbol{w}_{xy} \\ \boldsymbol{w}_f\end{bmatrix} < d_{sup}. \tag{33}$$

Although the approximation represents wider regions than true regions, the approximation worked in an experimental setting (i.e. ZMP trajectory did not leave true regions). It seems that the desired position constraint and the minimization of squared acceleration contributed to the result. To be more rigorous, Dimitrov et al. proposed the method that approximates the supporting region in the double supporting phase as a translation of a single footprint with time shift [24].

3) Desired position constraint depending on foot placements: The desired position associated with a footstep should be moved according to the modification of the footstep. The desired position constraint (24) is redefined by

$$\left[1, t'(t_{des}), \ldots, t'^n(t_{des}), -S\right]\begin{bmatrix}\boldsymbol{w}_i \\ \boldsymbol{w}_f\end{bmatrix} = p_{des}. \tag{34}$$

4) The QP problem of foot placement optimization: Changing footsteps is not acceptable without any perturbations. For this purpose, Dimitrov et al. applied the l_1 plus l_2 cost function for footstep optimization [24]. The l_1 norm cost function makes the solution converge strongly to zero than the l_2 norm cost function and is known as *lasso* in machine learning literature. One of the methods of l_1 norm optimization is the parameterization by non-negative variables: $\Delta x_{foot} = w_x^+ - w_x^- \ s.t. \ w_x^+ \geq 0, \ w_x^- \geq 0$. The linearly constrained quadratic programming problem for both ZMP and foot placements is defined by

$$\begin{aligned}\min \quad & \boldsymbol{w}^T H \boldsymbol{w} + \frac{\mu}{2}\boldsymbol{w}_f^T \boldsymbol{w}_f + \epsilon \mathbf{1}^T \boldsymbol{w}_f \\ & = \begin{bmatrix}\boldsymbol{w} \\ \boldsymbol{w}_f\end{bmatrix}^T \begin{bmatrix}H & \mathbf{0} \\ \mathbf{0} & \mu I\end{bmatrix}\begin{bmatrix}\boldsymbol{w} \\ \boldsymbol{w}_f\end{bmatrix} + \begin{bmatrix}\mathbf{0} \\ \epsilon\mathbf{1}\end{bmatrix}^T \begin{bmatrix}\boldsymbol{w} \\ \boldsymbol{w}_f\end{bmatrix} \\ \text{s.t.} \quad & \boldsymbol{w}_f \geq 0 \\ & (31), (32), (33), (21), (22), (23), (34),\end{aligned} \tag{35}$$

where μ and ϵ are tuning parameters, and the selection matrix is defined as

$$S = \begin{bmatrix}1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1\end{bmatrix}.$$

V. STABILIZING CONTROLLER

A. Inclination Error Controller

The error between the reference calculated by the planner and the actual state is compensated by a feedback controller.

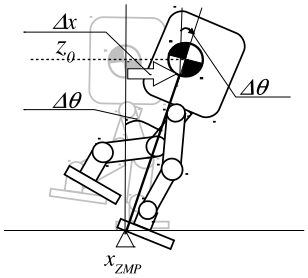

Fig. 4: Upper body inclination error

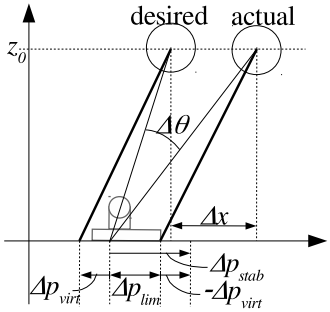

Fig. 5: Error model of LIPs

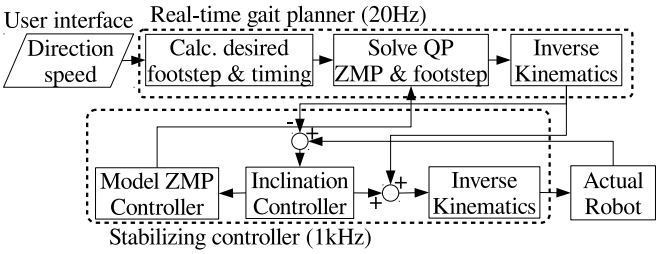

Fig. 6: Software system

As shown in Fig. 4, we assume that the inclination of the upper body estimated from the IMU sensor can be treated as a horizontal displacement of an inverted pendulum. The modified ZMP Δp_{stab} to recover the state error Δx is computed with the PD control law as follows

$$\Delta p_{stab} = -k_x \Delta x - k_v \Delta \dot{x}. \tag{36}$$

To determine feedback gains k_x and k_v, assigning one pole of the system to ω makes the best COM-ZMP regulator [25].

B. Model ZMP Controller

The inclination error illustrated in Fig. 4 is modeled by the difference between two LIPs as shown in Fig. 5. If the robot can completely track the modified ZMP Δp_{stab}, the inclination error Δx would converge to zero. However, ZMP is saturated by the supporting region. Instead of moving the actual COG, the desired COG moves closer to the actual COG to reduce the residual error caused by the saturation. It is performed by modifying ZMP of the desired LIP calculated as:

$$\Delta p_{virt} = -(\Delta p_{stab} - \Delta p_{lim}), \tag{37}$$

where Δp_{lim} denotes the limited ZMP. We call Δp_{virt} the virtual ZMP, because the ZMP is not applied to the actual robot and is only applied to the desired model. The virtual ZMP Δp_{virt} accelerates the desired trajectory of COG.

The long-term stability of the desired trajectory is guaranteed by the gait planner. If additional torque Δp_{virt} is simply added to the model, guaranteed stability will be broken. Therefore, we apply virtual ZMP to the gait planner by adding the stability criterion (18) as

$$b_{cyc} = -\xi(0) + e^{-\omega t_{e+1}}(\xi_{target} - \xi_{virt}(t_{e+1})) \tag{38}$$

where ξ_{virt} denotes DCM depending on virtual ZMP. The trajectory of virtual ZMP is calculated by a prediction by (1) to the end of time t_{e+1}.

VI. EXPERIMENT

A. Experimental Hardware

As shown in Fig. 1, to verify our proposed algorithms, we conducted experiments on our legged robot platform E2-DR. The E2-DR was designed to walk on two legs in narrow spaces of real plants. The height and weight of the robot were equal to 1.68 m and 85 kg, respectively. The robot has a total of 33 dofs including 8 dofs for each arm, 6 dofs for each leg, 2 dofs for its torso, 1 dof for each end-effector and 1 dof for its head. Further details are shown in [26].

B. Software Implementation

Fig. 6 illustrates our software system. The walking direction and speed were input as commands through an external PC. Two processes were run on an onboard computer. One was a real-time gait planner and the other was a stabilizing controller. In the gait planner, the desired footsteps and timing was calculated from the command direction and speed by using a heuristic conversion. Next, the ZMP trajectory and foot placements were optimized by solving the QP problem described in section IV. Finally, the reference of COG, ZMP and all joint angles were calculated by inverse kinematics (IK).

The stabilizing controller determined the modified ZMP Δp_{zmp} and the virtual ZMP Δp_{virt} as shown in section V. The modified ZMP was converted into position and orientation of the robot's feet. Then, all joint angles were calculated by IK. Calculations of IK were duplicated in both the gait planner and the stabilizing controller. Because the software was designed for common usage for both dynamic motion in a wide area and quasi-static motion in narrow spaces such as climbing ladders [26]. For quasi-static motion, kinematic difficulties are likely infeasible rather than dynamic difficulties. Therefore, we developed an optimization based IK solver for the offline gait planner. For dynamic motion in a wide area, kinematic difficulties might not occur or can be avoided by simple methods such as the limitation of footsteps. We may be able to skip the IK calculation in the real-time gait planner, but it is currently computed.

C. Result

We conducted a push recovery experiment in which the robot was walking on the spot and the robot's back was pushed by a bar equipped with a force sensor. Fig. 7 shows the result of the experiment in which the robot walked 44 steps. The duration of one step was 0.75 sec with 0.25 sec for the double supporting phase and 0.5 sec for the swing phase. Fig. 7 (a) shows the applied force measured by the attached force sensor on the bar to push the robot. The measured force data was shorter than other data measured from sensors attached to the robot, because the measurement of the applied

force was started manually. The applied forces at each impact was quite large (> 500 N) because the tip of the bar was constructed from rigid urethane. The robot was pushed 9 times and Fig. 7 (b) shows that the inclination error in the roll direction increased due to the disturbances. As shown in Fig. 7 (c), except for the last 2 pushes, the calculated virtual input in the sagittal direction was around zero and the result of planning modified neither the reference ZMP trajectory nor the footstep placement. In the duration, as shown in Fig. 7 (d), actual ZMP went forward from the reference ZMP by the stabilizing controller.

Fig. 8 focuses on the second last push. The applied force is shown in Fig. 8 (a) and had an impulse of 16.5 Ns. As shown in Fig. 8 (b) and (c), the inclination error increased more than 2 degrees and the absolute virtual input was larger than 0.01 m. To compensate for the virtual input, the reference ZMP went forward 0.013 m as a result of optimization.

Fig. 9 focuses on the last push. The impulse of the applied force was 20.1 Ns. As shown in Fig. 9 (b) and (c), the inclination error increased more than 3 degrees and the absolute virtual input was close to 0.04 m. Then, as shown in Fig. 9 (d), the foot placement Δx_{foot} was modified twice about 0.17 m and 0.07 m as a result of optimization. Fig. 7 (f) shows the position of the foot in the sagittal direction went forward $0.17 + 0.07 = 0.24$ m.

As shown in Fig. 7 to 9, our proposed algorithm realized a hierarchical strategy for disturbance compensation which consists of no re-planing for weak disturbances, with re-planning of the ZMP reference for middle-level disturbances and with re-planning the foot placement for large disturbances. The foot placement was largely modified as 0.24 m by the difference of the impulse between modified and non modified which equals $20.1 - 16.5 = 3.6$ Ns. This sudden modification was intentionally formulated by the l_1 norm cost function described in section IV-C.4, because the modification of foot placements with respect to small disturbances makes the robot look unreliable.

The calculation time maximum was 0.013 sec and the average was 0.010 sec for all calculations of the real-time planner including a maximum of 0.005 sec and an average of 0.002 sec for the simultaneous optimization of ZMP, COG and foot placements using an Intel Core i7 2.4 GHz CPU. In the implementation, the planner included the optimization based IK to commonly use source code with offline motion generation which was designed for kinematically difficult movements. The calculation of IK is not necessary in the real-time planner, because kinematic difficulties rarely occur when a robot is walking in a wide area. Therefore, the sampling time of the real-time planner can be faster by eliminating the IK calculation.

VII. Conclusion

In this paper, we derived an analytical solution of DCM for a general input function and proposed a novel QP problem for simultaneous optimization of ZMP and foot placements based on the analytical solution. In the definition of the QP problem, ZMP trajectory was parameterized by the piecewise

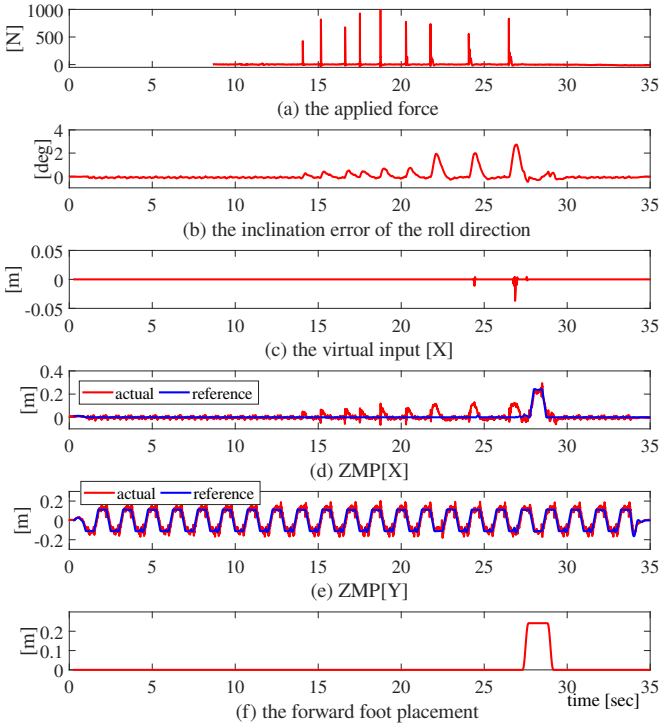

Fig. 7: The result of push recovery experiment.

relative-time polynomial and ZMP trajectory was optimized through parameter optimization. Our approach can handle different time window lengths only by the parameter T_i without any variation of the computational cost which only depends on the number of polynomial pieces and the degree of the polynomial. This functionality may be better when step timing is changed than other model predictive control methods with non-parametric input such as [24] which must vary the number of optimizing parameters by changing time.

The algorithm was validated by a push recovery experiment on real hardware. In the experiment, the algorithm realized a hierarchical strategy for disturbance compensation. The experiment was not wide enough to investigate the limitation and other problems of our methods. Therefore, we need further experiments such as changing the direction of the applied force and the walking speed.

As future work, we will apply faster implementation by removing the IK calculation from the real-time gait planner. Additionally, incorporating the proposed QP problem to the timing modification and the dynamic motion transition which was proposed by [15] will likely improve the performance of push-recovery.

References

[1] K. Nishiwaki and S. Kagami, "Simultaneous planning of CoM and ZMP based on the preview control method for online walking control," in *IEEE-RAS International Conference on Humanoid Robots*, 2011, pp. 745–751.

[2] J. Urata, K. Nishiwaki, Y. Nakanishi, O. Okada, S. Kagami, and M. Inaba, "Online walking pattern generation for push recovery and minimum delay to commanded change of direction and speed," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Ieee, oct 2012, pp. 3411–3416.

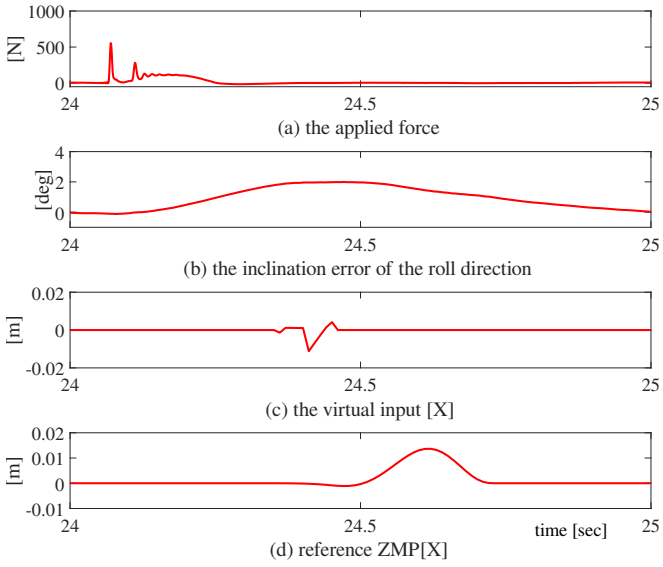

Fig. 8: The result focused on ZMP modified pushing

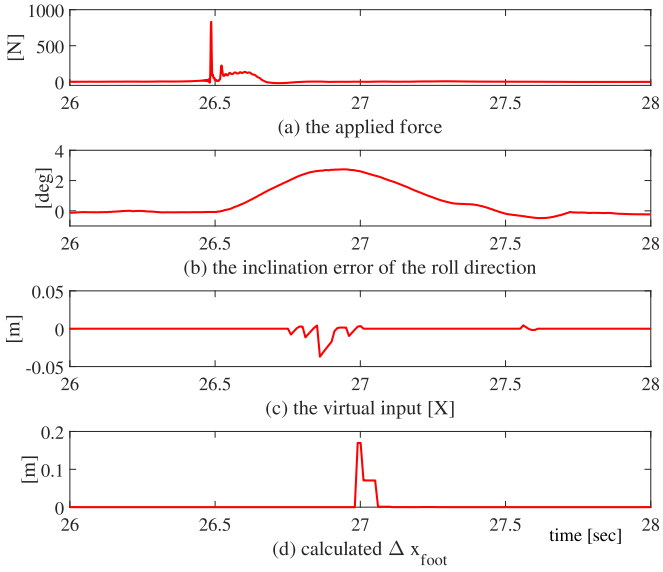

Fig. 9: The result focused on the footstep modified pushing

[3] M. Morisawa, F. Kanehiro, K. Kaneko, N. Mansard, J. Sola, E. Yoshida, K. Yokoi, and J. P. Laumond, "Combining suppression of the disturbance and reactive stepping for recovering balance," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 3150–3156.

[4] S. Feng, X. Xinjilefu, C. G. Atkeson, and J. Kim, "Robust Dynamic Walking Using Online Foot Step Optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 5373–5378.

[5] R. Wittmann, A.-C. Hildebrandt, D. Wahrmann, F. Sygulla, D. Rixen, and T. Buschmann, "Model-Based Predictive Bipedal Walking Stabilization," in *IEEE-RAS International Conference on Humanoid Robots*, 2016.

[6] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot -1st report: Walking gait pattern generation-," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1084–1091.

[7] T. Takenaka, T. Matsumoto, T. Yoshiike, and S. Shirokura, "Real time motion generation and control for biped robot -2nd report: Running gait pattern generation-," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1092–1099.

[8] T. Takenaka, T. Matsumoto, T. Yoshiike, T. Hasegawa, S. Shirokura, H. Kaneko, and A. Orita, "Real time motion generation and control for biped robot -4th report: Integrated balance control-," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1601–1608.

[9] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture Point: A Step toward Humanoid Push Recovery," in *IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 200–207.

[10] J. Pratt, T. Koolen, T. de Boer, J. Rebula, S. N. C. Cotton, J. Johnson, and M. Peter, "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, no. 10, pp. 1117–1133, 2011.

[11] J. Englsberger, C. Ott, and A. Albu-sch, "Three-dimensional bipedal walking control using Divergent Component of Motion," in *IEEE International Conference on Intelligent Robots and Systems*, 2013, pp. 2600–2607.

[12] M. A. Hopkins, D. W. Hong, and A. Leonessa, "Humanoid Locomotion on Uneven Terrain Using the Time-Varying Divergent Component of Motion," in *IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 0–6.

[13] R. J. Griffin, A. Asbeck, and A. Leonessa, "Disturbance Compensation and Step Optimization for Push Recovery," in *IEEE International Conference on Intelligent Robots and Systems*, 2016, pp. 5385–5390.

[14] J. Englsberger, G. Mesesan, and C. Ott, "Smooth trajectory generation and push-recovery based on Divergent Component of Motion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 4560–4567.

[15] T. Kamioka, H. Kaneko, M. Kuroda, C. Tanaka, S. Shirokura, M. Takeda, and T. Yoshiike, "Dynamic Gait Transition between Walking, Running and Hopping for Push Recovery," in *IEEE-RAS International Conference on Humanoid Robots*, 2017, pp. 1–8.

[16] W. L. Brogan, *Modern Control Theory (3rd Ed.)*. Prentice-Hall, Inc., 1991.

[17] D. C. Witt, "A Feasibility Study on Powered Lower-limb Prostheses," in *Proceedings of the Institution of Mechanical Engineers*, 1968, pp. 18–25.

[18] J. Englsberger and C. Ott, "Bipedal walking control based on Capture Point dynamics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 4420–4427.

[19] T. Koolen, T. Boer, J. Rebula, J. Pratt, and A. Goswami, "Capturability-Based Analysis and Control of Legged Locomotion , Part 1 : Theory and Application to Three Simple Gait Models," *International Journal of Robotics Research*, vol. 31, no. 9, 2012.

[20] B. C. Kuo, *Digital Control Systems*, 2nd ed. Oxford University Press, Inc., 1992.

[21] R. Tedrake, S. Kuindersma, R. Deits, and K. Miura, "A closed-form solution for real-time ZMP gait generation and feedback stabilization," in *IEEE-RAS International Conference on Humanoid Robots*, 2015, pp. 936–940.

[22] T. Kamioka, T. Watabe, M. Kanazawa, H. Kaneko, and T. Yoshiike, "Dynamic gait transition between bipedal and quadrupedal locomotion," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 2195–2201.

[23] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, "Learning, planning, and control for quadruped locomotion over challenging terrain," *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 236–258, nov 2010.

[24] D. Dimitrov, A. Paolillo, and P. B. Wieber, "Walking motion generation with online foot position adaptation based on 1 - and ∞ -norm penalty formulations," in *IEEE International Conference on Robotics and Automation*, 2011.

[25] T. Sugihara, "Standing stabilizability and stepping maneuver in planar bipedalism based on the best COM-ZMP regulator," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 1966–1971.

[26] T. Yoshiike, M. Kuroda, R. Ujino, H. Kaneko, H. Higuchi, S. Iwasaki, Y. Kanemoto, M. Asatani, and T. Koshiishi, "Development of experimental legged robot for inspection and disaster response in plants," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 4869–4876.