2017 IEEE-RAS 17th International Conference on
Humanoid Robotics (Humanoids)
Birmingham, UK, November 15-17, 2017

# Dynamic Gait Transition between Walking, Running and Hopping for Push Recovery

Takumi Kamioka[1], Hiroyuki Kaneko[2], Mitsuhide Kuroda[2], Chiaki Tanaka[2],
Shinya Shirokura[2], Masanori Takeda[2] and Takahide Yoshiike[1]

***Abstract*— Re-planning of gait trajectory is a crucial ability to compensate for external disturbances. To date, a large number of methods for re-planning footsteps and timing have been proposed. However, robots with the ability to change locomotion from walking to running or from walking to hopping were never proposed. In this paper, we propose a method for re-planning not only for footsteps and timing but also locomotion mode which consists of walking, running and hopping. The re-planning method of locomotion mode consists of parallel computing and a ranking system with a novel cost function. To validate the method, we conducted push recovery experiments which were pushing in the forward direction when walking on the spot and pushing in the lateral direction when walking in the forward direction. Results of experiments showed that the proposed algorithm effectively compensated for external disturbances by making a locomotion transition.**

## I. INTRODUCTION

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

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

Some methods for real-time re-planning have been proposed in this decade. Nishiwaki et al. proposed real-time planning of the center of gravity (COG) and the zero moment point (ZMP) trajectory based on linear inverted pendulum models (LIPs) [1], and Urata et al. extended it for footstep re-planning by a singular problem [2], [3]. Morisawa et al. proposed a re-planning method of the footstep based on an analytical solution by polynomial ZMP trajectory [4]. Santacruz and Nakamura proposed a footstep re-planning method based on boundary condition optimization [5]. Recently, some optimal control algorithms for footstep re-planning were formulated based on LIPs [6], [7].

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 the divergent component of motion (DCM) from an analysis of its dynamics [8] and proposed methods for real-time re-planning for both the COG (for walking [8] and running [9]) and footsteps [10]. Equivalent values to the DCM have been analyzed by Hof et al. as the extrapolated center of mass (XCoM) which was derived by the point to converge the center of mass [11], and by Pratt et al. as the capture point (CP) which was derived by the supporting point to converge the orbital energy [12], [13]. The DCM was extended to three dimensions (3D) and a real-time planner and controller based on the 3D DCM was proposed without direct planning of the COG [14], [15]. Griffin et al. proposed a real-time footstep optimization method based on the 3D DCM [16]. The amount of research for real-time planning of step timing has been far less than research for footsteps. One difficulty to modify step timing is non-linearity between the step timing and the COG. Approaches for the difficulty were to solve a non-linear optimization problem [17], [18], and linear approximation based on the DCM [19].

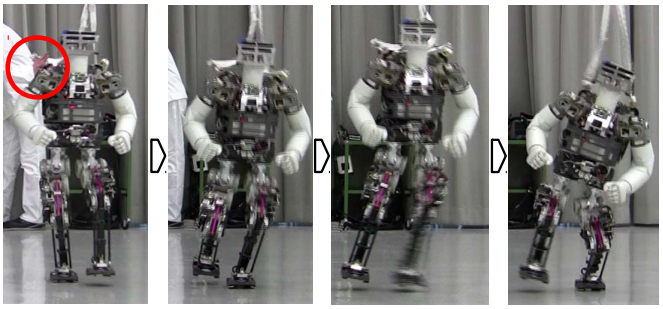

Fig. 1. A disturbance compensation by hopping motion.

There are some reports of experiments of push recovery for real robots [2]-[6], [13]. These studies were pushing only when robots were walking on the spot and most of the studies were pushing only in the forward direction. Basically, bipedal robots have a hard time keeping balance with respect to disturbances in the lateral direction when walking in the forward direction. In that case, the robot has to quickly change the walking direction from the forward to the lateral direction, but bipedal locomotion is intrinsically hard to move in the lateral direction because the supporting leg blocks the swing leg going in the lateral direction. One possibility to address this problem is a transition from walking to hopping by the supporting leg in the lateral direction as shown in Fig. 1. Additionally, the robot may compensate for larger disturbances if the robot has the capability to change its locomotion mode from walking to

[1]Honda Research Institute Japan Co., Ltd., 8-1 Honcho, Wako, Saitama, Japan. takumi_kamioka@n.f.rd.honda.co.jp
[2]Honda R&D Co., Ltd. 8-1 Honcho, Wako, Saitama, Japan.

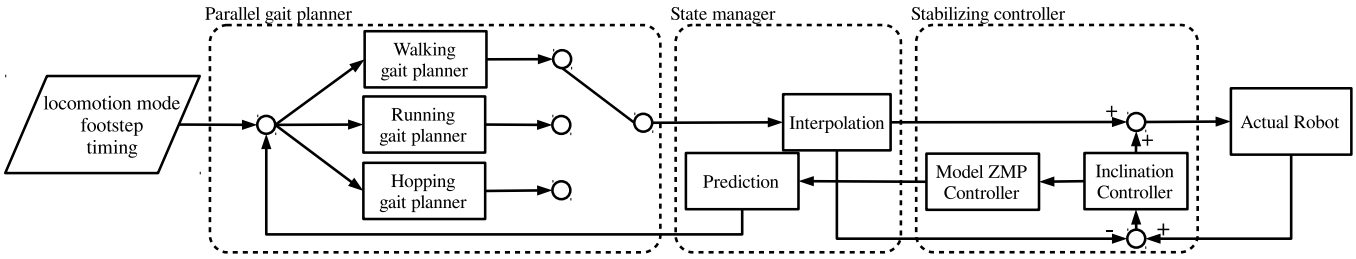


Fig. 2. System overview

running, because the length of a footstep when running can be longer than when walking. Switching technology from standing to other motions has been studied such as to stepping [20], [21] and to hopping [22]. However, there is no switching technology from walking to another motion.

We have been developing ASIMO [23], a humanoid robot that has three distinctive modes of locomotion: walking, running, and hopping on two legs, and can make a dynamic transition between walking, running, and hopping according to external commands based on unified planning and control algorithms. In this paper, we propose an extended planning algorithm for push recovery utilizing gait modification of not only for footsteps and timing but also locomotion mode. The modification of the locomotion mode is based on a parallel computation which calculates both walking, running and hopping and a ranking system with a novel cost function, and the modification of footsteps and timing is based on an approximated gradient descent algorithm.

The paper is organized as follows: Section II gives an overview of the proposed algorithms. The feedback controller which was originally proposed for ASIMO is given in Section III. The theory to calculate the stable trajectory of an inverted pendulum and its DCM is described in Section IV. The method to select the best locomotion mode is explained in Section V. Results of hardware experiments are shown in Section VI. Finally, we conclude the study and suggest future work in Section VII.

## II. SYSTEM OVERVIEW

Fig. 2 shows a schematic diagram of the proposed planning and control system. The system is divided into a parallel gait planner, a state manager and a stabilizing controller. These three components run different processes with different sampling periods. The parallel gait planner calculates a long-term stable trajectory based on the simplified dynamics model described in section IV according to the command which consists of desired locomotion modes, footstep positions and timing. The gait trajectory is calculated according to the locomotion mode which is defined as either walking, running or hopping in this paper. The planner simultaneously calculates multiple combinations of locomotion modes and selects the best locomotion mode according to the current state of the robot.

The stabilizing controller determines the angles of all joints at the moment to track the desired trajectory planned by the gait planner and is basically the same as our previous paper [10]. In this paper, we briefly review the key technology of the inclination controller and the model ZMP controller in section III.

To reduce computational cost, the parallel gait planner generates a long term scenario with a coarse sampling period and the stabilizing controller determines the joints angles at the moment with fast sampling. There are two problems caused by this time inconsistency: 1) the reference of the stabilizing controller holds between the sampling periods of the gait planner and 2) the reference will be discontinuous if the gait planner generates it from the current state of the robot. To solve these problems, the state manager works in the intermediate sampling period. For the first problem, the state manager interpolates the received trajectory from the parallel gait planner, and for the second problem, the state manager predicts the state at the start time of the planner from the newest state of the controller. Koenemann et al. pointed out the same problem of time inconsistency and also applied the prediction module [24]. For the experiment shown in section VI, sampling periods of the parallel gait planner, the state manager and the stabilizing controller were set at $20$ msec, $5$ msec and $0.5$ msec, respectively.

## III. GENERALIZED 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. As shown in Fig. 3, a horizontal displacement is approximated by $\Delta x \approx z_0 \Delta\theta$. The stabilizing torque $\tau_{stab}$ to recover the state error $\Delta x$ is computed with PD control law as follows

$$\tau_{stab} = -k_x \Delta x - k_v \Delta \dot{x}. \quad (1)$$

To determine the feedback gain $k_x$ and $k_v$, there are some studies based on LIPs. We applied a method based on the DCM with variable height of COG. The calculated stabilizing torque is distributed to the joint angle of legs and the rotation of the upper body. The distributing ratio depends on the locomotion phase (i.e. single or double supporting phase) [10].

### B. Model ZMP Controller

In the paper [10], the model ZMP controller is explained as that the stabilizing torque may not be generated because of

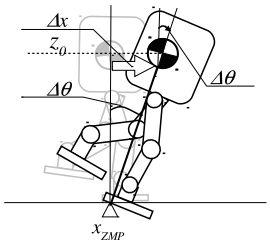


Fig. 3. Upperbody inclination error

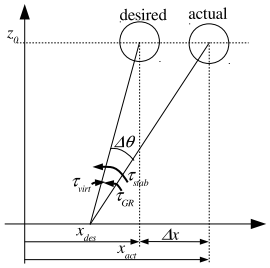


Fig. 4. Error model of linear inverted pendulum models

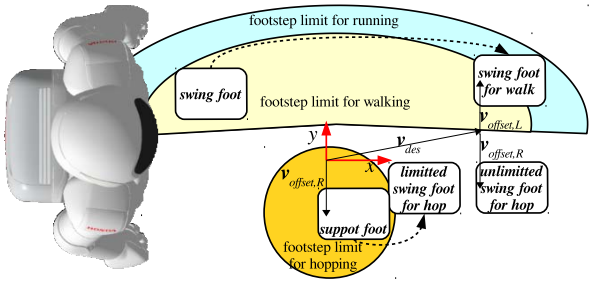


Fig. 5. Representations of desired footstep commands and limitations for locomotion mode. The two dimensional position of footsteps are represented by the sum of a parameter vector $\boldsymbol{v}_{des}$ and constant vectors $\boldsymbol{v}_{offset,i}, i \in R, L$. The parameter vector is commonly used between walking, running and hopping and the constant vector is determined by the length between hip joints. Positions of footstep are limited according to locomotion mode. The limit for running is wider than walking and only the limit for hopping allows to step on supporting foot.

a limitation, the model ZMP controller accelerates the upper body to reduce the residual error of stabilizing torque caused by the limitation. In this paper, we briefly introduce another aspect of the model ZMP controller.

The inclination error described in Fig. 3 is modeled by the difference between two LIPs as shown in Fig. 4. The equation of motion of the LIP with torque $\tau$ is

$$\ddot{x} = \omega^2 x + \frac{1}{mz}\tau, \tag{2}$$

where $m, z$ and $\omega$ denote the mass of the pendulum, the height of COG and the natural frequency, respectively. If the robot can completely generate the stabilizing torque $\tau = \tau_{stab}$, the error $\Delta x$ would converge to zero. However, the torque around ZMP is saturated, because the torque is equivalent to the ZMP according to the following conversion:

$$p_\tau = -\frac{1}{mg}\tau, \tag{3}$$

where $g$ denotes the gravity constant. Therefore, $\tau$ is limited proportional to the supporting region. Here, we call the ground reaction torque $\tau_{GR}$ the saturated stabilizing torque, because it will be generated around a point on the ground. Instead of moving the actual COG, the desired COG moves closer to the actual COG to reduce the residual error caused by the saturated torque. It is performed by applying a torque $\tau_{virt} = -(\tau_{stab} - \tau_{GR})$ which we call the virtual torque, because the torque is not applied to the actual robot and is only applied to the desired model. The virtual torque $\tau_{virt}$ accelerates the desired trajectory of COG.

The long-term stability of desired trajectory is guaranteed by the gait planner. If additional torque $\tau_{virt}$ is simply added to the model, guaranteed stability will be broken. Therefore, we apply virtual torque in the gait planner and generate stable trajectory including the virtual torque.

## IV. GENERALIZED GAIT PLANNER

### A. Overview

Fig. 6 illustrates a schematic diagram of the gait planner which is a general method for both walking, running and hopping proposed by us [9]. The planner receives the desired virtual torque described in section III-B and the command which consists of the desired locomotion mode, footsteps and timing. Footsteps and timing of the command are common for walking, running and hopping gait planners. However, as shown in Fig. 5, actual positions of swing feet are different depending on the locomotion mode because of the relative representation and footstep limits. These heuristically determined limits avoid kinematic constraints such as collisions between both legs. Furthermore, footstep timing is also limited according to the locomotion mode. For example, the duration of the single supporting phase when hopping should be shorter than when walking due to motor torque limitations. Long-term dynamic stability is guaranteed based on the linear time-variant inverted pendulum models (LTVIPs) and its DCM.

To calculate the proposed LTVIPs, the acceleration in height for COG, the intermediate foot trajectory of the footstep and the desired ZMP trajectory were heuristically determined from the command. By using the LTVIPs and the DCM, we can estimate the desired torque to guarantee long-term stability. The torque is limited by a support polygon and friction from the ground and is distributed to the ZMP and the acceleration of the flywheel which attached LTVIPs. If the torque can not be realized by the distribution, the footstep and the timing is modified. The detail is shown in following subsections.

### B. Divergent Component of Motion for LTVIPs

We proposed the LTVIPs for running gait planning [9] and gait transition between bipedal and quadruped locomotion [25]. The state equation of the LTVIPs is derived from (2) with the time-variant height of COG as follows:

$$\frac{d}{dt}\boldsymbol{x}(t) = \tilde{A}(z(t), \ddot{z}(t))\boldsymbol{x}(t) + \tilde{B}(z(t))\boldsymbol{u}(t), \tag{4}$$

where

$$\tilde{A}(z(t), \ddot{z}(t)) = \begin{bmatrix} 0 & 1 \\ \omega(t)^2 & 0 \end{bmatrix}, \quad \tilde{B}(z(t)) = \begin{bmatrix} 0 \\ -\frac{1}{mz(t)} \end{bmatrix},$$

$$\boldsymbol{x} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} \tau \end{bmatrix}, \quad \omega(t) = \sqrt{\frac{g + \ddot{z}(t)}{z(t)}}.$$

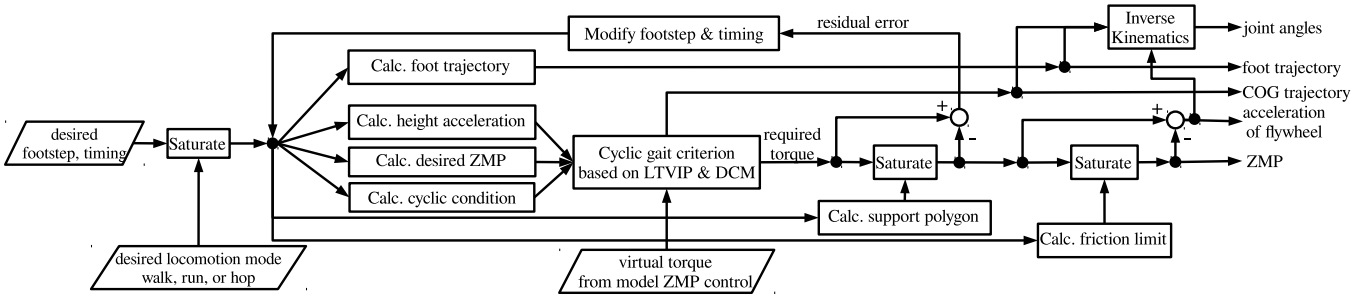


Fig. 6. Generalized gait planner

Furthermore, we can obtain the discrete time state equation by discretization of (4) as

$$\boldsymbol{x}[k+1] = A[k]\boldsymbol{x}[k] + B[k]\boldsymbol{u}[k] \tag{5}$$

where, time-variant elements of $A[k]$ and $B[k]$ were defined in [9]. The DCM is generally defined by the unstable part of a linearly divided state [8], [14]. The DCM of LTVIPs is defined with certain time steps $K$. From (5), we can obtain the state after $K$ by

$$\boldsymbol{x}[K] = \phi(K,0)\boldsymbol{x}[0] + \sum_{i=0}^{K-1} \phi(K,i+1)B[i]u[i] \tag{6}$$

where

$$\phi(k,j) = \begin{cases} A[k-1]A[k-2],\ldots,A[j] & \text{if } k > j \\ I & \text{otherwise.} \end{cases}$$

As a result of eigenvalue decomposition of $\phi(K,0)$, we let $\lambda_p$ and $\lambda_q$ be eigenvalues of $\phi(K,0)$, and we let $\Gamma_K$ be a square matrix that has eigenvectors in each column. If $\boldsymbol{u}[k] = \boldsymbol{0}$ for all $k$, the following linear transformation can be defined:

$$\begin{aligned} \begin{bmatrix} p[k] \\ q[k] \end{bmatrix} &= \Gamma_K^{-1}\boldsymbol{x}[k] \\ &= \Gamma_K^{-1}\phi(K,0)\Gamma_K\{p[0],q[0]\}^T \\ &= diag(\lambda_p,\lambda_q)\{p[0],q[0]\}^T. \end{aligned} \tag{7}$$

By the $n$-th repetition of the $\omega$ trajectory, $p[nk]$ and $q[nk]$ converge to $\lambda_p^n p[0]$ and $\lambda_q^n q[0]$, respectively. Because $\lambda_p < 1$ and $\lambda_q > 1$, we defined $p$ as the convergent component and $q$ as the divergent component of motion for the LTVIPs. We would like to note that this definition is time varying and different from another definition of a time-varying DCM proposed by Hopkins et al [15].

### C. Stable Trajectory Generation

Our basic idea of the stability criterion has been a *cyclic gait*, which means the same gaits connect sequentially [8]. Fig. 7 shows an example of a cyclic gait. In the figure, four footsteps and the desired ZMP trajectory represented by a solid line are divided into two types which are the target gait and the cyclic gait. The target gait starts from the current step to some arbitrary steps (two steps are selected in Fig. 7). The cyclic gait includes an arbitrary number of steps starting from

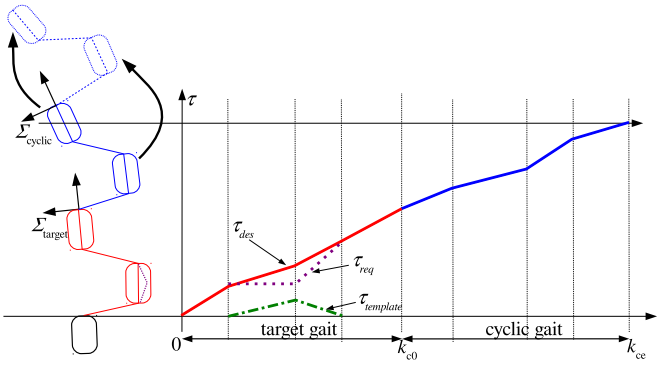


Fig. 7. An example of cyclic gait. The target gait and the cyclic gait are represented by the red and blue solid line, respectively. The required torque (magenta dotted line) is defined by summation of the desired torque (red solid line) and the template torque (green dashed-dotted line).

the end of the target gait. In the cyclic gait criterion, footsteps and the desired ZMP trajectory in the cyclic gait could sequentially continue without any modification. The target gait is the period to connect the cyclic gait by modifying footsteps and the ZMP trajectory. To satisfy this criterion, the states at both the start and the end of the cyclic gait should be relatively equal. We chose two steps for both the target and the cyclic gait to be consistent with the number of feet. For quadrupedal locomotion, four steps were chosen as the number of cyclic gait [25]. In the implementation, the number of steps of the command to the gait planner is three by copying the first cyclic gait to the second cyclic gait.

We applied the cyclic gait criterion based on the DCM. From (6) and (7), the DCM from $k_0$ to $k_e$ is calculated by

$$q[k_e] = \lambda_q q[k_0] + \bar{q}_u[k_e], \tag{8}$$

where

$$\bar{q}_u[k_e] = \boldsymbol{v}_q^T \sum_{i=k_0}^{k_e-1} \phi(k_e,i+1)B[i]\boldsymbol{u}[i]. \tag{9}$$

$\bar{q}_u$ is the DCM dependent on input $\boldsymbol{u}$. Let $\boldsymbol{q}[k] = \{q^x[k],q^y[k]\}^T$ be a two-dimensional vector of the DCM in the coordinate system $\Sigma_{target}$ which is associated with the cyclic gait. If the duration of the cyclic gait is defined by the start time $k_{c0}$ and the end time $k_{ce}$, the stability constraint of the cyclic gait is represented by the DCM at the start and

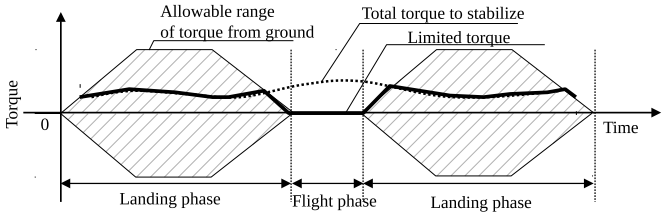


Fig. 8. Torque limitation and distibution to a flywheel

the end times with the coordinate transformation as follows:

$$R_z(\boldsymbol{q}[k_{c0}] + \boldsymbol{o}) = \lambda_q \boldsymbol{q}[k_{c0}] + \bar{\boldsymbol{q}}_u[k_{ce}]$$
$$\boldsymbol{q}[k_{c0}] = (R_z - \lambda_q I)^{-1}(\bar{\boldsymbol{q}}_u[k_e] - R_z \boldsymbol{o}), \quad (10)$$

where $R_z$ and $\boldsymbol{o}$ are, respectively, the rotation matrix and the vector for transforming from $\Sigma_{target}$ to the relatively equal coordinate system $\Sigma_{cyclic}$. For example, in Fig. 7, both coordinate systems are put on the toe of the supporting foot of each cyclic gait at the start.

By substituting (8) into (10), we obtain the equality constraint for the input trajectory during the target gait from 0 to $k_{c0}$.

$$\lambda_q \boldsymbol{q}[0] + \bar{\boldsymbol{q}}_u[k_{c0}] = (R_z - \lambda_q I)^{-1}(\bar{\boldsymbol{q}}_u[k_{ce}] - R_z \boldsymbol{o}). \quad (11)$$

Here, we call the right-hand side of (11) the target DCM, $q_{target}$. To satisfy (11), we apply a simple modification of the desired torque trajectory which was converted from the desired ZMP trajectory by (3). As shown in Fig. 7, the required torque to stabilize at time $k$ is defined as the sum of the virtual torque $\tau_{virt}$, the desired torque $\tau_{des}$ and the template torque $\tau_{template}$ as follows:

$$\tau_{req}[k] = \tau_{virt}[k] + \tau_{des}[k] + \alpha \tau_{template}[k], \quad (12)$$

where $\alpha$ is a scalar parameter. In the definition, the desired torque is modified along the template torque with parameter $\alpha$, hence we can control when and how to modify the ZMP trajectory by the design of the shape of the template torque. Consequently, the DCM $\bar{q}_u[k]$ depending on the required torque is calculated by the following summation:

$$\bar{q}_u[k] = \bar{q}_{virt}[k] + \bar{q}_{des}[k] + \alpha \bar{q}_{template}[k]. \quad (13)$$

Therefore, we obtain the parameter $\alpha$ by substituting (13) into (11)

$$\alpha = \frac{q_{target} - (\lambda_q q[0] + \bar{q}_{virt}[k_{c0}] + \bar{q}_{des}[k_{c0}])}{\bar{q}_{template}[k_{c0}]}. \quad (14)$$

The required torque calculated by (12) guarantees the cyclic gait, however a robot when jumping could not generate any torque from the ground. Therefore, as shown in Fig. 8, the torque is saturated as

$$\tau_{ground} = Limit_{fric}(\tau_{req}), \quad (15)$$

where $Limit(.)$ denotes a saturate function. We use the body rotation as a flywheel model to perform the residual torque $\tau_{req} - \tau_{ground}$. The angular accelaration of flywheel $\ddot{\theta}$ is converted to the torque $\tau_{wheel}$ by

$$\tau_{wheel} = I_{wheel} \ddot{\theta}, \quad (16)$$

where $I_{wheel}$ denotes the inertia matrix of the flywheel. The acceleration of the flywheel is calculated by

$$\tau_{wheel} = \tau_{req} - \tau_{ground} \quad (17)$$
$$\ddot{\theta} = I_{wheel}^{-1}(\tau_{req} - \tau_{ground}). \quad (18)$$

Our robot does not have joints at the position of COG around the $x$ and $y$ axis. So, we approximate the angle of the hip joints to the flywheel.

### D. Footstep and timing modification

As shown in (3), the input torque from ground $\tau_{req}$ has a limitation due to the support polygon. If a large disturbance occurs, the required torque to stabilize it will exceed the limit of the support polygon. The residual error defined by $\tau_{err} = \tau_{req} - Limit_{support}(\tau_{req})$ can be reduced by solving the following optimization problem with footstep and timing:

$$\min_{\boldsymbol{f}_i, T_i} \frac{1}{2} \left(\tau_{err,x}^2 + \tau_{err,y}^2\right) \quad (19)$$

where, $\boldsymbol{f}_i$ and $T_i$ denotes the two dimensional position of footstep and duration of the $i$th step, respectively. The modification of the footstep position $\boldsymbol{f}_i$ effects heuristically determined desired ZMP trajectory and desired torque $\tau_{des}$, and the modification of step duration increases or decreases the start time of cyclic gait $k_{c0}$. The optimization problem is not easy to solve, because the calculation of the error $\tau_{err}$ is nonlinear to both the footstep and the duration, and is not enough fast to make frequent calculations. So, we apply the approximated gradient descent algorithm. The approximated gradient is calculated as follows:

$$\Delta \boldsymbol{f}[i] = \frac{\tau_{err}(\boldsymbol{f}[i]) - \tilde{\tau}_{err}(\boldsymbol{f}[i] + d\boldsymbol{f})}{d\boldsymbol{f}}, \quad (20)$$
$$\Delta T[i] = \frac{\tau_{err}(T[i]) - \tilde{\tau}_{err}(T[i] + dT)}{dT}. \quad (21)$$

where, $d\boldsymbol{f}$ and $dT$ are heuristically determined small values to evaluate the gradient and $\tilde{\tau}_{err}(.)$ denotes the approximated residual error calculated based on the linear time-invariant inverted pendulum proposed by the paper [8].

It is a problem that the cost function (19) reduces to zero only by the modification of the step duration, because the modification of the step duration $T$ affects the residual error of both the $x$ and $y$ axis. Therefore, we conducted modifications in the order of 1) step duration and 2) footsteps.

## V. LOCOMOTION MODE SELECTION

### A. Parallel Gait Ranking

As shown in Fig. 2, our system simultaneously calculates multiple gaits which are calculated from different locomotion modes reflecting the calculation of the acceleration of COG height, the foot trajectory, and limitations which consist of the footstep, the step duration, the support polygon and the ground friction. The best gait is selected from calculated

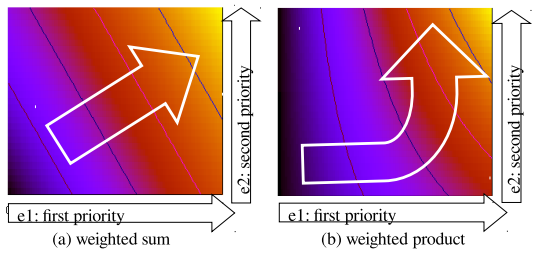


Fig. 9. Comparison of scalarization of multi-objective cost function. (a) weighted sum scalarization. (b) weighted product scalarization. The white arrow illustrates the direction to improve the cost function.

gaits in terms of a cost function. We implemented parallel computing with the message passing interface (MPI) [26].

### B. Cost function

We evaluated two criteria of robot motion: 1) stability and 2) energy consumption. The first criterion of stability is crucially important to avoid falling down and can be evaluated by $e_{stability} =$ (19). In this cost function, if there is no disturbance, all costs between walking, running and hopping gaits are equal. To avoid running all the time, we add the second criterion of energy consumption to the cost. Energy consumption is approximately estimated by the sum of the acceleration of the height of COG as follows [27]:

$$e_{energy} = \sum_{k=0}^{k_{c0}} |\ddot{z}[k]|. \tag{22}$$

The stability of locomotion is guaranteed if and only if the cost function (19) equals to zero. If there exists no locomotion mode with $e_{stability} = 0$, the robot will fall over. Therefore, we apply scalarization by the weighted product to represent a definite priority between objectives as

$$e = w_1 e_{stability}(w_2 e_{energy} + 1) \tag{23}$$

where, $w_1$ and $w_2$ denotes weight parameters. Fig. 9 shows the comparison between the weighted sum and the weighted product. Compared to weighted sum, weighted product strongly represents priority according to the non-linear gradient. This characteristic was very suitable for our hierarchical and multi-objective problem.

### C. Continuous gait connection

Discretely changing locomotion may cause discontinuous gait trajectory. To avoid this, we constrained the timing to change the locomotion mode with jumping (i.e. running and hopping). Changing to the jumping motion was enabled until a heuristically determined time in the single supporting phase. We divide the duration of a step by a pre-defined number and ratio, and call the divided duration 'step periods'. Limitations depending on time such as modification of footsteps, timing and locomotion modes are defined according to the step periods. Before the determined step period, if running or hopping locomotion is selected when walking, the trajectory of the height of COG is calculated to prepare

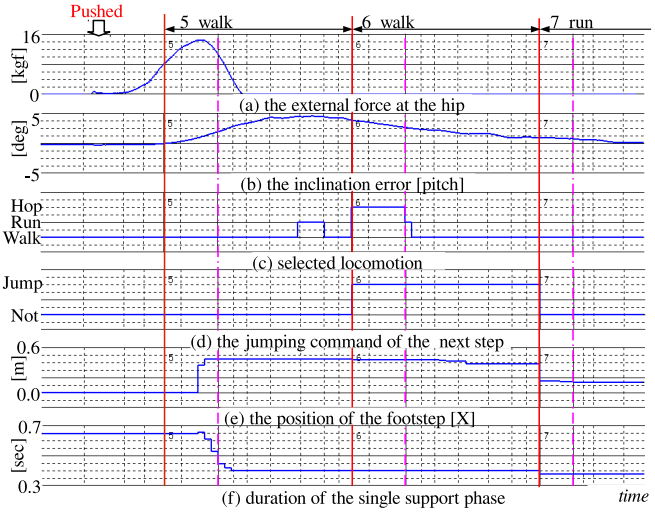


Fig. 10. Result of the forward pushing experiment.

jumping at the end of a single supporting phase and the footstep and timing is modified by the limitation according to the selected mode. If running or hopping locomotion for the next step is set after the determined time limit, the robot will jump after the next step.

## VI. EXPERIMENT

### A. Experimental Hardware

We used an experimental humanoid robot for the validation of our algorithm. Kinematic configurations and major specifications of the robot were the same as the current version of ASIMO [23]. Additionally, the robot was equipped with one-axis force sensors at the hip and both shoulders which were used only for measuring disturbances and not used for control.

### B. Push in the Forward Direction when Walking on the spot

We conducted a hardware experiment to evaluate the proposed algorithm by pushing the walking robot. Fig. 10 shows the result of the experiment in which the robot's hip was pushed from the back while walking on the spot. The horizontal axis of the figure indicates time: red solid lines, magenta dashed-dotted lines and black dashed lines indicate start time of steps, separators between double and single supporting phase and step periods, respectively. The top numbers indicate steps and the observed motion is noted at the right of the number. We pushed the robot at the end of its 4th step with a peak of 14.6 N and an impulse of 28 Ns. The proposed footstep and timing modification algorithm modified the length of the swing footstep to be longer and the duration of steps to be shorter. Nevertheless the running motion was selected at the end of the 5th step, the robot did not run because the jumping motion was limited by time in the step. Although the hopping motion was selected at the beginning of the 6th step, the running motion was selected at that time to decide jumping and the robot ran in the 7th step. The maximum inclination error was approximately 5°

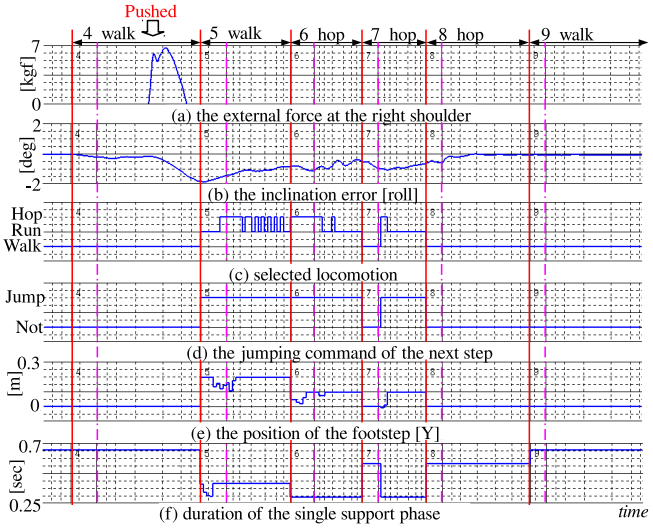


Fig. 11. Result of the side pushing experiment.

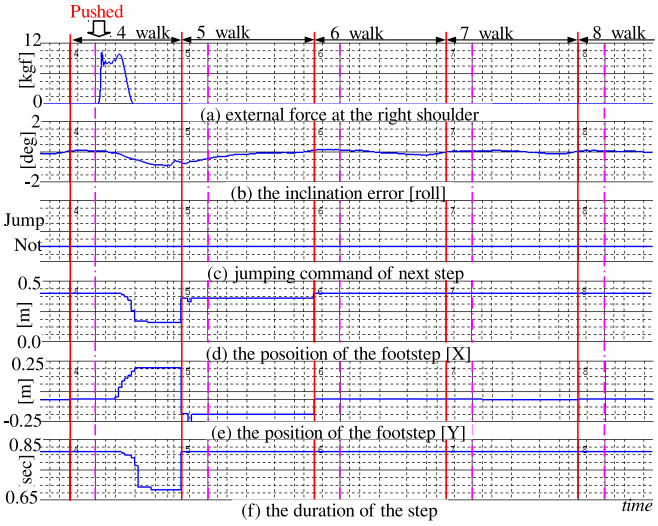


Fig. 12. Another result of the side pushing experiment

and was gradually converged to zero by the sequence of these motions.

There are very few papers which reported the amount of compensated disturbances in push recovery experiments. To the best of the author's knowledge, the largest impulse in those papers was 22 Ns reported by Urata et al. [3] which applied the footstep modification in fast sampling. The ratio of the impulse to the weight of the robot was 0.42 Ns/kg and was less than the ratio 0.58 Ns/kg in our experiment. It is likely that the proposed algorithm of only footsteps and timing without the locomotion mode selection algorithm would fall over by even weaker disturbances than 22 Ns, because the footstep modification is 20 times slower than [3]. These results suggest that the transition from walking to running effectively compensated for the large disturbance. It should be noted that a fair comparison of robustness between methods from results of push recovery experiments is very difficult because whether the robot falls over or not, does not only depend on the force pattern but also timing of when the robot is pushed.

### C. Push in the Lateral Direction when Walking

Fig. 11 shows the result of the experiment in which the robot's right shoulder was pushed from right to left when walking forward with the footsteps 0.4 m. The pushing force had a peak of 5.9 N and an impulse of 10 Ns. In the 4th step, the robot was supported by its right foot. The 5th position of the swing footstep could not change, because it was pushed when in the late phase of the step. In the next step, the lateral direction of the swing foot could not be modified to the left because the left foot was the support foot, therefore the hopping motion was selected to move left in two steps and the length of the jumped swing foot in the 6th and the 7th steps were 0.2 m and 0.1 m, respectively. Fig. 1 shows snapshots of these sequences of this experiment. This result suggests that our proposed algorithm effectively compensates for disturbances in the lateral direction which might be difficult to be compensated only by walking.

Fig. 12 shows another result for the same experiment shown in Fig. 11. Nevertheless, the pushing force which had a peak of 10.3 N and an impulse of 14 Ns was stronger than the result of Fig. 11, the disturbance was compensated only by footstep position and timing modification without any transition of the locomotion mode. The difference probably depends on the time when the disturbance was applied. As shown in Fig. 12 (a), the disturbance was applied at the first half of the single supporting phase, therefore, the motion to compensate for the disturbance could be taken quickly (i.e. the footstep position and the step time was modified in the 4th step). By contrast, the result of Fig. 11 could not take any motion to compensate for the disturbance in the 4th step and the inclination error increased exponentially, because the disturbance was applied at the last half of the single supporting phase.

Although, we show the desired push recovery motions in these experiments, many different motions which include the result of falling over were observed in the same kind of experiments. One of the reasons why the variation of results might be the oscillation of the result of the locomotion mode selection which was observed in Fig. 11 (c). For the 5th to the 6th step, the selected locomotion mode oscillated between Hop and Run. Owing to the experiment, we hypothesized that the difference of the cost function between hopping and running is small, because the footstep distance of hopping is limited to shorter ranges than the distance of running. Furthermore, the inclination error in Fig. 11 (b) is likely to converge at the beginning of the 7th step and a walking motion was selected. However, at the end of the aerial phase in the 7th step, the inclination error increased and a hopping motion was selected because a hopping motion is more difficult to stabilize due to a lack of robustness. Therefore, a more sophisticated hopping motion will increase

the robustness.

## VII. CONCLUSIONS

In this paper, we proposed an algorithm for gait re-planning not only for footsteps and timing but also locomotion mode. The algorithm for footsteps and timing modification depends on the approximated gradient descent algorithm and the algorithm for locomotion mode selection consists of parallel computing and a ranking system. The proposed algorithm was validated by real hardware experiments. The stabilizing ability only by the modification of footsteps and timing is probably not stronger than the current state of the art methods such as [3]. However, the robot compensated larger disturbance than the results of such methods by making a transition from walking to running. Furthermore, the robot kept its balance by making a transition to hopping motion with respect to the disturbance in the lateral direction when walking forward.

We have to note that the robot fell over many times in the same kind of experiments. So, further studies are needed to apply robots to real environments. As future work, 1) to speed up the sampling period of the parallel gait planner and 2) to sophisticate the hopping motion will improve performance, since the effectiveness of faster re-planning was shown in [3] and the hopping motion of our system was not sufficient. Additionally, one of the ideas is adding the transition to quadrupedal [25] to the candidate of the locomotion mode selection. We consider that the motion to make the transition to quadrupedal is the ultimate motion to compensate for large disturbances, because a human also touches the ground if extremely disturbed.

## REFERENCES


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

[2] J. Urata, K. Nishiwaki, Y. Nakanishi, K. Okada, S. Kagami, and M. Inaba, "Online Decision of Foot Placement using Singular LQ Preview Regulation," in *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, 2011, pp. 13–18.

[3] 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 *Proceedings of Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 3411–3416.

[4] 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 *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 3150–3156.

[5] C. Santacruz and Y. Nakamura, "Reactive Stepping Strategies for Bipedal Walking based on Neutral Point and Boundary Condition Optimization," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2013, pp. 3095–3100.

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

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

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

[9] 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 *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1092–1099.

[10] 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 *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1601–1608.

[11] A. Hof, M. Gazendam, and W. Sinke, "The condition for dynamic stability." *Journal of biomechanics*, vol. 38, no. 1, pp. 1–8, 2005.

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

[13] 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.

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

[15] M. A. Hopkins, D. W. Hong, and A. Leonessa, "Humanoid Locomotion on Uneven Terrain Using the Time-Varying Divergent Component of Motion," in *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 266–272.

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

[17] Z. Aftab, T. Robert, and P. B. Wieber, "Ankle, hip and stepping strategies for humanoid balance recovery with a single Model Predictive Control scheme," in *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, 2012, pp. 159–164.

[18] P. Kryczka, P. Kormushev, N. G. Tsagarakis, and D. G. Caldwell, "Online Regeneration of Bipedal Walking Gait Pattern Optimizing Footstep Placement and Timing," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 3352–3357.

[19] M. Khadiv, A. Herzog, S. A. a. Moosavian, and L. Righetti, "Step Timing Adjustment : A Step toward Generating Robust Gaits," in *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, 2016.

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

[21] S. Dafarra, F. Romano, and F. Nori, "Torque-controlled stepping-strategy push recovery: Design and implementation on the iCub humanoid robot," in *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, 2016, pp. 152–157.

[22] K. Yamamoto, "Control strategy switching for humanoid robots based on maximal output admissible set," *Robotics and Autonomous Systems*, vol. 81, pp. 17 – 32, 2016.

[23] [Online]. Available: http://asimo.honda.com/asimo-specs/

[24] J. Koenemann, A. D. Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz, N. Mansard, and A. Motivations, "Whole-body Model-Predictive Control applied to the HRP-2 Humanoid," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 3346–3351.

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

[26] M. P. Forum, "Mpi: A message-passing interface standard," Knoxville, TN, USA, Tech. Rep., 1994.

[27] C. Brasseur, A. Sherikov, C. Collette, D. Dimitrov, and P.-B. Wieber, "A Robust Linear MPC Approach to Online Generation of 3D Biped Walking Motion," in *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, 2015, pp. 595–601.