The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Real Time Motion Generation and Control for Biped Robot -4th Report: Integrated Balance Control-

Toru Takenaka, Takashi Matsumoto, Takahide Yoshiike, Tadaaki Hasegawa,
Shinya Shirokura, Hiroyuki Kaneko and Atsuo Orita

Abstract—**A controller for biped running has to consider varying vertical ground reaction force while satisfying the horizontal ground reaction force and moment limits. We propose a design technique for feedback gains to stabilize the upper body position under varying vertical ground reaction force. We also propose an extended model ZMP control method which uses horizontal and rotational acceleration of the upper body and step duration change to generate moments to handle disturbances too large to be handled by ground reaction force control. Combining these techniques, robust biped running is achieved.**

I. Introduction

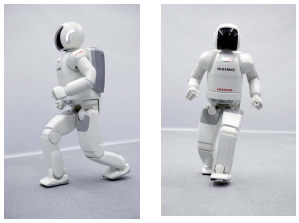

Fig. 1 Running biped robot system (ASIMO)

For biped robots [1][2](Fig. 1) to exist around and collaborate with human, they need abilities to react robustly against events including avoiding collision with previously unknown obstacles and maintaining balance under external disturbances by taking steps.

Feedback control techniques with preset reference gait pattern have been proposed to achieve balance under disturbances. Hirai et al. [1] proposed methods which use rotation of the foot and relative motion of the legs, Nagasaka et al. [6] proposed to accelerate the upper body of the robot to make the actual ZMP follow the desired ZMP trajectory to achieve balance. These techniques can only try to resist against disturbances, and they cannot achieve balance under larger disturbances.

Realtime gait pattern modification techniques have been developed recently which generates new trajectories frequently using the recent state of the robot, and thus are more robust against previously unknown disturbances. Nishiwaki et al. [7] and Wieber [15] proposed methods to update the upper body trajectory using preview control from the current inclination of the upper body to a stable future state. Diedam et al. [16] extends these methods by considering foot positioning as well. Tajima et al. [10] extended these methods to a running robot. With these methods, new trajectories are generated from the current actual robot state to a goal state. This is an undesirable behavior because the robot does not try to follow the original trajectory at all even though it can do so physically.

Hirai et al.[1] proposed methods to modify the desired ZMP trajectory based on the upper body angle error by accelerating the desired upper body position (model ZMP control). With the model ZMP control, new gait pattern is not generated merely from the current robot state. The new gait pattern is generated from computing the portion of the upper body stabilizing moment which could not be generated due to the support polygon. This method resists to a disturbance as much as allowed by support polygon, and also modifies the future gait pattern to bring the robot to a stable state in the future.

Pratt et al.[11] and Stephens [13] modeled biped robots with a linear inverted pendulum and a flywheel and analyzed boundaries of different balancing strategies such as ankle and hip strategies, rotation of the upper body and step positioning. Hyon et al.[9], Atkeson [12] and Hoffman et al.[17] also achieve balance by stepping, ankle and hip strategies. The initial and goal states of the robot have to be at rest with these approaches. Sugihara [14] discussed foot positioning to extend admissible region of ZMP, but their approach does not contain notion of future stability. Kajita et al. [8] proposed stabilizing techniques for bipedal running. In their method the upper body trajectory during the flight phases follows preset trajectory and ankle torque is controlled during the support phases as in [1]. This approach falls into the first group where the controller tries to follow the desired trajectory which is not updated in real time.

In this paper, we extend previously proposed model ZMP control. Previously, the trajectory modification was achieved merely from modifying the horizontal position of the upper body, and we extend it to include step duration change and rotation of the upper body in addition. We report results of applying our methods to bipedal running on a real robot in which the vertical ground reaction force changes largely. It is shown that varying upper body stabilizing moment dependent on the vertical ground reaction force, and distribution of it to the two strategies are essential.

The remainder of the paper is organized as follows. In section II, a general overview of the system is given. In section III, control of upper body inclination is explained. In section IV,

Fundamental Research Center, Honda Research & Development, 8-1 Honcho, Wako-shi, Saitama, Japan.

distribution of the ground reaction moments into two different strategies is explained. In section V, control of ground reaction force is explained. In section VI, model ZMP control is explained. In section VII, design of posture stabilizing moment for running motion is explained. Results and conclusions are shown in section VIII and IX respectively.

II. System Overview

In this paper, a gait pattern is a set of trajectories for the desired ZMP, the feet and the upper body.

1. Given a command to move, step position and duration are decided (Fig. 2(a)).
2. Given parameters above, design the desired ZMP and feet trajectories. Then design the upper body trajectory which satisfies the desired ZMP trajectory without causing the upper body to diverge (Fig. 2(b)).
3. Feed the gait pattern into the real robot, and stabilize it while it is following the gait pattern (Fig. 2(c)).

The details of the stability control (Fig. 2(c)) are

1. Modify the desired ground reaction moment to generate a moment to cancel the upper body inclination error (Fig. 2(d)).
2. Distribute the desired posture stabilizing moment into the ground reaction force control and model ZMP control (Fig. 2(e)).
3. Follow the desired ground reaction force so that the motion generated in Fig. 2(b) is achieved (Fig. 2(f))
4. Accelerate the upper body to maintain balance against large disturbances which cannot be absorbed by ground reaction force control (Fig. 2(g)).
5. Control the angle of each joint to follow the modified feet and upper body trajectories (Fig. 2(h)).

The output of the stability control system is the desired angle for each joint of the real robot. The real robot has a six axis force sensor between the ankle joint and the foot of each leg and an accelerator and a gyroscope to estimate its inclination. There are rubber bushes between the force sensor and the foot [1]which absorb landing impact and function as a mechanical low pass filter within ground reaction force control feedback loop. Each joint of the robot is position-controlled.

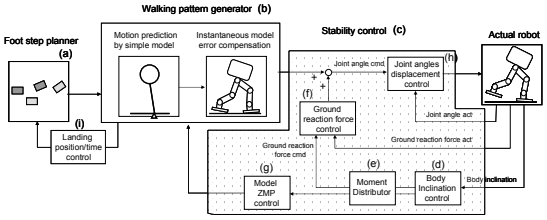

Fig. 2 System overview

III. Upper Body Inclination Error Control

To make the real robot follow a desired posture, the error between the desired posture and the actual robot is compensated by an additional posture stabilizing moment. Assuming that the joints and links are rigid and each joint is satisfying the desired angle, the inclination error of the upper body is computed from the accelerometer and the gyroscope. The inclination error is treated as the horizontal displacement of an inverted pendulum. Thus, the angle error θ_{incl}^{err} is treated as horizontal position error x_{err} for the rest of this paper.

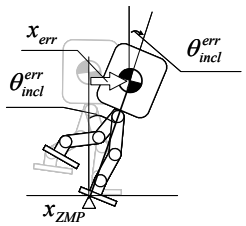

Fig. 3 Upper body inclination error

The desired ground reaction moment to recover x_{err}, M_{stab}, is computed then added to the desired ground reaction moment around desired ZMP. M_{stab} is computed with PD control as follows.

$$M_{stab} = K_x x_{err} + K_v \dot{x}_{err} \quad (1)$$

One way to determine the values of K_x and K_v is from the critical damping of the robot regarded as a linear inverted pendulum (Fig. 3). The details on how to automatically design the values of K_x and K_v while running are explained in section VII.

IV. Distribution of Ground Reaction Moment

A biped robot cannot generate ground reaction moments exceeding the limits determined by the vertical ground reaction force and the foot geometry. Thus, to achieve required ground reaction moment M_{stab} derived in the previous section, both the ground reaction force control and modification of the desired ZMP by upper body acceleration are employed.

The robot tries to generate M_{cmpl} equivalent to M_{stab} by ground reaction force control. This moment cannot be generated if it is larger than the maximum moment, the vertical acceleration of the CoG multiplied by the distance from the desired ZMP to the edge of the support polygon. Then, the remaining portion is achieved by modifying the desired ZMP, $M_{\text{model\_zmp}}$.

$$M_{\text{model\_zmp}} = M_{stab} - M_{cmpl} \quad (2)$$

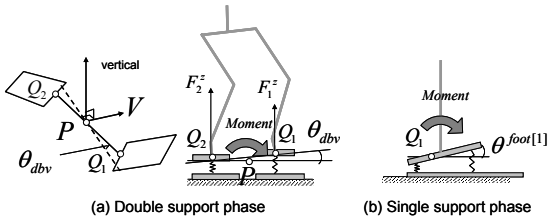

Fig. 4 Ground reaction force controller for single and double support phases

V. Ground Reaction Force Control

Input to the ground reaction force control system are the desired foot position, the desired ZMP and the desired center of the ground reaction forces for each foot, Q_1 and Q_2. The ground reaction force controller tries to generate a moment about the desired ZMP, M_{act}, equal to M_{cmpl}. Two controllers are shown in Fig. 4 for double support and single support phases.

During the double support phases, a moment about the desired ZMP is generated by modifying the ground reaction force at each foot. Vector V is horizontal to the ground and perpendicular to the line containing Q_1 and Q_2. The controller outputs θ^{dbv}, relative angle of the two feet, about the desired ZMP to control the ground reaction moment. For forward straight walking gait pattern, the ZMP trajectories are usually designed to start at the heel and leave at the toe. Thus, during the double support phases, Q_1 and Q_2 are usually located at the toe of the behind leg and at the heel of the front leg respectively.

During the single support phases, a moment about the desired center of the ground reaction force is generated by rotating the stance foot. Each foot is rotated by $\theta^{foot[n]}$ about Q_n where n is either left or right.

The ground reaction moment is influenced by the deformation of the rubber bushes. To cancel it, the deformation is computed from the desired ground reaction moment M_{cmpl} and added to the output.

VI. Model ZMP Control

The upper body stabilizing moment cannot be generated exceeding the limit which depends on the foot size and vertical ground reaction force. To generate a moment beyond this limit, the model ZMP control [1] is used by accelerating the upper body. In this paper, it is extended to control the upper body angle and step duration change.

A. Model ZMP Control of an Inverted Pendulum

The basic principle of model ZMP control is explained using walking in which the vertical ground reaction force is almost constant. The model ZMP generates the portion of M_{stab} which could not be generated by ground reaction force control, $M_{\text{model\_zmp}}$. The desired ZMP is modified by accelerating the upper body $x^{ZMP}_{mdfd\_cmd}$. The difference between $x^{ZMP}_{mdfd\_cmd}$ and the actual ZMP, x^{ZMP}_{act}, which is fixed to x^{ZMP}_{cmd} by ground reaction force control generates a moment to recover the inclination of the upper body (Fig.5). Note that physical feasibility of the modified desired ZMP trajectory is not important. By intentionally loosing balance, a recovering moment is generated on the real robot.

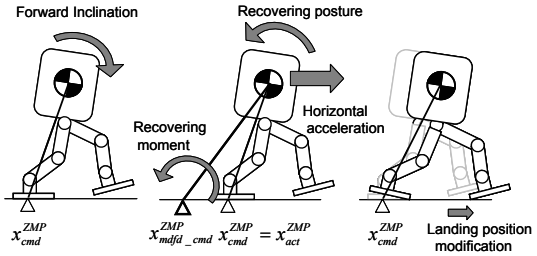

Fig. 5 Model ZMP control (horizontal)

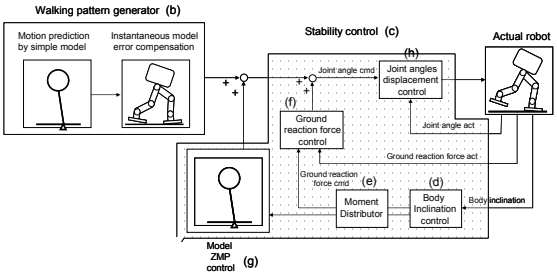

Fig. 6 Inverted pendulum model for Model ZMP control

The acceleration of the upper body is computed from an linear inverted pendulum model(Fig. 6(g)) and added to the original upper body trajectory. The inverted pendulum has mass m_{pend} and height constant h. The inverted pendulum is used to compute the horizontal displacement x_{pend} required to generate the desired moment M_{pend}.

$$\ddot{x}_{pend} = \frac{g}{h} x_{pend} + \frac{1}{m_{pend} h} M_{pend} \tag{3}$$

Substituting $M_{\text{model\_zmp}}$ into M_{pend}, the upper body position is recovered.

$$M_{pend} = M_{\text{model\_zmp}} \tag{4}$$

To prevent the inverted pendulum from diverging, a feedback control similar to Eq. (1) is employed. Adding the pendulum stabilizing term M^{stab}_{pend}, Eq. (4) becomes

$$M_{pend} = M_{\text{model\_zmp}} + M^{stab}_{pend} \tag{5}$$

Adding M^{stab}_{pend} to M_{pend}, the robot may loose balance. To cancel this effect, the ground reaction moment equivalent to M^{stab}_{pend} is generated. In case this cannot be generated without being limited for a long period of time, the inverted pendulum diverges. Step position is modified to prevent divergence of the inverted pendulum (Fig. 7)

To think about the divergence of the inverted pendulum in the future, divergent component[3] is considered. Setting current

time to 0 and time step to ΔT, and letting $q(k)$ be the divergent component at time k and x_{zmp} be the ZMP input,

$$q(k) = e^{\omega_0 k\Delta T} q(0) + e^{\omega_0 k\Delta T}\left(e^{-\omega_0 \Delta T} - 1\right)\sum_{i=0}^{k-1} e^{-\omega_0 i\Delta T} x_{zmp}(i) \quad (6)$$

where $\omega_0 = \sqrt{g/h}$, g is gravitational acceleration constant.

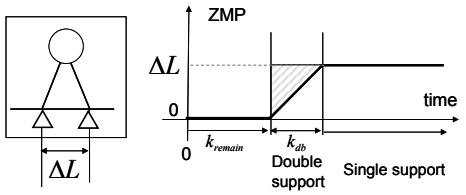

Fig. 7 Step position control of linear inverted pendulum

Modifying step position by ΔL (Fig. 7(left)) causes the desired ZMP trajectory to change during the double support phase as shown in Fig. 7(right). k_{db} is the duration of the next double support phase as a number of time steps, k_{remain} is the time to it as a number of time steps. At time $(k_{remain} + k_{db})$, the divergent component of the linear inverted pendulum model has to equal ΔL to prevent inverted pendulum from diverging. The details of how to determine ΔL are explained in the appendix.

The desired step position is modified frequently, and thus the divergence of the inverted pendulum model due to M_{pend} is suppressed.

B. Extended Model ZMP Control of an Inverted Pendulum

Step duration plays an important role in modifying ZMP without violating kinematic constraints. But the method explained previously cannot be used to change the step duration. Also, it cannot be directly applied to motions which involve large vertical acceleration changes.

To this end, we propose a new method to input $M_{\text{model\_zmp}}$ as virtual external moment $M_{virtual}$ into the gait pattern generator (Fig. 2(b)). Inside the gait pattern generator, the three mass dynamics model [3][4] is accelerated instead of the inverted pendulum in (Fig. 6(g)). It is important to note that $M_{virtual}$ is virtual, and thus the inverted pendulum of the three mass model is accelerated without modifying the desired ZMP. The moment generated by the inverted pendulum is M_{pend}. Similarly, M_{feet} stands for the moment generated by the foot and the total moment, M_{total}, equivalent to the desired ZMP, is

$$M_{total} = M_{pend} + M_{feet} \quad (7)$$

Adding $M_{virtual}$,

$$M_{total} = M_{pend} + M_{feet} + M_{virtual} \quad (8)$$

Fixing M_{total} and M_{feet}, the three mass model has to accelerate in the opposite direction as $M_{virtual}$. Eq. (8) can be rewritten into the following by introducing M_{pend}^{mdl} which has equivalent value to M_{pend}.

$$M_{total} = M_{pend}^{mdl} + M_{feet} + M_{virtual} \quad (9)$$

$M_{virtual}$ is a moment due to an external disturbance which does not get generated on the real robot. Thus, the actual moment the robot experiences is

$$M_{total}^{mdl} = M_{pend}^{mdl} + M_{feet} \quad (10)$$

The real robot tries to generate M_{total} in Eq. (9). However, due to the lack of $M_{virtual}$ which was expected when the gait pattern was generated, the robot experiences M_{total}^{mdl}. As a result, the difference of the desired and actual moments, $M_{virtual}$, is generated to recover balance as we have intended.

The inverted pendulum of the three mass model converges by satisfying the boundary condition at each step as explained in [3][4]. To satisfy the boundary condition, the desired ZMP trajectory of the current gait has to be modified. If the modified desired ZMP goes out of the support polygon, the step position and duration can be changed as explained in [3] (Fig. 2(i)). An advantage of this approach over the simple model ZMP control is that the inverted pendulum of the three mass model is a more accurate dynamics approximation of the real robot than the linear inverted pendulum, thus generating motions with less dynamics approximation errors. An example scenario is a big forward push while a biped robot is walking in place. Virtual external moment $M_{virtual}$ is given to the inverted pendulum of the three mass model, and it accelerates the robot forward. The robot would try to connect the boundary condition calculated from the cyclic gait pattern of walking in place. The desired ZMP trajectory of the current step has to be modified largely forward and goes out of the support polygon. Then the step position and duration of the current and next step need to be modified. As a result, the robot would take the next few steps forward and then go back to walking in place afterwards.

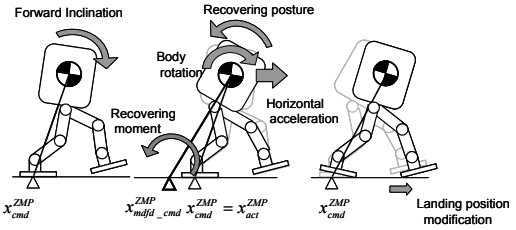

Fig. 8 Model ZMP control (horizontal+rotational)

C. Model ZMP Control with an Inverted Pendulum and a Flywheel

The model ZMP control can recover not only the position of the upper body but also its angle by generating a recovery moment. Adding $M_{virtual}$ to the running gait pattern generator with an inverted pendulum and a flywheel models proposed in [4](Fig. 8),

$$M_{total} = M_{pend} + M_{wheel} + M_{feet} + M_{virtual} \quad (11)$$

$M_{virtual}$ is distributed to the inverted pendulum and the

flywheel following the two rules below.

The flywheel generates $M_{virtual}$ when available horizontal ground reaction force is small.

Otherwise, distribute the high frequency components of $M_{virtual}$ to the flywheel and the low frequency components to the inverted pendulum.

The first rule is equivalent to the rule used in [4] to assign the difference between the commanded moment, M_{pend}, and the moment generated by the inverted pendulum with the horizontal ground reaction force limit, M_{pend}^{mdfd}, to the flywheel.

$$M_{wheel} = M_{pend} - M_{pend}^{mdfd} \tag{12}$$

The flywheel accelerates during the flight phases of running to recover the horizontal displacement of the upper body. In case the robot is given a large push while walking, the flywheel is used thus tilting the upper body first. Then it is accelerated horizontally to remove the low frequency component of the disturbance. Finally, step position and duration are changed to satisfy the kinematic constraints of the robot.

VII. Design of Posture Stabilizing Moment for Running Motion

A feedback control scheme for stabilizing the upper body using the divergent component of motion (Eq. (1)) proposed in [3] is explained.

The motion of the inverted pendulum can be decomposed into naturally converging component p_{err} and diverging component q_{err}.

$$\begin{pmatrix} p_{err} \\ q_{err} \end{pmatrix} = \begin{pmatrix} 1 & -1/\omega_0 \\ 1 & 1/\omega_0 \end{pmatrix} \begin{pmatrix} x_{err} \\ \dot{x}_{err} \end{pmatrix} \tag{13}$$

where $\omega_0 = \sqrt{g/h}$, h is the height of the inverted pendulum, and g is the gravitational acceleration constant. Omitting the naturally converging term , the moment required to stabilize can be simply written as,

$$M_{stab} = K_q q_{incl}^{err} \tag{14}$$

However, for motions accompanying varying vertical acceleration such as running, fixed gains cannot guarantee the upper body inclination error to be corrected. We propose a design method to solve this issue.

A cyclic gait pattern is prepared as the next gait to switch to at the end of the current gait as explained in [4]. The divergent component at the boundary of the cyclic gait has to be matched at the end of the current gait. The feedback gains are determined such that the inverted pendulum has the required divergent component at the end of the current gait.

The following notations are used in later discussions.

m_{pend} : The mass of the pendulum.

x_{err} : The horizontal position of the pendulum.

$\ddot{z}_{pend}$: The vertical acceleration of the pendulum.

h : The height of the pendulum.

g : The gravitational constant.

M_{pend} : The moment about the fixed end of the pendulum

M_{pend} and the horizontal acceleration of the pendulum are related as follows.

$$\ddot{x}_{err} = \frac{(g + \ddot{z}_{pend})}{h} x_{err} + \frac{1}{h \cdot m_{pend}} M_{pend} \tag{15}$$

where $\ddot{z}_{pend}$ is designed as explained in [4]. Rewriting Eq. (15) as a state equation in discrete time,

$$\mathbf{x}_{k+1} = A(k)\mathbf{x}_k + B(k)\mathbf{u}_k, \quad \mathbf{x} = \begin{pmatrix} x_{pend} & \dot{x}_{pend} \end{pmatrix}^T, \quad \mathbf{u} = M_{pend} \tag{16}$$

Let a matrix ϕ be the result of multiplying state transfer matrix $A(k)$ from the beginning of the cyclic gait pattern to one time step before the terminal time N_{cyc}.

$$\phi = A(N_{cyc} - 1)A(N_{cyc} - 2)\cdots A(0) \tag{17}$$

Then let Γ_{cyc} be formed by columns of the eigenvectors of ϕ. Using Γ_{cyc}, the convergent component p_{err} and divergent component q_{err} are defined.

$$\begin{pmatrix} p_{err} \\ q_{err} \end{pmatrix} = \Gamma_{cyc}^{-1} \mathbf{x}_0 \tag{18}$$

Refer to [4] for detailed derivation of Eq. (15)-(18).

Note that p_{err} and q_{err} to correspond to eigenvalues λ_1 and λ_2 where $\lambda_1 < 1$ and $\lambda_2 > 1$, and $\mathbf{x}_0$ is the initial state of the cyclic gait pattern.

Let $(x_{err}^k, \dot{x}_{err}^k)^T$ be the state of the inverted pendulum at time k. Note time at the beginning of the current gait is 0 and is N at the end of the current gait. Let $\varphi(N, j)$ be the result of multiplying the state transfer matrix from time j to N. Also, let be the result of multiplying state transfer matrices from time j to N.

$$\varphi(N, j) = A(N-1)A(N-2)\cdots A(j) \tag{19}$$

Suppose that $(x_{err}^k, \dot{x}_{err}^k)^T$ is the state of the inverted pendulum at time k, and an impulse M_{pend} is given to the system for one time step. The expected values of the convergent and divergent components at time N, $\begin{pmatrix} p_{err}^{k+1(N)} & q_{err}^{k+1(N)} \end{pmatrix}^T$, can be written in terms of the state of the pendulum at time k+1, $(x_{err}^{k+1}, \dot{x}_{err}^{k+1})^T$.

$$\begin{aligned} \begin{pmatrix} p_{err}^{k+1(N)} \\ q_{err}^{k+1(N)} \end{pmatrix} &= \Gamma_{cyc}^{-1} \varphi(N, k+1) \begin{pmatrix} x_{err}^{k+1} \\ \dot{x}_{err}^{k+1} \end{pmatrix} \\ &= \Gamma_{cyc}^{-1} \varphi(N, k+1) \left\{ A(k) \begin{pmatrix} x_{err}^{k} \\ \dot{x}_{err}^{k} \end{pmatrix} + B(k) M_{pend} \right\} \\ &= \Gamma_{cyc}^{-1} \varphi(N, k) \begin{pmatrix} x_{err}^{k} \\ \dot{x}_{err}^{k} \end{pmatrix} + \Gamma_{cyc}^{-1} \varphi(N, k+1) B(k) M_{pend} \end{aligned} \tag{20}$$

Similarly, $\begin{pmatrix} p_{err}^{k\ (N)} & q_{err}^{k\ (N)} \end{pmatrix}^T$ are the expected values of the convergent and divergent components at time N starting from time k without any impulse.

$$\begin{pmatrix} p_{err}^{k\ (N)} \\ q_{err}^{k\ (N)} \end{pmatrix} = \Gamma_{cyc}^{-1} \varphi(N, k) \begin{pmatrix} x_{err}^{k} \\ \dot{x}_{err}^{k} \end{pmatrix} \tag{21}$$

If the state of the inverted pendulum is unchanged between time k and k+1,

$$\left(x_{err}^{k}, \dot{x}_{err}^{k}\right)^{T} = \begin{pmatrix}1 & 0\end{pmatrix}^{T} \tag{22}$$

$$\left(x_{err}^{k+1}, \dot{x}_{err}^{k+1}\right)^{T} = \begin{pmatrix}1 & 0\end{pmatrix}^{T} \tag{23}$$

Substituting Eq. (22)(23) into Eq. (20)(21), the ratio of $q_{err}^{k+1(N)}$ to $q_{err}^{k\ (N)}$ is

$$\alpha = \frac{q_{err}^{k+1(N)}}{q_{err}^{k\ (N)}} = \frac{\begin{pmatrix}0 & 1\end{pmatrix}\Gamma_{cyc}^{-1}\varphi(N,k+1)\begin{pmatrix}1\\0\end{pmatrix}}{\begin{pmatrix}0 & 1\end{pmatrix}\Gamma_{cyc}^{-1}\varphi(N,k)\begin{pmatrix}1\\0\end{pmatrix}} \tag{24}$$

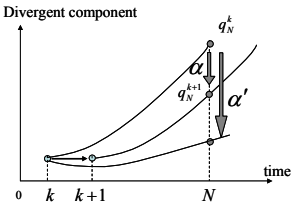

Fig. 9 Divergent component of time k and k+1

Holding the same state from time k to k+1, the divergent component at time N becomes α times larger. M_{pend} required to suppress the divergent component is

$$M_{pend} = \frac{q_N^{k+1} - q_N^{k}}{\begin{pmatrix}0 & 1\end{pmatrix}\Gamma_{cyc}^{-1}\varphi(N,k+1)B(k)} \tag{25}$$

Eq. (25) is the moment required to hold the same state. To force it to converge,

$$\frac{q_N^{k+1}}{q_N^{k}} = \alpha' \quad \left(\alpha' = \alpha e^{-\lambda\Delta T}\right) \tag{26}$$

With α', the divergent component is expected to decay at a time constant of $1/\lambda$. Computing M_{pend} from Eq. (25)(26) and substituting with M_{stab},

$$M_{stab} = \frac{\alpha \cdot e^{-\lambda\Delta T} - 1}{\begin{pmatrix}0 & 1\end{pmatrix}\Gamma_{cyc}^{-1}\varphi(N,k+1)B(k)}\begin{pmatrix}0 & 1\end{pmatrix}\Gamma_{cyc}^{-1}\varphi(N,k)\begin{pmatrix}x_{err}^{k}\\ \dot{x}_{err}^{k}\end{pmatrix} \tag{27}$$

The gains in Eq. (1) can be computed from Eq. (27) and used in the upper body inclination error control loop.

VIII. Results

A. Model ZMP Control

A push is given to ASIMO while walking in place with step duration of 500 ms (100 ms long double support phase and 400 ms long single support phase). Horizontal force 100 N is given to a point on the upper body at1140 mm from ground, for 100 ms at the beginning of a double support phase. Fig. 10 shows the force added, the estimated upper body inclination θ_{incl}^{err}, the inverted pendulum displacement, the angle of the flywheel, stabilizing moment for the inverted pendulum, stabilizing moment for the flywheel, the modified step position of the current gait. Fig. 11 is a time series of the postures of ASIMO. It can be observed that the flywheel starts rotating to generate a moment followed by the inverted pendulum. The duration of the single support phase is modified to 340 ms and a forward step with length of 197 mm is taken to satisfy the kinematic constraints of the robot.

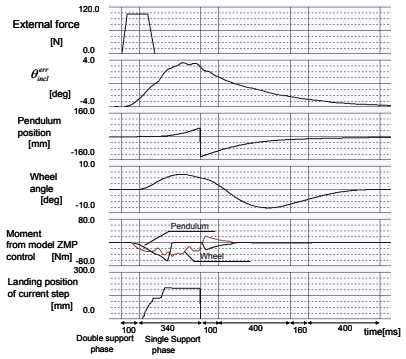

Fig. 10 Model ZMP control with pendulum and wheel (Simulation 100N)

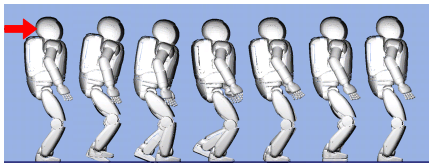

Fig. 11 Posture of model ZMP control with pendulum and wheel

Another experiment is conducted in which a push of 50 N is given to the same robot walking at 1.55 km/h for 110 ms (Fig. 12,13). Fig. 12 shows that the upper body inclines backward due to the external force and the step position is changed. Also it can be observed that the robot converges to its cyclic trajectory from the next step.

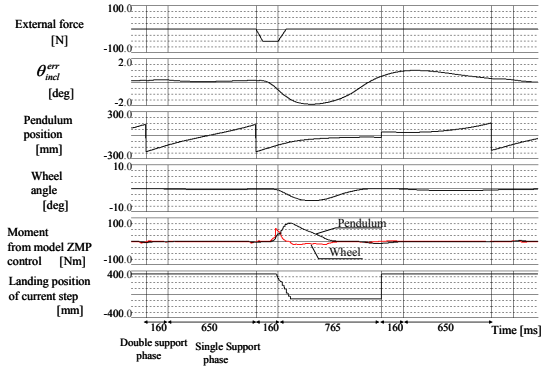

Fig. 12 Experiment of model ZMP control with pendulum and wheel while walking

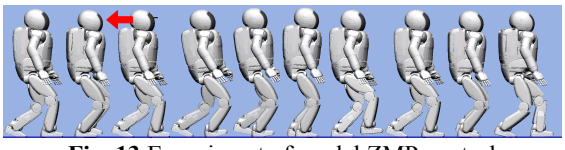

Fig. 13 Experiment of model ZMP control with pendulum and wheel while walking

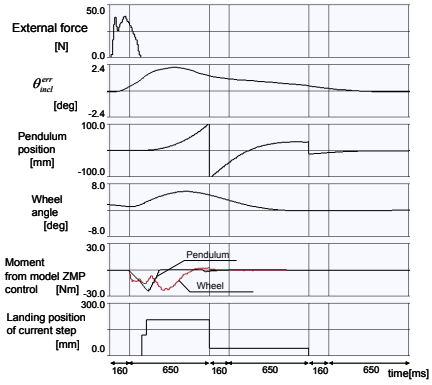

Fig. 14 Experiment of model ZMP control with pendulum and wheel (Real robot 40N)

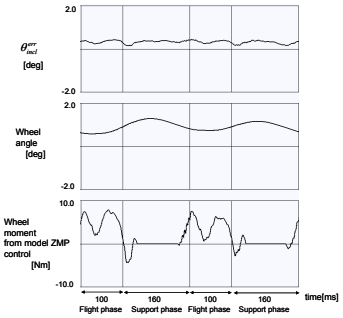

Fig. 15 Experiment of running at 10km/h

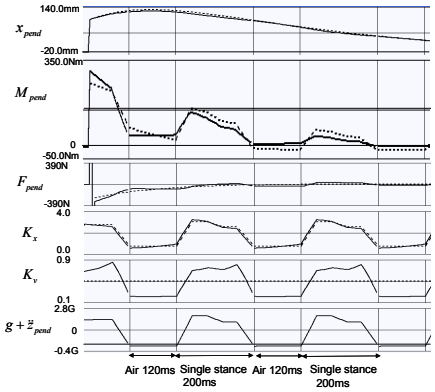

Fig. 16 Experiment of variable gain of inclination feedback

Fig. 14 is the result of pushing the real robot while walking in place. The ground reaction moment limit is modified to 20 mm in the form of equivalent ZMP. This is for safety reasons as well as to make it easy for the model ZMP control to take place instead of the ground reaction force control. The input force was 40 N, and the duration of a step was 810 ms. Fig. 14 shows that the inverted pendulum and flywheel models are used to recover the upper body position. It is also observed that the step position is changed for two steps as well.

Finally, effectiveness of using the model ZMP control for biped running is tested. The moment required to recover the upper body horizontal displacement is generated by the flywheel when available horizontal ground reaction force is near 0. The model ZMP control stabilizes the upper body angle while running and robust running is achieved. Fig. 15 shows the estimated upper body angle error θ_{incl}^{err}, the flywheel angle and the moment input to the flywheel from the model ZMP control while running at 10km/h. It can be seen that the flywheel accelerates around the flight phases to stabilize the upper body and is stabilized during the support phases. The upper body inclination error stays within 0.5 deg, and the validity of the proposed model ZMP control is verified.

B. Upper Body Position Stabilizing Control

To show the validity of the proposed variable gain design, a step disturbance is given to an inverted pendulum which is jumping vertically on simulation.

The proposed variable feedback gains are compared against a fixed set of gains, K_x^{fix} and K_v^{fix}, with the following feedback control to prevent the inverted pendulum from diverging due to its vertical acceleration $\ddot{z}_{pend}$.

$$M_{stab} = (K_x^{fix} - \frac{\ddot{z}_{pend}}{g})x_{pend} + K_v^{fix}\dot{x}_{pend} \quad (28)$$

Fig. 16 shows, from the top, the horizontal position of the inverted pendulum x_{pend}, the stabilizing moment for the inverted pendulum M_{pend}, the horizontal ground reaction force F_{pend}, the variable feedback gains K_x and K_v, the vertical ground reaction force $(g+\ddot{z}_{pend})$. The result of the proposed method is shown with plain lines and the compared method is shown with dotted lines. The pendulum is given horizontal displacement of 100 mm and horizontal velocity of 500 mm/s. K_x is similar to the vertical ground reaction force for both methods, K_v becomes small during the flight phases for the proposed method and stays constant for the other method. It can be also observed that is similar for both methods while M_{pend} of the proposed method converges more smoothly. Also, M_{pend} of the compared method has negative values during the flight phase generating more moments than required to recover balance.

IX. CONCLUSIONS

In this paper, we proposed control techniques to maintain balance under external disturbances for biped robots under rapidly varying vertical acceleration as well as the horizontal ground reaction force and moment limits. Based on ground reaction force control, a feedback control method to stabilize the upper body position under varying vertical ground reaction force while running is proposed. In addition, the model ZMP control is extended to use the rotational motion of the upper body and step duration change. This extension enabled stable upper body control under the horizontal ground reaction force and moment limits.

Combining these techniques, we achieved a biped robot system which react flexibly to external disturbances. Our system can use ground reaction force control to recover balance from relatively small disturbances. If the disturbance is too large to handle by the ground reaction force control, it accelerates its upper body to maintain balance and then take steps to come to rest. Stable running is achieved using similar techniques as well.

REFERENCES

[1] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka,"The Development of Honda Humanoid Robot", In Proceedings of the 1998 IEEE International Conference on Robotics and Automation, Leuven, Belgium, May. 1998, pp. 1321-1326.

[2] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, "The intelligent ASIMO: System overview and integration", In Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002, pp.2478-2483.

[3] 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.

[4] 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.

[5] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real Time Motion Generation and Control for Biped Robot -3rd Report: Gait Pattern Modification to Compensate Approximated Dynamics Error-", In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009.

[6] K. Nagasaka, M. Inaba, and H. Inoue, "Stabilization of Dynamic Walk on a Humanoid Using Torso Position Compliance Control", In Proceedings of 17th Annual Conference on Robotics Society of Japan, pp. 1193-1194, 1999.

[7] K. Nishiwaki and S. Kagami, "High Frequency Walking Pattern Generation based on Preview Control of ZMP" , In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, pp.2667--2672, Orlando, Florida, May, 2006.

[8] S. Kajita, T. Nagasaki, K.Kaneko, K. Yokoi, and K. Tanie, "A Running Controller of Humanoid Biped HRP-2LR", In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp.618-624, Barcelona, Spain, April, 2005.

[9] S. H. Hyon and G. Cheng, "Disturbance Rejection for Biped Humanoids", In Proceedings of 2007 IEEE International Conference on Robotics and Automation Roma, Italy, 10-14 April 2007.

[10] R.Tajima, D.Honda and K.Suga, "Fast Running Experiments Involving a Humanoid Robot", In Proceedings of IEEE International Conference on Robotics and Automation, 2008, pp.1571-1576

[11] J.Pratte, J.Carff, S.Drakunov and Ambarish Goswami, "Capture Point: A Step toward Humanoid Push Recovery", In Proceedings of the 2006 IEEE International Conference on Humanoids, 2006, pp. 200-207.

[12] C. G. Atkeson and B. Stephens, "Multiple Balance Strategies From One Optimization Criterion", In Proceedings of the 2007 IEEE International Conference on Humanoids, 2007.

[13] B. Stephens, "Humanoid Push Recovery", In Proceedings of the 2007 IEEE International Conference on Humanoids, 2007.

[14] T. Sugihara, "Standing Stabilizability and Stepping Maneuver in Planar Bipedalism based on the Best COM-ZMP Regulator", In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, pp.1966-1971, Kobe, Japan, May, 2009.

[15] P. B. Wieber, "Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations", In Proceedings of the 2006 IEEE International Conference on Humanoids, 2006, pp. 137-142.

[16] H. Diedam, D. Dimitrov, P.B.Wieber, K. Mombaur and M. Diehl, "Online Walking Gait Generation with Adaptive Foot Positioning through Linear Model Predictive Control", In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, pp.1121-1126.

[17] A. Hoffmann, M. Popvic and H. Herr, "Exploiting Angular Momentum to Enhance Bipedal Center-of-Mass Control", In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, pp.4423-4429, Kobe, Japan, May, 2009.

APPENDIX

As expressed in Eq. (13), the divergent component is a variable depending on the same coordinate system as position. Therefore before computing the divergent component in the present coordinate system, the divergent component $q_{\Delta L}(k)$ in the coordinate which advanced to ΔL is calculated. Letting the divergent component at time of next double support phase k_{remain} in the coordinate which advanced to ΔL be $q_{\Delta L}(k_{remain})$, and $k > k_{remain}$,

$$\begin{aligned} q_{\Delta L}(k) &= e^{\omega_0(k-k_{remain})\Delta T} q_{\Delta L}(k_{remain}) \\ &+ e^{\omega_0(k-k_{remain})\Delta T}\left(e^{-\omega_0\Delta T}-1\right)\sum_{i=k_{remain}}^{k-1} e^{-\omega_0 i\Delta T} x_{zmp}(i) \end{aligned} \tag{29}$$

To compute the second term of Eq. (29) $U1(k)$, the effect of the shaded triangle in Fig. 7 is computed as follows.

$$U1(k) = -\Delta L\left\{(e^{-\omega_0\Delta T}-1)\sum_{i=0}^{k_{db}-1} e^{-\omega_0 i\Delta T}(1-\frac{i}{k_{db}})\right\} e^{\omega_0(k-k_{remain})\Delta T} \tag{30}$$

If let $q_{\Delta L}(k)$ be 0 in the future in the coordinate which advanced to ΔL, the inverted pendulum will converge to rest after landing next step. Thus, $q_{\Delta L}(k_{remain})$ should satisfy next equation.

$$q_{\Delta L}(k_{remain}) = \frac{-1}{e^{\omega_0 k\Delta T}} U1(k) \tag{31}$$

The divergent component at time of k_{remain} in present coordinate $q_0(k_{remain})$ becomes

$$q_0(k_{remain}) = q_{\Delta L}(k_{remain}) + \Delta L \tag{32}$$

Letting input to the system be $x_{zmp}(i) = M_{pend}/(m_{pend}g)$ at current time step and 0 from the next time step, $q_0(k_{remain})$ can be computed as follows.

$$q_0(k_{remain}) = q_0(0)e^{\omega_0 k_{remain}\Delta T} + (e^{-\omega_0\Delta T}-1)\frac{M_{pend}}{m_{pend}g} e^{\omega_0 k_{remain}\Delta T} \tag{33}$$

Here, $q_0(0) = x^0_{pend} + \dot{x}^0_{pend}/\omega_0$, $(x^0_{pend}, \dot{x}^0_{pend})^T$ is the current state of the inverted pendulum. From Eq. (30)-(33), ΔL becomes

$$\Delta L = \frac{(x^0_{pend} + \frac{\dot{x}^0_{pend}}{\omega_0}) + (e^{-\omega_0\Delta T}-1)\frac{M_{pend}}{m_{pend}g}}{\left\{1 + (e^{-\omega_0\Delta T}-1)\sum_{i=0}^{k_{db}-1} e^{-\omega_0 i\Delta T}(1-\frac{i}{k_{db}})\right\} e^{-\omega_0 k_{remain}\Delta T}} \tag{34}$$