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

Real-time Running and Jumping Pattern Generation for Bipedal Robots based on ZMP and Euler's Equations

Barkan Ugurlu and Atsuo Kawamura

Abstract— **This paper is presenting a method to generate real-time running and jumping trajectories that can be applied to bipedal humanoid robots. The proposed method is based on maintaining the overall dynamic balance by using the ZMP stability criterion throughout support phases. To be able to reach this goal, we utilize ZMP equations in spherical coordinates, so that the rate change of angular momentum terms in ZMP equations are included naturally by using Eulerian equations of motion for unsymmetrical bodies. Thus, undesired torso angle fluctuation is expected to be more restrainable comparing to other methods in which angular momentum information is ignored or zero-referenced. Applying the aforementioned technique, we primarily simulated running motion on a dynamic 3-D simulator. Secondarily, one-legged jumping experiments were conducted on the actual bipedal robot. In conclusion, we obtained repetitive and successful running and jumping cycles which satisfactorily verify the proposed method.**

I. INTRODUCTION

Starting from 1970s, humanoid robots have evolved in many aspects and they are still expected to be adapted within the social human environment. Considering such dynamic human environment, people may walk, run, jump and interact with others in various cases. Therefore, a well-adapted humanoid robot should function such human-like dexterity. Since there are several reports about reliable dynamic bipedal walking[5], and human-humanoid interaction[6], jumping and running motions seem to be the current tasks for achieving human-like dexterity goal. In addition, studying such fast motions will force researchers to develop more robust hardware structures to overcome the fragility problem.

In order to improve humanoid robot technology, Raibert and his colleagues have developed fundamental control laws of running motion[7]. Their robots were actuated by hydraulic systems and they demonstrated impressive experiments. Applying a similar approach in control laws and considering energy efficiency, Ahmadi and Buehler succeeded monopod running with a spring supported mechanism[11]. However these types of robots are not adequate for usual humanoid activities. What is more, Raibert's control laws were efficient for robots with foot points whose mass is cumulated around their hips[20].

Kajita et al. proposed Resolved Momentum Control (RMC) method which is an offline pattern generation technique based on the total momenta resolution[4]. In spite of the fact that RMC covers dominant robot dynamics; it is relatively difficult to manipulate the robot's control strategy in real-time. Nonetheless, Toyota's research group employed online ZMP (Zero Moment Point) resolution technique into humanoid jumping and running using a humanoid robot with very light leg structures[1]. In both [1] and [4], angular momentum information about CoM (Center of Mass) is substituted as zero.

However, forcing angular momentum around CoM to be zero tends to create unnatural torso rotation. For example, in [18], we simulated RMC method using a realistic robot model. As the result, the robot's torso angle is fluctuated to maintain its angular momentum, varying from 2 degrees to 20 approximately 3 times a second. Hence, an experiment using this method may harm our robot.

Recently, researchers also pointed out the importance of angular momentum as well. In [2], angular momentum is firstly zero referenced then included in the updated pelvis link velocity. As a similar example, Sugihara and Nakamura enhanced boundary relaxation method for 3-D hopping motion planning[3]. In their ZMP-based approach, rate change of angular momentum terms are firstly ignored, then included in the inverse kinematics stage. Dissimilarly to these two methods, we defend that angular momentum information could be directly included during the CoM trajectory generation stage instead of subsequent updates. In this manner, we believe it is more appropriate to inject necessary amount of torques about CoM for preventing undesired torso angle fluctuations.

Considering the aforementioned facts, we composed a method to generate real-time running and jumping trajectories, which may ensure the overall balance. In the proposed method, we utilized ZMP equations in spherical coordinates, so that angular momentum is naturally included in our dynamic equations for support phases. Moreover, flight phase trajectories are derived using projectile motion dynamics. Since the angular momentum is included using Euler's equations, we name it Eulerian ZMP Resolution (EZR).

Authors group firstly proposed this method for 2-D case to generate jumping motion on a planar robot[16]. Subsequently, the method is enhanced for 3-D case in order to generate motions for bipedal robots which are assumed to be symmetrical about three axes[15]. In this paper, we extend our method by considering unsymmetrical robot geometry to be able to reach more realistic and general results.

In the present paper, Eulerian ZMP Resolution is explained in section II. Simulation and experimental results are discussed in section III and finally the paper is concluded in section IV.

B. Ugurlu and A. Kawamura are with Division of Advanced Physics, Department of Electrical and Computer Engineering, Yokohama National University, 79-5 Tokiwadai, Hodogaya-ku, Yokohama 240-8501 Japan. barkanu@kawalab.dnj.ynu.ac.jp

II. Eulerian ZMP Resolution: Real-time Bipedal Motion Generation

Our proposed method, EZR, is a trajectory generation technique for motions, which include single support phases and flight phases such as running, jogging and jumping. Here, the main idea of trajectory planning is to obtain real-time joint motions that ensure desired ZMP profiles through the support phase and ensure predetermined projectile motion through the flight phase, consecutively. In Fig. 2, significant moments from a jumping sequence are illustrated.

A. Support Phase Trajectory Generation

During the support phase, we consider the robot as an unsymmetrical body, rotating about a fixed point(foot center) and it is in contact with the floor through a foot which has a rectangular shape. Fig. 1 illustrates such modeling. In this model, CoM position is defined in the spherical coordinate frame by using the parameters CoM length, r, zenith angle, θ and azimuth angle, ϕ[9]. Utilizing such model enables us to combine rate change of angular momentum terms with inertial forces terms in ZMP equations. Deriving the z-axis CoM trajectory, z, by 5^{th} order polynomials[10] and determining proper ZMP references, we are able to obtain θ and ϕ angles' trajectories in real-time as we solve ZMP equations iteratively. Perfoming a conversion from spherical coordinates to Cartesian coordinates gives us x-axis and y-axis CoM trajectory.

Firstly, let us analyze ZMP equations[1]:

$$X_{zmp} = \frac{x(\ddot{z}+g) - z\ddot{x}}{(\ddot{z}+g)} - \frac{\dot{L}_y}{m(\ddot{z}+g)} \tag{1}$$

$$Y_{zmp} = \frac{y(\ddot{z}+g) - z\ddot{y}}{(\ddot{z}+g)} + \frac{\dot{L}_x}{m(\ddot{z}+g)} \tag{2}$$

In (1) and (2), x, y and z stand for CoM position in Cartesian frame while L_x and L_y symbolize roll axis and pitch axis angular momentum about CoM. One dot and two dots represent first and second derivatives with respect to time. Further, m is the total mass and g is the gravitational acceleration. In EZR method, we are going to group ZMP equations into two parts: Inertial Forces Terms and Rate

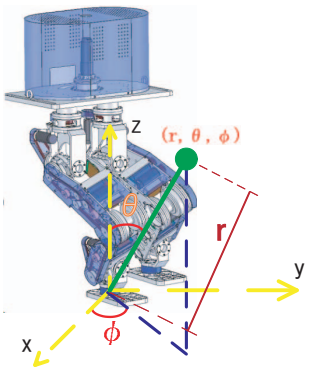


Fig. 1. One Mass Model and Spherical Coordinate System on MARI-3

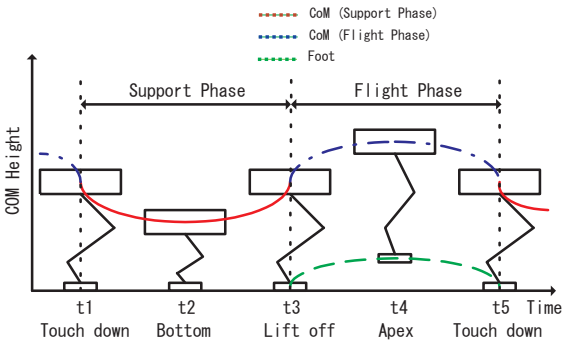


Fig. 2. A Complete Jumping Sequence

Change of Angular Momentum Terms which are the first and second terms in (1) and (2) respectively. Henceforward, we are going to express these terms in the spherical coordinate frame, namely using the parameters r, θ and ϕ.

1) Inertial Forces Terms: In order to express inertial forces terms by using spherical coordinate frame's parameters, we perform necessary coordinate transformation for CoM position. Subsequently, we differentiate position expression two times and obtain CoM acceleration.

$$\begin{aligned} x &= r sin\theta \cos\phi \\ \ddot{x} &= \ddot{r} sin\theta \cos\phi + \dot{r}\dot{\theta}\cos\theta\cos\phi \\ &+\dot{r}\dot{\theta}\cos\theta\cos\phi + r\ddot{\theta}\cos\theta\cos\phi \\ &-r\dot{\theta}^2 sin\theta\cos\phi - r\dot{\theta}\dot{\phi}\cos\theta\sin\phi \\ &-\dot{r}\dot{\phi}sin\theta\sin\phi - r\ddot{\phi}sin\theta\sin\phi \\ &-r\dot{\phi}\dot{\theta}\cos\theta\sin\phi - r\dot{\phi}^2 sin\theta\cos\phi \\ &-\dot{r}\dot{\phi}sin\theta\cos\phi \end{aligned} \tag{3}$$

$$\begin{aligned} y &= r sin\theta \sin\phi \\ \ddot{y} &= \ddot{r} sin\theta \sin\phi + \dot{r}\dot{\theta}\cos\theta\sin\phi \\ &+\dot{r}\dot{\theta}\cos\theta\sin\phi + r\ddot{\theta}\cos\theta\sin\phi \\ &-r\dot{\theta}^2 sin\theta\sin\phi - r\dot{\theta}\dot{\phi}\cos\theta\cos\phi \\ &-\dot{r}\dot{\phi}sin\theta\cos\phi - r\ddot{\phi}sin\theta\cos\phi \\ &-r\dot{\phi}\dot{\theta}\cos\theta\cos\phi - r\dot{\phi}^2 sin\theta\sin\phi \\ &-\dot{r}\dot{\phi}sin\theta\cos\phi \end{aligned} \tag{4}$$

$$\begin{aligned} z &= r\cos\theta \\ \ddot{z} &= \ddot{r}\cos\theta - 2\dot{r}\dot{\theta}sin\theta - r\ddot{\theta}sin\theta - r\dot{\theta}^2\cos\theta \end{aligned} \tag{5}$$

2) Rate Change of Angular Momentum Terms: Supposing that the friction between the floor and the foot sole is sufficient and there is no foot rotation, we can consider the support phase motion as a rotation of a rigid body about a fixed point as previously stated. For this case, Euler's equations of motion for unsymmetrical bodies might give us insights,

$$\begin{aligned} \tau_x = \dot{L}_x &= I_{xx}\dot{\omega}_x - (I_{yy} - I_{zz})\omega_y\omega_z - (\omega_y^2 - \omega_z^2)I_{yz} \\ &-(\omega_x\omega_y + \dot{\omega}_z)I_{xz} + (\omega_x\omega_z - \dot{\omega}_y)I_{xy} \\ \tau_y = \dot{L}_y &= I_{yy}\dot{\omega}_y - (I_{zz} - I_{xx})\omega_z\omega_x - (\omega_z^2 - \omega_x^2)I_{xz} \\ &-(\omega_y\omega_z + \dot{\omega}_x)I_{xy} + (\omega_y\omega_x - \dot{\omega}_z)I_{yz} \\ \tau_z = \dot{L}_z &= I_{zz}\dot{\omega}_z - (I_{xx} - I_{yy})\omega_x\omega_y - (\omega_x^2 - \omega_y^2)I_{xy} \\ &-(\omega_z\omega_x + \dot{\omega}_y)I_{yz} + (\omega_z\omega_y - \dot{\omega}_x)I_{xz} \end{aligned} \tag{6}$$

in which I_{xx}, I_{yy} and I_{zz} are moments of inertia about principle axes, I_{xy}, I_{xz} and I_{yz} are products of inertia, ω_x,

ω_y and ω_z are angular velocities about principle axes and τ_x, τ_y and τ_z are rate change of angular momentums (torques) about roll pitch and yaw axes.

In [8], a gyroscopic pendulum which is suspended at a fixed point is studied. In this model, Timoshenko and Young demonstrated the CoM angular velocity vector in terms of given set of angles, by using the orientation that is defined between fixed point's coordinate frame and CoM frame. Following a similar approach and considering the spherical coordinate frame's rotation set[9], we obtained CoM angular velocity and angular acceleration vectors as below.

$$\omega_x = -\dot{\theta} sin\phi \quad , \quad \omega_y = \dot{\theta}\cos\phi \quad , \quad \omega_z = \dot{\phi} \tag{7}$$

$$\begin{aligned} \dot{\omega}_x &= -\ddot{\theta}\sin\phi - \dot{\theta}\dot{\phi}\cos\phi \\ \dot{\omega}_y &= \ddot{\theta}\cos\phi - \dot{\theta}\dot{\phi}\sin\phi \\ \dot{\omega}_z &= \ddot{\phi} \end{aligned} \tag{8}$$

If we combine (7) and (8) in (6), it is possible to obtain rate change of angular momentum terms as shown below.

$$\begin{aligned} \dot{L}_y = I_{yy}(\ddot{\theta}\cos\phi - \dot{\theta}\dot{\phi}\sin\phi) + (I_{zz} - I_{xx})\dot{\theta}\dot{\phi}\sin\phi \\ -(\dot{\phi}^2 - \dot{\theta}^2\sin^2\phi)I_{xz} + \ddot{\theta}\sin\phi I_{xy} - (\dot{\theta}^2\sin\phi\cos\phi + \ddot{\phi})I_{yz} \end{aligned} \tag{9}$$

$$\begin{aligned} \dot{L}_x = -I_{xx}(\ddot{\theta}\sin\phi + \dot{\theta}\dot{\phi}\cos\phi) - (I_{yy} - I_{zz})\dot{\theta}\dot{\phi}\cos\phi \\ -(\dot{\theta}^2\cos^2\phi - \dot{\phi}^2)I_{yz} - \ddot{\theta}\cos\phi I_{xy} - (\ddot{\phi} - \dot{\theta}^2\sin\phi\cos\phi)I_{xz} \end{aligned} \tag{10}$$

3) Solving ZMP Equations for θ and ϕ Trajectories: Before solving ZMP equations, we define τ_{zmpx} and τ_{zmpy} which come out from cross multiplications in (1) and (2) to eliminate fractional expressions.

$$\begin{aligned} \tau_{zmpx} &= mX_{zmp}(\ddot{z} + g) = mx(\ddot{z} + g) - mz\ddot{x} - \dot{L}_y \\ \tau_{zmpy} &= mY_{zmp}(\ddot{z} + g) = my(\ddot{z} + g) - mz\ddot{y} + \dot{L}_x \end{aligned} \tag{11}$$

Furthermore, we may eliminate some trigonometric terms by introducing τ_a and τ_b.

$$\begin{aligned} \tau_a &= \tau_{zmpx}\cos\phi + \tau_{zmpy}\sin\phi \\ \tau_b &= \tau_{zmpx}\sin\phi - \tau_{zmpy}\cos\phi \end{aligned} \tag{12}$$

$$\begin{aligned} r &= z\sec\theta, & \mu_2 &= \cos^2\theta \\ \mu_1 &= \sin\theta\cos\theta, & \mu_4 &= \cos^2\phi \\ \mu_3 &= \sin\phi\cos\phi, & J_g &= mgr\sin\theta \\ J_p &= I_{yz}\sin\phi + I_{xz}\cos\phi, & J_\beta &= J_\alpha\mu_1 + J_m \\ J_m &= I_{yz}\cos\phi - I_{xz}\sin\phi, & \mu_5 &= \sin^2\phi = 1 - \mu_4 \\ \dot{r} &= \dot{z}\sec\theta + z\dot{\theta}\tan\theta\sec\theta, & J_\alpha &= mr^2, \dot{J}_\alpha = 2mr\dot{r} \end{aligned} \tag{13}$$

In (13), we present some repeating parameters to ease our calculations. Finally, if we insert (3), (4), (5), (9) (10), (12) and (13) into (11), we obtain the following equations:

$$\begin{aligned} \tau_a &= -\ddot{\theta}\left(J_\alpha + 2I_{xy}\mu_3 + I_{xx}\mu_5 + I_{yy}\mu_4\right) \\ &+ \dot{\phi}^2 J_\beta - \dot{\theta}\dot{J}_\alpha + J_g + \ddot{\phi}J_m \end{aligned} \tag{14}$$

$$\begin{aligned} \tau_b &= \ddot{\theta}\left(I_{xy}\cos 2\phi + (I_{xx} - I_{yy})\,\mu_3\right) + J_m(\dot{\theta}^2 - \dot{\phi}^2) \\ &+ \dot{\theta}\dot{\phi}(2J_\alpha\mu_2 + I_{yy} + I_{xx} - I_{zz}) + \ddot{\phi}J_\beta + \dot{\phi}\dot{J}_\alpha\mu_1 \end{aligned} \tag{15}$$

In order to solve (14) and (15) for θ and ϕ trajectories, we arrange them as stated below.

$$\begin{aligned} \Gamma_1 &= \dot{\theta}\dot{\phi}J_m(I_{zz} - I_{xx} - I_{yy} - 2J_\alpha\mu_2) \\ &+ \tau_b J_m - \tau_a J_\beta + J_m^2(\dot{\phi}^2 - \dot{\theta}^2) + J_\beta J_g \\ &+ \dot{\phi}^2 J_\beta^2 - \dot{J}_\alpha(\dot{\theta}J_\beta + \dot{\phi}\mu_1 J_m) \end{aligned} \tag{16}$$

$$\begin{aligned} \Gamma_2 &= J_m\left[\mu_3(I_{xx} - I_{yy}) + I_{xy}\cos 2\phi\right] \\ &+ J_\beta(J_\alpha + 2I_{xy}\mu_3 + I_{xx}\mu_5 + I_{yy}\mu_4) \end{aligned} \tag{17}$$

$$\begin{aligned} \Gamma_3 &= \tau_a + \ddot{\theta}(J_\alpha + 2I_{xy}\mu_3 + I_{xx}\mu_5 + I_{yy}\mu_4 \\ &- \dot{\phi}^2 J_\beta + \dot{\theta}J_\alpha - J_g) \end{aligned} \tag{18}$$

$$\ddot{\theta} = \frac{\Gamma_1}{\Gamma_2} \qquad \ddot{\phi} = \frac{\Gamma_3}{J_m} \tag{19}$$

Equation (19) describes a pair of second order differential equations. These equations can be solved by using Runge-Kutta method[10] in an iterative fashion when following parameters are designed.

- z-axis CoM trajectory : z
- X_{zmp} and Y_{zmp} trajectories
- Initial values: $\theta_{[0]}$, $\phi_{[0]}$, $\dot{\theta}_{[0]}$ and $\dot{\phi}_{[0]}$
- Support Phase Time Interval: (t_{start}, t_{start}+t_{width})
- Swing Leg Parameters: Stride, Foot Height

B. Flight Phase Trajectory Planning

In the flight phase, a well-designed foot trajectory ensures a safe landing and stable initial conditions for the next support phase. Moreover, adjusting z-axis of foot trajectory may extend the flight time as well. Fig. 3 illustrates a flight phase where dashed green and blue lines indicates CoM and foot trajectories respectively. ψ_y is the angle between z and y axes when the robot is in the air. As we mentioned previously, we cannot use ZMP here. Thus, we employ projectile motion dynamics in this phase.

Firstly, z-axis CoM trajectory follows a parabola simply formulated as below.

$$z_{CoM} = z_{lo} + v_{zlo}t - \frac{1}{2}gt^2 \tag{20}$$

where z_{CoM}, z_{lo}, v_{zlo} and t indicate z-axis CoM trajectory, z-axis lift-off position, z-axis lift-off velocity and time variable respectively. Furthermore, z-axis foot trajectory can be

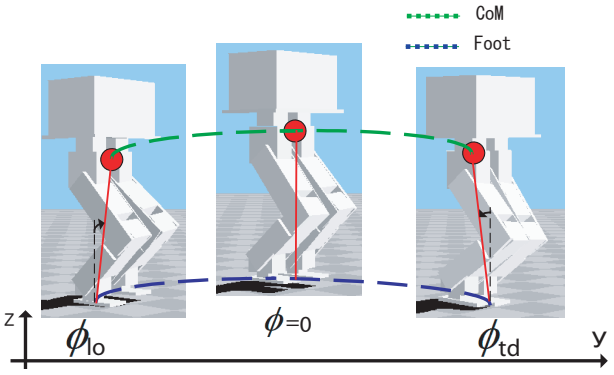


Fig. 3. Illustration of a Flight Phase

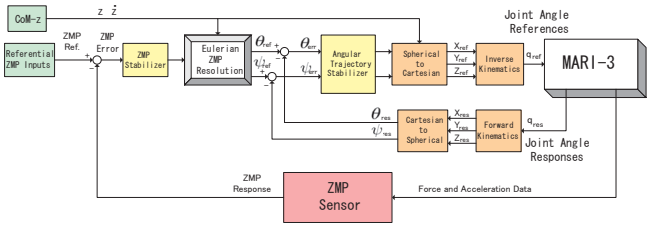


Fig. 4. Overall Control Frame, Support Phase

calculated by 5^{th} order polynomials[10]. Below, we explain the flight phase motion planning for y-z plane. An identical approach is executed for x-z plane as well.

For y-axis CoM trajectory, we use the momentum conservation as there is no forces acting on our robot through this axis[2].

$$\dot{y}_{CoM} = \frac{-y_{CoM[k-1]} + y_{CoM[k+1]}}{2\Delta t} = \upsilon_{ylo} \tag{21}$$

Here, υ_{ylo} symbolizes y-axis lift-off velocity. Since (21) appears to be in a tridiagonal form, it can be solved in a similar fashion as performed in [2]. If the discrete event q is set as $q = \frac{FlightTime}{\Delta t} - 1$, we might define boundary conditions, $y_{CoM[0]}$ and $y_{CoM[q]}$, as follows:

$$y_{CoM[0]} = -\sin\psi_{y_{lo}} \qquad y_{CoM[q]} = -\sin\psi_{y_{td}} \tag{22}$$

As it may clearly be seen, $\psi_{y_{lo}}$ and $\psi_{y_{td}}$ angles are the most crucial part in our trajectory planning. By setting these initial and terminal values, we can accelerate and decelerate the motion. Unlike the support phase, we define ψ_y angular trajectory by employing 5^{th} order polynomials[10].

Using the aforementioned parameters, foot location planning for y-axis can simply calculated by the following formulae.

$$y_{foot} = y_{CoM} - (z_{CoM} - z_{foot}) \tan\psi_y \tag{23}$$

C. Control Block

Our control strategy is based on updating calculated reference commands by feedback loops through the support phase. During the flight phase, we only consider CoM and foot trajectories within an open loop control strategy. However, either the control system runs closed loop or open loop, there are servo control blocks at the actuator level, in order to ensure exact joint positioning. In the support phase, ZMP reference is chosen as the command to ensure dynamic balance. Fig. 4 displays the general control frame through the support phase. In this frame subscripts ref, res and err symbolizes reference, response and error values respectively. Firstly, ZMP reference is compared with ZMP response and due to ZMP error reference value is updated in the ZMP stabilizer block. ZMP is measured by 6 axis F/T sensors by using the method proposed in [12]. Subsequently, updated ZMP reference and previously determined z-axis CoM trajectory are inserted to Eulerian ZMP Resolution block and angular trajectories(θ, ϕ) are obtained. These calculated angle trajectories are stabilized in Angular Trajectory Stabilizer block. As next step, spherical coordinates (r, θ, ϕ) are converted into Cartesian coordinates (x, y, z) and CoM trajectory is extracted. Finally, inverse kinematics block gives us each joint angles' references using CoM trajectory. These angles' references are given as inputs to each joint's servo controller in which both position control, velocity control and disturbance rejection are applied[16].

III. Simulation and Experimental Results

To be able to validate Eulerian ZMP Resolution method, we conducted one-legged jumping experiments on our actual bipedal robot MARI-3. Its mechanical and electrical specifications are disclosed in [14]. Furthermore, we also simulated running motion on a 3-D dynamic simulator[13] using the realistic model of MARI-3. In addition, we assumed that the robot is symmetrical for simplicity.

A. Running Simulation

In order to verify Eulerian ZMP Resolution method, we simulated running motion on ROCOS[13] by using the realistic model of our bipedal robot MARI-3. In this simulation, x-axis and y-axis ZMP references are used to generate motions on both sagittal and lateral planes. Results may be seen from Fig. 5 to Fig. 7.

In Fig. 5(a), GRF (Ground Reaction Force) response is plotted for both right leg(red) and left leg(green). In this figure, it is possible to see successful flight phases when both of the legs' GRF responses become zero. What is more, touch down impact can also be observed.

Undesired roll axis torso angle is displayed in Fig. 5(b). Since the angular momentum around CoM is included in our dynamic ZMP equations, we obtained considerably small torso angle fluctuations. Comparing to the case reported in [18], certain amount of undesired torso angle is reduced. In addition, torso angles on other axes are smaller and therefore not included.

CoM trajectories for y-axis and z-axis are illustrated in Fig. 6(a) and Fig. 6(b) respectively. In these figures, dot lines indicate reference trajectories while solid lines indicate measured trajectories. Trajectory responses follow their references well. However considerably small differences occur at moments of touch down. Additionally, the trajectory shown in Fig. 6(b) is plotted with respect to the left foot sole center. The z-axis CoM trajectory with respect to the right foot sole center is symmetrical to this result. We both use these z-axis trajectories since single support phases are switched between two legs.

ZMP response and reference for x-axis is illustrated in Fig. 7(a). In this figure, solid red, dot green, solid blue and dot purple lines are indicating right foot's ZMP response, right foot's ZMP reference, left foot's ZMP response and left foot's ZMP reference respectively. In addition, support polygon is also indicated with dark dot lines. As it may be seen, ZMP responses are always inside the support polygon and they are around their references. Same thing can be observed for y-axis ZMP which is displayed with respect right foot sole center in Fig. 7(b). Please note that, ZMP is

undefined during flight phases and we substituted zero ZMP value when the robot's foots has no contact with the floor.

B. One-Legged Jumping Experiment on Bipedal MARI-3

Having completed running simulations, we applied the same theory into our bipedal robot MARI-3 and conducted one-legged jumping experiments. In order to perform these experiments, firstly we defined an initial position in which the robot balances on its right foot without any external support. Since we aimed vertical jumping, x-axis and y-axis ZMP references are set to zero.

In these experiments, we used a jumping trajectory, in which the flight time, T_f, is planned as 0.12 [s] and support time, T_s, is planned about 0.75 [s]. As the result, we obtained successful jumping cycles within these values. However, as we did not consider compliance effect on the foot, only T_f is a bit longer than we planned. Since this 0.02 [s] time difference is relatively small, it did not affect our trajectory generation.

From Fig. 8 to Fig. 9, one legged jumping experimental results can be seen. GRF response on the right foot sole is displayed in Fig. 8(a) . In this figure, zero GRF period indicates a successful flight phase. Additionally, touch down impact may also be observed. Fig. 8(b) shows the z-axis CoM trajectory during the experiment. Firstly, we pull the CoM at the bottom and this action is indicated as pre-jump phase. After that, CoM is manipulated upwards and when it reaches the maximum value it performs a jump. Having landed on the floor, the robot turns back to the bottom position for the next cycle.

Measured x-axis ZMP and y-axis ZMP values can be observed in Fig 9(a) and Fig. 9(b) respectively. We set these values' references around zero as we conducted vertical jumping experiments. These values are almost well matched with their references and they are always inside the support polygon even under the effect of touch down impact.

We repeated this experiment several times and obtained the same results in each trial.

C. Comparison: EZR Versus Other Jumping Trajectory Planning Methods

In this subsection, we compare EZR-based jumping trajectory planning with our past experiments[1].

In [19], authors experimented jumping motion on a one legged planar robot. During this experiment, the robot is modeled as a point mass, which is in contact with the floor through a virtual spring. Until touch down, the robot follows a pre-designed sinusoidal trajectory. After touch down, force feedback is applied. Demerit of this method is the difficulty of setting the virtual spring constant. If the virtual spring is set to be stiff, the robot cannot maintain its balance after touch down. If it is not, then the robot oscillates vertically. Further, the backward jumping distance was about 7.7 [cm].

Two legged jumping experiment on MARI-3 is reported in [17]. This experiment is aimed at regulating angular momentum vector about CoM by controlling torso angle. Even though the robot landed on the floor stably without oscillating on the vertical axis, backward jumping distance was measured about 15.8 [cm]. As our bipedal robot MARI-3 has larger mass and inertia properties comparing to the one legged planar robot, backward jumping distance is dramatically increased.

Applying Eulerian ZMP Resolution, we fixed the backward jumping problem as it is measured only about 3.7 [cm]. Fig. 10 illustrates backward jumping distances in these 3 distinct experiments.

[1]These experiments' and latest experiment's videos are available on http://www.kawalab.dnj.ynu.ac.jp/~barkanu/jumpcompare.wmv

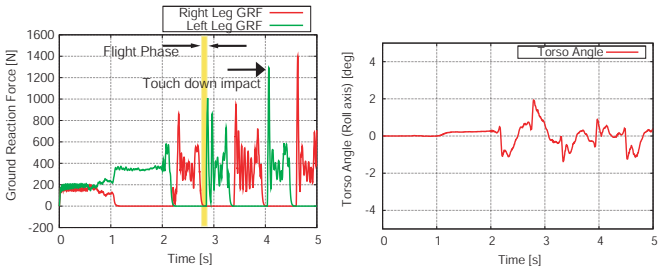
IV. Conclusion

In conclusion, we obtain a systematic way of generating ZMP-based CoM trajectories for bipedal motion generation, without ignoring or referencing angular momentum information. Unlike other methods in which angular momentum around CoM is forced to be zero, certain amount of undesired torso angle fluctuation is reduced. This certainly helps us to obtain more stable jumping and running motions.

Moreover, Eulerian ZMP Resolution method is verified by running simulations and one legged jumping experiments on MARI-3. We also compared our method's jumping performance with other available methods. In this comparison, we can see that undesired backward jumping problem is fixed.

In Eulerian ZMP Resolution, it may be observed that we could manipulate the robot by using ZMP input command. Since the input command is ZMP, this method could be used in any humanoid motion planning which includes single support phases and/or flight phases. Furthermore, we believe that Eulerian way of resolution could be applied to other methods which represent robot dynamics by considering rate change of angular momentum vector.

Having obtained successful running simulation and one legged jumping experimental results, our next work is to employ EZR method to succeed high speed running motions on the bipedal robot MARI-3.

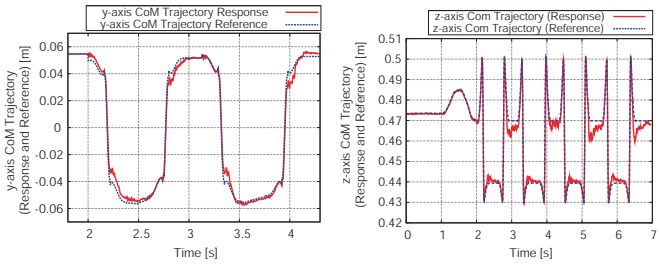


(a) Ground Reaction Force (b) Roll axis Torso Angle

Fig. 5. Running Simulation, GRF and Torso Angle (Roll)

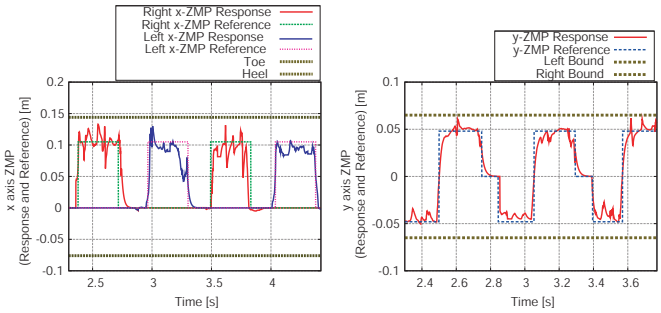
References

[1] R. Tajima, and K. Suga, "Motion having a flight phase: Experiments involving a one-legged robot," in *Proc. IEEE Conf. Intl. Rob. and Sys.*, Beijing, China, 2006, pp. 1726-1731.



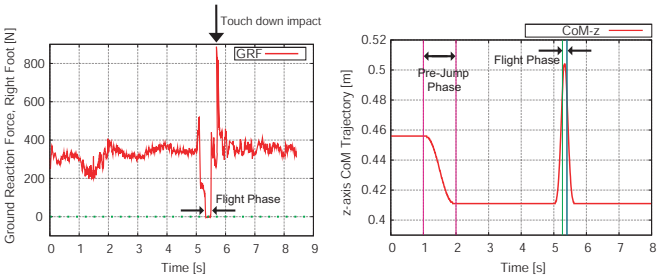
(a) y-axis CoM Trajectory (b) z-axis CoM Trajectory

Fig. 6. Running Simulation, y-axis and z-axis Trajectories



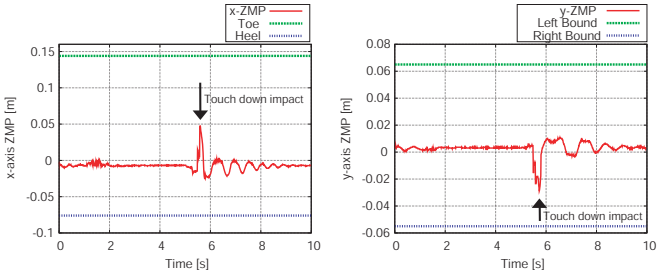
(a) x-axis ZMP (b) y-axis ZMP

Fig. 7. Running Simulation, x-axis and y-axis ZMP



(a) Ground Reaction Force (b) z-axis CoM Trajectory

Fig. 8. Jumping Experiment on MARI-3, GRF and z-axis CoM Trajectory



(a) x-axis ZMP Response (b) y-axis ZMP Response

Fig. 9. Jumping Experiment on MARI-3, x and y axis ZMP Responses

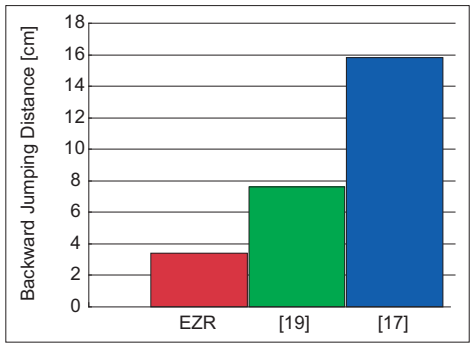


Fig. 10. Backward Jumping Distances of EZR, [17], [19]

[2] S. Kajita, K. Kaneko, M. Morisawa, S. Nakaoka, and H. Hirukawa, "ZMP-based biped running enhanced by toe springs," in *Proc. IEEE Conf. Rob. and Aut.*, Rome, Italy, 2007, pp. 3963-3969.

[3] T. Sugihara, and Y. Nakamura, "Enhancement of boundary condition relaxation method for 3D hopping motion planning of biped robots," in *Proc. IEEE Conf. Int. Rob. and Sys.*, CA, USA, 2007, pp. 444-449.

[4] T. Nagasaki, S. Kajita, K. Kaneko, K. Yokoi, and K. Tanie, "A running experiment of humanoid biped," in *Proc. IEEE Conf. Intl. Rob. and Sys.*, Sendai, Japan, 2004, pp. 136-141.

[5] K. Nishiwaki, and S. Kagami, "High frequency walking pattern generation based on preview control of ZMP," in *Proc. IEEE Conf. Rob. and Aut.*, Orlando, US, 2006, pp. 2667-2672.

[6] K. Yokoyama, H. Handa, T. Isozumi, Y. Fukase, K. Kaneko, F. Kanehiro, Y. Kawai, F. Tomita, and H. Hirukawa, "Cooperative works by a human and humanoid robot," in *Proc. IEEE Conf. Rob. and Aut.*, Taipei, Taiwan, 2003, pp. 2985-2991.

[7] M. H. Raibert, *Legged robot that balance*, MA: MIT Press, 1986.

[8] S. Timoshenko and D. H. Young *Advanced Dynamics*, McGraw-Hill Book Company, Inc., 1948.

[9] P. Moon and D. E. Spencer *Field Theory Handbook, Including Coordinate Systems, Differential Equations and Their Solutions*, New York, Springer Verlag, Inc., 1988.

[10] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery *Numerical Recipes in C: The Art of Scientific Computing* 2^{nd} *Edition*, Cambridge University Press, 1992.

[11] P. Gregorio, M. Ahmadi, and M. Buehler, "Design, control and energetics of an electrically actuated legged robot," in *IEEE Trans. Systems, Man and Cybernetics*, vol. 27, no. 2, pp. 626-634, Aug 1997.

[12] Q. Li, A. Takanishi and I. Kato, "A biped walking robot having a ZMP measurement system using universal force-moment sensors," in *Proc. IEEE Conf. Intl. Rob. and Sys.*, Osaka, Japan, 1991, pp. 1568-1573.

[13] Y. Fujimoto, and A. Kawamura "Simulation of an autonomous biped walking robot including environmental force interaction," in *IEEE Rob. and Aut. Magazine*, vol. 5, no. 2, pp. 33-42, 1998.

[14] A. Kawamura, and C. Zhu "The development of biped robot MARI-3 for fast walking and running," in *Proc. IEEE Conf. Intl. Rob. and Sys.*, Beijing, China, 2006, pp. 599-604.

[15] B. Ugurlu, and A. Kawamura, "Eulerian ZMP Resolution: Real-time Jogging and Jumping Trajectory Planning for Bipedal Robots," in *Proc. IEEE Conf. Adv. Intl. Mechat.*, Singapore, 2009, pp. 150-155.

[16] B. Ugurlu, and A. Kawamura, "Real-time jumping trajectory generation for a one legged jumping robot," in *Proc. IEEE Conf. Ind. Elec. and Con.*, Orlando, USA, 2008, pp. 1668-1673.

[17] T. Maeda, B. Ugurlu, and A. Kawamura, "Two legged jumping simulation and experiment on biped robot MARI-3," in *Adv. Motion Con.*, Trento, Italy, 2008, pp 301-305.

[18] B. Ugurlu, and A. Kawamura, "Research on maximum speed of one legged jumping robot," in 25^{th} *RSJ Proc.*, 1G35, 2007.

[19] H. Yamamoto, A. Kawamura, and C. Zhu "Experiments on one leg planar jumping robot," in 24^{th} *RSJ Proc.*, 2F17, 2006. (in Japanese)

[20] F. Xu, and A. Kawamura "3-D simulation of a one legged jumping robot using force control," in Proc. of Japan Institute of Electrical Engineers, Technical Meeting on Industrial Instrumentation and Control, IIC-08-70, Tokyo, Japan, 2008. (in Japanese)