

# ZMP-based Online Jumping Pattern Generation for a One Legged Robot

Barkan Ugurlu, *Student Member, IEEE*, and Atsuo Kawamura, *Fellow, IEEE*

**Abstract**—This paper is aimed at presenting a method to generate online jumping patterns, which can be applied to one legged jumping robots and optionally to humanoid robots. Our proposed method is based on ensuring the overall dynamic balance through the complete jumping cycle. To be able to reach this goal, we discretized the ZMP equation in polar coordinates so that we are able to include angular momentum information in a natural way. 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. Moreover, we unified support and flight phases in terms of motion generation. Having obtained successful simulation results and vertical jumping experiments in our previous work, we conducted forward jumping experiments. As the result, we obtained successful and repetitive jumping cycles, which satisfactorily verify the proposed method.

**Index Terms**—angular momentum, euler’s equations, jumping robot, polar coordinates, zmp.

## I. INTRODUCTION

RESEARCH on humanoid robots appears to be one of the most promising topics in the field. Starting from 1970s, humanoids have evolved in many aspects and they are still expected to be adapted within the social human environment. Requirements vary between different applications, such as, nuclear power plants, government installations, factories and so on[17]. Perhaps, the most intriguing case is the future soccer match between famous professional players and autonomous humanoid robots[27]. Considering these high expectations, one may think that humanoid roboticists should accelerate the advancements.

In a dynamic human environment, people may change their movement from walking to running in various situations. In a corresponding manner, a well-adapted humanoid robot should be able to run when necessary. As it may be known, running motion includes flight phases so that jumping motion appears to be a very important index in humanoid robot technology. Furthermore, humanoid robots may have to perform small jumps in order to stabilize their locomotion when they are subjected to external forces. Considering these facts, jumping motion deserves to be carefully analyzed and it gives us insights in order to improve current humanoid robots.

B. Ugurlu and A. Kawamura are with the Department of Electrical and Computer Engineering, Yokohama National University, Yokohama, 240-8501 JAPAN. e-mail: ({barkanu, kawamura}@kawalab.dnj.ynu.ac.jp)

Manuscript received November 21, 2008; revised February 17, 2009. Accepted for publication September 1, 2009.

Copyright ©2009 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org

Apart from fast mobility issues, humanoid roboticists perform research on many other applications as well. Examples vary from stair climbing[7] to developing special shoes for humanoids[8]. Unifying these exciting properties within a big picture, humanoid robot technology enhances cumulatively.

Besides humanoids, there are other types of robots to carry out human-robot interaction[9][10]. An impressive project was executed in Swiss National Exposition, Expo.02, where mobile robots served as tour guides[10]. However, most of the mobile robots have wheeled mechanisms. Therefore, they can only be employed in designated areas. On the other hand, multi-legged mobile robots could be used in different types of areas and they have versatile walking capabilities[24]. Notwithstanding, they are more suitable for outdoor activities rather than the indoor activities. All in all, requirements explained in [17] may more likely be realized by employing humanoid robots.

In order to improve humanoid robot technology, Raibert and his colleagues have developed fundamental control laws of running motion[11]. Their robots were actuated by hydraulic systems and they demonstrated impressive experiments. Applying a similar approach in control laws and considering energy efficiency, Gregorio et. al succeeded monopod running with a spring supported mechanism[15]. However these types of robots are not adequate for usual humanoid activities. What is more, Raibert’s control laws are efficient for robots with foot points whose mass is cumulated around their hips[26].

On account of these facts, researchers mainly focused on the idea of achieving running and jumping motion on general humanoids. Kajita et. al suggested an offline pattern generation technique based on the total momentum resolution, called RMC (Resolved Momentum Control)[4] and demonstrated jumping experiments on humanoids. Despite the fact that RMC covers dominant robot dynamics; it is relatively difficult to manipulate robot’s control strategy in real-time. Nevertheless, Toyota’s research group adapted Kagami’s online ZMP(Zero Moment Point)[18] resolution technique[14] into humanoid jumping and performed impressive experiments[1]. Our research group also reported real-time jumping experiments on a one legged robot[22] and two legged jumping on a bipedal robot[21].

However, both Kajita and Toyota research group consider angular momentum references as zero in their dynamic models. If the robot’s leg structure has lighter mass, this assumption may hold valid but in many cases angular momentum should not be zero referenced. When zero-referencing the angular momentum about CoM(Center of Mass), the following problems may occur:

- Leg structure must be designed as light as possible.

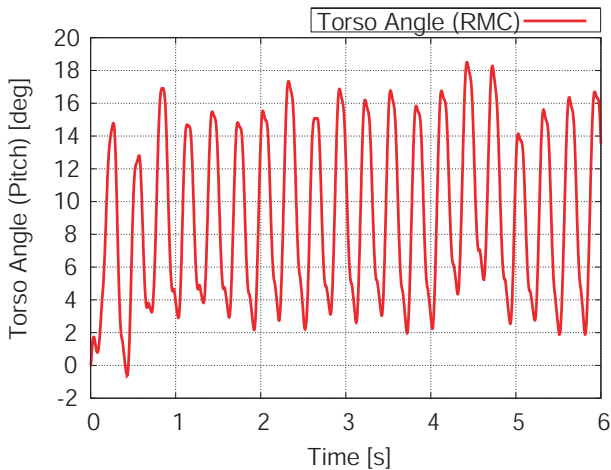


Fig. 1. Resolved Momentum Control simulation result. In this simulation, angular momentum about CoM is referenced as zero. This plot shows the torso angle fluctuation of our one legged robot.

- It causes undesired torso angle (defined in Fig. 2(b)) fluctuation and vibrations[25].
- Because of these undesired vibrations, complicated and cascaded feedback control strategies are required rather than simple stabilizers[3].

For instance, we also simulated RMC technique using a realistic robot model[25]. Torso angle fluctuation in this simulation can be examined in Fig. 1. Since the angular momentum about CoM is forced to be zero, the robot's torso angle fluctuates to maintain its angular momentum. As it may be seen in the figure, torso angle is varying from 2 degrees to 20 approximately 3 times a second and as such experiments using this method could harm our robot. This problem is examined in [2] as well.

Considering the facts mentioned above, we composed a method to generate online jumping patterns which may ensure the overall dynamic balance through a complete jumping sequence. In our proposed method, the ZMP equation is discretized in polar coordinates so that angular momentum is smoothly included in our dynamic equations. This method shares common ground with Toyota's ZMP discretization method in cartesian coordinate frame, however the main difference is that we are able to comprise angular momentum information and therefore decreased undesired torso angle fluctuation.

This paper is intended to be the follow up report of [22] through the helpful discussions and necessary improvements. Improvements include a more complete way of generating forward jumping patterns and additional blocks in the control frame. Moreover, this paper also presents forward jumping experiments as the continuation of vertical jumping experiments that we previously reported in [22]. In the present paper, our one legged jumping robot and its 3-D simulation model is introduced in section II. Our proposed online jumping pattern generation method is explained in section III. Experimental results are discussed in section IV and finally the paper is concluded in section V.

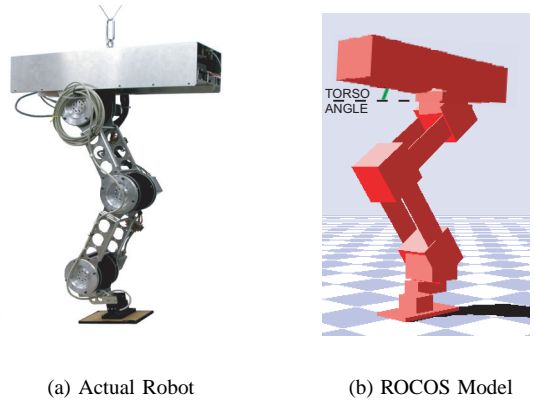


Fig. 2. One Legged Jumping Robot KEN-2 and ROCOS Model

## II. ONE LEGGED JUMPING ROBOT KEN-2

In order to explore jumping dynamics, we developed a planar one legged experimental jumping robot, called KEN-2. Its mechanical and electronic hardware structure is very similar to the actual humanoid biped robot MARI-3[20]. Thus, acquired knowledge and experience can directly be adapted to humanoid running research.

TABLE I  
MECHANICAL SPECIFICATIONS OF KEN-2

Size	Upper Leg length: 300 [mm] Lower Leg length: 300 [mm] Ankle-sole length: 170 [mm] Toe-heel length: 200 [mm]
Weight	Leg: 7.4523 [kg] Body: 6.1306 [kg] Total : 13.5829 [kg]

Table I summarizes the mechanical specifications of KEN-2. It has 3 DOF (degrees of freedom) at its ankle, knee and hip about pitch axis. Hence, its motion is constrained on the x-z plane. It might be noted that torques about pitch axes are generating forward running motion in bipedal robots as well. Moreover, KEN-2 is equipped with a 6-axis universal F/T (Force/Torque) sensor at the ankle and a gyro sensor in the upper body. Especially, placing the F/T sensor at the ankle provides a good measurement for ZMP[16]. Main controller is a 32-bit built-in Renesas SH-4 microcontroller. Data communication is performed via RON (Robot Network) bus, which is a serial communication bus system, developed in our laboratory. Fig. 2 shows the actual robot and its ROCOS[19](Robot Control Simulator) simulation model. ROCOS is a 3-D dynamic simulator, also developed in our laboratory.

## III. REAL-TIME JUMPING PATTERN GENERATION

### A. Overall Concept

Here, the main idea of generating jumping pattern is to obtain real time joint motions that ensure desired ZMP profile through the support phase and ensure predetermined CoM~foot trajectory through the flight phase, consecutively.

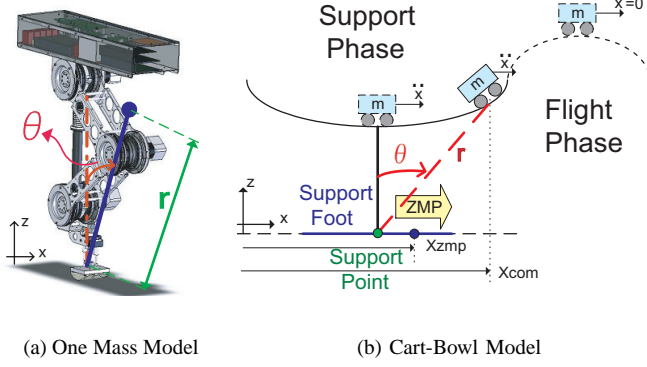


Fig. 3. Possible Presentations for Modeling

In Fig. 4, significant moments from a jumping sequence are illustrated.

During the support phase, we consider the robot as a 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. 3(a) illustrates such modeling. In this model, main parameters are CoM length,  $r$ , and angle between vertical  $z$ -axis and horizontal  $x$ -axis,  $\theta$ . We derive  $r$  by using 5<sup>th</sup> order polynomials as shown below.

$$r(t) = n_5(t - t_s)^5 + n_4(t - t_s)^4 + n_3(t - t_s)^3 + n_2(t - t_s)^2 + n_1(t - t_s) + n_0 \quad (1)$$

$$n_5 = \frac{12(r_{02} - r_{01}) - 6t_w(v_{02} + v_{01}) + t_w^2(a_{02} - a_{01})}{2t_w^5} \quad (2)$$

$$n_4 = \frac{30(r_{01} - r_{02}) - 2t_w(8v_{01} + 7v_{02}) + t_w^2(3a_{01} - a_{02})}{2t_w^4} \quad (3)$$

$$n_3 = \frac{20(r_{02} - r_{01}) - 4t_w(3v_{01} + 2v_{02}) + t_w^2(a_{02} - 3a_{01})}{2t_w^3} \quad (4)$$

$$n_2 = 0.5a_{01}, \quad n_1 = v_{01}, \quad n_0 = r_{01} \quad (5)$$

Here,  $r(t)$  is the CoM length function,  $t$ ,  $t_s$  and  $t_w$  are the time variable, start time and time width,  $r_{02}$ ,  $v_{02}$ ,  $a_{02}$ ,  $r_{01}$ ,  $v_{01}$  and  $a_{01}$  are final and initial CoM length, velocity and acceleration values respectively.  $n_i$  symbolizes the 5<sup>th</sup> order polynomial's coefficients ( $i=0\dots5$ ). These initial and final values are chosen by performing numerous simulations which fit our robot's mechanical structure and desired motion. Particularly,  $v$  values are determined by considering planned flight time and  $a$  values are chosen to be near gravity at the moment of planned lift off.

In the support phase, angular trajectory,  $\theta$ , is derived by given ZMP references as we express ZMP equation in polar coordinates and solve the tridiagonal equations in discrete domain. Having calculated both  $r$  and  $\theta$  parameters, we are able to obtain CoM trajectory by performing a conversion from polar coordinates to Cartesian coordinates. Consequently,

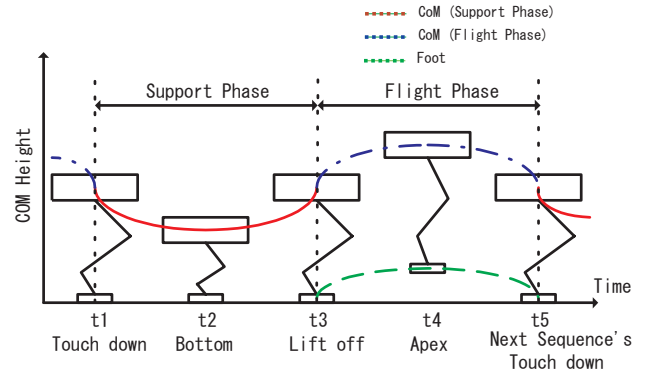


Fig. 4. A Complete Jumping Sequence

we insert CoM trajectory into inverse kinematics block and compute all joint motions.

In the flight phase, we are not able to use the ZMP criterion, as there is no contact with the floor. Therefore, we do not need to treat trajectories in polar coordinates. Considering projectile motion dynamics, we express  $z$ -axis CoM trajectory in terms of lift off velocity. For  $x$ -axis CoM position, we use the translational momentum conservation since there is no force acting on the robot about that axis.

For a simpler explanation, we can consider the cart-bowl model, shown in Fig. 3(b). Suppose that a cart is travelling on a two dimensional bowl, which is attached to the ground through a support foot and we want to calculate this cart's trajectories both in  $x$  and  $z$  axes. If the distance from support point to the cart position ( $r$ ) is given and the zero moment point of the system based on inertial forces and torques ( $\ddot{x}$ ,  $\ddot{z} + g$ ,  $\dot{L}_y$ ) through the support foot is measurable (ZMP), it is possible to derive the  $\theta$  angle indicated on the figure. Having obtained both  $\theta$  and  $r$ , we are able to calculate this cart's position both in  $x$  and  $z$  axes. Furthermore, we can also calculate the moment at which the cart leaves the bowl. When the cart is flying, it has no contact with the floor or any other effector. It is only subjected to gravitational force, which is vertical. Using this concept, we can define the next moment that the cart lands on the second bowl. Repeating this sequence as long as the cart keeps moving, it is possible to obtain its (CoM) trajectory for a complete cycle, from one bowl to another, in a unifying manner.

To sum up, we can manipulate the robot's motion using ZMP command input in the support phase and foot trajectory in the flight phase.

### B. Dynamic Balance Concept ZMP

The ZMP is a point on the floor, in which the total moment acting on the robot due to gravity, inertial forces and torques equals to zero[18]. In order to ensure dynamic balance, the ZMP response must be within the convex hull of the support polygon. Since KEN-2 has only one foot, the support polygon is the foot sole area. Therefore, if the actual ZMP is within the boundaries of foot sole area, we can say that the system is dynamically equilibrated.

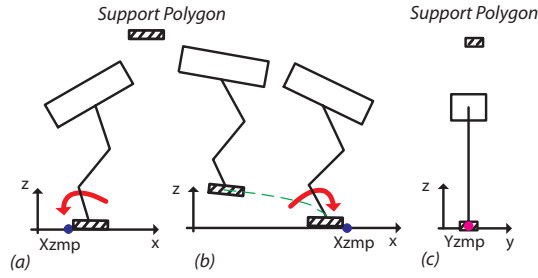


Fig. 5. Possible unstable cases on x-z plane and static balance on y-z plane

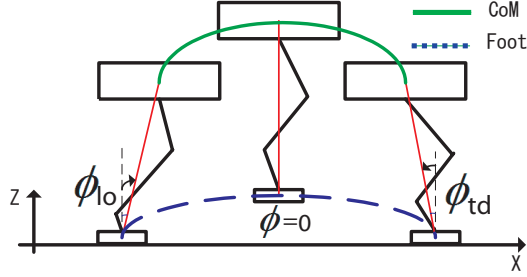


Fig. 6. Illustration of Flight Phase

In Fig. 5, three different cases are displayed. In case (a), a possible unstable case for the support phase is illustrated. Here, the x-axis ZMP is at the edge of the support polygon. Because of this, the robot cannot maintain its balance and falls over.

When the robot is in the air, the ZMP is undefined. Here, the flight phase trajectory is important as it provides safe landing. If the robot cannot follow the flight phase trajectory, the ZMP response at touch down may leave the support polygon and then the robot falls over. In Fig. 5, case (b) displays such instability situation.

As previously stated, KEN-2 cannot generate motion on the z-y plane. However, it is near symmetrical about this plane and y-axis ZMP is always within the support polygon unless some external force is applied. Case (c) in Fig. 5 explains this fact.

### C. Support Phase Trajectory Planning

1) *Motion Constraint on KEN-2:* As mentioned previously, KEN-2 has 3 actuators, which are able to generate torques only about pitch axis because of its mechanical structure. This however, makes its reference pitch axis angular momentum expression independent from other planes. Please note that, we do not limit our robot's motion and it can be moved on other planes if it is subjected to external forces or disturbances.

If the friction between the robot's foot sole and the floor is sufficient, we can consider the support phase motion as a rotation of a rigid body about a fixed point as displayed in Fig. 3(a). For this case, Euler's equations of motion might give us insights,

$$\begin{aligned} I_{xx}\dot{\omega}_x - (I_{yy} - I_{zz})\omega_y\omega_z &= \tau_x \\ I_{yy}\dot{\omega}_y - (I_{zz} - I_{xx})\omega_z\omega_x &= \tau_y \\ I_{zz}\dot{\omega}_z - (I_{xx} - I_{yy})\omega_x\omega_y &= \tau_z \end{aligned} \quad (6)$$

in which  $I_{xx}$ ,  $I_{yy}$  and  $I_{zz}$  are diagonal elements of inertia matrix around CoM,  $\omega_x$ ,  $\omega_y$  and  $\omega_z$  are angular velocities about roll, pitch and yaw axes and  $\tau_x$ ,  $\tau_y$  and  $\tau_z$  are rate change of angular momentums (torques) about roll, pitch and yaw axes. Because the motion is constrained on the x-z plane, our robot is not able to generate torques about roll and yaw axis. Therefore,  $\tau_x$ ,  $\tau_z$ ,  $\omega_x$  and  $\omega_z$  are set to zero. Moreover, the robot cannot move either on x-y plane or on z-y plane, so that y-axis ZMP is not available in our case.

In this case, rate change of pitch axis angular momentum,  $\dot{L}_y$ , can be expressed as below [12]:

$$\begin{aligned} \omega_x &= \omega_z = 0 \\ \tau_x &= \tau_z = 0 \\ \tau_y &= \dot{L}_y = I_{yy}\dot{\omega}_y = I_{yy}\ddot{\theta} \end{aligned} \quad (7)$$

As we explain later, (7) enables us to include angular momentum in our dynamic ZMP equation naturally as it is expressed in terms of  $\theta$  angle and pitch axis diagonal inertia element,  $I_{yy}$ . This inertia element is easy to calculate and we can update its value in each control cycle.

2) *ZMP Equation in Polar Coordinates:* In [1], the x-axis ZMP equation is expressed as below.

$$\dot{L}_y = mx(\ddot{z} + g) - mX_{zmp}(\ddot{z} + g) - mz\ddot{x} \quad (8)$$

where  $m$ ,  $g$ ,  $L_y$  and  $X_{zmp}$  symbolize the total mass, gravitational acceleration, pitch axis angular momentum about CoM and x-axis ZMP. Moreover,  $x$  and  $z$  indicate CoM position while  $\ddot{x}$  and  $\ddot{z}$  indicate CoM acceleration. Hence, CoM position and acceleration can be expressed as follows.

$$\begin{aligned} x &= r\sin\theta \\ z &= r\cos\theta \\ \ddot{x} &= \ddot{r}\sin\theta + 2\dot{r}\dot{\theta}\cos\theta + r\ddot{\theta}\cos\theta - r\dot{\theta}^2\sin\theta \\ \ddot{z} &= \ddot{r}\cos\theta - 2\dot{r}\dot{\theta}\sin\theta - r\ddot{\theta}\sin\theta - r\dot{\theta}^2\cos\theta \end{aligned} \quad (9)$$

Placing (7) and (9) in (8) and simplifying the resultant equation we obtain

$$\tau_{ZMP} = -(mr^2 + I_{yy})\ddot{\theta} - 2mrr\dot{\theta} + mgr\sin\theta, \quad (10)$$

in which  $\tau_{ZMP} = mX_{zmp}(\ddot{z} + g)$ , indicating the torque generated by the foot when ZMP does not coincide with the foot sole center. Please note that, angular momentum is not referenced as it is directly included in (10). In order to obtain  $\theta$  angular trajectory in our planar system model, (10) is discretized with the sampling time  $\Delta t$  (1 [ms]) step by step in the following equations [14].

$$\dot{\theta}_{[k]} = \frac{\theta_{[k+1]} - \theta_{[k-1]}}{2\Delta t} \quad (11)$$

$$\ddot{\theta}_{[k]} = \frac{\theta_{[k+1]} - 2\theta_{[k]} + \theta_{[k-1]}}{\Delta t^2} \quad (12)$$

Above, (11) and (12) display Euler's differentiation technique in discrete domain. Combining (11) and (12) in (10),

and considering  $\sin \theta \simeq \theta$  for smaller values, we can obtain the following general form of tridiagonal equations.

$$D_{[k]} = A_{[k]}\theta_{[k-1]} + B_{[k]}\theta_{[k]} + C_{[k]}\theta_{[k+1]}, \quad (13)$$

where

$$D_{[k]} = \tau_{ZMP} \ddot{z}_{[k]} + g \quad (14)$$

$$\sigma_{[k]} = -\frac{mr_{[k]}^2 + I_{yy}[k]}{\Delta t^2} \quad (15)$$

$$A_{[k]} = \sigma_{[k]} + \frac{mr_{[k]}\dot{r}_{[k]}}{\Delta t} \quad (16)$$

$$B_{[k]} = -2\sigma_{[k]} + mgr_{[k]} \quad (17)$$

$$C_{[k]} = \sigma_{[k]} - \frac{mr_{[k]}\dot{r}_{[k]}}{\Delta t}. \quad (18)$$

Equation (13) can efficiently be solved as follows[13]:

$$\begin{bmatrix} D_{[0]} \\ D_{[1]} \\ \vdots \\ D_{[p-1]} \end{bmatrix} = \begin{bmatrix} B_{[0]} & C_{[0]} & 0 & \dots & 0 \\ A_{[1]} & B_{[1]} & C_{[1]} & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & \vdots & 0 \\ 0 & \dots & A_{[p-2]} & B_{[p-2]} & C_{[p-2]} \\ 0 & \dots & 0 & A_{[p-1]} & B_{[p-1]} \end{bmatrix} \begin{bmatrix} \theta_{[0]} \\ \theta_{[1]} \\ \vdots \\ \theta_{[p-1]} \end{bmatrix} \quad (19)$$

In (19),  $p$  can be calculated as  $p = \frac{T_s}{\Delta t}$  (and rounding as needed), where  $T_s$  symbolizes support time. Finally, we define boundary conditions,  $\theta_{[0]}$  and  $\theta_{[p-1]}$  as follows:

$$\theta_{[0]} = \phi_{td} \quad \theta_{[p-1]} = \phi_{lo} \quad (20)$$

in which  $\phi_{td}$  and  $\phi_{lo}$  are touch down and lift off angles from the flight phase, displayed in Fig. 6. We discuss these angles in the next subsection.

#### D. Flight Phase Trajectory Planning

In the flight phase, a well-designed foot trajectory ensures a safe landing and proper initial conditions for the next support phase. Moreover, adjusting z-axis of foot trajectory may extend the flight time as well. Fig. 6 illustrates a flight phase where solid(green) and dot(blue) lines indicate CoM and foot trajectories respectively.  $\phi$  is the angle between vertical z-axis and horizontal x-axis when the robot is in the air. As we mentioned previously, we cannot use ZMP as it is undefined in the air. In addition, we aim to achieve zero rotation at the moment of lift off so that the whole kinetic energy is converted into potential energy. All in all, based on the aforementioned reasons, we employ projectile dynamics through the flight phase.

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

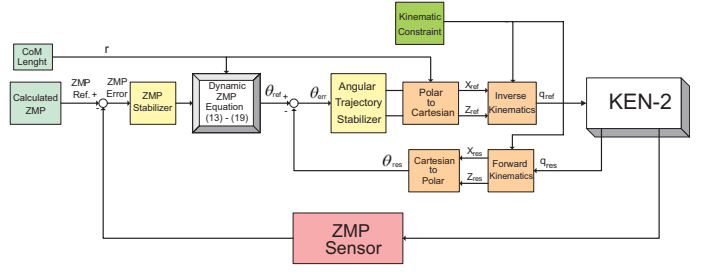


Fig. 7. Overall Control Frame, Support Phase

$$z_{CoM} = z_{lo} + v_{zlo}t - \frac{1}{2}gt^2 \quad (21)$$

where  $z_{CoM}$ ,  $z_{lo}$  and  $v_{zlo}$  indicate z-axis CoM trajectory, z-axis lift off position and z-axis lift off velocity respectively. Furthermore, z-axis foot trajectory can be calculated by a 5<sup>th</sup> order polynomial, in a similar way as calculated in equations from (1) to (5).

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

$$\dot{x}_{CoM} = \frac{-x_{CoM}[k-1] + x_{CoM}[k+1]}{2\Delta t} = v_{xlo} = E_{[k]} \quad (22)$$

In (22),  $v_{xlo}$  symbolizes x-axis lift off velocity which is constant during the flight phase. As performed in (13), (22) can efficiently be solved as follows[13]:

$$\begin{bmatrix} E_{[0]} \\ E_{[1]} \\ \vdots \\ E_{[q-1]} \end{bmatrix} = \frac{1}{2\Delta t} \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ -1 & 0 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & \dots & 0 \\ \dots & 0 & -1 & 0 & 1 \\ 0 & \dots & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_{CoM}[0] \\ x_{CoM}[1] \\ \vdots \\ x_{CoM}[q-1] \end{bmatrix} \quad (23)$$

Similarly,  $q$  can be calculated as  $q = \frac{T_f}{\Delta t}$  (and rounding as needed), where  $T_f$  symbolizes flight time. Finally, we define boundary conditions,  $x_{CoM}[0]$  and  $x_{CoM}[q-1]$  as follows:

$$x_{CoM}[0] = \sin \phi_{lo} \quad x_{CoM}[q-1] = \sin \phi_{td} \quad (24)$$

As it may clearly be seen,  $\phi_{lo}$  and  $\phi_{td}$  angles are the most crucial part in our flight phase trajectory planning. By setting these initial and terminal values, we can accelerate and decelerate the robot's motion. In the flight phase, the angular trajectory,  $\phi$ , is defined by a 5<sup>th</sup> order polynomial, which can be calculated in a similar way as done in equations (1) to (5).

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

$$x_{foot} = x_{CoM} - (z_{CoM} - z_{foot}) \tan \phi \quad (25)$$

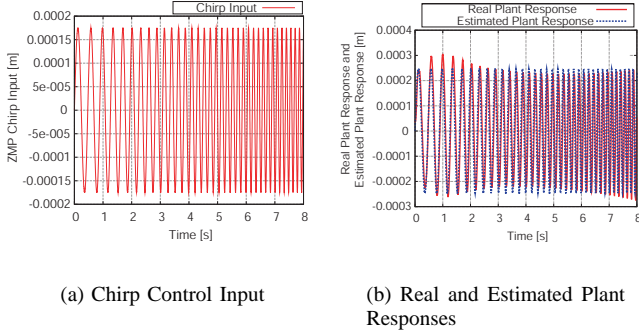


Fig. 8. System Identification

### E. 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.

1) *Static Balance*: When KEN-2 is motionless, it is in balance if joint servos are on and no external force is applied. If joint servos are off, it cannot carry its own weight and collapse. Moreover, it is symmetrical about lateral plane. Hence, it is also in balance through this plane unless some external force is applied.

2) *Overall Control Concept*: In the support phase, ZMP reference is chosen as the command input in order to ensure dynamic balance. Fig. 7 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, which is a PI type controller. ZMP is measured by a 6-axis F/T sensor by using the method proposed in [16]. Subsequently, updated ZMP reference and previously determined CoM length are inserted to dynamic ZMP equation and  $\theta$  angular trajectory is obtained. This calculated  $\theta$  angle reference is stabilized in Angular Trajectory Stabilizer block (PI controller). As next step, polar coordinates  $(r, \theta)$  are converted into Cartesian coordinates  $(x, z)$  and CoM trajectory is extracted. Finally, inverse kinematics block gives us the each joint angle's references ( $q_{ref}$ ) using CoM~foot trajectory. These joint angle references are given as inputs to each joint's servo controller in which both position control, velocity control and disturbance rejection are applied. Detailed explanation about joint servo controllers and disturbance observer are given in [22].

3) *System Identification and Controller Design*: In order to determine stabilizer block(controller) parameters, we estimated plant models by using MATLAB System Identification Toolbox. For instance, to be able to estimate the plant model between ZMP input and ZMP response, firstly feedback loops are removed. After that, we apply a chirp signal as ZMP input command and run the system open loop. This chirp signal's

TABLE II  
ZMP BASED METHODS' COMPARISON

	Comp Time	Ang Mom	Mech Indp
Toyota[1]	○	×	△
Erbatur[5]	△	×	○
Ohnishi[6]	△	×	○

amplitude is chosen to be very small so that the robot does not move. As next step, this open loop experiment data and control input are evaluated in the system identification toolbox and the estimated first order plant model is obtained as follows.

$$P_{zmp} = \frac{1.742}{26.63s + 1} \quad (26)$$

In Fig. 8(a), input chirp signal is plotted. Real and estimated plant responses are displayed in Fig. 8(b). Here, estimated plant model fits about 77.1% with real plant model.

Having estimated the plant model between ZMP input and ZMP response, we consider PI controller in the ZMP stabilizer block and assigned controller parameters as  $K_p = 9.031$  and  $K_i = 1.509$ . Implementing this controller, gain margin and phase margin are determined to be infinity and 78.2 degrees. These values indicate that the stabilizer block may provide stability. As for the angular trajectory stabilizer block, a similar approach is followed.

### F. Comparison with Other ZMP Based Methods

In this subsection, we compare our method with other recent ZMP based trajectory generation methods. Considering our particular purpose, we define comparison parameters as Computation Time (Comp Time), angular momentum inclusion (Ang Mom) and mechanism independency (Mech Indp). Table II summarizes the resulting comparison. In Table II, ×, △ and ○ symbolizes unfavorable, likely unfavorable and favorable.

Kagami and his colleagues proposed discretizing ZMP in Cartesian coordinate frame to obtain x-axis CoM trajectory for a given ZMP reference and predetermined z-axis CoM trajectory[14]. Toyota group also used this technique in [1] and performed stable jumping experiments. As different approaches using ZMP concept, Erbaturo treated ZMP equations using Fourier series approximation[5] and Ohnishi et al. proposed linear inverted pendulum mode[6].

In all methods, angular momentum information is either ignored or referenced as zero. As previously stated, forcing angular momentum around CoM causes undesired torso angles. Concerning computation time, Toyota group's method is quite effective. On the other hand, in [5] and [6], resulting equations are hyperbolic and unlikely to be calculated in real-time using built-in microprocessors. From the point of view mechanism independency, Erbaturo's and Ohnishi's methods can be applied to any humanoid type. However, Toyota group used a robot with a toe joint, which helps dynamic balance but creates redundancy.

## IV. EXPERIMENTAL RESULTS

Having obtained successful forward jumping simulation results and vertical jumping experimental results in [22],

we conducted several forward jumping experiments. In these experiments, the robot performed forward jumping once in approximately 5 seconds, as such we were able to demonstrate our method's ability to generate forward jumping patterns. The results are shown from Fig. 9 to Fig. 16.

In the experiment, flight time, forward displacement and foot height is planned as 0.09 [s], 2.25 [cm] and 2.0 [cm] respectively. Fig. 9 and Fig. 10 show planned hopping motion and estimated hopping motion. Since the robot cannot be equipped with position sensing to be able to measure such small displacements, they are estimated by inserting joint angle response into forward kinematics.

Ground reaction force (GRF) is displayed in Fig. 11. Here, zero GRF indicates a successful flight phase. It is also possible to see the touch down impact, which is measured to be about 600 [N]. Comparing to our previous experiments, this touch down impact is reduced by making the CoM height minimum, right after the collision. This approach is firstly proposed in [3]. However, our method is empirical as we make the CoM height as low as we could. On the other hand, GRF response at the touch down might be more than 600 [N] as universal F/T sensors response slowly. Hence, they might miss the peak of the GRF.

Fig. 12 illustrates z-axis CoM trajectory with respect to the foot sole center. In this figure, we can see that its peak value is well conserved in each flight. Both GRF and z-axis CoM trajectory results are very similar to results that we reported in [22].

As previously mentioned, ZMP is our command input in the proposed method. When we change the ZMP input command, the robot moves forward and based on the designated pattern it performs forward jumping. In Fig. 13, dot line (black) shows our reference x-axis ZMP input command and solid line (red) shows the measured x-axis ZMP response. Here, it may be seen that ZMP response is following the ZMP command in each jumping cycle. What is more, the response always lies within support polygon (toe-heel) which is demonstrated with dot line (blue), and solid (purple) line. Please note that, our robot is not able to move on either x-y plane or z-y plane, therefore, we did not measure y-axis ZMP response.

Angular momentum about pitch axis result can be seen in Fig. 14. In this figure, angular momentum reaches to  $0.25 [kgm^2/s]$  then decrease to a very small value at the moment of lift off. During the flight phase, its value remains unchanged as the angular momentum about CoM is naturally conserved. Then due to the touch down impact, it reaches to  $-0.4 [kgm^2/s]$  for a fairly short period. Lastly, it becomes zero as the robot is not in motion. This result also proves our approach for considering angular momentum as well.

Finally, we may observe torso angle fluctuations for two distinct cases. Primarily, we omitted the rate change of angular momentum term in equation (8) and conducted a forward jumping experiment. As it may be observed in Fig. 16, the robot's torso angle fluctuated between +20 and -20 degrees. Consequently, the robot lost its balance and this is symbolized as a mat area in the figure. Secondly, we did not omit the rate change of angular momentum term and repeated the same experiment. In this case, the maximum torso angle is measured

around 5.4 degrees as it can be seen in Fig. 15. Moreover, the robot never did not lose its balance during the forward jumps. Comparing to the first case, we obtained a significant decrease in torso angle fluctuation when applying the proposed method.

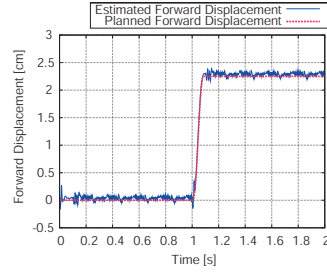


Fig. 9. Forward Displacement

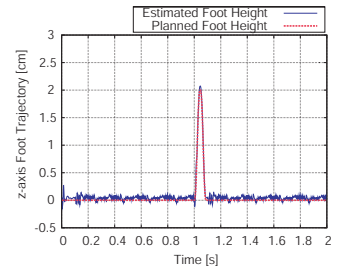


Fig. 10. Foot Height

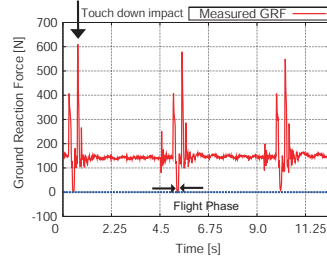


Fig. 11. Ground Reaction Force

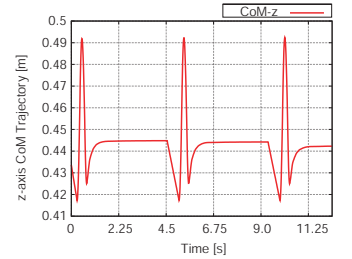


Fig. 12. z-axis CoM Trajectory

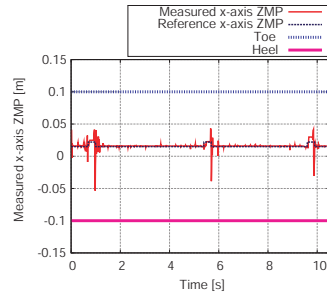


Fig. 13. x-axis ZMP

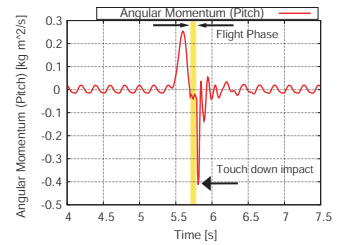


Fig. 14. Angular Momentum (Pitch)

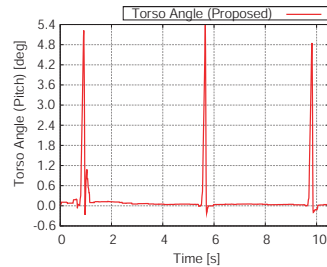


Fig. 15. Torso Angle (Proposed)

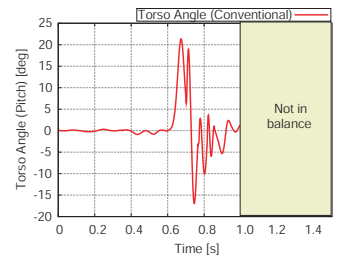


Fig. 16. Torso Angle (Conventional)

## V. CONCLUSION

As conclusion, we obtain a systematic way of generating CoM trajectory based on ZMP, without referencing angular momentum information. Unlike other methods in which angular momentum about CoM is forced to be zero, a great

amount of undesired torso angle fluctuation is decreased. This certainly helps us to obtain more feasible jumping and running motion.

Moreover, our proposed method is verified by performing several forward jumping experiments as stated. It may be observed that we could manipulate the robot just by commanding the ZMP input. Since the command input is ZMP, this method might be applied to any planar humanoid motion, which includes single support phases.

Recently we extended this method for 3-D case, in the name of "Eulerian ZMP Resolution" to be able to obtain trajectories for bipedal jogging and running[23]. In this method, we can both manipulate motions on sagittal and lateral planes by commanding x-axis and y-axis ZMP inputs. Therefore, our next report will be about bipedal jogging and running.

## REFERENCES

- [1] R. Tajima, and K. Suga, "Motion having a flight phase: Experiments involving a one-legged robot," in *Proc. IEEE Conf. Intelligent Robots and Systems*, Beijing, China, pp. 1726-1731, 2006.
- [2] S. Kajita, K. Kaneko, M. Morisawa, S. Nakaoka, and H. Hirukawa, "ZMP-based biped running enhanced by toe springs," in *Proc. IEEE Conf. Robotics and Automation*, Rome, Italy, pp. 3963-3969, 2007.
- [3] S. Kajita, N. Nagasaki, K. Kaneko, K. Yokoi, and K. Tanie, "A running controller of humanoid biped HRP-2LR," in *Proc. IEEE Conf. Robotics and Automation*, Barcelona, Spain, pp. 618-624, 2005.
- [4] S. Kajita, T. Nagasaki, K. Kaneko, K. Yokoi, and K. Tanie, "A hop towards running humanoid biped," in *Proc. IEEE Conf. Robotics and Automation*, New Orleans, US, pp. 629-635, 2004.
- [5] K. Erbaturo, and O. Kurt, "Natural ZMP trajectories for biped robot reference generation," *IEEE Trans. on Industrial Electronics*, vol. 56, no. 3, pp. 835-845, 2009.
- [6] N. Motoi, T. Suzuki, and K. Ohnishi, "A bipedal locomotion planning based on virtual linear inverted pendulum mode," *IEEE Trans. on Industrial Electronics*, vol. 56, no. 1, pp. 54-61, 2008.
- [7] C. Fu, and K. Chen, "Gait synthesis and sensory control of stair climbing for a humanoid robot," *IEEE Trans. on Industrial Electronics*, vol. 55, no. 5, pp. 2111-2120, 2008.
- [8] H. Minekata, H. Seki, and S. Tadokuma, "A study of energy-saving shoes for robot considering lateral plane motion," *IEEE Trans. on Industrial Electronics*, vol. 55, no. 3, pp. 1271-1276, 2008.
- [9] T. Takeda, Y. Hirata, and K. Kosuge, "Dance step estimation method based on HMM for dance partner robot," *IEEE Trans. on Industrial Electronics*, vol. 54, no. 2, pp. 699-706, 2007.
- [10] B. Jensen, N. Tomatis, L. Mayor, A. Drygajlo, and R. Siegwart, "Robot meet humans - Interaction in public spaces," *IEEE Trans. on Industrial Electronics*, vol. 52, no. 6, pp. 1530-1546, 2005.
- [11] M. H. Raibert, *Legged robot that balance*, Cambridge, MA: MIT Press, pp. 57-79, 1986.
- [12] S. Timoshenko and D. H. Young *Advanced Dynamics*, McGraw-Hill Book Company, Inc., pp. 359-366, 1948.
- [13] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery *Numerical Recipes in C: The Art of Scientific Computing 2<sup>nd</sup> Edition*, Cambridge University Press, 1992.
- [14] S. Kagami, T. Kitagawa, K. Nishiwaki, T. Sugihara, M. Inaba, and H. Inoue "A fast dynamically equilibrated walking trajectory generation method of humanoid robot," in *Autonomous Robots*, Kluwer Academic Publishers, vol.2, issue 1, pp. 71-82, 2002.
- [15] 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, 1997.
- [16] 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. Int. Sys. and Robots*, Osaka, Japan, pp. 1568-1573, 1991.
- [17] P. Gupta, V. Tirth, and R. K. Srivastava "Futuristic Humanoid Robots: An Overview," in *Proc. IEEE Conf. Industrial and Information Systems*, Sri Lanka, pp. 247-254, 2006.
- [18] M. Vukobratovic, and B. Borovac, "Zero-Moment Point - Thirty Five Years of its Life," in *Int. Journal of Humanoid Robotics*, Vol 1, No. 1, pp. 157-173, 2004.
- [19] Y. Fujimoto, and A. Kawamura "Simulation of an autonomous biped walking robot including environmental force interaction," in *IEEE Robotics and Automation Magazine*, vol. 5, no. 2, pp. 33-42, 1998.
- [20] A. Kawamura, and C. Zhu "The development of biped robot MARI-3 for fast walking and running," in *Proc. IEEE Conf. Intelligent Robots and Systems*, Beijing, China, pp. 599-604, 2006.
- [21] T. Maeda, B. Ugurlu, and A. Kawamura, "Two legged jumping simulation and experiment on biped robot MARI-3," in *Advanced Motion Control*, Trento, Italy, pp 301-305, 2008.
- [22] B. Ugurlu, and A. Kawamura, "Real-time jumping pattern generation for a one legged jumping robot," in *Proc. IEEE Conf. Industrial Electronics and Control*, Orlando, US, pp. 1668-1673, 2008.
- [23] B. Ugurlu, and A. Kawamura "Real-time running and jumping pattern generation for bipedal robots based on ZMP and Euler's equations," in *Proc. IEEE Conf. Intelligent Robots and Systems*, St. Louis, US, to be presented, 2009.
- [24] B. Ugurlu, C. M. Uzundere, H. Temeltas, and A. Kawamura, "Mechanical and electrical design of a two segmental eight-legged mobile robot for planetary exploration," in *Proc. Int. Conf. Climing and Walking Robots Association*, Coimbra, Portugal, pp. 1061-1068, 2008.
- [25] B. Ugurlu, and A. Kawamura, "Research on maximum speed of one legged jumping robot," in *25<sup>th</sup> Japan Robotics Society*, IG35, 2007.
- [26] 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)
- [27] UEFA official web-site, "We are the robots" [Online]. Available: <http://www.uefa.com/magazine/news/kind=1024/newsid=78991.html>



applicable. He is a student member of IEEE, the IEE of Japan and Robotics Society of Japan.



an associate professor, and became a professor in 1996. Prof. Kawamura is a member of the IEE of Japan, IEEE(Fellow), Robotics Society of Japan, the Institute of Electronics, Information and Communication Engineers, the Society of Instrument and Control Engineering.

**Barkan Ugurlu (S'08)** was born in Cankiri, Turkey in 1981. He received his B.E. in electrical engineering with high honor from Yildiz Technical University, Turkey and his M.E. in mechatronics engineering from Istanbul Technical University, Turkey in 2004 and 2006, respectively. He is currently a Ph.D. candidate in the department of electrical and computer engineering, Yokohama National University, Japan. His research interests are bipedal robot dynamics, bipedal motion planning and all dynamic systems where Eulerian equations of motion are

**Atsuo Kawamura (S'77-M'81-SM'96-F'02)** was born in Yamaguchi, Japan in 1953. He received his B.E., M.E. and Ph.D. degrees all in electrical engineering from University of Tokyo in 1976, 1978 and 1981, respectively. Upon receiving his Ph.D., he joined the department of electrical and computer engineering at University of Missouri-Columbia as a postdoctoral fellow, and was an assistant professor there from 1983 through 1986. In 1986 he joined the department of electrical and computer engineering at Yokohama National University, Yokohama, Japan as