ZMP-Based Biped Running Control

The HRP-2LR Humanoid Biped Robot

Humanoid robots are currently one of the most exciting topics in the field of robotics, and many projects on the topic exist [1]–[6]. Most of the projects focus on biped walking. Reliable dynamic biped walking has already been demonstrated. Watching those successful demonstrations, we can ask a natural question, "Can we build a humanoid robot that can run?"

We believe this is worthwhile as a technical challenge for the following reasons. First, enabling humanoid robots to run increases their mobility. For example, jumping over large obstacles or a crevasse in the ground might be realized by a derivative of running control. Second, studying extreme dynamic tasks will give insights into how to improve the robot's mechanical design. Current robots are too fragile to operate in environments that are not carefully controlled. Even when humanoids operate at low speeds, they must be treated with care. We hope to overcome this fragility in the process of developing a running humanoid.

Running robots have been studied extensively by Raibert and his colleagues [7]. Their famous hopping robots, which are driven by pneumatic and hydraulic actuators, performed various actions, including somersaults [8]. Using a similar control strategy, Hodgins simulated a running human for use in computer graphics [9]. Ahmadi and Buehler studied running monopods from a standpoint of energy efficiency. Their ARL Monopod II is an electrically powered running robot that weighs 18 kg and can run at 1.25 m/s with a power expenditure of only 48 W [10].

All of these robots have a spring mechanism to retrieve kinetic energy during running cycles. It is obvious that these springs help running, but they might prevent ordinary humanoid activities such as walking and carrying objects. Since our intention is to add a running function to a versatile humanoid robot, we started with a mechanism without springs. A similar approach is taken by Gienger et al. [1], Nagasaka et al. [11], and Chevallereau et al. [12]. In 2005, Honda announced that they developed the next-generation ASIMO that can run at 6 km/h (http://world.honda.com/ASIMO). So far, its technical details have not yet been disclosed.

In this article, we define running as a biped locomotion with flight phase. We start with a biped robot that has already performed a reliable zero-moment point (ZMP) based walk. Since the ZMP can not be defined during the flight phase, our main interest is how to expand the ZMP concept to running control. Considering this, we present the details of the running biped HRP-2LR: its hardware, the pattern generation, the stabilization, and the experimental result. The following section describes the hardware of HRP-2LR. Then, we plan a running motion for a simple inverted pendulum and its center of mass (CoM). This running motion is then transformed into a running pattern for HRP-2LR by the algorithm shown in the next section. Following that, we explain the stabilizer, a real-time controller to realize a given running pattern; the experimental result is given; and the article is concluded.

Humanoid Biped HRP-2LR

HRP-2LR is a 12 degrees of freedom (DoF) biped robot with legs of humanoid configuration. The robot was originally developed in the humanoid robotics project (HRP) as a prototype leg module in 2001 [13] and remodeled in 2003 for running. Figure 1 shows HRP-2LR running.

BY SHUUJI KAJITA, TAKASHI NAGASAKI, KENJI KANEKO, AND HIROHISA HIRUKAWA

1070-9932/07/$25.00©2007 IEEE

Table 1 shows the important dimensions of HRP-2LR. Its total weight is 31.0 kg and the height is 1.27 m. The body contains a three-axes acceleration sensor, three gyros for angular velocity sensing, twelve servo drivers, and a CPU board (Pentium III, 933 MHz). Each foot is equipped with a six-axes force sensor and rubber bushing, which protects the sensor and robot from the touchdown impact. HRP-2LR is actuated by dc servo motors with harmonic drive gears, as shown in Table 2. Pulleys are used to modify the final reduction ratio of each joint so that the total reduction can be modified based on experiments. For the detailed specifications of the robot hardware, see [13].

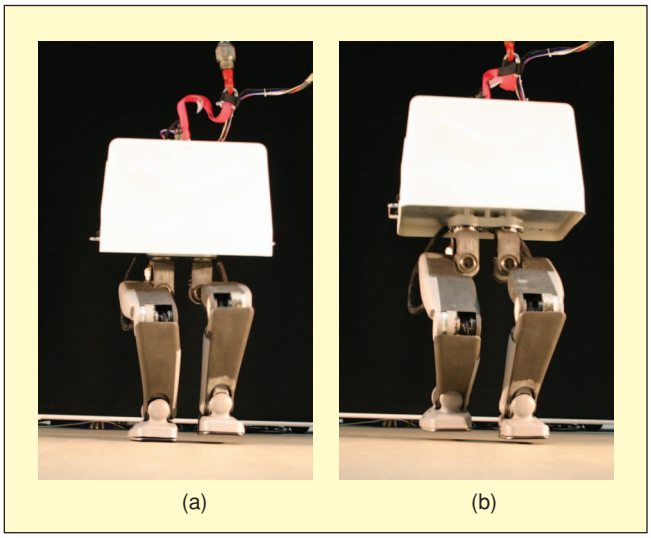

Figure 1. *HRP-2LR running. (a) Support phase and (b) flight phase.*

Table 1. Specifications of HRP-2LR.

	Six DOF/Leg (Hip: 3 Knee: 1 Ankle: 2)	
Size	Upper leg length:	300 mm
	Lower leg length:	300 mm
	Ankle-sole height:	93 mm
	Length between hip joints:	120 mm
	Toe-heel length:	170 mm
Weight	Legs: 8.6 kg/leg × 2 legs:	17.2 kg
	CPU, I/O boards and drivers:	7.0 kg
	Body structure:	6.8 kg
	Total:	31.0 kg

Table 2. Actuators and harmonic drive gears (HDG).

Joint		**Actuator**	**Ratio of HDG**
Hip	Yaw	dc 20 W	1:160
	Roll	dc 90 W	1:160
	Pitch	dc 90 W	1:120
Knee	Pitch	dc 150 W	1:100
Ankle	Pitch	dc 90 W	1:160
	Roll	dc 70 W	1:160

ZMP-Based Running Pattern Generation

Vertical CoM Trajectory

First, we design the vertical CoM motion assuming a simple hopping robot of Figure 2. Its dynamics is given by

$$M\ddot{c}_z = f_z - Mg\,, \tag{1}$$

where M is the mass of the robot, c_z is the height of the CoM, and g is gravity acceleration.

We prescribe the vertical floor reaction force f_z, which is the only input of this system. For one hopping cycle, it was specified to be

$$f_z = \begin{cases} F_0 & 0 \le t < \lambda T_s \\ F_0\left\{1 - \left(\frac{t-\lambda T_s}{(1-\lambda)T_s}\right)^2\right\} & \lambda T_s \le t < T_s \\ 0 & T_s \le t < T_s + T_f, \end{cases} \tag{2}$$

where T_s and T_f are the periods of the support phase and the flight phase, respectively. From the touchdown, $t = 0$ to λT_s, the floor reaction force is kept constant at F_0, then from $t = \lambda T_s$ to T_s, it is decreased to zero. From $t = T_s$ to $T_s + T_f$, the robot is in the flight phase, thus the floor reaction force is zero.

To realize a steady hopping motion, the average floor reaction force must coincide with the force due to gravity

$$\frac{1}{T_s + T_f}\int_0^{T_s+T_f} f_z\,dt = Mg. \tag{3}$$

Substituting (2) into (3), we obtain

$$F_0 = \frac{3}{2+\lambda}\left(1 + \frac{T_f}{T_s}\right)Mg\,. \tag{4}$$

A dynamically consistent floor reaction force that will realize a steady hopping motion is given by (2) and (4). By integrating (1) with this floor reaction force, we can easily obtain a reasonable motion of the CoM. Figure 3 shows an example CoM trajectory and the floor reaction force.

We can modify the hopping motion by modifying λ in the interval $[0,\ 1]$. By choosing $\lambda = 1$, we get a rectangular shaped floor reaction force, which has discontinuity at the lift off. When $\lambda < 1$, we get a smooth force pattern, as illustrated in Figure 3(b). By using such a smooth pattern, we can expect to reduce the mechanical stress on the robot. In the following simulations and experiments, we consistently use the hopping pattern of $\lambda = 0.9$.

Horizontal CoM Trajectory

To calculate CoM trajectories in three-dimensional (3-D) space, force balance will be used. Figure 4 illustrates the force acting on a running robot modeled by a simple inverted pendulum. We assume that its supporting foot is located at $[p_x\ p_y\ 0]$, which corresponds to the ZMP [14], which is determined in advance. For straight-line running at a constant speed, the ZMP of the jth step can be specified as

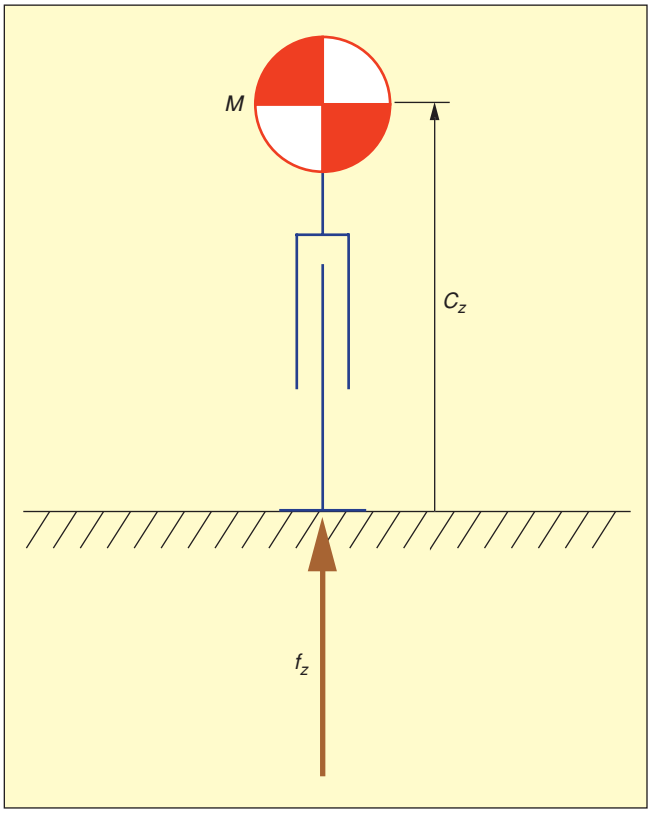

Figure 2. *Model to design a hopping trajectory.*

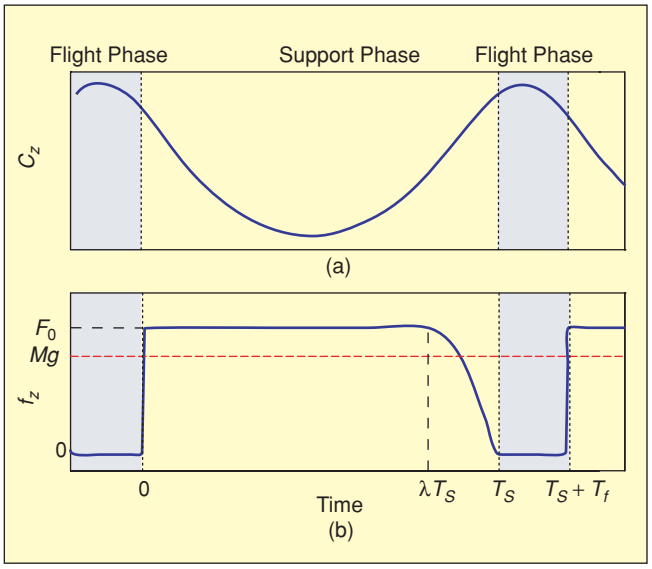

Figure 3. *Floor reaction force and vertical motion.*

$$p_{x(j)} = s_d\, j, \tag{5}$$
$$p_{y(j)} = -0.5\, y_d (-1)^j, \tag{6}$$

where s_d is the desired step length, and y_d is the lateral distance between the feet.

As shown in Figure 4, the floor reaction force vector must point the CoM at $[c_x\ c_y\ c_z]$, and therefore,

$$p_x = c_x - \frac{c_z \ddot{c}_x}{g + \ddot{c}_z}, \tag{7}$$
$$p_y = c_y - \frac{c_z \ddot{c}_y}{g + \ddot{c}_z}. \tag{8}$$

Our goal is to calculate the horizontal CoM trajectory $[c_x\ c_y]$ to satisfy the above equations using c_z, p_x, and p_y. For this purpose, the method proposed by Nishiwaki and Kagami [15] is used. In the following explanation, the sagittal CoM trajectory c_x will be calculated. The same technique can be used for the lateral motion c_y.

First, discretize (7) with a small time step Δt. This is done by making the following substitutions:

$$c_x(i\Delta t) \rightarrow c_{x(i)},$$
$$\ddot{c}_x(i\Delta t) \rightarrow \frac{c_{x(i-1)} - 2c_{x(i)} + c_{x(i+1)}}{\Delta t^2}. \tag{9}$$

The discretized equation is

$$p_x = \alpha_i c_{x(i-1)} + \beta_i c_{x(i)} + \alpha_i c_{x(i+1)},$$
$$\alpha_i := -\frac{1}{\Delta t^2}\frac{c_z(i\Delta t)}{g + \ddot{c}_z(i\Delta t)},$$
$$\beta_i := 1 + \frac{2}{\Delta t^2}\frac{c_z(i\Delta t)}{g + \ddot{c}_z(i\Delta t)}. \tag{10}$$

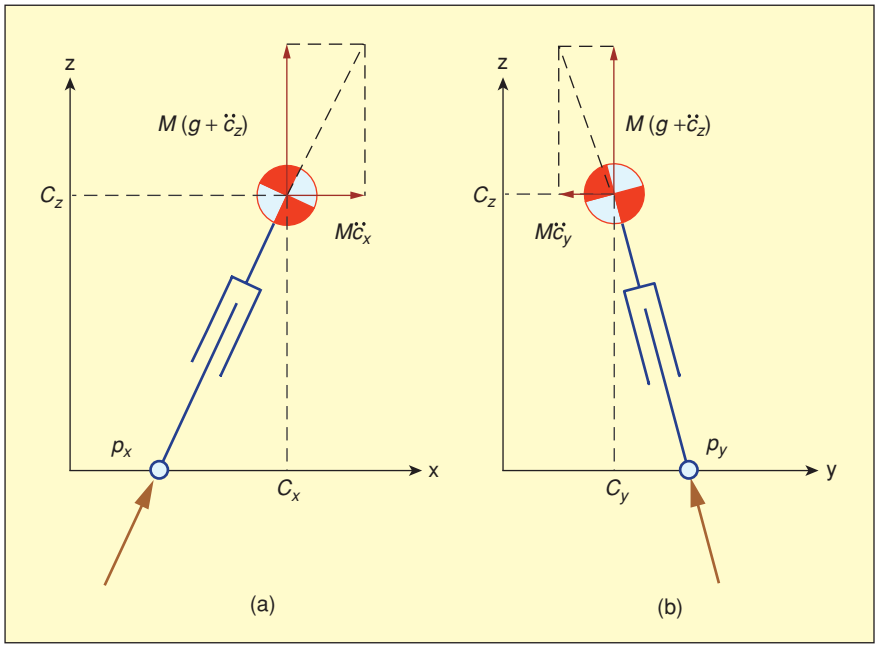

Figure 4. *Horizontal force balance during the support phase.*

The following matrix equation is obtained by choosing $N_s = T_s/\Delta t$ (and rounding as needed) and using (10)

$$\overbrace{\begin{pmatrix} p_x \\ p_x \\ \vdots \\ p_x \end{pmatrix}}^{\boldsymbol{P}} = \overbrace{\begin{bmatrix} \alpha_1 & \beta_1 & \alpha_1 & & \mathbf{0} \\ & \alpha_2 & \beta_2 & \alpha_2 & \\ & \ddots & \ddots & \ddots & \\ \mathbf{0} & & \alpha_{N_s} & \beta_{N_s} & \alpha_{N_s} \end{bmatrix}}^{\boldsymbol{K}} \begin{pmatrix} c_{x(0)} \\ c_{x(1)} \\ c_{x(2)} \\ \vdots \\ c_{x(N_s)} \\ c_{x(N_s+1)} \end{pmatrix}. \tag{11}$$

During the flight phase, the robot has no horizontal acceleration, i.e., $\ddot{c}_x = 0$. Therefore,

$$0 = c_{x(i-1)} - 2c_{x(i)} + c_{x(i+1)}\,. \tag{12}$$

The following matrix equation is obtained by choosing $N_f = T_f/\Delta t$ (and rounding as needed) and using (12)

$$\overbrace{\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}}^{\mathbf{0}} = \overbrace{\begin{bmatrix} 1 & -2 & 1 & & \mathbf{0} \\ & 1 & -2 & 1 & \\ & \ddots & \ddots & \ddots & \\ \mathbf{0} & & 1 & -2 & 1 \end{bmatrix}}^{\boldsymbol{J}} \begin{pmatrix} c_{x(N_s)} \\ c_{x(N_s+1)} \\ c_{x(N_s+2)} \\ \vdots \\ c_{x(N_s+N_f+1)} \end{pmatrix}. \tag{13}$$

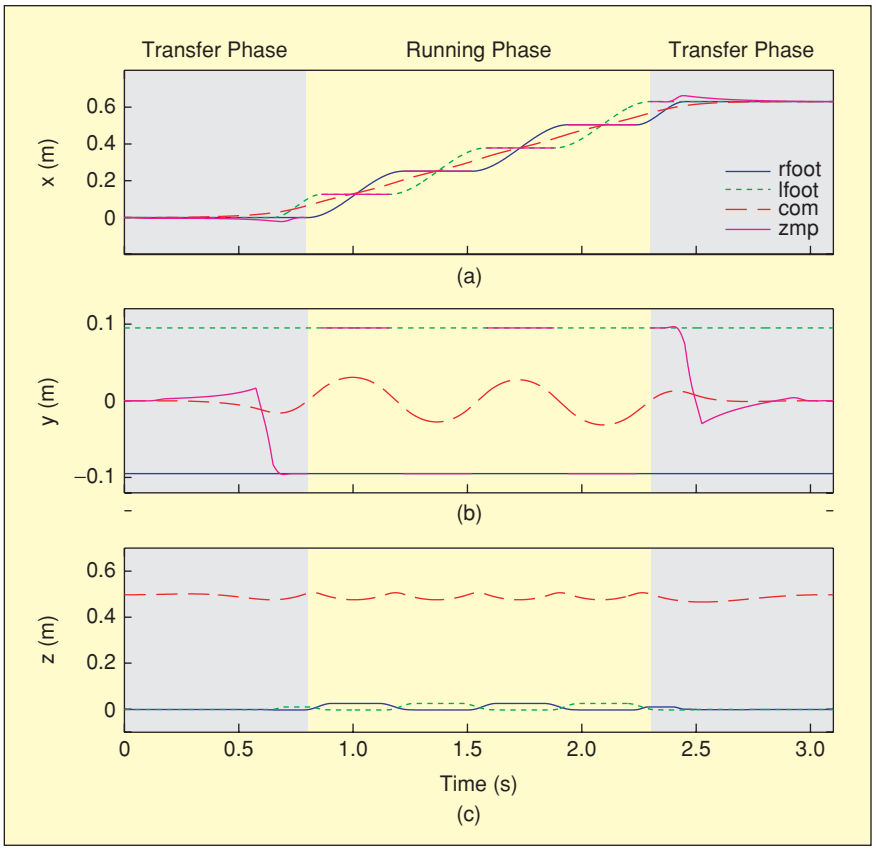

Figure 5. *CoM and foots trajectories for running.*

A matrix equation for m step running (m support periods and $m+1$ flight periods) is obtained by combining (11) and (13)

$$\begin{pmatrix} \mathbf{0}' \\ \boldsymbol{P}_1 \\ \mathbf{0} \\ \boldsymbol{P}_2 \\ \vdots \\ \mathbf{0}' \end{pmatrix} = \mathbf{X} \begin{pmatrix} c_{x(0)} \\ c_{x(1)} \\ c_{x(2)} \\ \vdots \\ c_{x(N)} \end{pmatrix}$$

$$N := mN_s + (m+1)N_f$$

$$\mathbf{X} := \begin{bmatrix} \boldsymbol{J}_0' & & & & & \mathbf{0} \\ & \boldsymbol{K}_1 & & & & \\ & & \boldsymbol{J}_1 & & & \\ & & & \boldsymbol{K}_2 & & \\ & & & & \ddots & \\ \mathbf{0} & & & & & \boldsymbol{J}_m' \end{bmatrix}, \tag{14}$$

where the terms with $'$ indicate the modifications used to satisfy boundary conditions. The boundary conditions are the initial and the terminal CoM speeds

$$\dot{c}_{x(0)} = v_0\,, \tag{15}$$

$$\dot{c}_{x(N)} = v_N\,. \tag{16}$$

The terminal CoM speed introduces the constraint

$$c_{x(N+1)} = c_{x(N)} + v_N \Delta t. \tag{17}$$

By substituting (17) into (12), the following equation, which corresponds to the bottom row of the matrix equation (14), is obtained

$$-v_N \Delta t = c_{x(N-1)} - c_{x(N)}\,. \tag{18}$$

The first row of (14) is also obtained in the same way using the initial CoM speed v_0.

Since the matrix $\boldsymbol{X}$ is a square matrix, the CoM trajectory may be obtained by solving (14)

$$\begin{pmatrix} c_{x(0)} \\ c_{x(1)} \\ c_{x(2)} \\ \vdots \\ c_{x(N)} \end{pmatrix} = \boldsymbol{X}^{-1} \begin{pmatrix} \mathbf{0}' \\ \boldsymbol{P}_1 \\ \mathbf{0} \\ \boldsymbol{P}_2 \\ \vdots \\ \mathbf{0}' \end{pmatrix}. \tag{19}$$

Although $\boldsymbol{X}$ is a matrix of row dimension in the thousands, it is tridiagonal and can be solved efficiently [16].

Figure 5 shows an example CoM pattern for five steps of running. In

Figure 5(a) and (b), the sagittal and lateral ZMP trajectories are indicated by bold lines. The CoM trajectories are plotted using dashed lines. For the transfer phases to start and finish the running (shaded area), a different algorithm was used [17].

Joint Trajectories

In this section, we describe the transformation from the CoM trajectory into the joint trajectories that can be applied to the actual biped robot. The basic idea is to generate the robot motion so that the resultant momentum becomes the desired value. We call this method resolved momentum control [18].

Full Body Kinematics

Figure 6 illustrates the configuration of HRP-2LR and the reference frames for pattern generator. The body translation is represented by the 3-D vector $\boldsymbol{p}_B$, which indicates the midpoint of the hip joints with respect to the origin of the ground-fixed frame O. The body posture is represented by a 3×3 matrix $\boldsymbol{R}_B$. The joint angles of each leg are given as a six-dimensional (6-D) vector $\boldsymbol{q}_{\text{leg}_i}$ for the right ($i = 1$) and left ($i = 2$). The position and orientation of the feet are represented as $\boldsymbol{p}_i$, $\boldsymbol{R}_i$ ($i = 1, 2$).

The speed of a link in space is given by

$$\boldsymbol{v}_i = \dot{\boldsymbol{p}}_i \tag{20}$$

$$\omega_i = (\dot{\boldsymbol{R}}_i \boldsymbol{R}_i^T)^\vee, \qquad (i = 1, 2, B)\,. \tag{21}$$

The operator $^\vee$ translates a skew-symmetric 3×3 matrix into a 3-D vector.

The relationship of velocity between the feet and the leg joint can be expressed as

$$\begin{bmatrix} \boldsymbol{v}_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{E} & -\widehat{\boldsymbol{r}}_{B\to i} \\ \boldsymbol{0} & \boldsymbol{E} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_B \\ \omega_B \end{bmatrix} + \boldsymbol{J}_{\text{leg}_i}\, \dot{\boldsymbol{q}}_{\text{leg}_i}, \tag{22}$$

where $\boldsymbol{E}$ is a 3×3 identity matrix and $\boldsymbol{J}_{\text{leg}_i}$ (6×6) is the Jacobian matrix calculated from the leg configuration. The operator $\widehat{\ }$ translates a 3-D vector into a skew-symmetric 3×3 matrix, which is equivalent to a cross product. The vector $\boldsymbol{r}_{B\to i}$ points from the body to foot i

$$\boldsymbol{r}_{B\to i} := \boldsymbol{p}_i - \boldsymbol{p}_B\,.$$

When the speeds of the body and the feet were given, we can calculate the joint speed from (22) as follows,

$$\dot{\boldsymbol{q}}_{\text{leg}_i} = \boldsymbol{J}_{\text{leg}_i}^{-1} \begin{bmatrix} \boldsymbol{v}_i \\ \omega_i \end{bmatrix} - \boldsymbol{J}_{\text{leg}_i}^{-1} \begin{bmatrix} \boldsymbol{E} & -\widehat{\boldsymbol{r}}_{B\to i} \\ \boldsymbol{0} & \boldsymbol{E} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_B \\ \omega_B \end{bmatrix}. \tag{23}$$

Note that the singularity of the leg Jacobian $\boldsymbol{J}_{\text{leg}_i}$ must be avoided. Singularities occur when, for example, the knee joint is fully extended.

Momentum Equations

Consider the linear momentum $\mathcal{P}$ and the angular momentum $\mathcal{L}$ around the CoM as the dynamic state of the entire robot system. These quantities are calculated as

$$\begin{bmatrix} \mathcal{P} \\ \mathcal{L} \end{bmatrix} = \begin{bmatrix} M\boldsymbol{E} & -M\,\widehat{\boldsymbol{r}}_{B\to c} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_B \\ \omega_B \end{bmatrix} + \sum_{i=1}^{2} \begin{bmatrix} \boldsymbol{M}_i \\ \boldsymbol{H}_i \end{bmatrix} \dot{\boldsymbol{q}}_{\text{leg}_i}\,, \tag{24}$$

where M is the total robot mass, $\boldsymbol{I}$ is the total inertia tensor around the CoM, and $\boldsymbol{M}_i$ and $\boldsymbol{H}_i$ are the inertia matrices, which indicate how the joint speeds affect the total momentum. The vector $\boldsymbol{r}_{B\to c}$ points from the body to the CoM

$$\boldsymbol{r}_{B\to c} := \boldsymbol{c} - \boldsymbol{p}_B\,.$$

A momentum equation results by substituting (23) into (24)

$$\begin{bmatrix} \mathcal{P} \\ \mathcal{L} \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}_B^* \\ \boldsymbol{H}_B^* \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_B \\ \omega_B \end{bmatrix} + \sum_{i=1}^{2} \begin{bmatrix} \boldsymbol{M}_i^* \\ \boldsymbol{H}_i^* \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_i \\ \omega_i \end{bmatrix} \tag{25}$$

where $\boldsymbol{M}_B^*$, $\boldsymbol{H}_B^*$, $\boldsymbol{M}_i^*$, and $\boldsymbol{H}_i^*$ are modified inertia matrices.

Relating Momentum and Joint Speed

To realize a robot motion that is equivalent to the running pattern of a simple inverted pendulum, the desired linear and angular momenta are specified as

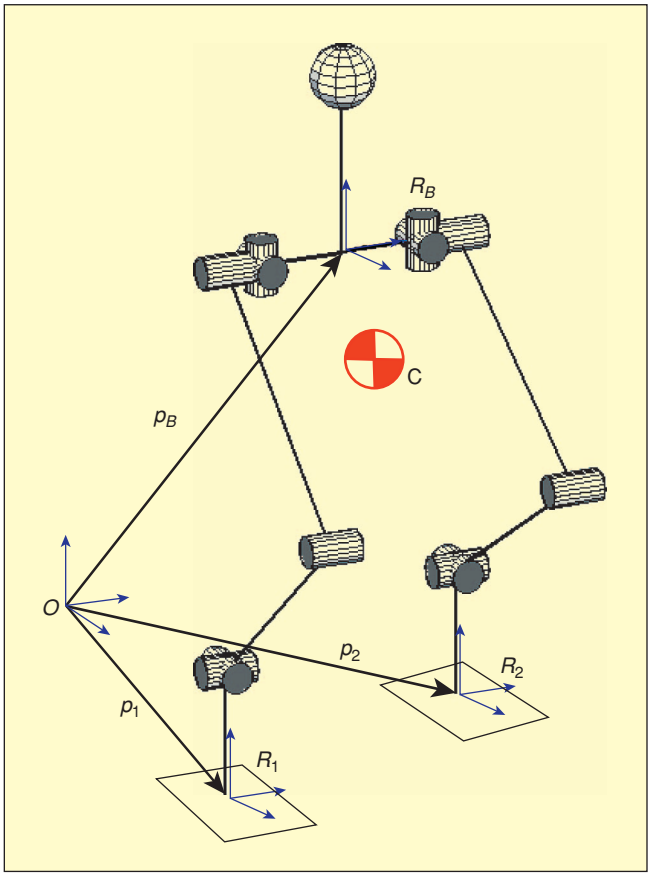

Figure 6. *Reference frames for running pattern generation.*

Studying extreme dynamic tasks will give insights into how to improve the robot's mechanical design.

$$\begin{bmatrix} \mathcal{P}^d \\ \mathcal{L}^d \end{bmatrix} = \begin{bmatrix} M\dot{\boldsymbol{c}}^d \\ \mathbf{0} \end{bmatrix}, \tag{26}$$

where $\dot{\boldsymbol{c}}^d$ is the CoM speed for running. We assume the foot speeds $[\boldsymbol{v}_i^d \ \omega_i^d]$ have already been determined as the running pattern of the pendulum. By using the momentum equation (24), the body velocity that may realize the desired motion may be calculated as

$$\begin{bmatrix} \boldsymbol{v}_B^d \\ \omega_B^d \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}_B^* \\ \boldsymbol{H}_B^* \end{bmatrix}^{-1} \left(\begin{bmatrix} \mathcal{P}^d \\ \mathcal{L}^d \end{bmatrix} - \sum_{i=1}^{2} \begin{bmatrix} \boldsymbol{M}_i^* \\ \boldsymbol{H}_i^* \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_i^d \\ \omega_i^d \end{bmatrix} \right). \tag{27}$$

Finally, the joint speeds may be obtained by substituting $[\boldsymbol{v}_i^d \ \omega_i^d]$ $(i = 1, 2, B)$ into (23). The joint trajectories can be obtained by numerical integration from an appropriate initial condition.

However, the robot body performs a complicated twisting in the running pattern when (27) is used. Although the body twist occurs to realize the strict zero angular momentum, it requires excessive joint speeds and torques. To mitigate this problem, we decided to allow nonzero yaw angular momentum since its effect can be canceled by the floor friction during the support phase. We also decided to accept nonzero roll angular momentum since the leg motion around roll axis is relatively small. The zero pitch angular momentum was kept since the swing leg motion creates considerable angular momentum around this axis. Consequently, a selection matrix was introduced to neglect the angular momentum demand for the roll and yaw axes

$$\boldsymbol{S} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{28}$$

Using this matrix, the calculation of the body speed is modified as

$$\begin{aligned} \begin{bmatrix} \boldsymbol{v}_B^d \\ \omega_B^d \end{bmatrix} &= \left(\boldsymbol{S} \begin{bmatrix} \boldsymbol{M}_B^* \\ \boldsymbol{H}_B^* \end{bmatrix} \right)^{\dagger} \mathcal{X} \\ \mathcal{X} &:= \boldsymbol{S} \left(\begin{bmatrix} \mathcal{P}^d \\ \mathcal{L}^d \end{bmatrix} - \sum_{i=1}^{2} \begin{bmatrix} \boldsymbol{M}_i^* \\ \boldsymbol{H}_i^* \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_i^d \\ \omega_i^d \end{bmatrix} \right), \end{aligned} \tag{29}$$

where † denotes the pseudoinverse operation. In this case, further modification may be introduced by using the null space of the pseudoinverse [18].

In practice, instead of (26), the target linear momentum was specified as

$$\mathcal{P}^d = K_p(\boldsymbol{c}^d - \boldsymbol{c}) + M\dot{\boldsymbol{c}}^d, \tag{30}$$

where K_p is a feedback gain and $\boldsymbol{c}$ is the CoM of the generated pattern. The first term of the right side of (30) modifies the momentum so that the generated pattern follows the reference CoM against the error of numerical calculation and the effect of time derivative of Jacobian.

Figure 7 shows a resulting running pattern that was generated using this method. Note the cyclic pitching of the body posture to realize the zero angular momentum around the CoM. Using this motion, the robot behaves as a point mass in the direction of running. On the other hand, since the roll and yaw angular momenta were not specified to be zero, these momenta will appear as disturbances for the rolling and the yawing motions. These motions must be compensated by a real-time stabilization, which is explained in the next section.

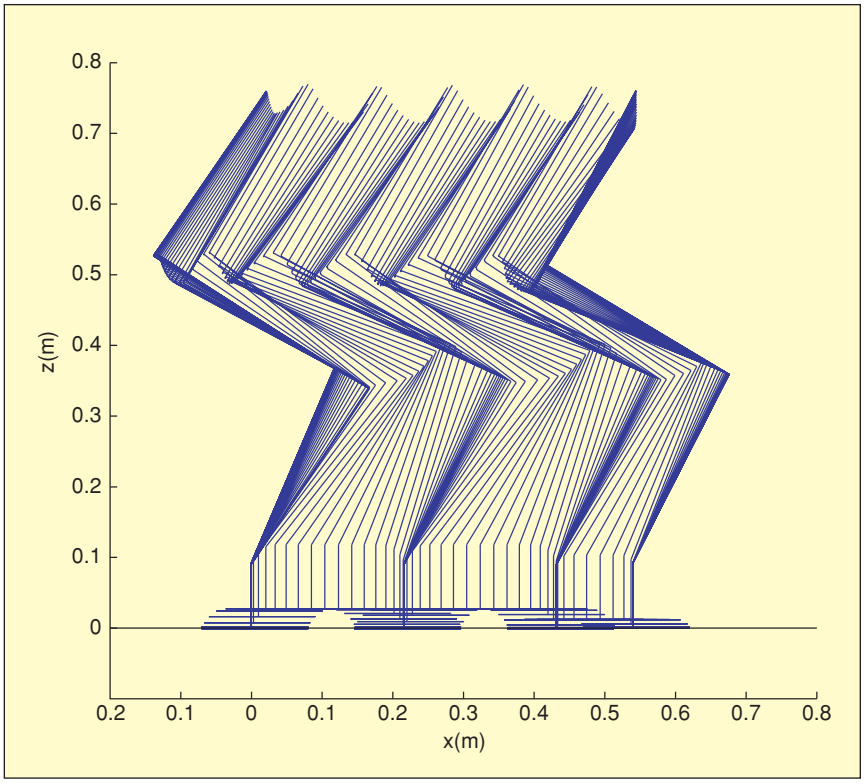

Figure 7. *Stick picture of the running pattern (left leg is omitted).*

Running Stabilizer

The running stabilizer modifies a running pattern so that the robot can continue running despite modeling errors and disturbances. The stabilizer takes the sensor information and the running pattern and updates the servo reference at a rate of 200 Hz. Table 3 lists the data of the running pattern that are required by the stabilizer. These data for an entire running period must be prepared in advance of a running experiment.

Architecture of the Stabilizer

Figure 8 shows the architecture of the stabilizer. The block labeled "F.K." in the left of the figure takes the running pattern and calculates the reference configuration of the robot as

$$\boldsymbol{R}_B^d = \boldsymbol{R}_{rpy}(\phi^d, \theta^d, \psi^d), \tag{31}$$

$$\left[\boldsymbol{p}_1^d, \boldsymbol{R}_1^d, \boldsymbol{p}_2^d, \boldsymbol{R}_2^d\right] = FK(\boldsymbol{p}_B^{in}, \boldsymbol{R}_B^d, \boldsymbol{q}^d), \tag{32}$$

where $\boldsymbol{R}_B^d$ is the reference body posture and $\boldsymbol{p}_B^{in}$ is the nominal body position. The function $\boldsymbol{R}_{rpy}(\cdot)$ calculates a rotation matrix corresponding to a given roll-pitch-yaw angles, and the function $FK()$ calculates the forward kinematics of the robot.

Since HRP-2LR cannot sense its absolute body position, the nominal body position $\boldsymbol{p}_B^{in}$ is set to the origin

$$\boldsymbol{p}_B^{in} = \boldsymbol{O}.$$

The other block labeled "F.K." in the bottom of Figure 8 takes sensor data and calculates the real configuration of the robot as

$$\boldsymbol{R}_B = \boldsymbol{R}_{rpy}(\phi, \theta, \psi), \tag{33}$$

$$[\boldsymbol{p}_1, \boldsymbol{R}_1, \boldsymbol{p}_2, \boldsymbol{R}_2] = FK\left(\boldsymbol{p}_B^{in}, \boldsymbol{R}_B, \boldsymbol{q}\right), \tag{34}$$

where $\boldsymbol{R}_B$ is the current body posture. $\boldsymbol{R}_B$ is calculated from roll-pitch-yaw angles (ϕ, θ, ψ) estimated by the Kalman filter, which has the acceleration sensor and the gyro sensors as its inputs. The vector $\boldsymbol{q}$ represents the current joint angles measured by the encoders.

The block labeled "Stabilizing Algorithms" in the center of Figure 8 takes the reference and the real robot configurations, the running pattern, and the sensor data as its inputs. This block calculates the robot configuration for the servo reference $[\boldsymbol{p}_i^s, \boldsymbol{R}_i^s]$ $(i = 1, 2, B)$ so that the robot can successfully perform the given running pattern. The stabilizing algorithms contain the following control components:

- linear inverted pendulum (LIPM) control
- body posture control
- two DOF torque controller
- foot torque control
- foot vertical force control.

In the following subsections, the first two components are explained. The remaining components are explained in [19].

The block labeled "I.K" in the right of Figure 8 takes the outputs of stabilizing algorithms and converts them into joint angles using inverse kinematics (IK) as

$$\boldsymbol{q}^s = IK\left(\boldsymbol{p}_B^s, \boldsymbol{R}_B^s, \boldsymbol{p}_1^s, \boldsymbol{R}_1^s, \boldsymbol{p}_2^s, \boldsymbol{R}_2^s\right), \tag{35}$$

where $\boldsymbol{q}^s$ is a servo reference for the proportional derivative (PD) joint servo controller of HRP-2LR. The function $IK()$ is an IK of the robot.

Linear Inverted Pendulum Control

Figure 9 shows the reference and the real robot configurations calculated by (31), (32), (33), and (34). To represent the inclination of the entire robot, define the vector $\boldsymbol{r}$ that points from the midpoint of the two feet to the body

$$\boldsymbol{r} := \boldsymbol{p}_B^{\text{in}} - (\boldsymbol{p}_1 + \boldsymbol{p}_2)/2 \tag{36}$$

$$\boldsymbol{r}^d := \boldsymbol{p}_B^{\text{in}} - \left(\boldsymbol{p}_1^d + \boldsymbol{p}_2^d\right)/2. \tag{37}$$

Table 3. Data of the Running Pattern.

q^d	Joint angles $q^d := [q_{leg_1}^T q_{leg_2}^T]^T$
ϕ^d, θ^d, ψ^d	Body posture (R_B^d in roll-pitch-yaw)
ω_B^d	Body angular velocity
f_i^d, τ_i^d	Foot force and moment
phase	Running phase (support leg) (Right, Left, Double, Flight)

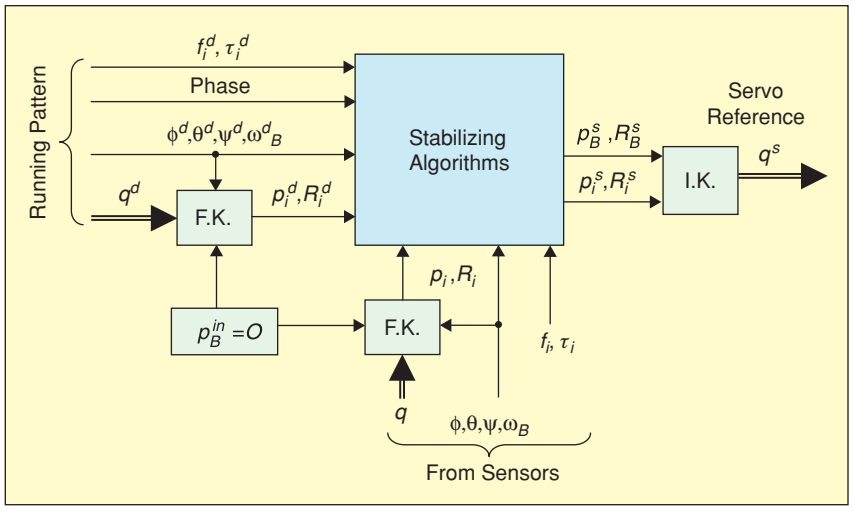

Figure 8. *Architecture of the stabilizer. F.K.: Forward Kinematics, I.K.: Inverse Kinematics.*

To realize a steady hopping motion, the average floor reaction force must coincide with the force due to gravity.

Define the error between $\boldsymbol{r}$ and $\boldsymbol{r}^d$ as

$$\begin{aligned}\Delta \boldsymbol{r} &= \boldsymbol{r} - \boldsymbol{r}^d \\ &:= [\Delta x,\ \Delta y,\ \Delta z]^T . \end{aligned} \tag{38}$$

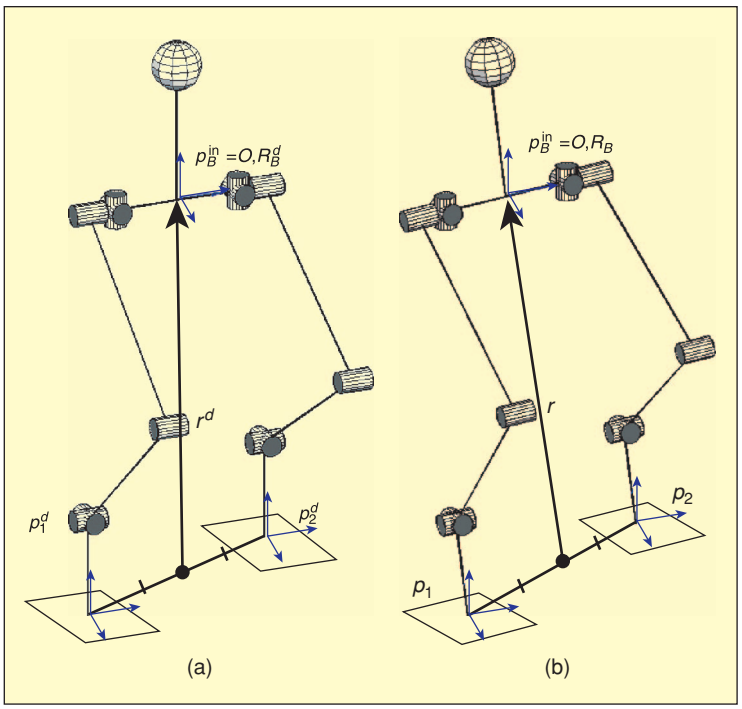

Figure 9. *Robot representations in the stabilizer.*

Both $\boldsymbol{r}$ and $\boldsymbol{r}^d$ point approximately at the CoM, and the dynamics associated with $\Delta \boldsymbol{r}$ are approximately linear. Therefore, it is assumed that the error vector $\Delta \boldsymbol{r}$ behaves as a 3-D LIPM when the robot is on the ground [20]

$$\Delta \ddot{x} = \frac{g}{z_c}\Delta x + \frac{1}{M z_c}\tau_y \tag{39}$$

$$\Delta \ddot{y} = \frac{g}{z_c}\Delta y - \frac{1}{M z_c}\tau_x , \tag{40}$$

where g is gravity acceleration, z_c is the nominal height of the CoM, and M is the total mass. τ_y and τ_x are the pitch and roll torque, which act from the floor to the robot.

The dynamics of (39) and (40) allow the design of a simple linear feedback law

$$\tau_y = -k_x \Delta x - d_x \Delta \dot{x} , \tag{41}$$

$$\tau_x = k_y \Delta y + d_y \Delta \dot{y} . \tag{42}$$

These torques are converted into the foot displacement by a torque controller. This becomes possible by the compliant elements in the HRP-2LR's feet that guarantee the linear torque-displacement relationship. The torque controller is described in [19].

Body Posture Control

The body posture controller calculates supplemental body rotation so that the actual posture (ϕ, θ) becomes (ϕ^d, θ^d) as specified by the running pattern. In the current implementation, the yaw rotation ψ is neglected. The control law is

$$\Delta \ddot{\phi} = k_B(\phi^d - \phi) + d_B\left(\omega_{Bx}^d - \omega_{Bx}\right) , \tag{43}$$

$$\Delta \ddot{\theta} = k_B(\theta^d - \theta) + d_B\left(\omega_{By}^d - \omega_{By}\right) , \tag{44}$$

where $\Delta\phi$ and $\Delta\theta$ are the supplemental body rotation, k_B and d_B are the feedback gains, ω_{Bx} and ω_{By}

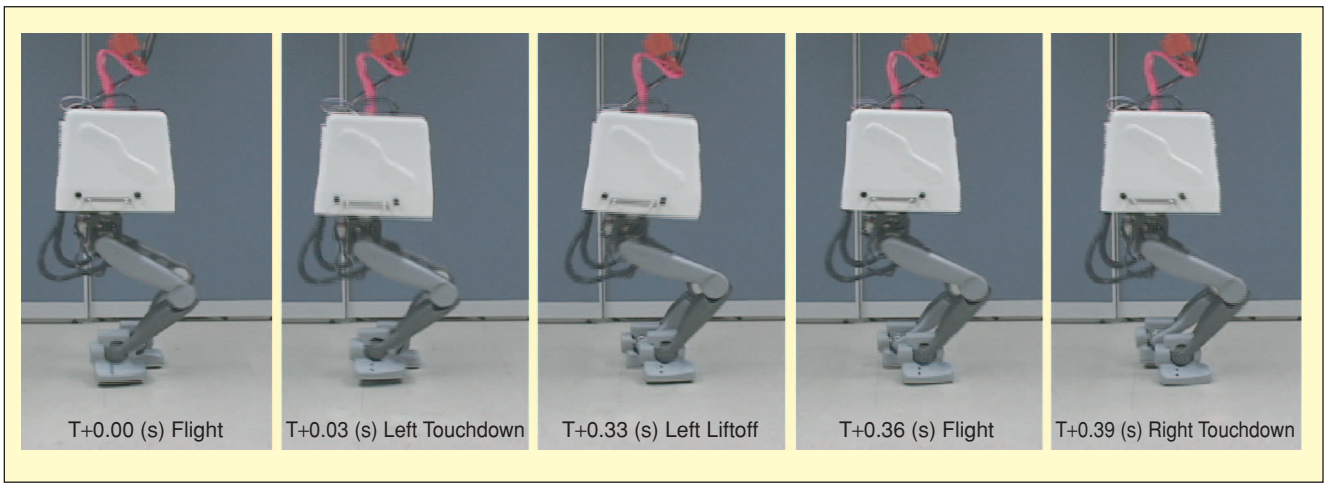

Figure 10. *Running experiment of HRP-2LR. The robot is running from left to right.*

are angular velocities measured by gyros, and ω^d_{Bx} and ω^d_{By} are the corresponding references.

By integrating (43) and (44) twice, the additional body rotation angle $\Delta\phi$ and $\Delta\theta$ are obtained, which allows the reference body orientation to be determined as

$$\boldsymbol{R}^s_B = \Delta\boldsymbol{R}_B\boldsymbol{R}^d_B \tag{45}$$

$$\Delta\boldsymbol{R}_B := \boldsymbol{R}_{rpy}(\Delta\phi, \Delta\theta, 0)\,. \tag{46}$$

To let the body rotate around its CoM, the origin of the base frame is also changed

$$\boldsymbol{p}^s_B = \boldsymbol{p}^{in}_B + \boldsymbol{R}^d_B\bar{\boldsymbol{c}} - \boldsymbol{R}^s_B\bar{\boldsymbol{c}}, \tag{47}$$

where $\bar{\boldsymbol{c}}$ is the CoM of the body represented with respect to the base frame.

The calculated body configuration $(\boldsymbol{p}^s_B, \boldsymbol{R}^s_B)$ is translated into joint angles by $IK()$. The result is that the robot adjusts its body posture by using all of the leg joints.

Experiments

A running pattern was designed with a support duration of 0.3 s, flight duration of 0.06 s, and speed of 0.25 m/s (0.9 km/h). By applying the stabilizer, HRP-2LR could successfully run. Figure 10 shows a running experiment of HRP-2LR. The flight phase is at time T+0.36 s. Due to slips between the robot's sole and the ground, the actual traveled distance was shorter than planned by 68%. Thus, the running speed of the robot was 0.16 m/s (0.58 km/h).

Data from a typical running experiment are plotted in Figures 11 and 12. Figure 11(a) shows the errors in the body posture control described previously. The maximum absolute errors are 5° in pitch (thin line) and 3° in roll (bold line).

Figure 11(b) shows the errors in the LIPM control described previously. The maximum errors are 3 cm in both directions.

Although these errors are not small, the robot could run with good stability and repeatability. The longest running pattern we tested had 21 steps with designed step length of 0.072 m. This pattern was tested several times, and it was confirmed that our stabilizer works well.

Figure 12 shows the vertical reaction force during the running experiment. A flight phase exists when both of the vertical forces are zero. At each landing, the robot suffers huge impacts of approximately 1,000 N, which is more than three times the robot's weight. The vertical dotted lines indicate the phase transitions specified by the running pattern. Comparing with the force data, we can be see that the actual running motion has a delay of 20–30 ms from the running pattern.

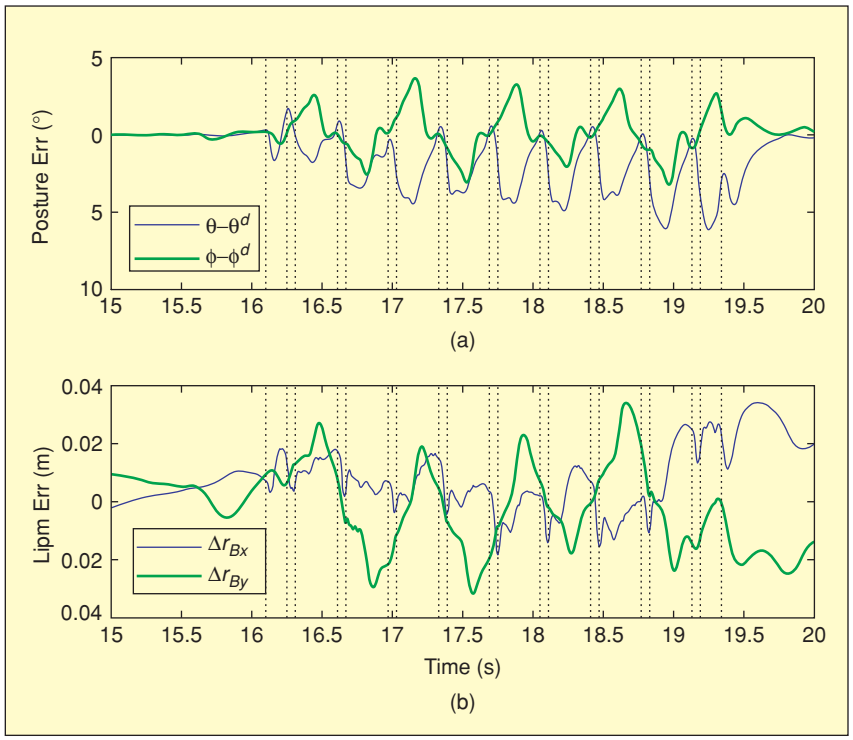

Figure 11. *Error in posture and horizontal displacement.*

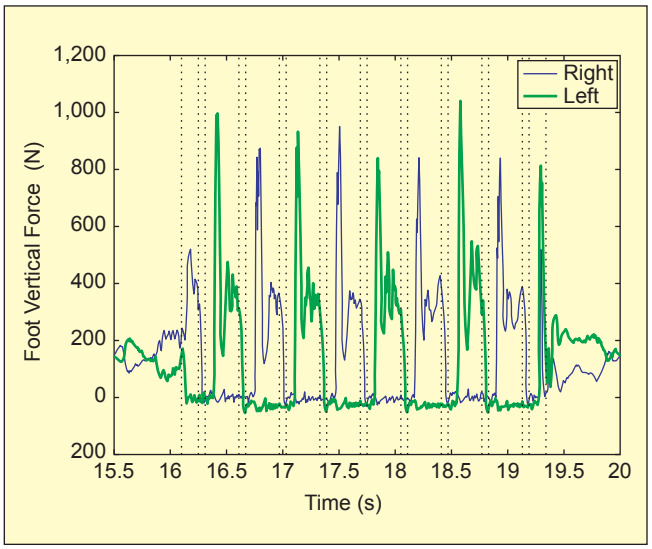

Figure 12. *Vertical force at running.*

Table 4. Running pattern and result.

Planned	Support time T_s:	0.3 s
	Flight time T_f:	0.06 s
	Step length s_d:	0.09 m
	Traveled distance:	0.81 m
	Average speed:	0.25 m/s
	Maximum foot height:	0.025 m
	Steps:	9 (step)
Result	Traveled distance:	0.55 m
	Average speed:	0.16 m/s

Conclusions

In this article, a running humanoid robot, HRP-2LR, was presented along with its running pattern generation, its controller, and the experimental result. Applying the proposed controller, HRP-2LR could successfully run with average speed of 0.16 m/s, repeating flight phases of 0.06 s and support phases of 0.3 s. Future work consists of the realization of faster running and online running pattern generation.

Acknowledgments

The authors would like to thank the employees of Kawada Industries, Inc., especially Jiro Sakurai, Toshikazu Kawasaki, and Takakatsu Isozumi for their excellent technical support. They also thank Fumio Kanehiro, Kiyoshi Fujiwara, Kensuke Harada, Hajime Saito, Mitsuharu Morisawa, and Shinnichiro Nakaoka of Humanoid Research Group, AIST for their helpful advice.

Keywords

Biped robot, running control, ZMP.

References

[1] M. Gienger, F. Löffler, and F. Pfeiffer, "Toward the design of a biped jogging robot," in *Proc. 2001 ICRA*, pp. 4140–4145.

[2] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The development of Honda humanoid robot," in *Proc. 1998 ICRA*, pp. 1321–1326.

[3] H. Inoue, S. Tachi, Y. Nakamura, K. Hirai, S. Hirai, K. Tanie, K. Yokoi, and H. Hirukawa, "Overview of humanoid robotics project of METI," in *Proc. Int. Symp. Robotics*, 2001, pp. 1478–1482.

[4] K. Nishiwaki, T. Sugihara, S. Kagami, F. Kanehiro, M. Inaba, and H. Inoue, "Design and development of research platform for perception-action integration in humanoid robot: H6," in *Proc. Int. Conf. Intelligent Robots Systems*, 2000, pp. 1559–1564.

[5] J. Yamaguchi, E. Soga, S. Inoue, and A. Takanishi, "Development of a bipedal humanoid robot—Control method of whole body cooperative dynamic biped walking," in *Proc. 1999 ICRA*, pp. 368–374.

[6] J. Kim and J. Oh, "Walking control of the humanoid platform KHR-1 based on Torque feedback control," in *Proc. 2004 ICRA*, pp. 623–628.

[7] M. Raibert, *Legged Robots that Balance*. Cambridge, MA: MIT Press, 1986.

[8] R.R. Playter and M.H. Raibert, "Control of a biped somersault in 3D," in *Proc. IFToMM-jc Int. Symp. Theory Machines Mechanisms,* Nagoya, Japan, 1992, pp. 669–674.

[9] J.K. Hodgins, "Three-dimensional human running," in *Proc. 1996 ICRA*, pp. 3271–3277.

[10] M. Ahmadi and M. Buehler, "The ARL monopod II running robot: control and energetics," in *Proc. 1999 ICRA*, pp. 1689–1694.

[11] K. Nagasaka, Y. Kuroki, S. Suzuki, Y. Itoh, and J. Yamaguchi, "Integrated motion control for walking, jumping and running on a small bipedal entertainment robot," in *Proc. 2004 ICRA*, pp. 3189–3194.

[12] C. Chevallereau, E.R. Westervelt, and J.W. Grizzle, "Asymptotically stable running for a five-link, four-actuator, planar bipedal robot," *Int. J. Robot. Res.,* vol. 24, no. 6, pp. 431–464, 2005.

[13] K. Kaneko, S. Kajita, F. Kanehiro, K. Yokoi, K. Fujiwara, H. Hirukawa, T. Kawasaki, M. Hirata, and T. Isozumi, "Design of advanced leg module for humanoid robotics project of METI," in *Proc. 2002 ICRA*, pp. 38–45.

[14] M. Vukobratović and J. Stepanenko, "On the stability of anthropomorphic systems," *Math. Biosci.*, vol. 15, pp. 1–37, 1972.

[15] S. Kagami, T. Kitagawa, K. Nishiwaki, T. Sugihara, M. Inaba, and H. Inoue, "A fast dynamically equilibrated walking trajectory generation method of humanoid robot," *Autonomous Robots*, vol. 12, no. 1, pp. 71–82, 2002.

[16] S.A. Teukolskey and W.H. Press, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge, U.K.: Cambridge Univ. Press, 1993.

[17] T. Nagasaki, S. Kajita, K. Kaneko, K. Yoki, and K. Tanie, "A running experiment of humanoid biped," in *Proc. IROS2004*, 2004, pp. 136–141.

[18] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Resolved momentum control: Humanoid motion planning based on the linear and angular momentum," in *Proc. 2003 IROS*, pp. 1644–1650.

[19] S. Kajita, T. Nagasaki, K. Kaneko, K. Yokoi, and K. Tanie, "A running controller of humanoid biped HRP-2LR," in *Proc. 2005 ICRA*, pp. 618–624.

[20] S. Kajita, O. Matsumoto, and M. Saigo, "Real-time 3D walking pattern generation for a biped robot with telescopic legs," in *Proc. 2001 ICRA*, pp. 2299–2306.

Shuuji Kajita received the M.E. and Dr.E. degrees in control engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 1985 and 1996, respectively. In 1985, he joined the Mechanical Engineering Laboratory, National Institute of Advanced Industrial Science and Technology (AIST), Ministry of International Trade and Industry. During 1996-1997, he was a visiting researcher at the California Institute of Technology. Since 2007, he has been the leader of the Humanoid Robotics Group of Intelligent Systems Research Institute, AIST, Japan. His research interests include robotics and control theory. He is a member of the IEEE Robotics and Automation Society.

Takashi Nagasaki received the M.E. degree in engineering from the University of Tsukuba, Japan, in 2002. Since 2006, he is with Human Resocia Co., Ltd. His research interests include robotics and control engineering. He is a member of the Robotics Society of Japan.

Kenji Kaneko received the B.E., M.E., and Ph.D. degrees from Keio University, Tokyo Japan, in 1988, 1990, and 1997, respectively, all in electrical engineering. In 1990, he joined the Mechanical Engineering Laboratory, National Institute of Advance Industrial Science and Technology (AIST), Ministry of International Trade and Industry, Tsukuba, Japan, as a researcher. He is currently a senior researcher at the Humanoid Research Group of Intelligent systems research Institute, AIST, Tsukuba. His current research interests include humanoid system integration, motion control, and macro-micro teleoperation. He is a member of the Robotics Society of Japan.

Hirohisa Hirukawa received the Ph.D. degree from Kobe University, Kobe, Japan, in 1987, after which he joined the Electrotechnical Laboratory, National Institute of Advance Industrial Science and Technology (AIST), Tsukuba, Japan, Ministry of Economy, Trade and Industry. From 1994 to 1995, he was a visiting scholar at Stanford University, California. Since 2004, he has been the Deputy Director of the Intelligent Systems Research Institute of AIST. His current research interests include humanoid robotics and motion planning.

Address for Correspondence: Shuuji Kajita, 1-1-1, Umezono, Tsukuba, Ibaraki 305-8568, Japan. Phone: +81 29 861 7280. Fax: +81 29 861 5444. E-mail: s.kajita@aist.go.jp.