

Aerial Posture Control for 3D Biped Running Using Compensator around Yaw Axis

Sang-Ho Hyon
Dept. of Mechatronics & Precision Eng.
Tohoku University
Aramaki-Aza-Aoba 01, Sendai 980-8579
Japan
(sangho@ieee.org)

Takashi Emura
Dept. of Mechatronics & Precision Eng.
Tohoku University
Aramaki-Aza-Aoba 01, Sendai 980-8579
Japan
(emura@emura.mech.tohoku.ac.jp)

Abstract— A new 3D biped prototype with small DOF, *SKIPPER*, was developed. As an important component of a running controller for this robot, an aerial posture controller is presented. This is the extended version of planar running controller, which was previously developed for a passive one-legged hopping robot, and plays important roll for orbital stabilization at the lowest control layer. The controller stabilizes three output functions defined as lateral leveling, holonomic constraint of pitch dynamics, and linearization of swing leg dynamics. Simulation results show that the robot achieves desired posture control for wide sets of initial conditions, naturally utilizing its compensator around yaw axis. In this paper, after describing the 3D biped model and its basic running control strategy, the details of the aerial posture controller are presented, as well as the simulation results.

I. INTRODUCTION

After Raibert's excellent works [1], running robots have been widely studied both experimentally [2][3][4][5][6] and theoretically [7][8][9]. On the other hand, recently there are many studies on biped humanoid robots with the aim of practical application. Many successful 3D biped walking motions have been realized (for example, Honda Humanoid Robot [10]). Therefore, enhancing the mobility of them is important target to be reached.

Running control of autonomous biped humanoid robot is included in such targets. Energy-efficient running control is especially crucial for autonomous humanoid robots because it directly extends the operation time.

In this context, there have been some fundamental researches on energy-efficient running control. Tompson and Raibert showed that spring-driven one-legged hopping robot can hop without any inputs, provided if the initial conditions were appropriately chosen [11]. Ahmadi and Buehler applied Raibert's algorithm to this robot and realized energy-efficient hopping in simulation and experiment [12][13]. François and Samson derived the elegant controller based on the general control method used in nonlinear oscillatory system [14].

Motivated from their works, Hyon and Emura proposed alternative controller based on its energy analysis [15]. The distinctive feature of this controller is

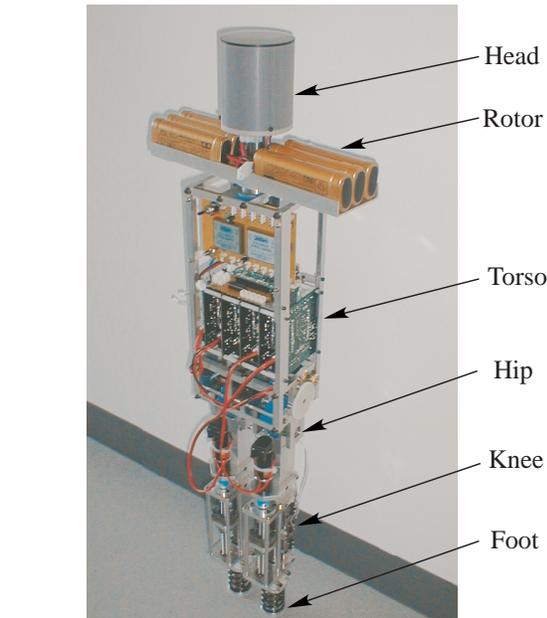


Fig. 1. 3D biped robot – *SKIPPER*

the dead-beat of both leg angle and angular velocity at given flight time. This enables the mechanical energy at touchdown to be non-dissipative. As a result, interesting quasi-periodic gaits, which can be seen in some Hamiltonian system, were found, and orbital stabilization was achieved.

This paper proposes an aerial posture control, which makes it possible to apply the planar touchdown control in the literature [15] to a 3D biped robot. The paper is organized as follows: Section II describes the model of 3D biped running robot, whereat, the hardware overview and the equation of motion of simplified model are given. Section III includes four subsections. Herein, the decoupling controller, lateral leveling controller, holonomic constraint control, and dead-beat touchdown controller are presented in order. Section IV gives simulation results. Section V concludes this paper.

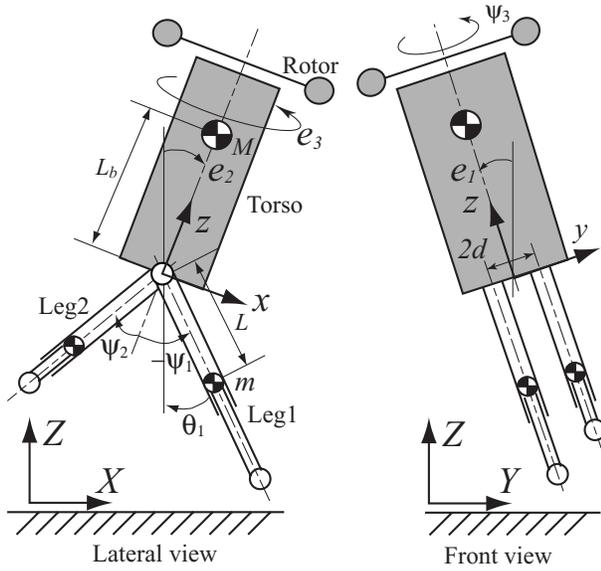


Fig. 2. Simplified model

II. MODEL OF 3D BIPED RUNNING ROBOT

A. Hardware Overview

Figure 1 shows the 3D biped prototype, named *SKIPPER*. The robot has telescopic legs swinging around pitch axis and a rotor rotating around yaw axis at the top of the torso. There is no actuator at each foot, that is, the robot stands on the ground with point contacts. The total DOF is five and it is relatively small compared to that of existing 3D biped robots. The main goal of this biped robot having a *small number of DOF* is to develop energy-efficient controller for various 3D motions including walking or running [16].

Overall height of the robot is 0.84[m] and the total weight is about 8[kg]. Five geared DC motors of 20[W] are installed on each joint. For the telescopic joint at the knee, high-lead ball screws are used to avoid self-locking. In addition, to assist the knee actuator, a coil spring is installed along the foot and parallel to the ball nut. It will enable the robot to perform both rigid and compliant walking or running. Main sensors are rotary encoders at each joint, three gyros in the head, and force sensors at the foot. To allow the robot to move three-dimensional space freely, all the control circuits and power supply are on board.

B. Simplified Model

Figure Fig. 2 shows the simplified model of *SKIPPER* described above. It is composed of four rigid bodies; torso, leg 1, leg 2, and rotor. Inertial coordinate system Σ_{XYZ} is set on the ground and local coordinate system Σ_{xyz} is set on the torso. The coordinates (e_1, e_2, e_3) rep-

resents the Roll, Pitch and Yaw angles of the torso. M includes the mass of torso, rotor and head. The foot is mass-less, and hence, has no effects on leg length. All principal axis of each rigid part are coincident with their center axis.

Table I shows the physical parameters, together with the values used in later simulations, where I_{ix}, I_{iy}, I_{iz} ($i = 0, 1, 2, 3$) are the principal inertia of each part.

C. Equations of motion at flight phase

The generalized coordinates are defined as the position of center of mass (COM) $x = (x_g, y_g, z_g)^T \in R^3$, the attitude of the torso $e = (e_1, e_2, e_3)^T \in R^3$, and joint angles $p = (\psi_1, \psi_2, \psi_3)^T \in R^3$.

The equations of motion can be derived through Lagrangian formulation.

$$\left(\begin{array}{c|c} \begin{bmatrix} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & M \end{bmatrix} & 0(3 \times 6) \\ \hline 0(6 \times 3) & \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix} \end{array} \right) \begin{pmatrix} \ddot{x} \\ \ddot{e} \\ \ddot{p} \end{pmatrix} = \begin{bmatrix} 0 \\ -Mg \\ 0 \\ H_1(e, p, \dot{e}, \dot{p}) \\ H_2(e, p, \dot{e}, \dot{p}) \end{bmatrix} + \begin{bmatrix} 0(3 \times 1) \\ 0(3 \times 1) \\ \tau \end{bmatrix}, \quad (1)$$

where $M = 2m + M_b$ is the total mass and g is gravity acceleration. $N_{11}, N_{21} = N_{12}^T$, and $N_{22} \in R^{3 \times 3}$ are the inertia matrix contains only e and p . H_1 and $H_2 \in R^{3 \times 1}$ are the nonlinear terms that contain only e, p, \dot{e} , and \dot{p} . The generalized forces are $\tau = (\tau_1, \tau_2, \tau_3)^T \in R^3$, where τ_1 and τ_2 are the hip torques and τ_3 is the torque of the rotor.

Since the COM of the robot moves along the path of trivial ballistic flight, we focus on the lower part of the Eq. (1), i.e. the dynamics about e, p , hereinafter.

D. Lateral stability and control strategy

Needless to say, lateral stability is important for 3D biped robot, even when moving forward. Clearly, the

TABLE I
ROBOT PARAMETERS

Variables	Unit	Values
L, L_b, d	m	0.1, 0.224, 0.04
M_b, m	kg	6.11, 1.126
I_{0x}, I_{0y}, I_{0z}	kgm ²	0.06, 0.06, 0.02
I_{1x}, I_{1y}, I_{1z}	kgm ²	0.004, 0.004, 0.001
I_{2x}, I_{2y}, I_{2z}	kgm ²	0.004, 0.004, 0.001
I_{3x}, I_{3y}, I_{3z}	kgm ²	0.035, 0.035, 0.07

lateral moment applied to the robot, which comes from the ground reaction force, is larger in running than in walking. However, controlling lateral motion is difficult for our robot because it has no actuator of Roll-axis [17].

There are two possible solutions to achieve lateral stability. One is controlling the ground reaction forces appropriately using leg thrust force of each leg during stance phase. However, this method may deteriorate the stability of forward (pitch) motion. Another solution is to keep the lateral level of the robot so as to *direct the bearing to the direction of falling during flight phase*. Rotor (Yaw-axis compensator) can be used for this purpose. The rotor can be also used at stance phase to compensate Yaw-axis moment arising from the leg swinging motion. These two usage of the rotor will achieve both the lateral stability and the bearing tracking. In this paper, we will focus on controlling at flight phase. It should be noted that we are not intend to use it explicitly, e.g. PD feedback. Managing of the rotor is entirely left to the controller.

If the leveling control works well, we only have to control a swinging leg appropriately using the controller for planar model proposed in [15]. This is our first control strategy for 3D biped running robot.

III. CONTROLLER

This section gives the details of the controller. First, the decoupling control is given in Section III-A. Next, lateral leveling control, holonomic constraint control, and touch down control are described in Section III-B, III-D and III-C respectively. And finally, the control input is calculated in Section III-E.

A. Decoupling control

The equation of motion Eq. (1) is highly nonlinear and formidable to derive posture controllers. Therefore, it is decoupled using a new control input.

Rewriting the concerned part of Eq. (1) as:

$$N_{11}\ddot{e} + N_{12}\ddot{p} + H_1 = 0 \quad (2)$$

$$N_{21}\ddot{e} + N_{22}\ddot{p} + H_2 = \tau \quad (3)$$

, and deleting \ddot{p} from Eq. (2) and Eq. (3) gives

$$\ddot{e} = \frac{N_{22}N_{12}^{-1}H_1 - H_2 + \tau}{(N_{21} - N_{22}N_{12}^{-1}N_{11})} \quad (4)$$

Defining the right hand side as a new control input $v \in R^3$ yields the decoupled control system:

$$\ddot{e} = v \quad (5)$$

Some remarks are in order.

Remark 1) Having obtained Eq. (5), it is easy to control e directly to desired values using e.g. PD feedback law. But we cannot tell what happens on joint

angles p and input torques τ . We have to choose carefully the variables to be controlled because the system is under-actuated.

Remark 2) The most of the interests of under-actuated system is to control more number of variables than that of inputs. However, different from space robots, given time (flight time) to finish control task for running robots is quite limited. Thus, the authors think it unrealistic to use nonholonomic control during flight phase.

B. Lateral leveling control

Turning to the direction of falling means keeping the level of local y-axis (Fig. 2). Here we call it *lateral leveling control*.

This can be formulated as the Z-component of the unit vector of y-axis are zero, that is,

$$\begin{aligned} & \left[E_e \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right]_Z \\ &= \cos e_1 \sin e_2 \sin e_3 + \sin e_1 \cos e_3 = 0 \end{aligned} \quad (6)$$

where E_e is the coordinate transformation matrix from Σ_{xyz} to Σ_{XYZ} .

This can be controlled using the idea of *Input-Output Linearization* [18]. Defining output function

$$\alpha := \cos e_1 \sin e_2 \sin e_3 + \sin e_1 \cos e_3 \quad (7)$$

and control it to be zero. For output zeroing, the following equation is used.

$$\ddot{\alpha} + k_{ad}\dot{\alpha} + k_{ap}\alpha = 0 \quad (8)$$

where $k_{ad} > 0$ and $k_{ap} > 0$ are the gains. On the other hand, expanding $\ddot{\alpha}$ gives

$$\ddot{\alpha} = a_1\ddot{e}_1 + a_2\ddot{e}_2 + a_3\ddot{e}_3 + a_4 \quad (9)$$

where a_i ($i = 1, 2, 3, 4$) is the nonlinear terms that contain only e, p, \dot{e} and \dot{p} .

Therefore, the condition that \ddot{e} must satisfy to achieve the control goal Eq. (6) is written as:

$$(a_1 \ a_2 \ a_3) \begin{pmatrix} \ddot{e}_1 \\ \ddot{e}_2 \\ \ddot{e}_3 \end{pmatrix} = -(a_4 + k_{ad}\dot{\alpha} + k_{ap}\alpha) \quad (10)$$

C. Holonomic constraint control

Integrating Eq. (2) gives three first integrals, i.e. angular momentums around COM. But this is nonholonomic constraint and is not easy to deal with. As mentioned in *Remarks* above, there are no time to apply some complex attitude controllers during flight phase for running robot. Also, it has no meaning to control

directly the attitude e . Instead, the following simple target dynamics are introduced.

$$\dot{e}_2 = \dot{e}_2(0) \quad (11)$$

$$\ddot{\psi}_1 = u_1 \quad (12)$$

Equation (11) means imposing the holonomic constraint of *virtual* angular momentum around Pitch-axis. Note that it suits normal biped running gait because this constraint holds approximately, when the both legs are swung symmetrically ($\psi_1 + \psi_2 = 0$). On the other hand, Eq. (12) is to control a swinging leg arbitrarily using a new control input u_1 . These target dynamics are used in touchdown control at the next section. Note that putting these two equations together with the above leveling control, three (the same number of control inputs) control goals were set.

Below we describe how to achieve these control goal. First, for Eq. (11), simply define output function and control it to be zero, as done in the previous section. The output function in this case is defined as

$$\gamma := \dot{e}_2 - \dot{e}_2(0) \quad (13)$$

Using

$$\dot{\gamma} + k_g \gamma = 0 \quad (14)$$

where $k_g > 0$ is the gain, Eq. (13) converges to zero. Therefore, merging Eq. (13) and Eq. (14) gives the following condition.

$$\ddot{e}_2 = -k_g \gamma \quad (15)$$

Next, Eq. (12) is treated as follows. Taking the first equation from Eq. (3) gets

$$\ddot{\psi}_1 = -(c_1 \ddot{e}_1 + c_2 \ddot{e}_2 + c_3 \ddot{e}_3 + c_4) \quad (16)$$

where c_i ($i = 1, 2, 3, 4$) is the nonlinear terms that contain only e, p, \dot{e} and \dot{p} . Therefore, substituting it to Eq. (12), we obtain

$$(c_1 \ c_2 \ c_3) \begin{pmatrix} \ddot{e}_1 \\ \ddot{e}_2 \\ \ddot{e}_3 \end{pmatrix} = c_4 + u_1 \quad (17)$$

D. Touchdown control of swing leg (dead-beat)

The purpose here is to dead-beat the absolute angle of the swing leg θ_1 (Fig. 2) and its velocity $\dot{\theta}_1$ to arbitral desired values at given fixed time T_v (flight time) as in [15].

Since Eq. (12) is the second order linear ODE, we can easily dead-beat ψ_1 and $\dot{\psi}_1$, by only once-switching of the constant inputs¹. Note that $\theta_1 = \psi_1 + e_2$ and $\dot{\theta}_1 = \dot{\psi}_1 + \dot{e}_2$.

¹Different from passive one-legged hopper in [15], *SKIPPER* prototype does not have hip spring at present. Therefore, here we used the simplest dead-beat controller.

Defining new variables

$$\Phi := \begin{pmatrix} e_2 \\ \dot{e}_2 \end{pmatrix}, \quad \Theta := \begin{pmatrix} \theta_1 \\ \dot{\theta}_1 \end{pmatrix} \quad (18)$$

and discretizing Eq. (12) using the piecewise constant inputs

$$u_1 = \begin{cases} \hat{u}_1, & \text{if } 0 \leq t < T_v/2 \\ \hat{u}_2, & \text{if } T_v/2 \leq t < T_v \end{cases} \quad (19)$$

, the dead-beat control inputs can be calculated as follows:

$$\begin{pmatrix} \hat{u}_1 \\ \hat{u}_2 \end{pmatrix} = B(T_v)^{-1} \{ \Psi(T_v) - A(T_v) \Psi(0) \} \quad (20)$$

where

$$A(T_v) = \begin{pmatrix} 1 & T_v \\ 0 & 1 \end{pmatrix}, \quad (21)$$

$$B(T_v) = \begin{pmatrix} \frac{3}{8} T_v^2 & \frac{1}{8} T_v^2 \\ \frac{1}{2} T_v & \frac{1}{2} T_v \end{pmatrix} \quad (22)$$

$$\Psi(T_v) = \Theta_d - A(T_v) \Phi(0) \quad (23)$$

, and Θ_d is the arbitral desired value to be set.

E. Calculation of control input

Using Eq. (10), Eq. (15) and Eq. (17) we get

$$\begin{pmatrix} a_1 & a_2 & a_3 \\ 0 & 1 & 0 \\ c_1 & c_2 & c_3 \end{pmatrix} \begin{pmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{pmatrix} = \begin{pmatrix} a_4 + k_{ad} \dot{\alpha} + k_{ap} \alpha \\ k_g \gamma \\ c_4 + u_1 \end{pmatrix} \quad (24)$$

where u_1 is the values in Eq. (19) and Eq. (20).

From this equation the desired angular acceleration of torso \ddot{e}_d can be calculated as:

$$\ddot{e}_d = \begin{pmatrix} a_1 & a_2 & a_3 \\ 0 & 1 & 0 \\ c_1 & c_2 & c_3 \end{pmatrix}^{-1} \begin{pmatrix} a_4 + k_{ad} \dot{\alpha} + k_{ap} \alpha \\ k_g \gamma \\ c_4 + u_1 \end{pmatrix} \quad (25)$$

Substituting this to \ddot{e} in Eq. (4) yields control input τ .

IV. SIMULATION

Using the controller described above, numerical simulations were carried out. This section demonstrates one of the examples, in which, the simulation starts from lift-off and terminates at touchdown. The initial conditions were set as Table II, where $\dot{e}_1(0) = -1$ rad/s means that the robot has initial angular momentum of Roll-axis at the instant of lift-off. This means the robot will fall down to the left without control. Dead-beat time is set to $T_v = 2z_g(0)/g = 0.3$ s (flight time) and desired touchdown values of swing leg is set to $\theta_{1d} = -1$ rad and $\dot{\theta}_{1d} = 3$ rad/s.

The time evolution of each variable are depicted in Fig. 3 to Fig. 7, where the transverse axis is time [sec].

TABLE II
INITIAL CONDITIONS

Variables	Unit	Values
$x_g(0), y_g(0), z_g(0)$	m	0, 0, 0.2
$e_1(0), e_2(0), e_3(0)$	rad	0, 0.01, 0
$\psi_1(0), \psi_2(0), \psi_3(0)$	rad	0.01, 0, 0
$\dot{x}_g(0), \dot{y}_g(0), \dot{z}_g(0)$	m/s	0, 0, 1.5
$\dot{e}_1(0), \dot{e}_2(0), \dot{e}_3(0)$	rad/s	-1, 0, 0
$\dot{\psi}_1(0), \dot{\psi}_2(0), \dot{\psi}_3(0)$	rad/s	0, 0, 0

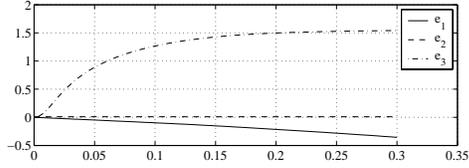


Fig. 3. Attitude of torso

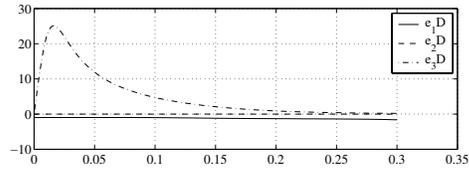


Fig. 4. Joint angles

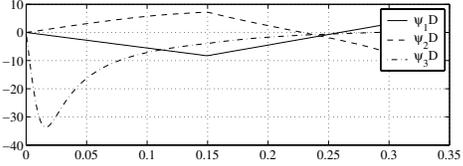
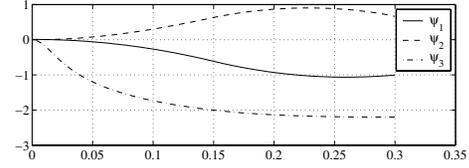


Fig. 6. Output functions

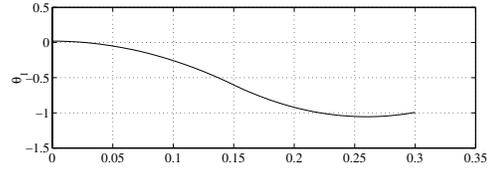


Fig. 7. Swing leg angle and velocity

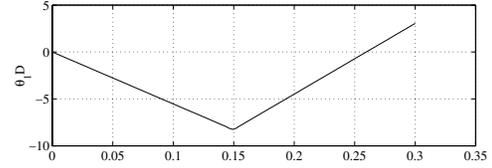


Figure 8 shows corresponding animation, where the robot turns to the left after lift-off. This animation was generated by the dynamics simulation software, named DADS (<http://www.cybernet.co.jp>). The upper graph of Fig. 3 and Fig. 4 show that the robot uses its rotor to turn to the left. The upper graph of Fig. 5 shows the robot needs relatively large Yaw-axis torque to turn and it is coincident with our intuition. The bottom graph depicts three components of angular momentum around COM. Here, only M_1 , the value about Roll-axis, is non-zero. It corresponds to the initial condition $\dot{e}_1(0) = -1$ rad/s. Figure 6 shows the convergence of output functions. The convergence rate of output functions can be changed by tuning gains. Note that γ is maintained to be zero because there are no disturbance at flight phase in this simulation. Figure 7 indicates that the angle and its velocity of swing leg reach desired values exactly at the given flight time. Consequently, the effectiveness of the controller was confirmed.

However, we found the controller has singularity at some configuration. For example, if the desired touchdown angle θ_{1d} is set to be -2 rad, hip angle ψ_1 passes through $\pi/2$, and then encounters singularity. Obviously it comes from invertibility of Eq. (25). Al-

Fig. 5. Input torques and angular momentums

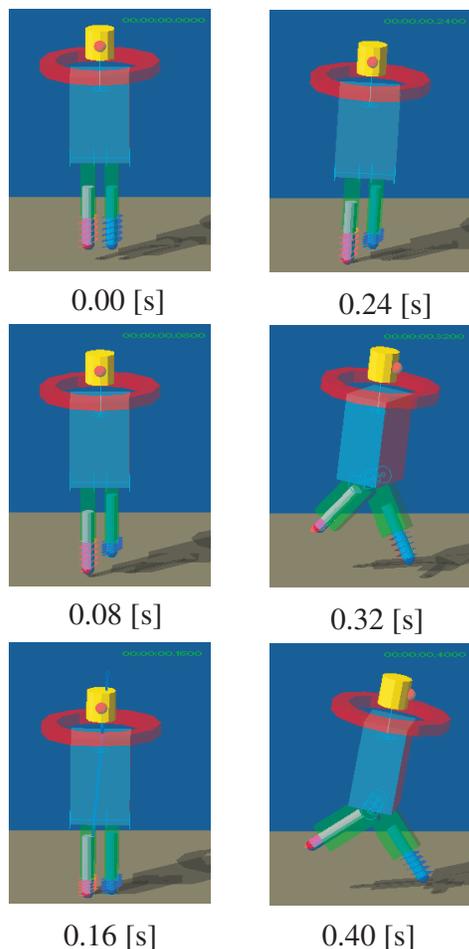


Fig. 8. Animation using DADS; starting from free fall, then turning to left (direction of falling) and touchdown

though singular configurations do not exist around normal workspace, the analysis and its avoidance is left as an important task.

V. CONCLUSION

In this paper, an aerial posture controller for our new 3D biped running robot, *SKIPPER*, was presented. Although the controller has singular configuration, simulation results show that the controller can achieve desired posture control in normal workspace.

The current work includes putting robustness to the controller and derivation of effective nonlinear controller at stance phase, which ensures the orbital stability of walking or running.

REFERENCES

[1] M.Raibert: *Legged Robots That Balance*, MIT Press, 1985.

[2] G.Zeglin, B.Brown: Control of a bow leg hopping robot, Proc. of IEEE ICRA, pp.793–798, 1998.

[3] D.Papadopoulos, M.Buehler: Stable running in a quadruped robot with compliant legs, Proc. of IEEE ICRA, pp.444–449, 2000.

[4] U.Saranli, et.al. RHex: a simple and highly mobile hexapod robot, Int. J. Robotics Research, vol.20, no.7, pp.616–631, 2001.

[5] H.Kimura, et.al.: Three-dimensional adaptive dynamic walking of a quadruped-rolling motion feedback to CPGs controlling pitch motion, Proc. of IEEE ICRA, May 2002.

[6] S.Hyon, T.Mita: Development of a Biologically Inspired Hopping Robot –“Kenken”, Proc. of IEEE ICRA, pp.3984–3991, May 2002.

[7] D.Koditschek, M.Buehler: Analysis of a simplified hopping robot, Int. J. Robotics Research, vol.10, no.6, pp.587–605, 1991.

[8] R.M’Closkey, J.Burdick: Periodic motions of a hopping robot with vertical and forward motion, Int. J. Robotics Research, vol.12, no.3, pp.197–218, 1993.

[9] W.Schwind, D.Koditschek: Control of forward velocity for a simplified planar hopping robot, Proc.of IEEE ICRA, pp.691–696, 1995.

[10] K.Hirai, et.al.: The development of Honda Humanoid Robot, Proc. of IEEE ICRA, pp.1321–1326, 1998.

[11] C.Thompson, M.Raibert: Passive dynamic running, Experimental Robotics I, eds. V.Hayward and O. Khatib., pp.74–83, Springer, 1989.

[12] M.Ahmadi, M.Buehler: Stable control of a simulated one-legged running robot with hip and leg compliance, IEEE Trans. on Robotics and Automation, vol.13, no.1, pp.96–104, 1997.

[13] M.Ahmadi, M.Buehler: The ARL Monopod II running robot: control and energetics, Proc. of IEEE ICRA, pp.1689–1694, 1999.

[14] C.François, C.Samson: A new approach to the control of the planar one-legged hopper, Int. J. Robotics Research, vol.17, no.11, pp.1150–1166, 1998.

[15] S.Hyon, T.Emura: Quasi-periodic gaits of passive one-legged hopper, Proc. of IEEE/RSJ IROS, pp.2625–2630, 2002.

[16] T.Emura, et.al: A biped hopper controlled around yaw axis by body-twisting motion, Proc. of ICOM, June 2003 (Accepted).

[17] A.Kuo: Stabilization of lateral motion in passive dynamic walking, Int. J. Robotics Research, vol.18, no.9, pp.917–930, 1999.

[18] H.Khalil: *Nonlinear Systems*, 2nd ed., Prentice-Hall, 1996.