# Analysis and Design of Jumping Robot System Using the Model Transformation Method

**Jin-Ho Suh† and Masaki Yamakita***

**Abstract** - This paper proposes the motion generation method in which the movement of the 3-links leg subsystem in constrained to slider-link and a singular posture can be easily avoided. This method is the realization of jumping control moving in a vertical direction, which mimics a cat's behavior. To consider the movement from the point of the constraint mechanical system, a robotics system for realizing the motion will change its configuration according to the position. The effectiveness of the proposed scheme is illustrated by simulation and experimental results.

**Keywords**: iterative learning control, linearization, LQ optimal control, model transformation method, nonholonomic, stochastic dynamic manipulability

## 1. Introduction

In general, cats sometimes jump toward a wall and kick it in an attempt to get to a higher-place such as a roof, thereby moving in a vertical direction, as indicated in Fig. 1. Moreover we know that the motion of a cat seems to be very skillful and efficient for jumping. In this case, a robot system to realize the motion changes its configuration according to the position, considering the movement from the point via the constraint mechanical system. Also, it has several phases on the ground and in the air for kicking with one leg. That is the system is under nonholonomic constraint due to the reservation of its momentum. Especially, the nonholonomic system has a constraint that is not integrable, and it is well known that it cannot be stabilized using smooth static state feedbacks.

The purpose of this paper is to analyze and construct the control law for the realization of a real robotic system mimicking the motion of a cat. To realize the considered robot system, the robot motion is supposed to be constrained in the $xy$-plane to simplify the problem. Moreover we propose a motion planning method in which the movement of the 3-link leg subsystem is constrained to a slider-link and a singular posture can be easily avoided. In the one leg phase with one of the nonholonomic systems, the direction for jumping is controlled using the input of the fore leg. In the air phase, one control method using state transformation and linearization is introduced to control the landing algorithm, which is also applied in order to improve the robustness of the proposed control method.

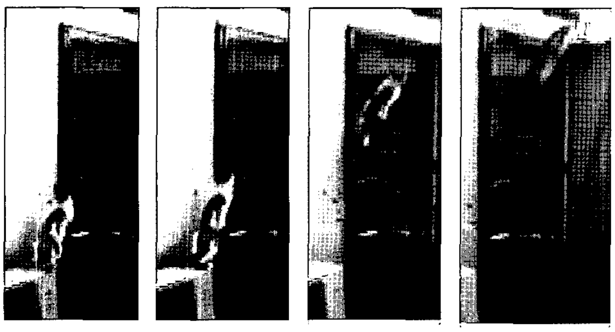


Fig. 1 Motion of the jumping cat

The organization of this paper is as follows. In Section 2, we derive the dynamic equation of a robotic system and the proposed control method in this paper is described in Section 3. In Section 4, we show the simulation and experimental results for the jumping control of a cat robot. These results illustrate the effectiveness of the proposed control algorithm in this paper. Moreover, the coupled tendon-driven system, which is supposed to be effective to concentrate the actuator power to principal joints, is also adapted. Finally, conclusions and recommendations or further works are drawn in Section 5.

## 2. System Modeling

### 2.1 System Modeling

Even though a real cat twists its body after jumping to a higher place such as a roof, we restrict the jumping motion in the sagittal plane in this paper as shown in Fig. 2, so that we can analyze and consider the fundamental control problems.

† Corresponding author: R&D, Pohang Institute of Intelligent Robotics (PIRO), Korea. (suhgang@donga.ac.kr)

* Dept. of Mechanical and Control System Engineering, Tokyo Institute of Technology, Japan. (yamakita@ac.ctrl.titech.ac.jp)

Received June 11, 2004 ; Accepted January 16, 2006

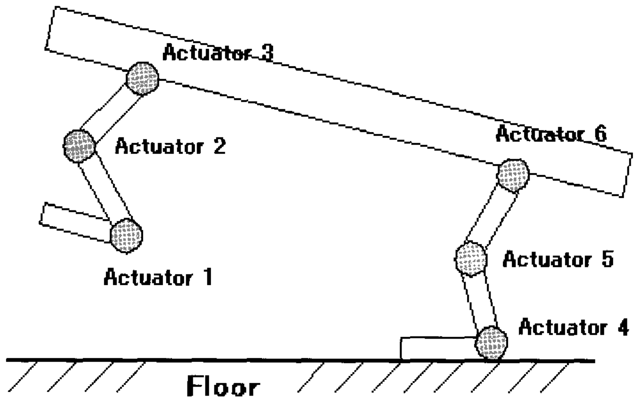



**Fig. 2** The considered robotic system

In order to realize the motion of a cat, which jumps from ground to wall and wall to roof, we consider a 7-link manipulator system as shown in Fig. 3. Then, the feature of this robot system can be summarized as follows:

(1) 6 rotational actuators at joints
(2) When the toe is contacting with the floor or wall, it is assumed that there is no slip between the toe and contact point.

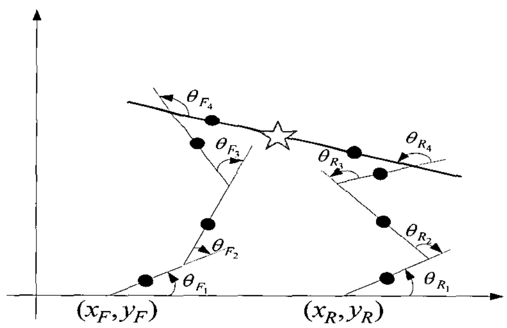



**Fig. 3** Coordinates of a jumping robot system

In order to derive the fundamental control law from the considered robot system in this paper, the whole system is divided into two 4-link serial links at points as shown in Fig. 3. Moreover a holonomic body constraint to keep the body as one straight link is also introduced.

The generalized coordinates of a jumping robot system as shown in Fig. 3 are defined as follows:

$$q = [q_F \quad q_R]^T \tag{1}$$

where

$$q_F = [x_F \; y_F \; \theta_{F_1} \; \theta_{F_2} \; \theta_{F_3} \; \theta_{F_4}]^T \tag{2}$$

$$q_R = [x_R \; y_R \; \theta_{R_1} \; \theta_{R_2} \; \theta_{R_3} \; \theta_{R_4}]^T \tag{3}$$

and $C_b(q)$ is also represented by

$$C_b(q) = \begin{bmatrix} x_F + l_{F_1}(C_{F_1} + C_{F_{12}}) + l_{F_3}C_{F_{123}} + (a_{F_4} + r_{F_4})C_{F_{1234}} \\ -x_R - l_{R_1}(C_{R_1} + C_{R_{12}}) - l_{R_3}C_{R_{123}} - (a_{R_4} + r_{R_4})C_{R_{1234}} \\ y_F + l_{F_1}(S_{F_1} + S_{F_{12}}) + l_{F_3}S_{F_{123}} + (a_{F_4} + r_{F_4})S_{F_{1234}} \\ -y_R - l_{R_1}(S_{R_1} + S_{R_{12}}) - l_{R_3}S_{R_{123}} - (a_{R_4} + r_{R_4})S_{R_{1234}} \\ \theta_{F_1} + \theta_{F_2} + \theta_{F_3} + \theta_{F_4} - \theta_{R_1} - \theta_{R_2} - \theta_{R_3} - \theta_{R_4} + \pi \end{bmatrix} \tag{4}$$

where

$$\begin{aligned} C_{F_1} &= \cos\theta_{F_1}, \quad S_{F_1} = \sin\theta_{F_1} \\ C_{F_{12}} &= \cos(\theta_{F_1} + \theta_{F_2}), \quad S_{F_{12}} = \sin(\theta_{F_1} + \theta_{F_2}) \\ C_{F_{123}} &= \cos(\theta_{F_1} + \theta_{F_2} + \theta_{F_3}), \\ S_{F_{123}} &= \sin(\theta_{F_1} + \theta_{F_2} + \theta_{F_3}) \\ C_{F_{1234}} &= \cos(\theta_{F_1} + \theta_{F_2} + \theta_{F_3} + \theta_{F_4}), \\ S_{F_{1234}} &= \sin(\theta_{F_1} + \theta_{F_2} + \theta_{F_3} + \theta_{F_4}) \end{aligned} \tag{5}$$

Though these coordinate systems are redundant and the system description becomes complex, the advantage for system coordination can be represented as follows:

(1) The 4-link dynamic equation is simpler than the 7-link one and we can use the same equation for each link.
(2) This coordinate system is very useful for judging the timing of switching the constraint on the toe, which will be mentioned later.

Moreover, for the following discussion, *Jacobian of body constraint* is defined as follows:

$$\frac{d}{dt}C_b(q) = J_b(q)\dot{q} = 0 \tag{6}$$

where

$$J_b(q) = \frac{\partial C_b(q)}{\partial q} \tag{7}$$

### 2.2 Variable Constraints

We assume that sufficient constraint force is exerted when the toe makes contact with the ground and the wall, and the holonomic constraint $C_v(q, \text{mode}) = 0$ is introduced according to the state of the system, where mode is an index that indicates the state of toe contact. For example, when only the hind toe is constrained to the floor or wall, then $C_v(q, \text{mode})$ becomes;

$$C_v(q, \text{mode}) = \begin{bmatrix} x_R - X_R(const) \\ y_R - Y_R(const) \end{bmatrix} \tag{8}$$

and we can calculate the *Jacobian* given by

$$\frac{d}{dt}C_v(q) = J_v(q)\dot{q} = 0 \tag{9}$$

where

$$J_v(q) = \frac{\partial C_v(q)}{\partial q} \tag{10}$$

In this example, the problem is how we can judge the mode. The reply of this problem lies in the understanding of the constraint force $\lambda_v = [\lambda_x\ \lambda_y]^T$ as shown in Fig. 4. In the case of the toe being on the ground, $\lambda_x$ is the horizontal constraint force and $\lambda_y$ is the vertical one, and if the $\lambda_y$ is equal to 0 and the acceleration upward is positive, the constraint should vanish, $\lambda = 0$, and the toe can move upward. When the toe is on the wall, the switching timing depends on the $\lambda_x$ and vise versa.

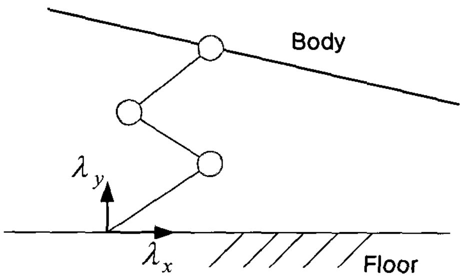


**Fig. 4** Constraint force

### 2.3 Dynamic Equation

In this section, the dynamic equation for redundant coordinate systems is considered. By ignoring the constraint, we can describe two 4-link dynamic equations for a jumping robot system as follows:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{11}$$

where

$$M = \begin{bmatrix} M_F & 0 \\ 0 & M_R \end{bmatrix},\ C = \begin{bmatrix} C_F & 0 \\ 0 & C_R \end{bmatrix},$$
$$G = \begin{bmatrix} G_F \\ G_R \end{bmatrix},\ \tau = \begin{bmatrix} \tau_F \\ \tau_R \end{bmatrix}$$

In order to change the position constraint to acceleration constraint, we will differentiate Eq. (5) and Eq. (8). Moreover, in order to keep the constraint, constraint forces, $J_b^T\lambda_b$ and $J_v^T\lambda_v$ are introduced, respectively. Also, the following simultaneous equations are used to express the system including all constraints;

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau - J_b^T\lambda_b - J_v^T\lambda_v, \tag{12}$$

$$J_b\ddot{q} = -\dot{J}_b\dot{q} \tag{13}$$

$$J_v\ddot{q} = -\dot{J}_v\dot{q} \tag{14}$$

From these equations, accelerating vector $\ddot{q}$, and constraint forces $\lambda_b$ and $\lambda_v$, can be calculated, respectively. Therefore, both constraint force and acceleration can be used for judging the mode change. Collision with a wall or something else is assumed to be perfectly inelastic, and the state will shift to that of the under constraint just after the collision, which is modeled as the effects of impulse forces.

In two constraint forces $\lambda_b$ and $\lambda_v$, the entire dynamic system is represented as follows:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau - J_r^T\lambda_r \tag{15}$$

$$J_r\ddot{q} = -\dot{J}_r\dot{q} \tag{16}$$

where

$$J_r = [J_b\ \ J_e]^T,\ \lambda_r = [\lambda_b\ \ \lambda_e]^T$$

The expression of collision is defined by new constraint force $J_i\dot{q} = 0$ and it is described as follows:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau - J^T\lambda_r - J_i^T \tag{17}$$

$$J_r\ddot{q} = -\dot{J}_r\dot{q} \tag{18}$$

$$J_i\ddot{q} = -\dot{J}_i\dot{q} \tag{19}$$

Therefore we can derive the following equation using Eq. (17) and Eq. (18)

$$\lambda_r = X^{-1}\{\dot{J}_r\dot{q} + J_r M^{-1}(\tau - C\dot{q} - G - J_i^T\lambda_i)\} \tag{20}$$

where

$$X = J_r M^{-1} J_r^T$$

So we can represent the dynamic equation from Eq. (17) as follows:

$$\begin{aligned} M\ddot{q} + C\dot{q} + G = \tau - J_r^T X^{-1} \cdot \\ \{\dot{J}_r\dot{q} + J_r M^{-1}(\tau - C\dot{q} - G - J_i^T\lambda_i)\} - J_i^T\lambda_i \end{aligned} \tag{21}$$

## 3. Jumping Control Algorithm

Since the initial configuration is very important for the jumping motion of a robot, it is determined by *stochastic dynamic manipulability measure*. For dynamic control of the jumping of a robot, we pay particular attention to the motion of the mass center and the proposed method is derived as if center of the gravity is moved by a spring connected to a virtual wall.

### 3.1 Model Transformation Method

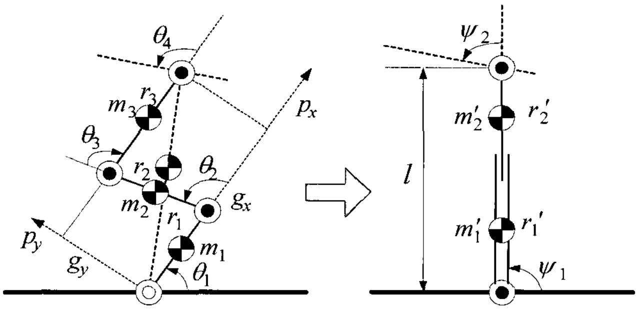



**Fig. 5** Model transformation for jumping motion

In the path design for jumping control, how should it just determine the desired value to the speed of the center of gravity? This is a very complex problem and the same thing can also be considered regarding the desired value of the rear leg. Therefore we consider the control to be applied constraint by moving 3-links of a leg using a virtual linear actuator as illustrated in Fig. 5.

Now, we consider the new coordinate as follows:

$$s=\begin{bmatrix}\psi_1\\ l\\ \psi_2\end{bmatrix}=\begin{bmatrix}\theta_1+\tan^{-1}(\frac{p_y}{p_x})\\ \sqrt{p_x^2+p_y^2}\\ -\theta_2-\theta_3-\theta_4+\pi+\tan^{-1}(\frac{p_y}{p_x})\end{bmatrix} \tag{22}$$

where $\theta_2>0$. Moreover we consider the constraint that the position of the gravity center of links is above the virtual linear actuator for one-to-one correspondence. Since we use new parameters $\varepsilon$ given by

$$\begin{aligned}\varepsilon(q)&=g_x p_y-g_y p_x\\ &=\xi_1 s_2+\xi_2 s_3+\xi_3 s_{23}\end{aligned} \tag{23}$$

Here the constraint condition and new coordinate using $s$ and $\varepsilon$ are defined by the following equation, respectively.

$$\varepsilon=0 \text{ and } x_s=[\phi \;\; l \;\; \psi \;\; \varepsilon]^T.$$

According to the relation for angular coordinate and velocity, the state is given by:

$$\dot{x}_s=\begin{bmatrix}J_s(q)\\ J_\varepsilon(q)\end{bmatrix}=J_{x_s}(q)\dot{q} \tag{24}$$

where

$$J_{x_s}=\begin{bmatrix}1 & J_2 & J_3 & 0\\ 0 & \frac{-l_1(l_2s_2+l_3s_{23})}{l} & \frac{-l_3(l_2s_3+l_1s_{23})}{l} & 0\\ 0 & -1+\frac{J_2}{a} & -1+\frac{J_3}{a} & -1\\ 0 & \xi_1c_2+\xi_3c_{23} & \xi_2c_3+\xi_3c_{23} & 0\end{bmatrix} \tag{25}$$

And the given parameters by the above equations are calculated as follows:

$$\xi_1=\frac{m_2a_2l_3+m_3l_2(l_2-a_1)}{M},\quad \xi_2=\frac{m_2a_2l_3+m_3l_2(l_3-a_3)}{M}$$
$$\xi_3=\frac{m_1a_1l_3+m_3l_1l_3+l_1l_3(m_2+m_3)}{M}$$

$$p_x=l_1+l_2c_2+l_3c_{23},\quad p_y=l_1+l_2s_2+l_3s_{23}$$
$$a=l_1^2+l_2^2+l_3^2+2l_1l_2c_2+2l_2l_3c_3+2l_1l_3c_{23}$$
$$J_2=l_2^2+l_3^2+l_1l_2c_2+2l_2l_3c_3+l_1l_3c_{23}$$
$$J_3=l_3^2+l_2l_3c_3+l_1l_3c_{23}$$

$$s_2=\sin\theta_2, s_3=\sin\theta_3, s_{23}=\sin(\theta_2+\theta_3)$$
$$c_2=\cos\theta_2, c_3=\cos\theta_3, c_{23}=\cos(\theta_2+\theta_3)$$

$rank J_x=4$ except for singular points $\theta_2=2n\pi$ and $\theta_3=2n\pi$ by model transformation as shown in Fig. 5 and it is assumed that $J_x^{-1}$ exists except for singular points. Therefore, we can know that model transformation is possible since $\ddot{q}$ is rewritten using Eq. (24) as follows:

$$\ddot{q}=J_x^{-1}(\ddot{x}-\dot{J}J_x^{-1}x) \tag{26}$$

### 3.2 Jumping Control

In this section, we discuss the jumping problem of a robot system with two legs. As the proposed robot system mimics the motion posture of a cat, it uses feedback control in order to converge upon a desired value that is decided by preparing itself for the next jumping motion.

#### *3.2.1 Standing with a hind leg*

In order to start the path in the desired direction, an angle needs to turn towards that direction. Therefore, we consider controlling the under-actuator link from the characteristic of the system. The control purpose in this paper is summarized as follows:

(1) To the angle that wants to jump an absolute angle of the rear leg,
(2) If possible, it is the angle of legs to desired posture.

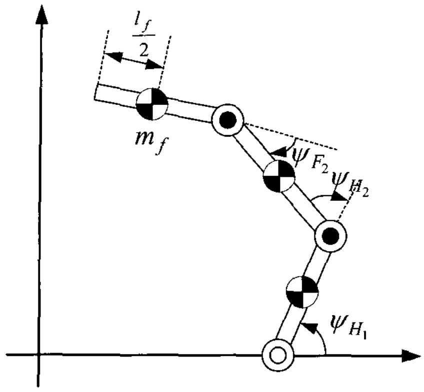



**Fig. 6** Coordinate configuration with one leg

We consider a 3-links model, which has fixed the lengths of legs $l_F$ and $l_H$, respectively. The structure of the proposed model is shown in Fig. 6. Using body angle $\theta_b$ for the relative angles of leg $\psi_{F_2}$ and $\psi_{H_2}$, the new state $x$ is considered as follows:

$$x=\begin{bmatrix}x_p & x_{a_1} & x_{a_2}\end{bmatrix}^T=\begin{bmatrix}\psi_{H_1} & \psi_{H_2} & \psi_{F_3}\end{bmatrix}^T \tag{27}$$

Therefore the quantity of motion $L(t)$ can be derived by

$$L(t)=L_0-\int_0^t G(x)dt=H_p\dot{x}_p+H_{a_1}\dot{x}_{a_1}+H_{a_2}\dot{x}_{a_2} \tag{28}$$

where

$$H_p(x)=\alpha_1+\alpha_2+\alpha_3+2\beta_1C_1+2\beta_2C_2+2\beta_3C_3$$
$$H_{a_1}(x)=\alpha_2+\alpha_3+\beta_1C_1+2\beta_2C_2+\beta_3C_3$$
$$H_{a_2}(x)=\alpha_3+\beta_2C_2+\beta_3C_3$$
$$G(x)=\frac{1}{2}\{(m_h+2m_b+2m_f)l_h\cos x_p$$
$$+(m_b+2m_f)l_b\cos(x_p+x_{a_1})$$
$$+m_fl_f\cos(x_p+x_{a_1}+x_{a_2})\}g$$

Here, $\alpha_i$ and $\beta_i$ $(i=1,2,3)$ are constants decided by the system. These parameters are defined by

$$\alpha_1=J_h+\{4(m_b+m_f)+m_h\}l_h^2,$$
$$\alpha_2=J_b+(m_b+4m_f)l_b^2,\ \ \alpha_3=J_f+m_fl_f^2$$
$$\beta_1=2(m_b+2m_f)l_hl_b,\ \ \beta_2=2m_fl_fl_b$$
$$\beta_3=2m_fl_fl_h,\ \ C_1=\cos(x_{a_1})$$
$$C_2=\cos(x_{a_2}),\ \ C_3=\cos(x_{a_1}+x_{a_2})$$

To consider input to have angular velocity of an under-actuated link, the given system is represented as follows:

$$\dot{x}=\begin{bmatrix}\frac{L(t)}{H_p(\psi)}\\ 0\\ 0\end{bmatrix}+\begin{bmatrix}-\frac{H_{a_1}(q)}{H_p(\psi)} & -\frac{H_{a_2}(q)}{H_p(\psi)}\\ 1 & 0\\ 0 & 1\end{bmatrix}u \tag{29}$$

$$\frac{L(t)}{dt}=G(x) \tag{30}$$

Therefore, the control input $u(t)$ pertaining to desired velocity $\dot{x}_{pd}$ of an under-actuated link is given by

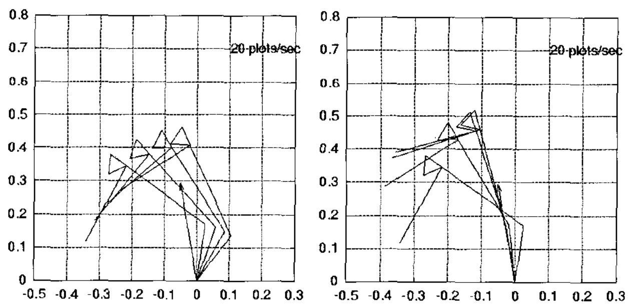
$$u(t)=J_a^+\{\dot{x}_{pd}-\frac{L(t)}{H_p}\}+(J_a^+J_a-I)\varepsilon \tag{31}$$

where

$$J_a=\begin{bmatrix}-\frac{H_{a_1}}{H_p} & -\frac{H_{a_2}}{H_p}\end{bmatrix} \tag{32}$$

and $J_a^+J$ exists zero space of $J_a$, and $\varepsilon$ shall decide that $\|\dot{x}_{ad}-u\|^2$ becomes the minimum to desired velocity $\dot{x}_{ad}$ of $\dot{x}_a$.

The simulation results for these conditions are shown in Fig. 7.




(a) without any control (b) with control

**Fig. 7** Movement of robot standing with a hind leg

In these results, an initial posture $x_0$ and a desired posture $x_f$ are given by

$$x_0=\begin{bmatrix}1.42 & 1.09 & 1.70\end{bmatrix}^T$$
$$x_f=\begin{bmatrix}1.75 & 0.10 & 1.57\end{bmatrix}^T$$

The quality of motion for an initial angle is $L_0=1.57$, and the desired velocity and $K_p$ are given by

$$\dot{x}_d=K_p(x(t)-x_f)$$
$$K_p=diag(20,30,20)$$

### 3.2.2 In the air

For control while in the air, we should prepare the controlling arbitrary posture for landing on the ground. Moreover, since the center of gravity may be considered by parabola movement in the air, the movement waiting when it jumps out is conserved and the jumping of a cat robot is exercised in the air. That is, this system has nonholonomic constraints resulting from the conservation law of movement.

In this paper, we will consider the method introduced by [7] in order to control the system that has drift terms. The procedure for the proposed control method is simply summarized as follows:

(1) The coordinate transformation
(2) The transformation to the time invariant nonlinear system
(3) The approximation by neighborhood of desired state

Moreover we will apply the learning control for an improvement of robustness [1]. The control purpose for jumping control in the air is summarized as follows: *landing on the roof on which height is set up beforehand with a desired posture from a fore limb.*

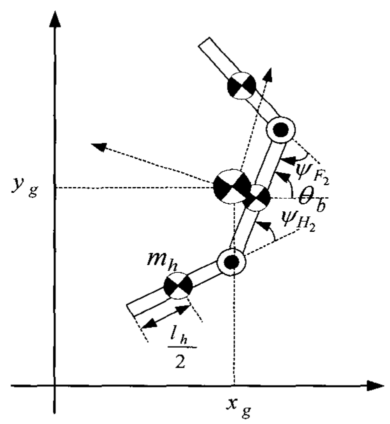



**Fig. 8** Coordinate configuration in the air

As shown in Fig. 8, we consider a 3-links model, which has fixed the length of legs, $l_F$ and $l_H$. Using body angle $\theta_b$ for the relative angle of legs $\psi_{F_2}$ and $\psi_{H_2}$, the new state $x$ is also rewritten by

$$x=\begin{bmatrix}\psi & \theta_b{}^T\end{bmatrix}=\begin{bmatrix}\psi_{F_2} & \psi_{H_2} & \theta_b\end{bmatrix}^T \tag{33}$$

For simplicity, we assume that the center of gravity is the center of each link. The circumference of the center of gravity in the air is described as follows:

$$L(t) = H_{aF}(\psi)\dot{\psi}_{F2} + H_{aB}(\psi)\dot{\theta}_b \tag{34}$$

Moreover, to consider the velocity of the link to be an input $u=[u_F \;\; u_H]^T$, the system is represented as follows:

$$\frac{dx}{dt}=\begin{bmatrix}0\\0\\\frac{L}{H_{aB}(\psi)}\end{bmatrix}+\begin{bmatrix}1 & 0\\0 & 1\\-\frac{H_{aF}(\psi)}{H_{aB}(\psi)} & -\frac{H_{aH}(\psi)}{H_{aB}(\psi)}\end{bmatrix}u$$
$$:= f(x)+g(x)u \tag{35}$$

where

$$H_{aF}(\psi)=h_F+h_1C_{a1}+h_3C_{a3} \tag{36}$$

$$H_{aH}(\psi)=h_H-h_2C_{a2}-h_3C_{a3} \tag{37}$$

$$H_{aB}(\psi)=h_B+2h_1C_{a1}+2h_2C_{a2}+2h_3C_{a3} \tag{38}$$

Here, $h_F,\cdots,h_3$ are constants to be decided by the system and these constants are given by

$$h_F=J_f+\frac{l_f^2m_f(m_b+m_h)}{4M}$$
$$h_H=J_h-\frac{l_h^2m_h(m_b+m_f)}{4M}$$
$$h_B=J_f+J_b+J_h+\frac{l_b^2(m_bm_f+m_hm_h+m_fm_h)}{4M}$$
$$+\frac{l_f^2m_f(m_b+m_h)+l_h^2m_h(m_b+m_f)}{4M}$$
$$h_1=\frac{l_bl_fm_f(m_b+2m_h)}{4M},\; h_2=\frac{l_bl_hm_h(m_b+2m_f)}{4M}$$
$$h=\frac{l_fl_hm_fm_h}{4M},\; M=m_f+m_b+m_h$$
$$C_{a1}=\cos\psi_{F_2},\, C_{a2}=\cos\psi_{H_2}\;\; C_{a3}=\cos(\psi_{F_2}+\psi_{H_2})$$

Since the state to realize at the time of landing is given by $x_f$, the error system of the state equation can consider a given error equation as follows:

$$\bar{x}=x-x_f=(\bar{q},\bar{\phi}) \tag{39}$$

Therefore, we can also consider coordinate transformation of time-varying and input transformation as follows:

$$\xi=\begin{bmatrix}\xi_\psi\\\xi_{\theta_b}\end{bmatrix}=\begin{bmatrix}\bar{\psi}\\\bar{\theta}_b-\frac{L}{H_b(0)}\times(t-t_f)\end{bmatrix} \tag{40}$$

$$\mu=[\mu_1\;\mu_2]^T=u \tag{41}$$

where $t_f$ is an initial vector and it is the landing time computed from an initial state and a desired state. Then, it is transferred by the time invariant nonlinear system as follows:

$$\frac{d\xi}{dt} = \begin{bmatrix} 0 \\ 0 \\ \frac{L}{H_{aB}(\psi)} - \frac{L}{H_{aB}(0)} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -\frac{\overline{H}_{aF}(\xi_\psi)}{\overline{H}_{aB}(\xi_\psi)} & -\frac{\overline{H}_{aH}(\xi_\psi)}{\overline{H}_{aB}(\xi_\psi)} \end{bmatrix} \mu$$

$$:= f(\xi) + g(\xi)\mu \tag{42}$$

In order to realize the control for this system, Eq. (22), we execute linearization around the desired state, and the state feedback is also given by *LQ optimal control* as follows:

$$\frac{d\xi}{dt} = f(0) + \left.\frac{\partial f}{\partial \xi}\right|_{\xi=0} + g(0)\mu$$

$$= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{\partial}{\partial \xi_F}\left(\frac{L}{H_{aB}(\xi_\psi)}\right) & \frac{\partial}{\partial \xi_H}\left(\frac{L}{H_{aB}(\xi_\psi)}\right) & 0 \end{bmatrix}_{\xi=0} \xi$$

$$+ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -\frac{\overline{H}_{aF}(0)}{\overline{H}_{aB}(0)} & -\frac{\overline{H}_{aH}(0)}{\overline{H}_{aB}(0)} \end{bmatrix} \mu \tag{43}$$

Though the weight matrices are defined by the following equations, the pole placement in a specified region is performed in order that the very late mode exists. The weight matrices are defined by

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 100 \end{bmatrix}, R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, d = 3 \tag{44}$$

In these simulation results, the desired value and an initial state are defined as follows:

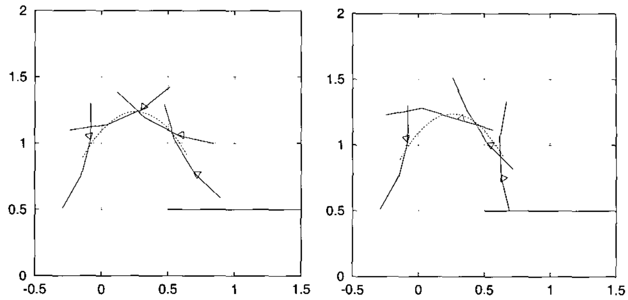
$$x = \begin{bmatrix} -0.1 & 0.1 & -\frac{\pi}{2} \end{bmatrix}^T, x_0 = \begin{bmatrix} -\frac{\pi}{12} & \frac{\pi}{12} & -\frac{5\pi}{12} \end{bmatrix}^T$$

The initial angular velocity of the center of gravity is referred to as 3[msec] in the direction of 60[degrees], and $L_0 = -0.4$, respectively. The parameters to be used in the simulation are indicated in Table 1 and the movement of the jumping robot in the air is shown in Fig. 9.

**Table 1** Variable parameters

| | $h_F$ | $h_B$ | $h_H$ | $h_1$ | $h_2$ | $h_3$ |
|---|---|---|---|---|---|---|
| $[m^2 g]$ | 0.011 | 0.037 | -0.004 | 0.011 | 0.011 | 0.004 |

Moreover the body angle $\theta_b$ and state $\xi$ are shown in Fig. 10 and Fig. 11, respectively. By controlling as shown in these results, it can check that it has landed and the state almost converges to 0 after the body angle has become perpendicular.




(a) without any control (b) with control

**Fig. 9** Movement of robot in the air

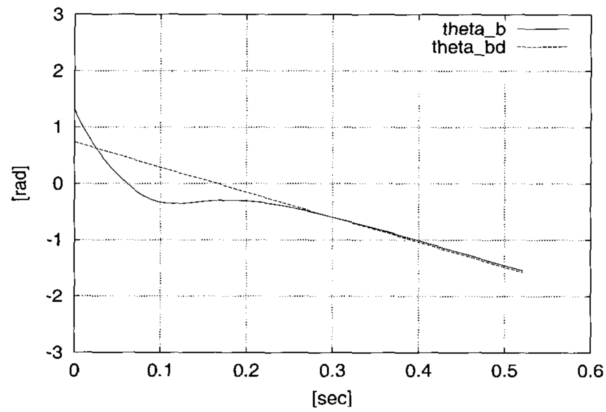



**Fig. 10** Body angle $\theta_b$

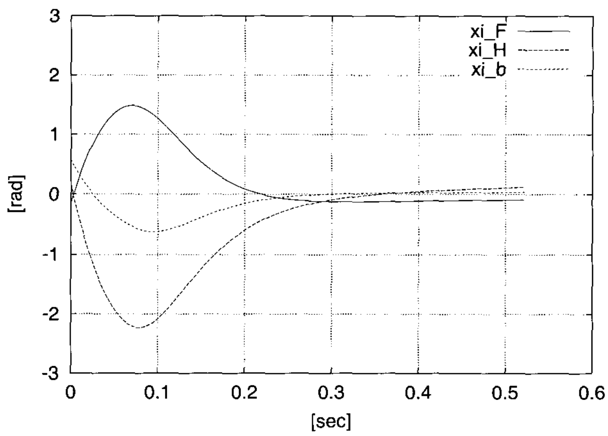



**Fig. 11** State $\xi$

### 3.3 Iterative Learning Control

In an actual system, since the input is given in many cases in torque input, the convergence of $\varepsilon$ is not guaranteed when sufficient input is not given or parameter

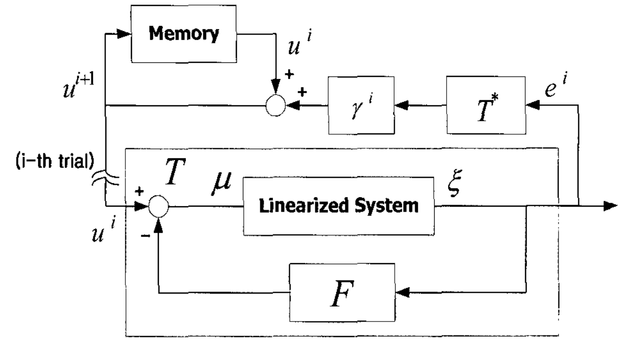



**Fig. 12** Iterative learning control

error exists. Therefore, we will realize *repetitive learning control* for a linearized system given by Eq. (43) [3]. Then, the system is realized by controlled minor loop feedback control pertaining to linearization, $u^i(t)$ which is a learning term generated by error signal in this system is added in Fig. 12.

That is,

$$\xi(t) = (A + FB)\xi(t) + u^i(t) \tag{45}$$

$$u^{i+1}(t) = u^i(t) + rz^i \tag{46}$$

where $\xi_d(t)$ is desired stat and $e^i(t)$ is defined by

$$e^i(t) = \xi_d(t) - \xi^i(t) \tag{47}$$

However, we specify $\xi_d = 0$ in this system. Multiplying the error by weight filter, $T_f$ in order to diminish the error of finial time, the criterion function $J^i$ is also defined as follows:

$$e^i_w(t) = W_{e\beta} e^{-(t_f - t)W_{e\alpha}} e^i(t) = T_f e^i \tag{48}$$

$$J^i = \int_0^{t_f} e_\omega^{iT}(t) e_\omega^i(t)dt = \left\| e_\omega^i \right\|^2 \tag{49}$$

where

$$W_{e\alpha} = \begin{bmatrix} \alpha_F & 0 & 0 \\ 0 & \alpha_H & 0 \\ 0 & 0 & \alpha_{\theta_b} \end{bmatrix}, \quad W_{e\beta} = \begin{bmatrix} \beta_F & 0 & 0 \\ 0 & \beta_H & 0 \\ 0 & 0 & \beta_{\theta_b} \end{bmatrix}$$

Then the control law is also given by

$$z^i(t) = T_p^* T_f^* T_f e^i \tag{50}$$

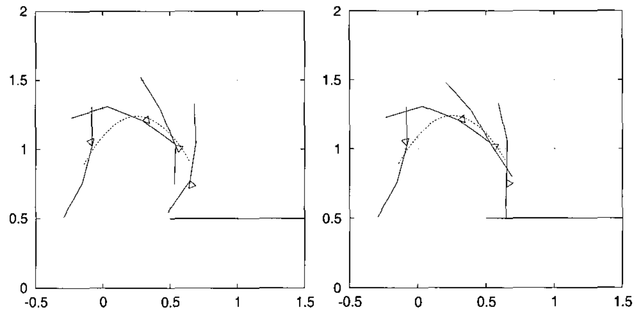
$$u^{i+1} = u^i + \gamma^i z^i, \tag{51}$$

$$\gamma^i = \frac{\langle T_p^* T_f^* T_f e^i, z^i \rangle}{\left\| T_f T_p z^i \right\|^2}, \tag{52}$$

In the system of 3-links in the air, we should execute numerical simulations adding disturbance as follows:

$$\dot{\xi}(t) = f(\xi) + g(\xi)u + w(t), \quad |w(t)| < 5 \tag{50}$$

where $\alpha = 5$ and $\beta = 1$.



(a) 1st trial (b) 10th trial

**Fig. 13** Movement of jumping control in the air

Therefore the movement of jumping control in the air applied an iterative learning control that is shown in Fig. 13. The criterion function is also indicated in Fig. 14. From this result, we can ensure that the criterion function is decreased about 30% and the landing is successful.

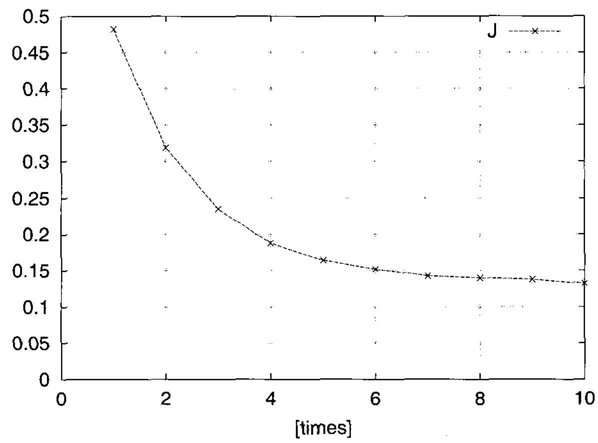



**Fig. 14** Criterion function

## 4. Simulation and Experimental Results

This section illustrates the performance of the proposed control algorithm for a jumping robot system using simulation and experimental results.

In the simulation, it is assumed that a fore leg and a rear leg have same parameters, respectively. Moreover the limit angles of links are designed by considering the experimental system of a jumping robot as indicated in Fig. 15.

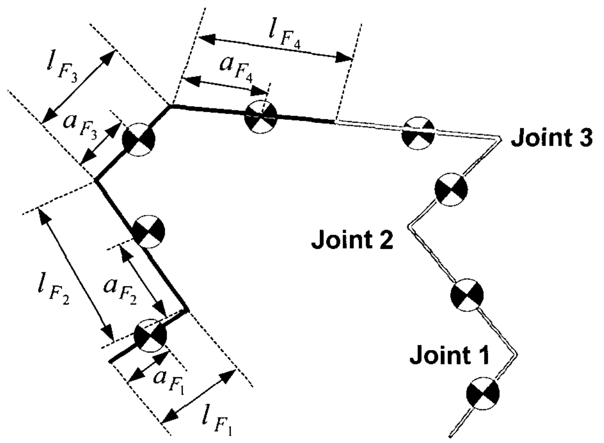



**Fig. 15** Parameters of jumping robot system

Moreover the system parameters used in optimization and the simulation parameters are determined as shown in Table 2. Also, we assume that both the wall and roof for jumping control are located at $x=-0.4[m]$ and $y=0.5[m]$, respectively. These results are presented in Fig. 16 and Fig. 17

**Table 2** System parameters for jumping robot system

| Link 1 | Link 2 | Link 3 | Link 4 |
|---|---|---|---|
| $m_1=0.15$ | $m_2=0.15$ | $m_3=0.195$ | $m_4=0.32$ |
| $l_1=0.09$ | $l_2=0.12$ | $l_3=0.09$ | $l_4=0.15$ |
| $a_1=0.045$ | $a_2=0.06$ | $a_3=0.045$ | $a_4=0.075$ |
| $J_1=\frac{m_1 l_1^2}{12}$ | $J_2=\frac{m_2 l_2^2}{12}$ | $J_3=\frac{m_3 l_3^2}{12}$ | $J_4=\frac{m_4 l_4^2}{12}$ |
| $V_1=0.02$ | $V_2=0.34$ | $V_3=0.34$ | $V_4=0.34$ |

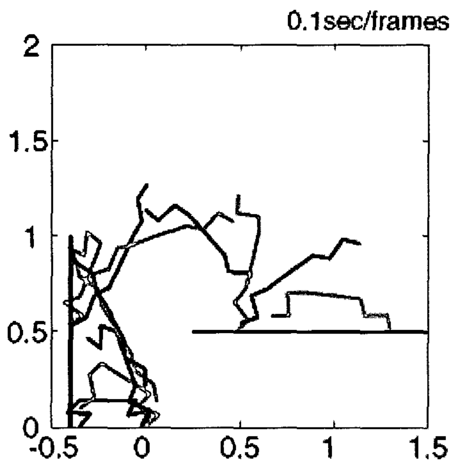



**Fig. 16** Simulation result of jumping robot system

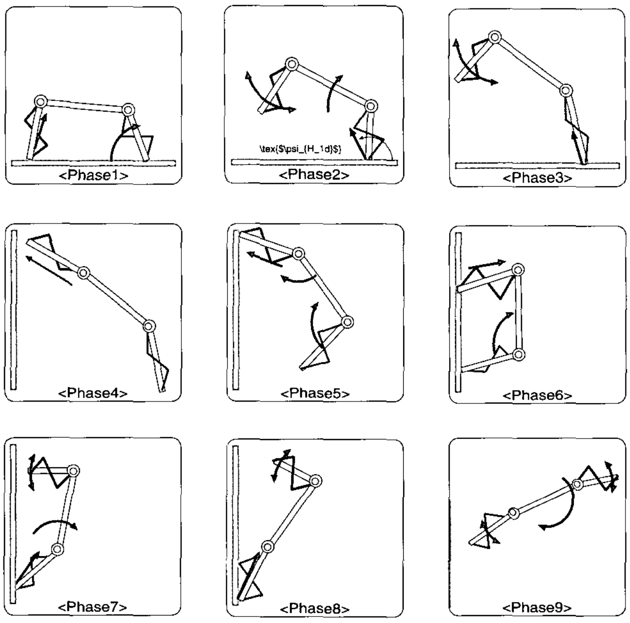



**Fig. 17** Schematic configuration of the jumping motion

For the experiment, we designed a 7-links jumping robot system as shown in Fig. 18 and the configuration of the entire system is illustrated by Fig. 19.

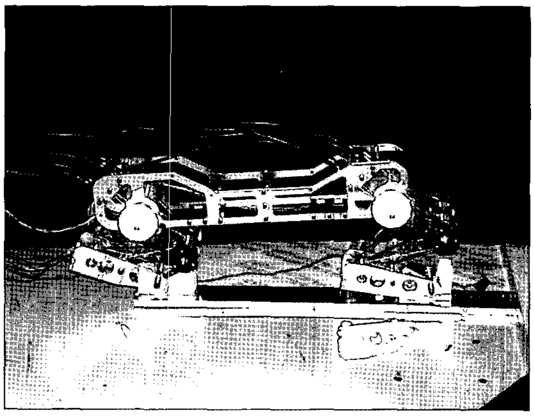


**Fig. 18** Experimental system

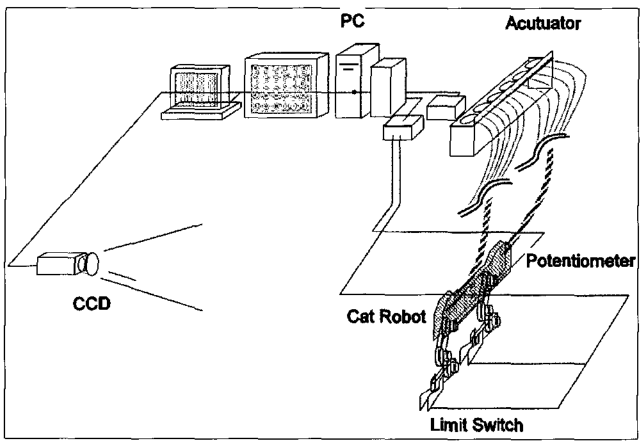



**Fig. 19** Configuration of jumping control system

Since motors are too heavy to be installed in the robot, the power of actuators is supplied from outside by wires as revealed in Fig. 20 and the weight of the wire is compensated by a contour weight. The position and toe angle are measured by a CCD camera, and the potentiometer used to encode the motors are useless due to the extension of wire in order to measure joint angles. Because of the wire extension, there are numerous time delays in the wire system.

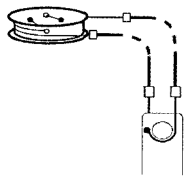
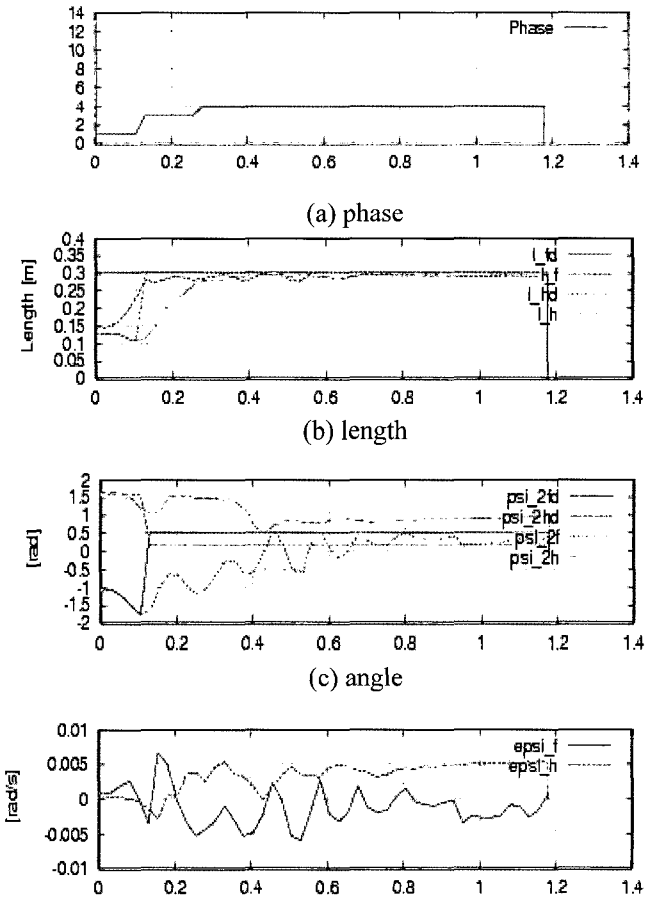


**Fig. 20.** Operator principle of motor and wire system

The problems due to time loss and disturbance of the wire system are too serious. Also, it is difficult to apply the proposed control method directly. Therefore, the learning control is applied in this experimentation and the experimental results are shown in Fig. 21.




(a) phase

(b) length

(c) angle

(d) constraint

**Fig. 21** Experimental results

## 5. Conclusion

In this paper, we proposed a jumping control method of a cat robot using model transformation. Moreover, we confirmed the effectiveness of the control law by simulation and experimental results, in which the jumping robot could jump towards a wall and land on it. For its initial posture evaluation, we used stochastic dynamic manipulability measure. In order to realize the real robot, the learning control was applied and significant progress was confirmed.

In these results, since the disturbances in wire system were very large, we faced difficulties when applying the proposed control method directly. Moreover the jumping motion has not yet been completed and further work should include the following conditions:

(1) Completion of the jumping with a real robot
(2) Learning control in task space
(3) Realization of 3-dimensional jumping robot and development of control into 3-dimensional system

## Acknowledgements

This work was supported by the Regional Technology Innovation program (RT104-02-06) of the Korean Ministry of Commerce, Industry and Energy.

## References

[1] M. Yamakita and K. Furuta, "Iterative Generation of Virtual Reference for a Manipulator", *J. of Robotica*, Vol. 9, pp. 71-80, 1991.

[2] M. Yamakita and Y. Omagari, "Jumping cat robot with kicking a wall", *Proc. of Adaptive Motion of Animals and Machines*, 2002.

[3] J. H. Suh, M. Yamakita, and S. B. Kim, "Jumping Control of Cat Robot System using Model Transformation", *Proc. of KIEE Conf.*, pp. 2427-2429, 2002.

[4] J. H. Suh, M. Yamakta, and K. S. Lee, "A Study on Motion Planning Generation of Jumping Robot System using Model Transformation Method", *J. of KSPE*, Vol. 21, No. 4, pp. 120-131, April, 2004.

[5] J. H. Suh, M. Yamakita, and K. S. Lee, "Study on Jumping Robot System using Model Transformation", *Proc. of KIEE Conf.*, pp. 228-235, May, 2004.

**Jin-Ho Suh**
He received his B.S. degree in Mathematics from Hanyang Univ. and his M.S. degree in Control Engineering from Pukyong National Univ. in 1993 and 1998, respectively. He also received his Ph.D. degree in Control Engineering from Tokyo Institute of Technology, Japan, in 2002. His research interests are robotics, nonlinear control for robotics, and the underwater robot.

**Masaki Yamakita**
He received the B.S. M.S. and Ph.D. from Tokyo Institute of Technology, Tokyo, Japan in 1984, 1986 and 1989, respectively. From 1989 he was a Research Associate in the Department of Control Engineering of Tokyo Institute of Technology and from 1993 he was a Lecture at Tokyohashi University of Technology. He is currently an Associate Professor in the Department of Control and System Engineering, Tokyo Institute of Technology, Japan, which he joined in 1995. His principle research interests are robotics, learning control, robust control and nonlinear control.