Final-state Control of A Two-link Cat Robot by Feedforward Torque Inputs

Zhiqiang WENG* and Hidekazu NISHIMURA**

*Graduate School of Science and Technology, Chiba University
**Dept. of Electronics and Mechanical Eng., Chiba University,
1-33 Yayoi-cho, Inage-ku, Chiba 263-8522, JAPAN
tel: +81-43-290-3194; fax: +81-43-290-3039; e-mail: nism@meneth.tm.chiba-u.ac.jp

Abstract: This paper deals with twisting motion of a falling cat robot by two torque inputs around her waist. The cat robot consists of two rigid columns and has internally two actuators at the joint to generate torque inputs around the normal coordinates. This system is a nonholonomic system whose angular momentum is conserved. We obtain the Lagrange's equation of motion with nonholonomic constraint and formulate the linear parameter varying system. Then, in order to get the twisting motion we apply the error learning method of final-state control with amplitude constraint of inputs. In simulation we show that the two-link cat robot can pose so that she lands on her feet, by using of the obtained feedforward torque inputs by the final-state control.

Key words : Motion Control, Two-link Cat Robot, Nonholonomic Constraint, Lagrange's Equation of Motion, Linear Parameter Varying System, Final-State Control, Input Constraint, Error Learning

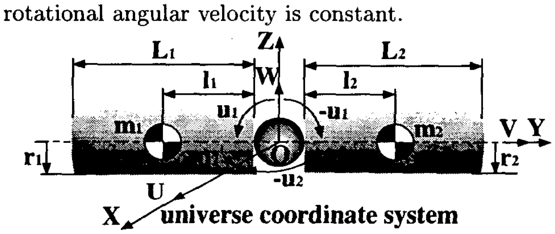
1. Introduction

A cat dropped from upside-down with no angular momentum can land on her feet. This is called the twisting motion of a falling cat. The system's angular momentum is conserved and there are constraint condition in the velocity terms. It is impossible to integrate the velocity constraint with respect to time, so that the falling cat motion is a first-order nonholonomic system. Furthermore it is impossible to make the system asymptotically stable smoothly by a time-invariant state feedback control[1].

In Ref. [2], although the proof uses only kinetic energy, it is shown that initial and final configurations of molecules which differ from each other by a rotation can also be connected to each other by a sequence of purely vibrational motions. The most striking instance is the well-known twisting motion of a falling cat. And in Ref. [3], "cat's problem" is defined as follows: find the most efficient way to realize the twisting motion of a falling cat. By differential geometry, Montgomery shows the cat's problem is equivalent to the isoholonomic problem if and only if the total angular momentum through the motion is zero. And the description of the isoholonomic problem in terms of nonholonomic Hamilton equation is the main mathematical result of Ref. [3].

There are several studies for this problem. In Ref. [4] and [5], from video observation, Kawamura *et.al.* paid attention to trunks of the cat which had not moved torsionally, but had turned both her front and rear bodies to the same direction at the same time. Then a cat was assumed as a two-link robot and the nonholonomic constraint condition was derived. They analyzed a falling cat's behavior by adopting DSS(dynamics system simulation). The falling cat motion has been clarified by the experimental setup using pneumatic actuators. However, the final state of the motion obtained has not been clarified.

In Ref. [6], the exact linearization technique was used for the state equation based on the conservation of the angular momentum with the angular velocity inputs. The feedback control was performed on condition that rotational angular velocity is constant.



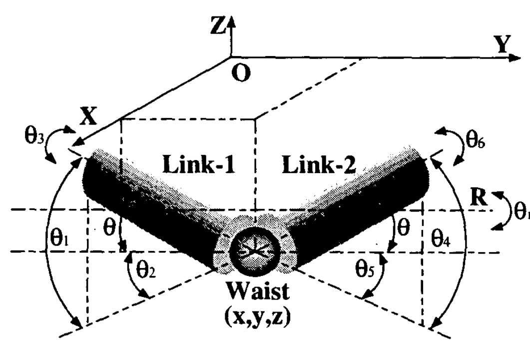


Fig.1 Modeling of two-link cat robot

On the other hand, in Ref. [7], a ski robot was developed to perform somersault and twist motion. The same formulation as Ref. [6] was performed by the nonholonomic constraint, and the inputs are angular velocities. They obtained feedforward inputs based on the Fourier bases algorithm.

0-7803-5976-3/00/$10.00 ©2000 IEEE.

This paper formulates the state equation which has torque inputs to the joint by using the nonholonomic constraint and Lagrange's function based on Ref. [8], with respect to the same two-link cat robot without torsion as the study of Kawamura *et.al.*[4]. Then, we transform the system into the linear parameter varying system. The feedforward torque inputs were obtained by the final state control with error learning to bring the system from an initial state to a final state in a desired time. It is verified by simulation that the objective motion control is realized by feedforward inputs based on the final-state control.

2. Modeling

In this section, we lead the dynamical model of the two-link cat robot without torsion as shown in Fig. 1. Two actuators are arranged around the waist to occur torque around U-axis and W-axis and torque inputs around the waist can bend the waist joint. The notations in this paper are shown as follows:

$S_i = \sin\theta_i$ $\quad C_i = \cos\theta_i$ $\quad$ (i = 1,2,3,4,5,6)

$\theta_{1,4}$(rad)	: angle around U-axis
θ_3(rad)	: angle around V-axis
$\theta_{2,5}$(rad)	: angle around W-axis
θ_r(rad)	: angle around R-axis
θ(rad)	: bending angle of robot
$m_{1,2}$(kg)	: mass of link
$r_{1,2}$(m)	: radius of link
$L_{1,2}$(m)	: length of link
$l_{1,2}$(m)	: distance from waist to center of gravity
$J_{1,3}$(kg·m^2)	: inertial moment around U or W-axis
$J_{2,4}$(kg·m^2)	: inertial moment around V-axis
J_r(kg·m^2)	: inertial moment around R-axis
u_1(N·m)	: torque input around U-axis
u_2(N·m)	: torque input around W-axis
$d_{1,2}$(N·m·s/rad)	: damping coefficient around waist
g(m/s^2)	: gravitational acceleration

On the assumption that only internal forces act on the two-link cat robot and that the parameters of link-1 are the same as those of link-2, it is clear that $\theta_1 = -\theta_4$, $\theta_2 = -\theta_5$, $\theta_3 = \theta_6$, $y = 0$.

2·1 Nonholonomic constraint condition

Ref. [4] gave the nonholonomic constraint condition

$$2J_2\dot{\theta}_3 \cos\theta_1 \cos\theta_2 + 2J_r\dot{\theta}_r = 0. \qquad (1)$$

Solving Eq. (1) for $\dot{\theta}_3$ yields

$$\dot{\theta}_3 = f_1(\theta_1,\theta_2)\dot{\theta}_1 + f_2(\theta_1,\theta_2)\dot{\theta}_2 \qquad (2)$$

where

$$f_1 = \frac{[(3r^2+L^2)+(3r^2-L^2)\cos^2\theta_1\cos^2\theta_2]\sin\theta_2}{-6r^2\cos\theta_1\cos\theta_2(1-\cos^2\theta_1\cos^2\theta_2)},$$

and

$$f_2 = \frac{[(3r^2+L^2)+(3r^2-L^2)\cos^2\theta_1\cos^2\theta_2]\sin\theta_1}{6r^2(1-\cos^2\theta_1\cos^2\theta_2)}.$$

Differential calculus to Eq. (2) yields

$$\ddot{\theta}_3 = f_1(\theta_1,\theta_2)\ddot{\theta}_1 + f_2(\theta_1,\theta_2)\ddot{\theta}_2 + f_{12}(\theta_1,\theta_2,\dot{\theta}_1,\dot{\theta}_2) \qquad (3)$$

where

$$f_{12} = \frac{\partial f_1}{\partial\theta_1}\dot{\theta}_1^2 + \frac{\partial f_2}{\partial\theta_2}\dot{\theta}_2^2 + (\frac{\partial f_1}{\partial\theta_2} + \frac{\partial f_2}{\partial\theta_1})\dot{\theta}_1\dot{\theta}_2.$$

2·2 Equation of motion with Lagrange's function

An equation of motion with Lagrange's function is described as Eq. (4), where T is the kinetic energy, V is the potential energy, and D is the loss energy.

$$\langle \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} + \frac{\partial D}{\partial \dot{q}}\ ,\ \delta q\rangle = \langle u\ ,\ \delta q\rangle \qquad (4)$$

where $q = [x\ z\ \theta_1\ \theta_2\ \theta_3]^T$, $L = T - V$, δq is the variation of q, and $\langle\ ,\ \rangle$ means the inner product. Then, from Eq. (2) we obtain

$$\delta\theta_3 = f_1(\theta_1,\theta_2)\delta\theta_1 + f_2(\theta_1,\theta_2)\delta\theta_2, \qquad (5)$$

and using Eqs. (3) and (5) yields

$$\left(\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_3}\right) - \frac{\partial L}{\partial \theta_3} + \frac{\partial D}{\partial \dot{\theta}_3}\right)\delta\theta_3 = J_2\ddot{\theta}_3\delta\theta_3$$
$$= J_2(f_1\ddot{\theta}_1 + f_2\ddot{\theta}_2 + f_{12})(f_1\delta\theta_1 + f_2\delta\theta_2). \qquad (6)$$

Using Eq. (6), Eq. (4) can be rewritten as follows:

$$\begin{aligned}
&(2m_1\ddot{x} - 2m_1l_1S_1S_2\ddot{\theta}_1 + 2m_1l_1C_1C_2\ddot{\theta}_2\\
&-2m_1l_1\left(\dot{\theta}_1^2+\dot{\theta}_2^2\right)C_1S_2 - 4m_1l_1\dot{\theta}_1\dot{\theta}_2C_2S_1)\delta x\\
&+(2m_1\ddot{z} - 2m_1l_1C_1\ddot{\theta}_1 + 2m_1l_1\dot{\theta}_1^2S_1 + 2m_1g)\delta z\\
&+(-m_1l_1S_1S_2\ddot{x} - m_1l_1C_1\ddot{z} + \left(m_1l_1^2+J_1\right)\ddot{\theta}_1 - m_1l_1gC_1\\
&+m_1l_1^2\dot{\theta}_2^2C_1S_1 + 2d\dot{\theta}_1 + J_2(f_1\ddot{\theta}_1 + f_2\ddot{\theta}_2 + f_{12})f_1)\delta\theta_1\\
&+(m_1l_1C_1C_2\ddot{x} + \left(m_1l_1^2C_1^2+J_1\right)\ddot{\theta}_2 - 2m_1l_1^2\dot{\theta}_1\dot{\theta}_2C_1S_1\\
&+2d\dot{\theta}_2 + J_2(f_1\ddot{\theta}_1 + f_2\ddot{\theta}_2 + f_{12})f_2)\delta\theta_2\\
&= u_1\delta\theta_1 + u_2\delta\theta_2.
\end{aligned} \qquad (7)$$

We can eliminate $\ddot{x}$ and $\ddot{z}$ from Eq. (7), then we can derive equations of motion as follows:

$$\begin{aligned}
&\left(m_1l_1^2S_1^2C_2^2 + J_1 + J_2f_1^2\right)\ddot{\theta}_1 + m_1l_1^2\left(\dot{\theta}_1^2+\dot{\theta}_2^2\right)C_1C_2^2S_1\\
&+(m_1l_1^2S_1C_1S_2C_2 + J_2f_1f_2)\ddot{\theta}_2 + 2d\dot{\theta}_1\\
&+\left(J_2f_{12}f_1/S_2 - 2m_1l_1^2\dot{\theta}_1\dot{\theta}_2S_1^2C_2\right)S_2 = u_1,\\
&\left(m_1l_1^2S_1C_1S_2C_2 + J_2f_1f_2\right)\ddot{\theta}_1 + m_1l_1^2\left(\dot{\theta}_1^2+\dot{\theta}_2^2\right)C_2C_1^2S_2\\
&+\left(m_1l_1^2C_1^2S_2^2 + J_1 + J_2f_2^2\right)\ddot{\theta}_2 + 2d\dot{\theta}_2\\
&+(J_2f_{12}f_2/S_1 - 2m_1l_1^2\dot{\theta}_1\dot{\theta}_2S_2^2C_1)S_1 = u_2.
\end{aligned}$$

It is worth while noting that the falling cat is unrelated to gravitation, because the gravitational term of the governing equation is disappeared.

The equation of motion for θ_1 and θ_2 can be described as follows:

$$M(\Theta)\ddot{\Theta} + D\dot{\Theta} + H\left(\Theta, \dot{\Theta}\right) = \boldsymbol{u}. \qquad (8)$$

where $M(\Theta)$, D are 2×2 matrices,

$$\boldsymbol{u} = [u_1\ u_2]^T, \quad \Theta = [\theta_1\ \theta_2]^T, \quad H\left(\Theta, \dot{\Theta}\right) = [H_1\ H_2]^T,$$

$$\begin{aligned} H_1 = {} & m_1 l_1^2 \left(\dot{\theta}_1^2 + \dot{\theta}_2^2\right) C_1 C_2^2 S_1 \\ & + (J_2 f_{12} f_1 / S_2 - 2 m_1 l_1^2 \dot{\theta}_1 \dot{\theta}_2 S_1^2 C_2) S_2, \end{aligned}$$

and

$$\begin{aligned} H_2 = {} & (J_2 f_{12} f_1 / S_1 - 2 m_1 l_1^2 \dot{\theta}_1 \dot{\theta}_2 S_2^2 C_1) S_1 \\ & + m_1 l_1^2 \left(\dot{\theta}_1^2 + \dot{\theta}_2^2\right) C_2 C_1^2 S_2. \end{aligned}$$

Since the nonlinear terms $H\left(\Theta, \dot{\Theta}\right)$ are remained, we transform the nonlinear system to a linear parameter varying system without approximation by adopting the following extended linearization[9]:

$$p \times \sin\theta_i = p \times \frac{\sin\theta_i}{\theta_i} \times \theta_i \qquad (i = 1, 2) \qquad (9)$$

where p is an arbitrary polynomial. In Eq. (9) when $\theta_i = 0$, it is assumed that $\sin\theta_i / \theta_i = 1$. By using Eq. (9), matrix $G\left(\Theta, \dot{\Theta}\right)$ that includes the nonlinear terms can be obtained so that the following relation is satisfied

$$H\left(\Theta, \dot{\Theta}\right) = G\left(\Theta, \dot{\Theta}\right)\Theta. \qquad (10)$$

By using Eqs. (2), (8), and (10), the state equation of the linear parameter varying system is obtained as follows:

$$\dot{\boldsymbol{x}}_s = A_s(\boldsymbol{x}_s)\boldsymbol{x}_s + B_s(\boldsymbol{x}_s)\boldsymbol{u}, \qquad (11)$$

where $A_s(\boldsymbol{x}_s)$ and $B_s(\boldsymbol{x}_s)$ include nonlinear terms, and are respectively shown as follows:

$$A_s(\boldsymbol{x}_s) = \begin{bmatrix} O_{2\times2} & \boldsymbol{I}_{2\times2} & O_{2\times1} \\ -M^{-1}G(\Theta, \dot{\Theta}) & -M^{-1}D & O_{2\times1} \\ O_{1\times2} & f_1 \quad f_2 & O_{1\times1} \end{bmatrix},$$

$$B_s(\boldsymbol{x}_s) = \begin{bmatrix} O_{2\times2} \\ M(\Theta)^{-1} \\ O_{1\times2} \end{bmatrix}, \quad \text{and} \quad \boldsymbol{x}_s = \begin{bmatrix} \Theta \\ \dot{\Theta} \\ \theta_3 \end{bmatrix}.$$

By the extended linearization method, we obtained the state equation of the linear parameter varying system whose coefficient matrices include the nonlinear terms.

3. Final-state control[10]

3·1 Formulation of augmented system with constraint of inputs

Hyperbolic tangent function

$$\boldsymbol{u} = f(\boldsymbol{v}) = \begin{bmatrix} \alpha_1 \tanh(v_1/\alpha_1) \\ \alpha_2 \tanh(v_2/\alpha_2) \\ \vdots \\ \alpha_m \tanh(v_m/\alpha_m) \end{bmatrix} \qquad (12)$$

is introduced to give constraint of control input amplitude, where $\boldsymbol{v}$ is $m \times 1$ vector.

Differentiation of Eq. (12) with respect to time yields

$$\frac{df(\boldsymbol{v})}{dt} = \frac{df(\boldsymbol{v})}{d\boldsymbol{v}} \frac{d\boldsymbol{v}}{dt}, \qquad (13)$$

where

$$\frac{df(\boldsymbol{v})}{d\boldsymbol{v}} = \mathrm{diag}\left[\frac{df(v_1)}{dv_1}\ \frac{df(v_2)}{dv_2}\ \cdots\ \frac{df(v_m)}{dv_m}\right],$$

and

$$\frac{d\boldsymbol{v}}{dt} = \left[\frac{dv_1}{dt}\ \frac{dv_2}{dt}\ \cdots\ \frac{dv_m}{dt}\right]^T.$$

If $df(\boldsymbol{v})/dt$ passes an integrator, $\boldsymbol{u} = f(\boldsymbol{v})$ are not possible to exceed $\pm\alpha_i (i = 1, 2, \cdots, m)$ with any arbitrary value of $d\boldsymbol{v}/dt$ because of the property of the hyperbolic tangent function. The block diagram is shown in Fig. 2.

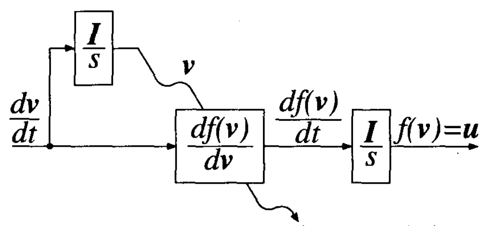


Fig.2 Block diagram of equation (13)

The system of Fig. 2 can be written as Eq. (14)

$$\dot{\boldsymbol{u}} = B_l(\boldsymbol{v})\dot{\boldsymbol{v}} \qquad (14)$$

where

$$B_l(\boldsymbol{v}) = \mathrm{diag}\left[\frac{df(v_1)}{dv_1}\ \frac{df(v_2)}{dv_2}\ \cdots\ \frac{df(v_m)}{dv_m}\right].$$

By using Eqs. (11) and (14), the augmented system is obtained as follows:

$$\dot{\boldsymbol{x}} = A_c(\boldsymbol{x})\boldsymbol{x} + B_c(\boldsymbol{x})\dot{\boldsymbol{v}} \qquad (15)$$

where

$$A_c(\boldsymbol{x}) = \begin{bmatrix} A_s(\boldsymbol{x}_s) & B_s(\boldsymbol{x}_s) \\ \boldsymbol{O}_{m\times n} & \boldsymbol{O}_{m\times m} \end{bmatrix},$$

$$B_c(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{O}_{n\times m} \\ B_l(\boldsymbol{v}) \end{bmatrix}, \quad \text{and} \quad \boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_s \\ \boldsymbol{u} \end{bmatrix}.$$

The augmented system, whose control inputs $\boldsymbol{u}$ do not exceed the ranges of $\pm\alpha_i$, and can have the value of zero at terminal points of the initial time and the final time, can be formulated.

3·2 Final-state control with error learning

Padé approximation method is used on the assumption of zero-order holding of input. Eq. (15) can be discretized in every sampling period Δt as Eq. (16).

$$\boldsymbol{x}(k+1) = A(k)\boldsymbol{x}(k) + B(k)\boldsymbol{v}_d(k) \qquad (16)$$
$$(k = 0, 1, \cdots, N)$$

where

$$A(k) = A_c(\boldsymbol{x}(k))\left(\boldsymbol{I} - \frac{\Delta t}{2}A_c(\boldsymbol{x}(k))\right)^{-1}\Delta t + \boldsymbol{I},$$

and

$$B(k) = B_c(\boldsymbol{x}(k))\left(\boldsymbol{I} - \frac{\Delta t}{2}A_c(\boldsymbol{x}(k))\right)^{-1}\Delta t.$$

Then, we introduce the method to obtain the feedforward inputs V_{nl} that can bring the system from the initial state $\boldsymbol{x}(0)$ to the final desired state $\boldsymbol{x}^o$ in a terminal time. The final state of the system $\boldsymbol{x}(N)$ that is brought by suitable feedforward inputs $\boldsymbol{v}_d(k)(k = 0, 1, \cdots, N)$ from the initial state $\boldsymbol{x}(0)$ at an objective time $N\Delta t$ can be written as follows:

$$\boldsymbol{x}(N) = A(N-1)A(N-2)\cdots A(0)\boldsymbol{x}(0) + U_{nl}V_{nl}, \quad (17)$$

where

$$U_{nl} = [A(N-1)\cdots A(1)B(0), \cdots, A(N-1)B(N-2), B(N-1)],$$

and

$$V_{nl} = \mathrm{col}(\boldsymbol{v}_d(0), \boldsymbol{v}_d(1), \cdots, \boldsymbol{v}_d(N-1)). \quad (18)$$

If the system reachability matrix U_{nl} is invariant, the feedforward input V_{nl} to reach the final desired state $\boldsymbol{x}^o$ after a control time $N\Delta t$ by the final-state control [11] is obtained as follows:

$$V_{nl} = U_{nl}^T(U_{nl}U_{nl}^T)^{-1}(\boldsymbol{x}^o - A(N-1)\cdots A(0)\boldsymbol{x}(0)). \quad (19)$$

However, because the reachability matrix U_{nl} changes with input V_{nl}, the actual final state $\boldsymbol{x}(N)$ can not reach the desired state $\boldsymbol{x}^o$ at an objective time $N\Delta t$ and the final error causes.

The input ΔV to compensate the final error vector $\boldsymbol{e} = \boldsymbol{x}^o - \boldsymbol{x}(N)$ is given as follows:

$$\Delta V = U_{nl}^T(U_{nl}U_{nl}^T)^{-1}\boldsymbol{e}. \quad (20)$$

To keep the change of U_{nl} small, and to improve convergent property of the final error by multiplying learning coefficient $\gamma(0 < \gamma \leq 1)$ by ΔV the input V_{nl} is renewed as follows:

$$V_{nl} + \gamma\Delta V \rightarrow V_{nl}. \quad (21)$$

After repeating the procedure until the sum of squares of final error $\boldsymbol{e}$ is small enough, the feedforward input V_{nl} will be obtained to bring the system to the desired state $\boldsymbol{x}^o$.

4. Simulation

4·1 Learning condition

Ref. [4] defined the twisting rate R_t of the two-link cat robot as follows:

$$R_t = \frac{\dot{\theta}_r + \dot{\theta}_3}{\dot{\theta}_3}. \quad (22)$$

Using nonholonomic constraint condition Eq. (2), we can rewrite Eq. (22) as follows:

$$R_t = 1 - \frac{\cos\theta}{1 - \left(\frac{1}{2} - \frac{2}{3}\left(\frac{l}{r}\right)^2\right)\sin^2\theta}, \quad (23)$$

where

$$\cos\theta = \cos\theta_1\cos\theta_2. \quad (24)$$

According to the analysis result of Ref. [4], to realize the same twisting rate, a fat cat needs to make her bending angle θ larger than a slim cat.

In experiments of Ref. [5], bending angle θ was assumed to be kept invariant, and twisting rate of the cat robot was given as $0.5(R_t = 0.5)$. Although the swing motion of the waist was controlled by a sinusoidal input, and the cat robot could change the posture [5], the final state was not clear. In this paper, we do not keep the bending angle θ invariant, and we aim at making the final error that includes final angular velocities small enough. The specifications of dynamical model in simulations are the same as those of the experimental setup of Ref. [5]. They are shown in Table 1.

Table 1 Specifications of a slim cat robot

Length of link (m)	L_1	0.45
	L_2	0.45
Center of gravity (m)	l_1	0.225
	l_2	0.225
Radius of link (m)	r_1	0.06
	r_2	0.06
Mass of link (kg)	m_1	2.2
	m_2	2.2
Inertia of link (kg·m^2)	J_1	0.0391
	J_2	0.00396
	J_3	0.0391
	J_4	0.00396
Joint damping (N·m·s/rad)	d_1	0.1
	d_2	0.1

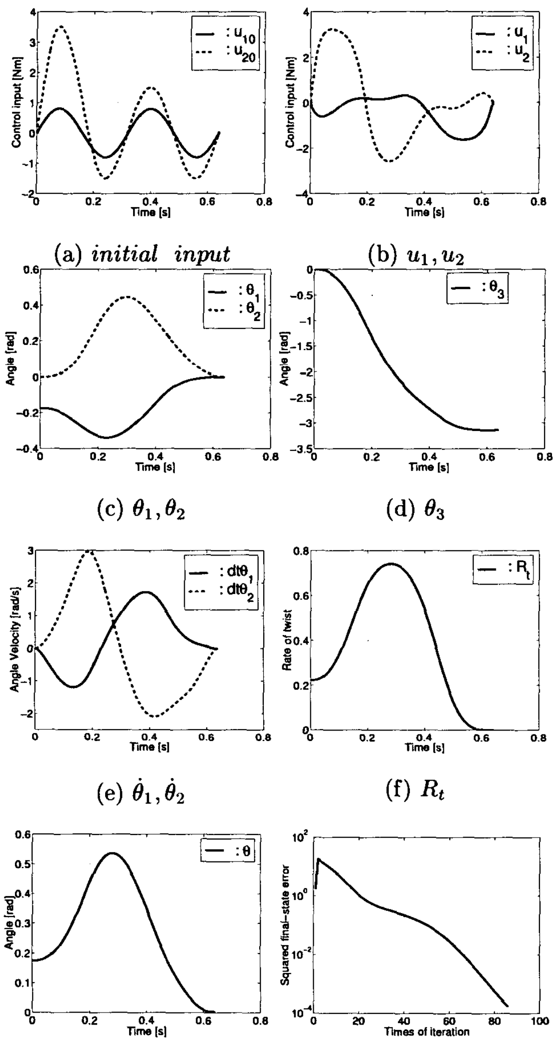
By referring to the photograph of the falling cat reported in Ref. [4], we can separate the twisting motion of a falling cat into two steps. The first step is that the cat turns θ_3 from 0° to −180°. At this point, the cat is almost in straight posture ($\theta = 0$). From the posture to the state that the cat bends her waist so that she can land on her feet is the second step. In this paper, because the motion of the second step is in the plane that lies at right angle at U-axis, it is considered that the motion is easily realized only by the torque input around U-axis.

Table 2 Learning condition

Sampling time	$\Delta t = 0.002$s
Control time	$T = 0.64$s
Coefficient of error learning	$\gamma = 0.1$
Initial state	$\boldsymbol{x}(0) = [-10^\circ\ 0\ 0\ 0\ 0\ 0\ 0]$
Final desired state	$\boldsymbol{x}^o$=[-0.001° 0 0 0 0 0 -180°]
Limit of control input	$\alpha_1 = 2$N·m $\quad \alpha_2 = 4$N·m
Tolerance error	2×10^{-4}

So we only pay attention to the first step. If the final states of θ_1 and θ_2 are equal to zero degree, and the denominator of Eq. (2) is equal to zero, it is impossible to calculate error learning. So the final state of $\theta_1 = -0.001°$ is given. The condition of learning is shown in Table 2.

4·2 Simulation of twisting motion of a slim cat



(a) *initial input* (b) u_1, u_2

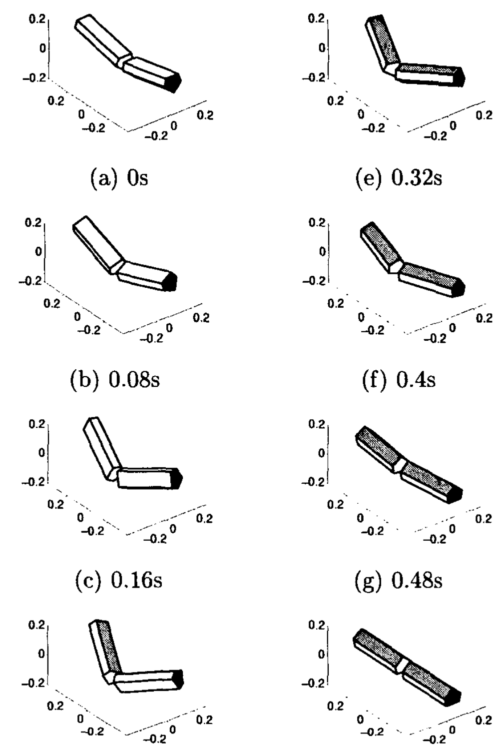
(c) θ_1, θ_2 (d) θ_3

(e) $\dot{\theta}_1, \dot{\theta}_2$ (f) R_t

(g) θ (h) *error*

Fig.3 Simulation for a slim cat

The result of simulation is shown in Fig. 3. The sinusoidal initial inputs u_{10}, u_{20} with two periods in control time 0.64s are given in Fig. 3(a). Particularly, it is necessary for u_{20} to be large in about first 0.2s. After 86 times of learning, torque inputs are obtained so that the squared final-state error converges to 2×10^{-4}, and the constraint of input amplitude is satisfied. From Fig. 3(c), (d), and (e), it is seen that $\theta_1 \cong 0$, $\theta_2 \cong 0$, $\theta_3 \cong -180°$, $\dot{\theta}_1 \cong 0$, $\dot{\theta}_2 \cong 0$ at the final time. The final error of θ_3 is about -0.5 degree. Furthermore, the change of the twisting rate of the cat is shown in Fig. 3(f). The maximum value of the bending angle is about 30 degree from Fig. 3(g). The animation of a falling cat every 0.08s is shown in Fig. 4. Although the shape of robot is cylinder actually, in order to distinguish the back from the belly, the robot is shown as pentagonal columns, and the back is hatched. From Fig. 4(d), we can see that the cat robot is bending backward. This is one of the reasons why the cat can realize the twisting motion.



(a) 0s (e) 0.32s

(b) 0.08s (f) 0.4s

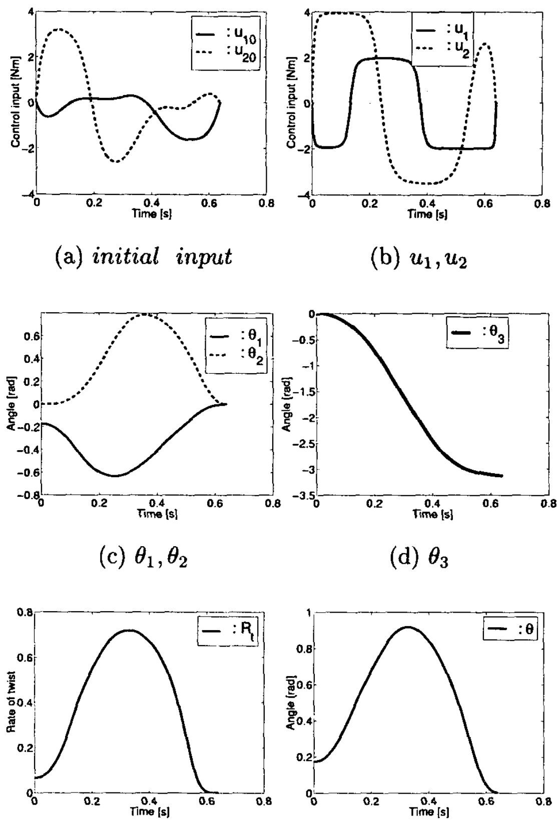
(c) 0.16s (g) 0.48s

(d) 0.24s (h) 0.56s

Fig.4 Animation of a slim cat

4·3 Simulation of twisting motion of a fat cat

In this section, the result of simulation of a fat cat whose $r_1 = r_2 = 0.12$m is shown. Initial inputs u_{10},u_{20} used are the inputs obtained for a slim cat. Fig. 5(b) shows the obtained torque inputs for the fat cat. From Fig. 5(b) it is seen that the constraint of input amplitudes are satisfied. Because larger torque inputs are necessary for a fat cat than a slim cat, the inputs are saturated. From Fig. 5(c) and (d), it is seen that $\theta_1 \cong 0$, $\theta_2 \cong 0$, $\theta_3 \cong -180°$ at the final time. The final error of θ_3 is about -1.1 degree. Furthermore, the final angular velocity of θ_3 is almost 0 rad/s. The change of the twisting rate of the cat is shown in Fig. 5(e). The maximum value of the bending angle is about 53° from Fig. 5(f). To secure the same twisting rate of the cat, the larger bending angle θ is necessary than a slim cat. The animation of a falling cat every 0.08s is shown in Fig. 6.

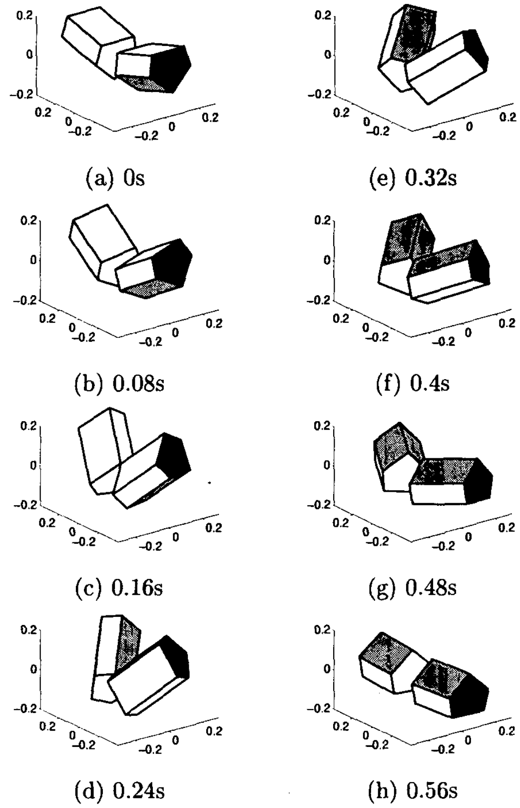


(a) *initial input* (b) u_1, u_2

(c) θ_1, θ_2 (d) θ_3

(e) R_t (f) θ

Fig.5 Simulation for a fat cat



(a) 0s (e) 0.32s

(b) 0.08s (f) 0.4s

(c) 0.16s (g) 0.48s

(d) 0.24s (h) 0.56s

Fig.6 Animation of a fat cat

5. Conclusions

By using nonholonomic constraint condition and Lagrange's function, we derived the state equation for expression of twisting motion of a falling cat strictly with torque inputs around waist. It was verified that although suitable initial inputs must be chosen, by the final-state control the torque inputs considering the input limit could be obtained to control the twisting motion of the cat robot so that the final angular velocities were small enough.

References

[1] Brockett, R.W., "Asymptotic Stability and Feedback Stabilization", *Differential Geometric Control Theory*, Birkhauser, Boston, Mass., pp.181-191, 1983.

[2] Guichardet, A., "On Rotation and Vibration Motion of Molecules", *Ann. Inst. H. Poincaré, Phys. Théor.*, Vol.40, No.3, pp.329-342, 1984.

[3] Montgomery, R., "Isoholonomic problems and applications", *Comm. Math. Phys.*, Vol.128, No.3, pp.565-592, 1990.

[4] Yamafuji, K., Kobayashi, T., Kawamura, T., "Elucidation of Twisting Motion of a Falling Cat and its Realization by a Robot"(in Japanese), *Journal of the Robotics Society of Japan*, Vol.10, No.5, pp.648-655, 1992.

[5] Kawamura, T., "Elucidation of Twisting Motion of a Falling Cat and its Realization by a Robot"(in Japanese), *Doctor thesis of the University of electro-communications*, 1992.

[6] Nakagawa, T., Sampei, M., Kiyota, H., "Analysis and Control of The Falling Cat Phenomenon through Simulation"(in Japanese), *Proceedings of Symposium on Sports Engineering*, JSME No.95-45, pp.204-207, 1995.

[7] Suzuki, H., Yoshida, K., "Development of Ski Robot Performance a Somersault and Twist"(in Japanese), *Proceedings of Dynamics and Design Conference '99*, JSME No.99-7, Vol.A, pp.263-266, 1999.

[8] Bloch, A., M., "Nonholonomic Mechanical System with Symmetry", *Arch. Rational Mech. Anal.*, Vol.136, No.1, pp.21-99, 1996.

[9] Friedland, B., "Advanced Control System Design", *Prentice Hall*, pp.110-112, 1996.

[10] Nishimura, H., "Motion Control of Inverted Pendulum via Final-state Control", *Proceedings of Pioneering International Symposium on MOVIC in Mechatronics*, pp.128-132, April 6-7, 1999, Tokyo.

[11] Totani, T., Nishimura, H., "Final-State Control Using Compensation Input", *Trans. of the Society of Instrument and Control Engineers*, Vol.30, No.3, pp.253-260, 1994.