Proceedings of the 2016 IEEE
International Conference on Robotics and Biomimetics
Qingdao, China, December 3-7, 2016

Research on Trajectory Planning of a Robot Inspired by Free-Falling Cat Based on Numerical Approximation*

Xingcan Liang, Linsen Xu, Lu Li, Wei Yu

Abstract— **A robot may be at the risk of falling from a high place when it works in an unknown complex environment, so the attitude control ability of the robot in the air should be considered to reduce the damage caused by the wrong dropping posture. When a cat drops in the upside-down posture with zero angular momentum, it can always right itself and land on its feet safely. Inspired by this biological phenomenon, we studied the optimal falling trajectory for adjusting the attitude of the falling cat robot. Firstly, we formulated a mathematical model based on the twisted model of two identical axially symmetric rigid cylinders, owing to nonintegrable conservation laws, a problem for controlling the attitude of the falling cat robot could be transformed to the problem of the nonholonomic motion planning. Then, taking the energy cost of the joint of the system as the optimal control function to find a control torque input, which was approximated by the spline approximation, with the method one can solve the problem that the initial and terminal values of control torque input are not zero. We found the optimal solution of control function through particle swarm optimization. Finally, we carried out numerical simulation, and the result shows the spline approximation is a valid method for trajectory planning of the falling cat robot.**

Keywords: **optimal trajectory; spline approximation; falling cat robot; particle swarm optimization**

I. Introduction

In modern times, the research and application of the robot have changed from job assignment in structured environment to the field of autonomous task in unstructured environment, such as interplanetary exploration, geological exploration, disaster assistance, counter-terrorism and anti-riot. Legged robots, relative to the wheeled and tracked robots, have an advantage in the unstructured environment, but there are still many restrictions on legged robots under a certain motion environment [1]. In order to expand the application area of legged robot, e.g. the landing of outer space probe, we must dispose of the safe landing of the legged robot when it drops from upper-air unintentionally [2]. Therefore, it is significant to study a robot that can adjust its attitude in the air through its own mechanical and control systems to make a safe landing come true.

As is known to us all, a cat can right itself and land on its feet safely when it drops from upside-down with zero angular momentum. Since there are no extra forces that tend to rotate the cat's body, the self-righting of the falling cat seems to violate the law of angular momentum conservation. Actually, that's not the truth. To interpret the amusing biological phenomenon, some attempts have been done at the end of 19^{th} century. Ge et al. introduced the history of the explanation of this phenomenon [3]. Many researchers proposed various assumptions to explain how a cat rotates its body in the air and proved that the angular momentum of a free falling cat is conserved, historically [4-6]. Among their theory, the most widely recognized one was proposed by Kane and Scher [6]. As for their explanation, they took a cat as the model, which to take the physiological constrains of the biological system into consideration, consists of two identical, axially symmetric rigid bodies connected by an unusual style of non-twistable joint, to explain a cat's rotation movement. Based on the dynamic Kane-Scher Model (KSM), a lot of researchers focused on the motion planning and optimization [3, 7-9].

Montgomery [10] proved the correction of the KSM by differential geometry. He believed that the problem of a falling cat is the same as the isoholonomic problem if and only if the total angular momentum through the turning motion is zero. In view of this, the dynamics of a falling cat problem becomes a prototypical example of nonholonomic system [11]. Nonholonomicity, which is characterized by the dimension of generalized coordinate is larger than that of control input, results from the cat subjected to nonintegrable conservation laws [12-13]. Nonholonomic system is an unusual kind of nonlinear system. Controlling the attitude and body orientation of a falling cat poses a big challenge because the number of degree of freedom is greater than total independent control variables, and there is no unified control theory that can deal with general, non-holonomic systems. Iwai et al. [14] improved the KSM. They released a degree of freedom by removing the no-twisted restriction and formulated a Twisted Kane-Scher Model (TKSM) of the falling cat. Using path integral reinforcement learning, Nakano el al. [15] studied the trajectory planning of attitude adjustment of the TKSM. They obtained the optimal trajectory of attitude angle with three different initial states, but the results show that this method is not good by virtue of the overshoot.

Despite the development of a number of techniques for planning the trajectories of attitude adjustment, no robotic systems have been built based on studies of free-falling cats.

* Research are supported by National Natural Science Foundation of China under Grant (No. 51405470) and Basic Applied Research of Changzhou City (No. CJ20159004).

Xingcan Liang is with the Department of Automation, University of Science and Technology of China, Hefei, Anhui Province 230026 China (phone: 13098203127; Email: lxcan@mail.ustc.edu.cn).

Linsen Xu and Lu Li are with the Department of Precision Manufacturing, Institute of Advanced Manufacturing Technology, Hefei Institute of Physical Science, CAS, Changzhou, Jiangsu Province 213164 China (e-mail: {lsxu & lli}@iamt.ac.cn).

Wei Yu is with the Department of Mechanical Engineering, Georgia Southern University, Statesboro, Georgia 30458 America (e-mail: wyu@georgiasouthern.edu).

The aim of this paper is to study the optimal trajectory of a falling cat robot, then using the trajectory planning to produce a robot, which can adjust its attitude in the air through its own mechanical and control systems, like a real falling cat, to land on its feet safely. In Ref. [3-10], the dimension of control inputs is not that of torque but of angular velocity, however, the common use one in robot control is control torque input. From Ref. [14], we know that if one add a “twisted-yes” condition to the classical KSM, the model will be more in line with falling cats in biology. The result in Ref. [15] cannot be used in engineering practice appropriately due to overshot.

In order to solve the issue above, we study the application of numerical approximation method—spline approximation—to obtain optimal trajectories of a falling cat robot based on TKSM. The article is organized as follows: In Section II, we consider a twisted model of the falling cat robot which is composed of two identical axial symmetric rigid bodies. We formulate the mathematical model, and in the case of the angular momentum of the falling robot system is zero, the problem for controlling the attitude of a robot will be transformed to the problem of the nonholonomic motion planning (NMP). In Section III, we take the consumed energy of each rotation joint of robot as the optimal control target function to find a control torque input. Using the control input parameterization, we convert the infinite dimensional optimal control problem to a finite one, in this step, the spline approximation is discussed to approximate a solution of an infinite-dimensional optimization problem. In Section IV, the particle swarm optimization (PSO) is proposed to find a solution to minimize the cost function discussed in Section III. In Section V, we carry out the numerical simulation based on the spline approximation. The results show that the method is a valid one for the trajectory planning of falling cat robots.

II. DYNAMICAL MODEL OF THE FALLING CAT ROBOT

During the dropping of cat, its spine bends in all directions sequentially, the result is that while the front and rear bodies of a cat finish one round movement, respectively, the entirety body of the cat turns a somersault in the opposite direction [6]. Inspired by this biological phenomenon, we consider the dynamical model of the falling cat robot consists of two symmetric rigid bodies, front body B_1 and rear body B_2, As Fig. 1 shows. They are torsion free when moving relative to each other at their connection point O. To better understand the whole model, we define the points, axes, and variables as follows:

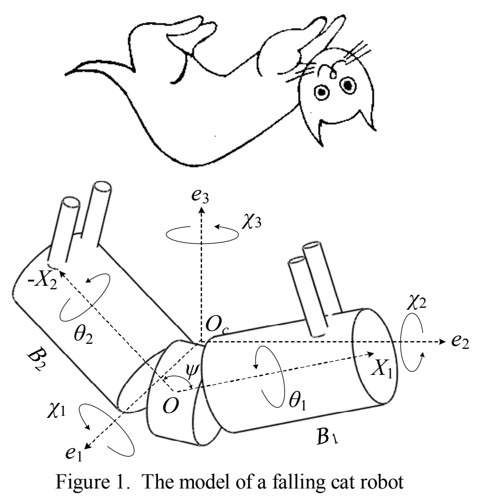

Figure 1. The model of a falling cat robot

O	Joint point of front body B_1 and rear body B_2
O_c	Centroid point of the entirety robot
OX_1	Centroid axis pointing from O to the head
OX_2	Centroid axis pointing from the tail to O
$\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$	Standard orthogonal base in $\boldsymbol{R}^3$
ψ	Bending angle of robot
θ_1, θ_2	Rotation angle of B_1 and B_2 around OX_1 and OX_2
χ_1, χ_2, χ_3	Rotation angle of the entire model around $O_c e_1, O_c e_2, O_c e_3$

We assume here that when a cat bends its spine, it not only allows the spine to bend in arbitrary plane, but also there exists torsion between B_1 and B_2, i.e. θ_1 and θ_2 can be unsynchronized. Due to without any external force (gravity and air resistance are not considered temporarily), centroid of the system is fixed during the cat robot’s falling. According to the law of angular momentum conservation, the total angular momentum is always 0 throughout the robot’s falling when initial angular momentum is 0, this restriction belongs to nonholonomic constraint. Thus, the system of robot is a nonholonomic system under above constraint. Moreover, we may take the ranges of attitude angle $(\psi, \theta_1, \theta_2)$ as

$$0 < \psi < \pi,\ 0 < \theta_1 < 2\pi,\ 0 < \theta_2 < 2\pi \tag{1}$$

Euler angle (χ_1, χ_2, χ_3), which is taken in a different way from that for the common Euler angle, should satisfies the constraint as follows:

$$0 < \chi_1 < 2\pi,\ 0 < \chi_2 < 2\pi,\ -\pi/2 < \chi_3 < \pi/2 \tag{2}$$

It is to be notice here that two rigid bodies are axially symmetric, therefore, we can put the mass and the inertia tensor of each rigid body in the form

$$m = m_1 = m_2 \tag{3}$$

$$\boldsymbol{A}_1 = \boldsymbol{A}_2 = diag(I_1, I_1, \nu I_1),\quad \nu = I_3 / I_1 \tag{4}$$

In order to derive the state equation of the system, we assume $\phi_1 = (\theta_1 + \theta_2)/2$, $\phi_2 = (\theta_1 - \theta_2)/2$, $\lambda = ml^2 / I_1$,here, l stands for the distance between the joint O and the centroid of each rigid body. let $\boldsymbol{x} = [\psi, \theta_1, \theta_2, \chi_1, \chi_2, \chi_3, \dot{\psi}, \dot{\phi}_1, \dot{\phi}_2]^{\mathrm{T}}$ be the state variable, $\boldsymbol{u} = [u_1, u_2, u_3]^{\mathrm{T}}$ the control torque input, with which one can change the attitude of the system arbitrarily, we note here that u_1 denotes the torque for changing the angle ψ , u_2 and u_3 for that for the angles θ_1 and θ_2 , respectively. Then, the state equation of the falling cat robot system can be expressed as [15]:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\psi\\ \theta_1\\ \theta_2\\ \chi_1\\ \chi_2\\ \chi_3\\ \dot{\psi}\\ \dot{\phi}_1\\ \dot{\phi}_2\end{bmatrix}=\begin{bmatrix}\dot{\psi}\\ \dot{\phi}_1+\dot{\phi}_2\\ \dot{\phi}_1-\dot{\phi}_2\\ f_1\\ f_2\\ f_3\\ f_4\\ f_5\\ f_6\end{bmatrix}+\begin{bmatrix}0\\0\\0\\0\\0\\0\\f_7\\0\\0\end{bmatrix}u_1+\begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\f_8\\f_9\end{bmatrix}u_2+\begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\f_8\\-f_9\end{bmatrix}u_3 \tag{5}$$

where

$$f_1=\nu\tan\chi_3\left(\frac{\sin\frac{\psi}{2}\cos\chi_1}{A}\dot{\phi}_1-\frac{\cos\frac{\psi}{2}\sin\chi_1}{B}\dot{\phi}_2\right)$$

$$f_2=\frac{\nu}{\cos\chi_3}\left(-\frac{\sin\frac{\psi}{2}\cos\chi_1}{A}\dot{\phi}_1+\frac{\cos\frac{\psi}{2}\sin\chi_1}{B}\dot{\phi}_2\right)$$

$$f_3=-\nu\left(\frac{\sin\frac{\psi}{2}\sin\chi_1}{A}\dot{\phi}_1+\frac{\cos\frac{\psi}{2}\cos\chi_1}{B}\dot{\phi}_2\right)$$

$$f_4=\frac{2\sin\frac{\psi}{2}\cos\frac{\psi}{2}}{1+\lambda\cos^2\frac{\psi}{2}}\left(\frac{\lambda}{4}\dot{\psi}^2-\frac{\nu^2}{A^2}\dot{\phi}_1^2+\frac{(1+\lambda)\nu^2}{B^2}\dot{\phi}_2^2\right)$$

$$f_5=\frac{\nu\sin\frac{\psi}{2}}{A\cos\frac{\psi}{2}}\dot{\psi}\dot{\phi}_1 \qquad f_6=-\frac{\nu\cos\frac{\psi}{2}}{B\sin\frac{\psi}{2}}\dot{\psi}\dot{\phi}_2$$

$$f_7=\frac{2}{I_1(1+\lambda\cos^2\frac{\psi}{2})} \qquad f_8=\frac{A}{2\nu I_1\cos^2\frac{\psi}{2}}$$

$$f_9=\frac{B}{2\nu I_1(1+\lambda)\sin^2\frac{\psi}{2}}$$

$$A=\cos^2\tfrac{\psi}{2}+\nu\sin^2\tfrac{\psi}{2} \qquad B=(1+\lambda)\sin^2\tfrac{\psi}{2}+\nu\cos^2\tfrac{\psi}{2}$$

As can be easily seen from the equation above, the dynamic equation of the system is strong coupling and highly nonlinear. We note here that in (5), the dimension of state variable $n=9$ is larger than that of torque input $m=3$, moreover, the rotational motion of the robot can be depicted by the nonlinear control system with nine state variables and three torque inputs during its falling, hence, the problem of attitude control of the robot is equivalent to the problem of NMP, i.e. given two configurations $\boldsymbol{x}_0$, $\boldsymbol{x}_\mathrm{f}\in\boldsymbol{R}^9$, find control torque input $\boldsymbol{u}(t)\in\boldsymbol{R}^3$, $t\in[0,T]$, of minimal cost, to drive the system (5) from $\boldsymbol{x}_0$ to $\boldsymbol{x}_\mathrm{f}$ at time T.

III. Cost Function Based on Spline Approximation

In order to reduce the energy consumption when a robot adjusts its attitude in the air, in terms of the minimal energy law, we choose the consumed energy of each rotation joint of the robot to minimize the cost function

$$J(\boldsymbol{u})=\int_0^T<\boldsymbol{u},\boldsymbol{u}>\mathrm{d}t \tag{6}$$

Suppose the system is controllable [13], therefore, there be some certain solution $\hat{\boldsymbol{u}}\in\boldsymbol{L}_2([0,T])$ for the minimization of cost function (6). Here, $\boldsymbol{L}_2([0,T])$ denotes the Hilbert space of measurable vector valued functions of the form $\boldsymbol{u}(t)$, $t\in(0,T)$. Therefore, we can rephrase the NMP problem as follows: given a NMP system of the form (5), find control torque input $\boldsymbol{u}(t)\in\boldsymbol{R}^3$ of minimum cost function (6) such that $\boldsymbol{u}(t)$ can drive the system (5) from $\boldsymbol{x}_0$ to $\boldsymbol{x}_\mathrm{f}$ at time T. Certainly, obtaining the exact solution of nonlinear optimization problem in an infinite dimensional space poses considerable computational difficulties.

To avoid solving the infinite dimensional problem, general method for solving $\boldsymbol{u}$ is using the control input parameterization method based on the Fourier basis to restrict the torque input to a finite dimensional space, however, there is a problem when we use it to approximate $\boldsymbol{u}$, i.e. the initial and final values of the control input obtained are not zero, which brings some difficulties for designing actuators in the engineering practice. For solving the above problem, we use the spline approximation as our numerical approximation method to approximate the control torque input $\boldsymbol{u}$.

A. *Spline approximation*

A smooth interpolation curve can be obtained by the spline function, which consists of polynomial pieces on subintervals joined together with certain continuity conditions. Without loss of generality, we assume that $n+1$ knots $t_0,t_1,\cdots,t_n$ have been specified and satisfy $t_0<t_1<\cdots<t_n$, a spline function of degree k $(k\ge 0)$ having knots $t_0,t_1,\cdots,t_n$ is a function S such that

1) On each interval $[t_{i-1},t_i]$, $i=1,2,\cdots,n$, S is a polynomial of degree $\le k$, and interpolate the value y_i at the point t_i,

2) S has a continuous $(k-1)$ st derivative on $[t_0,t_n]$.

Hence, S is a continuous piecewise polynomial of degree at most k having continuous derivatives of all orders up to $k-1$.

As the cubic spline ($k=3$), compared to other splines, with high precision and sufficient flexibility, it is a vital spline and often used in practice [16]. Therefore, we choose the cubic spline to approximate the control torque input in our experiment. There are $4n$ coefficients in the piecewise cubic polynomial, since there are 4 coefficients in each of the n cubic polynomials. On each subinterval $[t_i,t_{i+1}]$, there are 2 interpolation conditions, $S(t_i)=y_i$ and $S(t_{i+1})=y_{i+1}$, giving $2n$ conditions. The continuity of S' and S'' gives one condition at each interior knot, respectively, i.e. $S'_{i-1}(t_i)=S'_i(t_i)$ and $S''_{i-1}(t_i)=S''_i(t_i)$, accounting for $2(n-1)$ additional conditions. Therefore, there are $4n-2$ conditions in number for determining $4n$ coefficients. Two degrees of freedom remain, and there are various ways of using them to advantage. Here, given 2 conditions $S''(t_0)=0$ and $S''(t_n)=0$, with which we can obtain the smoothest possible interpolating function—natural cubic spline function.

B. Cost function

In the practical computation, given a collection of knots $0 = t_0 < t_1 < \cdots < t_N = T$, and the value of control input at those knots $\boldsymbol{\zeta} = [u_0, u_1, \cdots, u_N]^{\mathrm{T}}$, where N is a limited constant. Based on the natural cubic spline approximation, the control torque input can be written out as

$$\boldsymbol{u}(t) = S(\boldsymbol{\zeta}, t), \qquad t \in [0, T] \tag{7}$$

Applying the theory of penalty function, together with using (7), the cost function (6) is approximated as

$$J(\boldsymbol{\zeta}, \tau) = \int_0^T [S(\boldsymbol{\zeta}, t)]^2 \mathrm{d}t + \tau \parallel \boldsymbol{x}(T) - \boldsymbol{x}_{\mathrm{f}} \parallel^2 \tag{8}$$

Here, τ is a penalty factor and $\boldsymbol{x}(T) \in \boldsymbol{R}^9$ is the solution of system (5) at time $t = T$. Apparently, $\boldsymbol{x}(T)$ is a function of $\boldsymbol{\zeta} \in \boldsymbol{R}^N$, therefore, when τ is known, we can define $f(\boldsymbol{\zeta}) = \boldsymbol{x}(T)$, and then (8) can be rewritten to a function of $\boldsymbol{\zeta} \in \boldsymbol{R}^N$ as

$$J(\boldsymbol{\zeta}) = \int_0^T [S(\boldsymbol{\zeta}, t)]^2 \mathrm{d}t + \tau \parallel f(\boldsymbol{\zeta}) - \boldsymbol{x}_{\mathrm{f}} \parallel^2 \tag{9}$$

Ref. [12] shows that as $N, \tau \to \infty$, solutions obtained by solving (9) tend to solutions of (6). Our problem now is to find a suitable $\boldsymbol{\zeta} \in \boldsymbol{R}^N$ to minimize the cost function (9).

IV. PARTICLE SWARM OPTIMIZATION

Compared with the genetic algorithm, PSO algorithm does not use crossover and mutation, and it has a small number of parameters that need to be fixed, in addition, we need not any gradient message of the cost function when one use PSO to find the solution of (9), therefore, PSO algorithm will be discussed to find $\boldsymbol{\zeta}$ to minimize the cost function.

A. Theoretical basis of particle swarm optimization

The PSO algorithm, which is a member of swarm intelligence optimization algorithm, was first proposed by Kennedy and Eberhart in 1995 [17]. It originates in the research of a flock of birds collectively foraging for food. In PSO algorithm, a number of simple entities—the particles—are placed in the search space of some problem or function, the motion characteristic of each particle is represent by the current position (denoting the potential solution of objective function), the current velocity (determining the direction and distance for flying), and the fitness value (obtained by evaluating the objective function at its current location). PSO algorithm first initializes a group of particles in the solution space, and then the particles follow the best particle to search in the solution space. When a particle adjusts its trajectory according to its own flying experience and the flying experience of other particle in the search space, it can update its positon through individual best $\boldsymbol{P}_i$, which is a best position than any it has found by the particle i previously, and global best $\boldsymbol{P}_g$, which is the best position found so far by the whole population. We suppose that in the D dimensional search space, and the size of population is M, let $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \cdots, X_{iD})^{\mathrm{T}}$ be the current position of the target particle's index i, $\boldsymbol{V}_i = (V_{i1}, V_{i2}, \cdots V_{iD})^{\mathrm{T}}$ the current velocity, $\boldsymbol{P}_i = (P_{i1}, P_{i2}, \cdots, P_{iD})^{\mathrm{T}}$ the individual best, $\boldsymbol{P}_g = (P_{g1}, P_{g2}, \cdots, P_{gD})^{\mathrm{T}}$ the global best. The particle $i (i = 1, 2, \cdots, M)$ updates its positon and velocity by $\boldsymbol{P}_i$ and $\boldsymbol{P}_g$ in each iteration as follows [18]:

$$\boldsymbol{V}_i = \omega \boldsymbol{V}_i + c_1 r_1 (\boldsymbol{P}_i - \boldsymbol{X}_i) + c_2 r_2 (\boldsymbol{P}_g - \boldsymbol{X}_i) \tag{10}$$

$$\boldsymbol{X}_i = \boldsymbol{X}_i + \boldsymbol{V}_i \tag{11}$$

where ω is termed the inertia weight, c_1 and c_2 positive acceleration coefficients, r_1 and r_2 generate a uniform random number in the range $[0,1]$, respectively.

B. Optimization procedure

One can use the PSO algorithm to solve the unconstrained optimization problem (9), the specific steps are as follows:

Step 0:

① Set up initial and final state $\boldsymbol{x}_0, \boldsymbol{x}_{\mathrm{f}} \in \boldsymbol{R}^9$.

② Assign an appropriate number $\mathrm{c}_1, \mathrm{c}_2, \tau, \omega$,the size of population M ,the dimension of the position D , The maximum number of iterations C_t. Initialize a population array of particles with random positions and velocities.

③ Evaluate the fitness value of each particle using (9),and then the current position with respect to it is stored in the vector $\boldsymbol{P}_i$ ($i = 1, 2, \cdots, M$), the smallest of $\boldsymbol{P}_i$ is stored in the vector $\boldsymbol{P}_g$.

Step 1:

① Change the positon and velocity, according to (10) together with (11).

② For each particle, compared its new fitness value with that at $\boldsymbol{P}_i$, if it is smaller, set $\boldsymbol{P}_i$ equal to the current position.

③ Compared the fitness value at current $\boldsymbol{P}_i$ and $\boldsymbol{P}_g$, choose the position of smallest one as a new $\boldsymbol{P}_g$.

④ Repeat **Step 1** when the number of iterations is less than C_t, otherwise, exit loop.

V. SIMULATION RESULT

To achieve a safe landing by adjusting the robot's attitude in the air, the whole body of cat robot should turn a somersault in the opposite direction when its front body and rear body finish a circle coning motion, i.e. when the angles θ_1 and θ_2 change from 0 to -2π, respectively, the angle χ_2 changes from 0 to π.

A. Result based on natural cubic spline approximation

In this section, we use the natural cubic spline approximation to approximate the control torque inputs. Six parameters ($N = 5$) are used to construct the natural cubic spline function. In order to ensure that the initial and final values of the control inputs are zero, two ends of $\zeta = [u_0, u_1, \cdots, u_N]^{\mathrm{T}}$ are predefined as zero. In our simulation experiment, we take I_1=1.5833, $\nu = 0.3158$, λ=2.5263 [14]. After a number of trials, we have found other simulation parameters meeting our demand, as shown in Table I.

After 500 iterative computations, we can get the optimal parameterization vector ζ_1 = [0, 1.8478, 2.0056, 6.3784, 2.2741, 0]$^{\mathrm{T}}$ for u_1, ζ_2 = [0, -2.0944, 0.2825, -1.0456, 1.1589, 0]$^{\mathrm{T}}$ for u_2, and ζ_3 = [0, -2.3531, -0.4120, 2.3566, 1.1113, 0]$^{\mathrm{T}}$ for u_3, by the PSO algorithm, and the energy consumption of the optimal trajectory equals $J(\zeta) = 64.8075$ units energy. The results of simulation are shown in Fig. 2-5. Fig. 2 shows plots of the optimal control torque inputs. As can be seen, the curves of control inputs are very smooth, and two ends of the curves are zero, which denotes that each control input corresponding to the curve is start at zero value and end at zero also. Fig. 3 shows the optimal trajectories of attitude angles of the falling cat robot. We can see that the angles θ_1 and θ_2 change from 0 to -2π smoothly, in addition, θ_1 and θ_2 are unsynchronized, thus, the assumption that there exists torsion between front body and rear body is realized in our experiment. The optimal trajectories of Euler angles of the falling cat robot are shown in Fig. 4. It is seen that the angle χ_2 changes from 0 to π, and the angles χ_1 and χ_3 almost no change during its falling. From Fig. 3 and 4, we can easily see that when the front body and rear body of a falling cat robot finish one round coning movement, respectively, the entirety body of the robot turns a somersault in the opposite direction, i.e. the free-falling cat robot can adjust its attitude in the air to achieve a safe landing. Fig. 5 shows the optimal trajectories of three angular velocities. In Fig. 5, we can know that the initial and final values of three angular velocities are zero, the curves are very smooth with little fluctuation, the features mentioned above are more in line with the engineering practice.

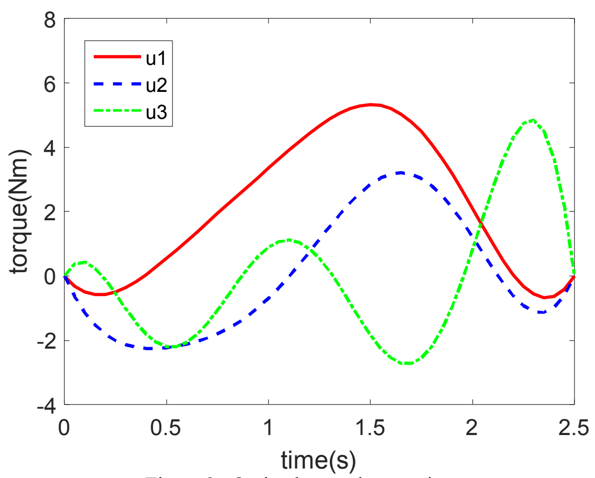

Figure 2. Optimal control torque input

TABLE I INITIALIZATION SETTING

Sampling time	$\Delta T = 0.05s$
Control time	$T = 2.5s$
Penalty factor	τ=194.85
Size of population	$M = 20$
Dimension of search space	$D = 18$
Acceleration coefficient	c_1=c_2=1.494
Inertia weight	$\omega = 0.729$
Maximum number of iterations	$C_t = 500$
Initial state	$\boldsymbol{x}_0 = [7\pi/12, 0,0,0,0,0,0,0,0]^{\mathrm{T}}$
Final desired state	$\boldsymbol{x}_r = [5\pi/6, -2\pi, -2\pi, \pi, 0,0,0,0,0]^{\mathrm{T}}$

B. Compared with Fourier-based methods

If we use the Fourier basis (Ref. [7]) to approximate inputs, although smooth input curves can be obtained, the start and end values of that are non-zero, as Fig. 6 shows, which makes it difficult to implement with typical servomotor. Compared with the method discussed above, we can obtain the control torque input with zero values at its initial and terminal points, moreover, our method requires only 2.5s, as compared to the Fourier-based method, which requires over 5.0s. In view of this, our method is a better one.

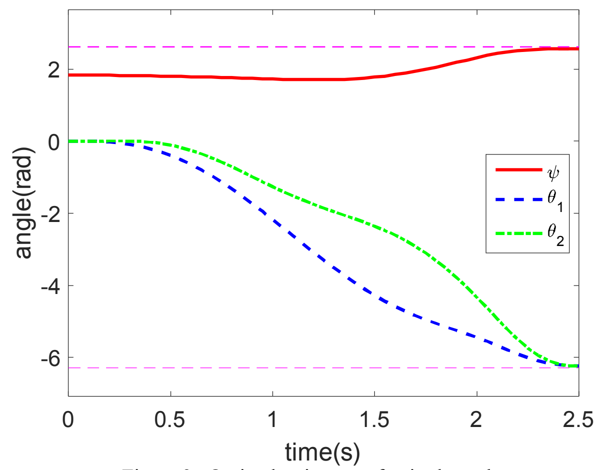

Figure 3. Optimal trajectory of attitude angle

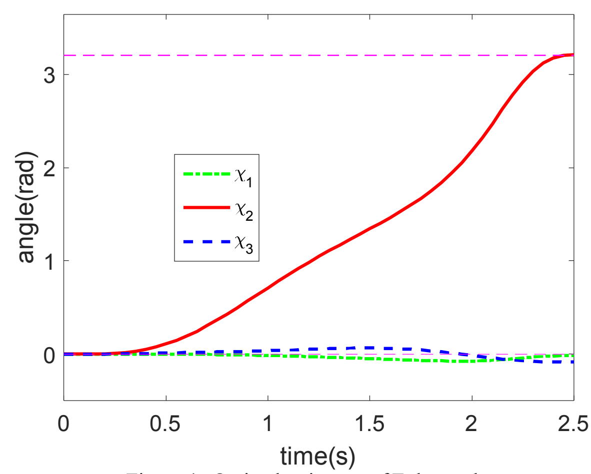

Figure 4. Optimal trajectory of Euler angle

VI. Conclusion and Future Work

The study of a falling cat is becoming a significant theme with the development of man-made spacecraft and exploratory researches of human turning motion in the case of zero gravity, but no robotic systems have been built based on their studies of the falling cat. In this paper, we have studied the optimal falling trajectory of attitude adjustment of the robot inspired by a free-falling cat with the spline approximation. As for analysis in the article and simulation result, we can know that the method based on the spline approximation is an effective one for trajectory planning of the falling cat robot, and PSO algorithm, which is applied to find an optimal solution of cost function, has a fast convergence rate and high precision. In addition, compared with other methods, we can obtain the control torque input with zero values at its initial and terminal points, and the decrease in over-turned time consumption is 50 percent. This is an excellent performance and it can easily implement in the engineering practice. It is worth mentioning that we have produced the prototype of a falling cat robot, as shown in Fig. 7. In the future, we will apply the trajectory date obtained in this paper to the prototype to achieve a real somersault in a real world.

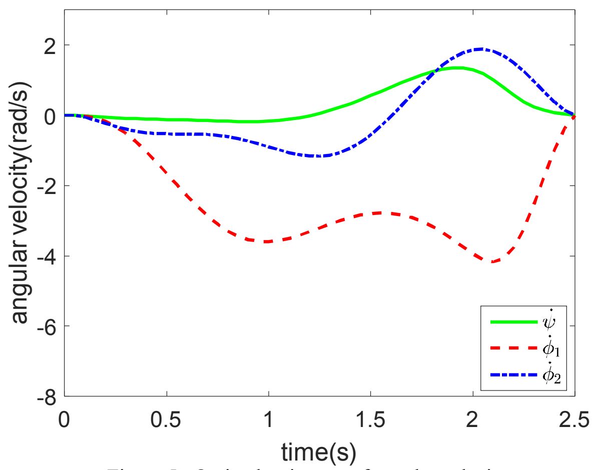

Figure 5. Optimal trajectory of angular velocity

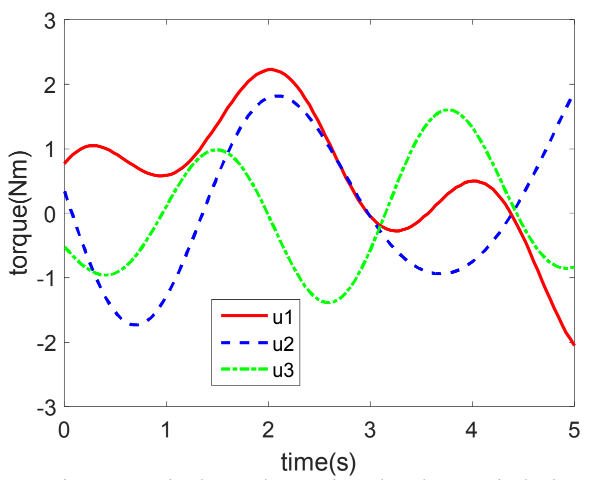

Figure 6. Optimal control torque input based on Fourier basis

Figure 7. The prototype of the robot inspired by a free-falling cat

References

[1] F. Y. Xu, J. Shen, and G. Jiang, "Kinematic and Dynamic Analysis of a Cable-climbing Robot," *International Journal of Advanced Robotic Systems*, vol. 12, no. 1, pp. 1-17, 2015.

[2] F. Y. Xu, X. Wang, and G. Jiang, "Experimental study on the dynamic performance of a cable-robot detecting system," *Transactions of the Institute of Measurement and Control*, vol. 38, no. 3, pp. 338-347, 2016.

[3] X. S. Ge, W. J. Zhao, and Y. Z. Liu, "Nonholonomic Motion Planning for a Free-Falling Cat Using Quasi-Newton Method," *Technische Mechanik*, vol. 31, no. 1, pp. 42-49, 2011.

[4] D. A. McDonald, "How Does a Falling Cat Turn Over," *American Journal of Physiology*, vol. 129, pp. 34-35, 1955.

[5] A. I. Loitsyansky, *Theoretical Mechanics*, Moscow: Saint Petersburg, 1953, pp. 48-57.

[6] T. R. Kane and M. P. Scher, "A Dynamical Explanation of the Falling Cat Phenomenon," *International Journal of Solids and Structures*, vol. 5, pp. 663-670, 1969.

[7] X. S. Ge and L. Q. Chen, "Optimal Control of Nonholonomic Motion Planning for a Free-Falling Cat," *Applied Mathematics and Mechanics*, vol. 28, no. 5, pp. 539-545, 2007.

[8] Z. Weng and H. Nishimura, "Final-State Control of a Two-Link Cat Robot," *Advance Robotics*, vol. 16, no. 4, pp. 325–343, 2002.

[9] X. Liang, L. Xu and L. Lu, "Research on trajectory planning of a robot inspired by free-falling cat based on modified quasi-Newton algorithm," in *Proceedings of IEEE International Conference on Mechatronics and Automation*, Harbin, China, 2016, pp. 552-557.

[10] R. Montgomery, "Gauge Theory of the Falling Cat," *Dynamics and Control of Mechanical Systems, American Mathematical Society*, vol. 1, pp. 193–218, 1993.

[11] R. W. Batterman, "Falling cats, Parallel Parking, and Polarized Light," *Studies in History and Philosophy of Modern Physics*, vol. 34, no.4, pp. 527-557, 2003.

[12] R. W. Brockett and L. Dai, "Non-holonomic Kinematics and the Role of Elliptic Functions in Constructive Controllability," *Nonholonomic Motion Planning*, *Springer US*, vol. 24, pp. 1-21, 1993.

[13] C. Fernandes, L. Gurvits and Z. X. Li, "Near-Optimal Nonholonomic Motion Planning for a System of Coupled Rigid Bodies," *IEEE Transaction on Automation Control*, vol. 39, no. 3, pp. 450–464, 1994.

[14] T. Iwai and H. Matsunaka, "The Falling Cat as a Port-Controlled Hamiltonian System," *Journal of Geometry and Physics*, vol. 62, pp. 279–291, 2012.

[15] D. Nakano, S. I. Maeda and S. Ishii, "Control of the falling cat motion by using path-integral reinforcement learning," *IEICE Technical Report*, vol. 112, no. 389, pp. 19-24, 2013. (in Japanese)

[16] D. R. Kincaid and E. W. Cheney, *Numerical analysis: mathematics of scientific computing*, Providence: American Mathematical Society, 2002.

[17] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, Perth, Australia, 1995, pp. 1942-1948.

[18] X. Gong and Z. Wang, *Proficient in matlab optimization calculation*, Beijing: Publishing House of Electronics Industry, 2012. (in Chinese)