A review of research methods on PLI robots

Jingzhe He[1], Hao Chen[2] , Yifan Lu[3]

1. College of IoT Engineering, Faculty of Informatics, Hohai University, Changzhou, 213022, China.
E-mail: 3435234264@qq.com

2. College of Mechanical and Electrical Engineering, Hohai University, Changzhou, 213022, China.
E-mail: 1971439481@qq.com

3. Business School of Hohai University, Changzhou, 213022, China.
E-mail: 3435234264@qq.com

Abstract: This paper mainly introduces a common power line inspection (PLI) robot model, and introduces several main solutions to the model, that is, to prove that its motion equation can converge to a given constant.

Key Words: PLI robot, PID controller, PSO algorithm, Linearization, first-order Euler discrete method

1 Introduction

The PLI (Power Line Inspection) robot takes the mobile robot as the carrier, the visible light camera, infrared thermal imager, and other detection instruments as the load system, and the multi-field information fusion of machine vision-electromagnetic field-GPS-GIS as the robot's navigation system for autonomous movement and autonomous inspection, PLI robot the functions of obstacle detection, identification and positioning, autonomous operation planning, autonomous obstacle crossing, autonomous inspection of transmission lines and their line corridors, automatic storage and remote wireless transmission of inspection images and data of the robot body, and ground remote wireless monitoring and control. At present, the application of line inspection robots is mainly used in occasions that need to inspect lines, such as transmission line inspection, pipeline inspection [1], etc., but it is also used for large equipment such as aircraft skin inspection [2]. This is also a manifestation of the concept not only limited to pipeline and power line inspections.

This paper mainly focuses on the trajectory of the line-following robot, and mainly discusses whether the robot's trajectory can converge to the given travel route under the given conditions based on a common model of the line-following robot.

2 Modeling of the PLI robot

In this section, we'll introduce the model of PLI robot and give its mathematical model. The line loading model of the PLI robot is shown in Fig. 1[3]. The balance adjustment parameters of the PLI robot are shown in Fig. 2[3].

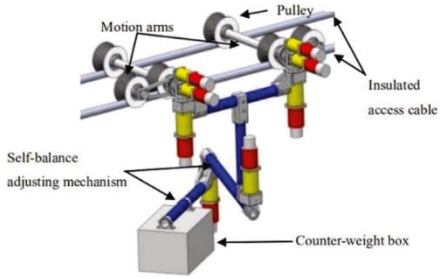
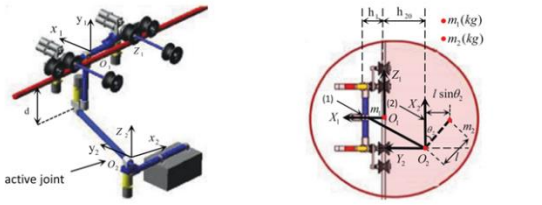
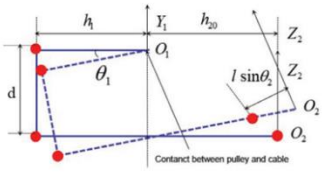


Fig. 1: Line-Loading of PLI robot.



(a) 3D model (b) Model in the $X_1O_1Z_1$ plane



(c) Model in the $X_1O_1Y_1$ plane

Fig. 2: Balance adjustment paramenters of the PLI robot.

Let θ_1 and θ_2 represent the deflection of the robot and the angle of the movable joint respectively. Let $m_1$and m_2 represent the mass of the robot body and the mass of the counterweight box, h_1 represents the horizontal distance between the cable and the center of mass of the body, h_{20} represents the counter weight box, l and d and D represent the length of the actuator bar and T-shaped base the height of. The Euler-Lagrange equation can besplayed as:

$$u_i = \frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\theta}_i}\right] - \frac{\partial L}{\partial \theta_i}, i = 1,\ldots,m \quad (1)$$

Where $u_i$is the external moment acting on the ith generalized coordinate; L can be expressed as:

*This work is supported by National Natural Science Foundation (NNSF) of China under Grant 00000000.

$$L = K - P \quad (2)$$

where K and P are the kinetic energy and potential energy of the PLI robot motion balance adjustment, respectively, and the calculation formula can be:

$$K = \frac{m_1 {h_1}^2 {\theta_1}^2}{2} + \frac{m_2 l^2 {\theta_2}^2}{2} + \frac{m_2[h^2+(-h_{20}+l\sin\theta_2)^2]{\theta_1}^2}{2} \quad (3)$$

$$P = -m_1 g h_1 \sin\theta_1 + m_2 g[(h_{20} + l\sin\theta_2)\sin\theta_1 - h\cos\theta_1] \quad (4)$$

where θ_1 is the angle between the cable and the PLI robot; θ_2 is the rotation angle of the actuator; m_1 and m_2 are the masses of the robot body and the counterweight box, respectively; l is the length of the actuator rod, and h is the height of the T-shaped base. h_{20} is the distance from the center of mass of the counterweight box to the cable when the PLI robot is in the horizontal position; g is the acceleration of gravity. We have:

$$m_1 h_1 = m_2 h_{20} \quad (5)$$

Substituting Eq. (5) into Eq. (4), we get:

$$P = m_2 g(-h\cos\theta_1 + l\sin\theta_2\sin\theta_1) \quad (6)$$

Substituting Eq. (3) and Eq. (6) into Eq. (2) to obtain the dynamic equation of the PLI robot motion balance adjustment, which can be expressed as:

$$u_1 = \{m_1 {h_1}^2 \ddot{\theta}_2 + m^2[h^2 + (-h_{20}+l\sin\theta_2)^2]\}\ddot{\theta}_1 + 2m_2 l(-h_{20} + l\sin\theta_2)(\cos\theta_2)\dot{\theta}_1\dot{\theta}_2 + m_2 g h \sin\theta_1 + m_2 g l \sin\theta_2 \cos\theta_1$$

$$u_2 = m_2 l^2 \ddot{\theta}_2 - m_2 l(-h_{20} + l\sin\theta_2)(\cos\theta_2)\dot{\theta}_1^2 + m_2 g l \cos\theta_2 \sin\theta_1 \quad (8)$$

where u_1 is the external disturbance; u_2 is the torque acting on the movable joint producing $\theta_2$$\theta$To study the balance adjustment progress of the PLI robot, let $u_1 = 0$,and define the state variable vector $[x_1, x_2, x_3, x_4] = [\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2]$. According to the dynamic equation, the state space model of the PLI robot motion balance adjustment can be obtained:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{-2m_2 l(-h_{20}+l\sin x_3)(\cos x_3)x_2 x_4 - m_2 g h \sin x_1 + m_2 g l \sin x_3 \cos x_1}{m_1 h_1^2 + m_2[h^2+(-h_{20}+l\sin x_3)^2]} \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = \frac{u_2 + m_2 l(-h_{20}+l\sin x_3)(\cos x_3)x_2^2 - m_2 g l \cos x_3 \sin x_1}{m_2 l^2} \end{cases} \quad (9)$$

The comprehensive idea is to use the Lagrange equation to construct the input quantities u_1, u_2, and by defining the state variable vector, the chain system can be obtained from the dynamic equation to achieve the original purpose. Next we will discuss a series of methods for proving the convergence of this system.

3 The research method

For the modeling of PLI robots in Section 2, there have been many and relatively complete studies have been formed. The following are some of the main research methods used:

3.1 The PID controller

PID(Proportion Integration Differentiation) controller is a linear controller, where P refers to proportion, I refers to integration, and D refers to differentiation. The principle of the PID controller is to construct the error value $e(t) = r(t) - y(t)$ (where $r(t)$is the given trajectory and $y(t)$ is the actual output value), and then e(t) is the actual output value. The Proportion(P), Integration(I) and Differentiation (D) of $e(t)$ constitute the control quantity through linear combination to control the controlled object.

The principle of PID control system is shown in Fig. 3:

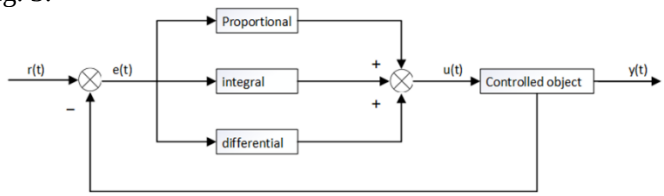


Fig. 3: The principle of PID control system

Its control law is[4]:

$$u(t) = k_P\left[e(t) + \frac{1}{T_i}\int_0^t e(t)\,dt + T_d\frac{de(t)}{dt}\right] \quad (10)$$

And Its transfer function is:

$$G_C(s) = k_p + \frac{k_p}{T_i}\cdot\frac{1}{s} + k_P\cdot T_d = k_p + k_i\cdot\frac{1}{s} + k_d\cdot s \quad (11)$$

Where k_p is the proportional control gain，$k_i = \frac{k_p}{T_i}$is the integral control gain t, and $k_d = k_P\cdot T_d$ is the differential control gain, all of which are adjustable parameters.

3.2 PSO algorithm

PSO(Particle Swarm Optimization), is an optimization method inspired by the ability of social organizations to face and solve complex problems, involving the individual capabilities of its members as well as group social dynamics. In PSO, particles are told about their neighborhood as well as the group. The particle adjusts its upcoming displacement using the best neighborhood solution as well as the best global solution. The particle displacement follows the following equation [5]:

$$\overrightarrow{V_{t+1}} = \omega\vec{V_t} + \overrightarrow{c_1} * rand\ () * (\vec{x}_{lbest} - \vec{x}_i) + \overrightarrow{c_2} * rand\ () * (\vec{x}_{Gbest} - \vec{x}_i) \quad (12)$$

$$\overrightarrow{x_{t+1}} = \overrightarrow{x_t} + \overrightarrow{V_{t+1}} \quad (13)$$

Eqs. (11) consists of three terms, where the first term ω is the inertia weight, which moderates the particle's current velocity. The second term, the cognitive component of displacement, (c1) regulates the cognitive behavior of particles. The last term represents social behavior, which is determined by a factor (c2); which defines the displacement relative to the group's optimal solution.

The PSO algorithm requires only a few parameters before running: adjustment factor values (ω, c_1, and c_2), number of particles, maximum number of iterations, and fitness function. The fitness function is used to evaluate the performance of the particles in each iteration. It is also

used to elect the best neighbor and best group solution.In this paper, the PSO is used to optimize the parameters of MFs in the GT2FLC(General type-2 fuzzy logic controller). For each iteration, the variables of the weight and velocity of the i[th] particle can be calculated by [6]:

$v_i = \omega v_i + c_1 r_1(pbest_i - \omega_i) + c_2 r_2\ (gbest - \omega_i)$

$\omega_i = v_i + \omega_i$ (14)

where v_i and ω_i are the velocity and weight of the ith particle respectively; $pbest_i$ is the optimal solution of all iterations for ith particle and $gbest$ is the best optimal solution of all iteration for entire swarm; ω is the inertia weight; c_1 and c_2 are the learning factors which are the positive constants; r_1 and r_2 are random numbers between 0 and 1.

3.3 The first-order Euler discrete method

Euler's method is a kind of numerical solution method of ordinary differential equations, and its basic idea is iteration. This paper mainly uses the first-order explicit Euler discrete method. The main method is to approximate the derivative of a point on the curve by the slope between two points near the point, that is:

$\dot{y} \approx \frac{y_{n+1} - y_n}{h}$(15)

where h is a sufficiently small constant, $\dot{y}$ is the derivative at y_n, and y_{n+1} is a point at distance h from y_n So as to achieve the purpose of transforming a continuous system into a discrete system.

4 Problem solving

4.1 The PID control system

A degenerate form PID controller, PD(Proportion Differentiation) controller [3] will be used here. The PD controller can be obtained by setting the integral coefficient $k_i = 0$ in the PID controller.

$\tau_c = k_p \cdot e + k_d \cdot \dot{e}$(16)

Where $e = \theta_d - \theta$, θ_d is the reference signal, k_p、k_d is the PD control gain

4.2 Linearization of the system

A For the linear system in (10), let $[x_1 \quad x_2 \quad x_3 \quad x_4]^T = [0 \quad 0 \quad 0 \quad 0]^T$, we can get:

$$\begin{cases} 0 = x_2 \\ 0 = \frac{-2m_2 l(-h_{20} + l\sin x_3)(\cos x_3)x_2 x_4 - m_2 gh \sin x_1 + m_2 gl \sin x_3 \cos x_1}{m_1 h_1^2 + m_2[h^2 + (-h_{20} + l\sin x_3)^2]} \\ 0 = x_4 \\ 0 = \frac{u_2 + m_2 l(-h_{20} + l\sin x_3)(\cos x_3)x_2^2 - m_2 gl \cos x_3 \sin x_1}{m_2 l^2} \end{cases}$$

(17)

The above formula can be transformed into:

$$\begin{cases} 0 = x_2 \\ h\sin x_1 = l\sin x_3 \cos x_1 \\ 0 = x_4 \\ u_{2(0)} - m_2 gl \cos x_3 \sin x_1 = 0 \end{cases} \quad (18)$$

The above formula shows that each x_1 satisfies $-\frac{\pi}{4} \le x_1 \le \frac{\pi}{4}\pi$, and has corresponding x_3 and $u_{2(0)}$. Therefore, the equation of state has a balance point, which is $[x_1, x_2, x_3, x_4]^T = [0,0,0,0]^T$. [7]

Therefore, the linear model of the PLI robot based on the nominal equilibrium point of the system is given by the following equation [8]:

$\dot{x} = Ax + Bu_2$（19）

Where matrix A、B can be expressed as:

$$A = \frac{\partial f}{\partial x}|_{x_0} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ a_{21} & 0 & a_{23} & 0 \\ 0 & 0 & 0 & 0 \\ a_{41} & 0 & 0 & 1 \end{bmatrix}$$

$$B = \frac{\partial f}{\partial u_2}\Big|_{u_2(0)} = [0 \quad 0 \quad 0 \quad b_4]$$

Where $a_{21} = \frac{-m_2 gh}{m_1 h_1^2 + m_2(h^2 + h_{20}^2)}$, $a_{41} = -\frac{g}{l}$,$b_4 = \frac{1}{m_2 l^2}$, $a_{23} = \frac{-m_2 gl}{m_1 h_1^2 + m_2(h^2 + h_{20}^2)}$

So the linear system can be written as:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = a_{21}x_1 + a_{23}x_3 \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = b_4 u_3 \end{cases}$$（19）

As a result, the complex system expression is greatly simplified, making the study of the model easier

4.3 PSO algorithm

The entire process of optimization is off-line calculation. The parameters c_1, c_2 and ω of the PSO are selected as 2.03, 2.03 and 0.9, respectively. The range of velocity v_i is [−0.1, 0.1]. The number of iterations and the size of population are selected as 200 and 70, respectively. The cost function is expressed by:

$MSE = \frac{1}{N}\sum_{i=1}^{N}(\theta_1 - \theta_d)^2$(20)

where $\theta_d$is the desired angle between the cable and the PLI robot.

The PLI robot is controlled by GT2FLC optimized by the PSO algorithm. The main role of the PSO algorithm here is to optimize GT2FLC.

The core code of PSO is as follows:

1: Initialize the population size $sizepop$, the maximum number of iterations A, velocities v_i andweights ω_i, ranges of velocities and weights, the inertia weight ω and learning factors c_1 and c_2.
2: **for** i = 1 : $sizepop$ **do**
3: apply the initially GT2FLC to the PLI robot and get θ_1
4: Compute the fitness $fit(\omega_i)$by Eq.(14)
5: $pbest_i = fit(\omega_i)$
6: **end for**
7: **for** $i = 1 : sizepop - 1$ **do**
8: **if** $fit(\omega_i) <$ fit(ω_{i+1}) **then**
9: $gbest = fit(\omega_i)$
10: **end if**
11: **end for**
12: **for** $i = 1$: A **do**
13: **for** $i = 1 : sizepop$ **do**
14: Update v_i and ω_i by Eq.(14) to get the new GT2FLC
15: apply the new GT2FLC to the PLI robot and get θ_1

16: Compute the fitness $fit(\omega_i)$ by Eq.(20)
17: **if** $fit(\omega_i) < pbest_i$ **then**
18: $pbest_i = fit(\omega_i)$
19: **end if**
20: **end for**
21: **for** $i = 1 : sizepop - 1$ do
22: **if** $pbest_i < pbest_{i+1}$ t **then**
23: $gbest = fit(\omega_i)$
24: **end if**
25: **end for**
26: **end for**
27: Output $gbest$

4.4 The FFDL model of the robotic system

Substituting $\dot{x}_i(i = 1,2,3,4)$ for $\frac{x_i^{(k+1)} - x_i^{(k+1)}}{T_s}$ in equation (9), the following discrete system can be obtained:

$$\begin{cases} x_1^{(k+1)} = x_1^{(k)} + T_s x_2^{(k)} \\ x_2^{(k+1)} = x_2^{(k)} + \frac{T_s[u_1^{(k)} - 2m_2 l\left(-h_{20} + l\sin x_3^{(k)}\right)\left(\cos x_3^{(k)}\right)x_2^{(k)} x_4^{(k)} - m_2 gh\sin x_1^{(k)} - m_2 gl\sin x_3^{(k)}\cos x_1^{(k)}]}{m_1 h_1^2 + m_2[h^2 + (-h_{20} + l\sin x_3^{(k)})^2]} \\ x_3^{(k+1)} = x_3^{(k)} + T_s x_4^{(k)} \\ x_4^{(k+1)} = x_4^{(k)} + \frac{T_s[m_2 l\left(-h_{20} + l\sin x_3^{(k)}\right)\left(\cos x_3^{(k)}\right)(x_2^{(k)})^2 + u_2^{(k)} - m_2 gl\cos x_3^{(k)}\sin x_1^{(k)}\}}{m_2 l^2} \end{cases}$$

(21)

Where Ts is the sampling time.

The discrete two-input two-output nonlinear system corresponding to the controlled object (2) can be described as:

$y(k+1) = f(y(k), \cdots, y(k-dy), u(k), \cdots, u(k-du))$(22)

Where $u(k)=[u_1(k) \quad u_2(k)]^T$, $y(k)=[y_1(k) \quad y_2(k)]^T$ are the input and output vectors of the system at time k, respectively. The terms du and $dy$are the unknown order of input and output, respectively. $f()$represents an unknown vector of smooth nonlinear functions.

This model will then be approximated and proven to converge using FFDL[9] technique.

5 Conclusion

The research on the line-following robot model mainly adopts the above method to solve the model mentioned in this paper. At present, the related research on the power line patrol robot has been relatively complete, and there are many solutions to the model research, such as particle swarm optimization algorithm, linearization and PID controller. These methods have proved the convergence of the model from different angles, and these methods have their own characteristics. For example, the PID controller has the characteristics of self-adaptation and has stronger anti-interference ability; the linearization scheme simplifies the research problem , which not only simplifies the research, but also overcomes the problem that the original nonlinear model is infeasible to calculate; using the PSO algorithm can effectively solve the problem of the model, as long as the relevant parameters of the PSO algorithm are set, the convergent solution can be obtained through the algorithm. These methods and algorithms may provide new ideas for the development of line-following robots in the application of line-following robots in the future.

From the research progress of line-following robots, there have been many solutions to the problem of trajectory tracking control of line-following robots. Although there are many methods, for actual simulation, if the advantages of these methods can be used to design a line patrol robot, it may be possible to design a line patrol that meets the needs and perfectly meets the needs for different occasions and different conditions robot.

References

[1] Michał Ciszewski, Tomasz Buratowski, Mariusz Giergiel,Modeling, Simulation and Control of a Pipe Inspection Mobile Robot with an Active Adaptation System,IFAC-PapersOnLine,Volume 51, Issue 22,2018,Pages 132-137,ISSN 2405-8963,

[2] Junjun Jiang and Congqing Wang, Modelling and Backstepping Motion Control of the Aircraft Skin Inspection Robot , CMES, vol.120, no. 1, pp. 105-121, 2019

[3] Tao Zhao, Jiahao Liu, Songyi Dian, Rui Guo, Shengchuan Li,Sliding-Mode-Control-Theory-Based Adaptive General Type-2 Fuzzy Neural Network Control for Power-line Inspection Robots, Neurocomputing,Volume 401,2020,Pages 281-294,ISSN 0925-2312,

[4] D. Li Nan. Research and application of modern design technology of PID control parameters. Zhejiang University of Technology, 2009.

[5] N. Rokbani, A. Abraham and A. M. Alimi, "Fuzzy Ant Supervised by PSO and simplified ant supervised PSO applied to TSP," 13th International Conference on Hybrid Intelligent Systems (HIS 2013), 2013, pp. 251-255, doi: 10.1109/HIS.2013.6920491.

[6] Zhao, Tao et al. 'Optimization Design of General Type-2 Fuzzy Logic Controllers for an Uncertain Power-line Inspection Robot'. 1 Jan. 2019 : 2203 – 2214.

[7] S. Dian, L. Chen, S. Hoang, M. Pu and J. Liu, "Dynamic balance control based on an adaptive gain-scheduled backstepping scheme for power-line inspection robots," in IEEE/CAA Journal of Automatica Sinica, vol. 6, no. 1, pp. 198-208, January 2019, doi: 10.1109/JAS.2017.7510721.

[8] Liu, J., Zhao, T. & Dian, S. General type-2 fuzzy sliding mode control for motion balance adjusting of power-line inspection robot. Soft Comput 25, 1033–1047 (2021).

[9] Lin Chen,Haibin Duan, Songyi Dia,Son Hoang,Event-based model-free sliding mode control for an inspection robot,Willey Online Library,2020