Research Article

Research of Pose Control Algorithm of Coal Mine Rescue Snake Robot

Yun Bai¹ and YuanBin Hou²

¹Engineering Training Center, Xi'an University of Science & Technology, Xi'an 710054, China

²College of Electrical and Control Engineering, Xi'an University of Science & Technology, Xi'an 710054, China

Correspondence should be addressed to Yun Bai; 494361962@qq.com

Received 10 November 2017; Revised 17 January 2018; Accepted 24 January 2018; Published 22 February 2018

Academic Editor: Luis Gracia

Copyright © 2018 Yun Bai and YuanBin Hou. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming at how to achieve optimal control of joint pitch angles in the process of the robot surmounting obstacle, taking the developed coal mine rescue snake robot as an experimental platform, a pose control algorithm based on particle swarm optimization weight coefficient of extreme learning machine (PSOELM) is proposed. In order to obtain the optimized hidden layer matrix of the extreme learning machine (ELM), particle swarm optimization (PSO) is applied to optimize the weight coefficient of hidden layer matrix. The simulation and experiment results prove that, compared with the ELM algorithm, the smaller mean square error (MSE) between the joint pitch angles of robot and the expected values is acquired by the PSOELM, which overcomes the shortcoming that traditional extreme learning machine cannot reach the best performance because of the random selection of the parameters of the hidden layer nodes. PSOELM is superior to ELM algorithm in control accuracy, fast searching for the optimal and stability. Optimal control of robot's joint pitch angles is achieved. The algorithm is applied to the surmounting obstacle control of the developed snake robot, and it lays the foundation for further implement of the coal mine rescue.

1. Introduction

The snake robot has the characteristics of good stability, multiple degrees of freedom, multimovement gaits, small cross-section, and so forth. The snake robot can walk on the rugged grounds, can move through caves, and has strong obstacle surmounting abilities. Therefore, compared with the traditional mobile robot driven by wheels or caterpillar tracks, faced with the complex environment after a coal mine disaster, the snake robot can quickly and reliably respond to rescue work. The robot can replace the rescue personnel to enter the site as a first responder. The site information can be collected, and scientific basis for a rapid rescue is provided. Hence developing a coal mine rescue snake robot has a very important practical significance. At the same time, multiple degrees of freedom and flexibility in the movement of the snake robot has brought opportunities and challenges for the research of its pose control. To allow the snake robot to move as flexibly as biological snakes, there are a variety of methods for pose control such as the discrete curve method,

dynamics and kinematics model generation method, neural network model generation method (e.g., CPG: central pattern generator) [1], and so on.

Discrete Curve Method. This method refers to that a curve which is defined according to biological snake gait is fitted by the body of snake robot; thus each joint rotating angle of the robot is obtained, also known as the “inverse kinematics method.” According to the serpenoid curve of serpentine locomotion proposed by Hirose [2], Wang et al. [3] used the specific parameters including the wave propagation rate and the number to disperse the serpenoid curve, and the governing equations of serpentine locomotion were obtained. Ye et al. [4] proposed a simple snake curve for rotational and lateral motion and thus made the motion control equation of snake robot simpler. Hatton and Choset [5] used a back curve (backbone curve) to extract the waveform of the snake robot and to fit the waveform by using the simulated annealing algorithm; hence multiple gaits for the snake robot were acquired. Xie et al. [6] developed a prototypical

underwater snake robot. The snake's body is composed of 16 light waterproof small servomotors, and the serpentine locomotion control of robot was realized by using the snake curve. The discrete curve method has the advantages of simple implementation and easy control. The disadvantage is that the jumping of joint angular velocity and torque can lead to the servomotors locking phenomenon and the damage of servomotors.

Dynamic and Kinematic Model Generation Method. Aiming at the snake robot with linkage type equipped with three driven wheels, Ostrowski and Burdick [7] established a dynamic model of serpentine locomotion by using the Lagrange method and analyzed the controllability of the snake robot. Liljebäck et al. [8] established a dynamic model of the snake robot by making use of the relationship between contact force and relative angle, thereby achieving control of obstacle-aided snake robot locomotion. Chen et al. [9] established a space kinematic model (spatial linkage model), and two kinds of three-dimensional motion of snake robot, side winding and lateral rolling, were implemented. Wei and Sun [10] established a kinematic model of snake robot based on orthogonal joints, achieving control of the bridge cable climbing gait. Guo et al. [11] put forward a velocity tracking control algorithm avoiding the singular posture based on the dynamic and control unified model. The algorithm was applied to motion control of a snake robot with passive wheels. Cheng et al. [12] presented the method of surmounting obstacle for snake robot with 16 P-R-T unit modules based on the kinematic model. The advantage of the dynamic and the kinematic model generation method is that complex and accurate gait control can be realized. The disadvantage is that the modeling process is difficult, and the controller must perform a large amount of calculation, and the adaptability to unknown environments is poor.

Neural Network Model (CPG) Generation Method. The CPG method has been widely used in the control of snake robots. Crespi and Ijspeert [13] made use of the CPG model and accomplished optimal control of the swimming and crawling of a snake robot. Lu et al. [14] proposed a cyclic inhibitory CPG controller and serpentine locomotion of a snake robot was successfully implemented. Aiming to resolve the problem of low efficiency and instability of parameter tuning in the cyclic inhibitory CPG control model, Lian et al. [15] presented a parameter optimization method of CPG model based on genetic algorithms, and the method was effectively applied to the gait control of a snake robot. Gao et al. [16] established a CPG motion control network by using Hopf oscillators and applied the network to the serpentine locomotion of a snake robot, which improved the environmental adaptability of the robot. CPG control has the advantage of easy integration of environmental information, so that the snake robot has the ability to adapt to the environment. The disadvantage is that the control parameters in CPG model need to be further optimized.

Based on the present methods for pose control of the snake robot, combined with the advantages of particle swarm

FIGURE 1: The orthogonal joint connection diagram.

optimization (PSO) algorithm and extreme learning machine (ELM), a pose control algorithm based on particle swarm optimization weight coefficient of extreme learning machine (PSOELM) is put forward. Taking the developed coal mine rescue snake robot as an experimental platform, and, in order to achieve optimal control of robot's joint pitch angles, the PSOELM algorithm is applied to the robot's surmounting obstacle behavior.

2. Design of the Coal Mine Rescue Snake Robot

2.1. Design of Mechanical Structure. The mechanical body of the coal mine rescue snake robot adopts orthogonal joint connection, which has four orthogonal joints, as is shown in Figure 1. The total length of the snake robot is 1 m, and it is composed of the head, the body, and the tail. Considering the rugged tunnel environment after the coal mine disaster, the robot is driven by the self-made blades wheels. Compared with snake robot driven by wheels or caterpillar tracks, this robot has better obstacle surmounting capabilities. The mechanical structure of the robot is shown in Figure 2.

The robot's characteristics are as follows.

(1) Unit module idea is adopted; the head of snake robot (6) is composed of a night vision device and sensor group, installed in the front end of the unit module (1).

(2) The body of snake robot (29) is composed of five unit modules (1, 2, 3, 4, and 5), five connecting plates (12, 13, 14, 15, and 16), four orthogonal joints (8, 9, 10, and 11), and two speed servomotors (27 and 28). Among them, each orthogonal joint consists of a horizontal and a vertical direction servomotor.

(3) The five unit modules are, respectively, installed on the five connecting plates and are connected successively through the four orthogonal joints to form the body of the snake robot. In which, the unit modules (2, 3, and 4) have the same structure, each having two direction servomotors, two direct current speed reduction motors, and two blades wheels; the unit modules (1, 5) are identical in structure, each having one direction servomotors, two direct current speed reduction motors, and two blades wheels.

(4) The tail of snake robot (7) consists of master controller, power source, obstacle avoidance module, and communication module, installed in the back end of the unit module (5).

FIGURE 2: Mechanical structure of the blades wheels snake robot.

FIGURE 3: The snake robot entering the simulated coal mine tunnel.

(5) The blades wheels (17, 18, 19, 20, 21, 22, 23, 24, 25, and 26) as moving mechanism, installed on both sides of the snake robot.

(6) The speed servomotor (28) is used to control the rotation of the blades wheels (17, 19, 21, 23, and 25) on the left side of the snake robot. The speed servomotor (27) is used to control the rotation of the blades wheels (18, 20, 22, 24, and 26) on the right side of the snake robot.

The snake robot enters the simulated coal mine tunnel, which is semioval tunnel, as shown in Figure 3.

2.2. Design of the Control System. The control system, as shown in Figure 4, is made up of the power source, environmental detection system, pose control system, master controller, mobile mechanism, obstacle avoidance module, communication module, and host computer system.

Among them, the power source includes E1 and E2, of which E1 is a 12.6v Li-PO rechargeable battery with charging current of 1300 mAh. E1 supplies power to servomotors and direct current speed reduction motors. E2 is 4.5v mercury-free alkaline battery and supplies power to the obstacle avoidance module.

The environmental detection system consists of a night vision device and sensor group. The night vision device can be set to night mode and general mode. The sensor group is composed of two ultrasonic sensors and an infrared sensor. Taking into account the special environment of coal mine tunnel, the night vision device is installed on the top of head, and the overall environment information of simulated coal mine tunnel can be obtained by the night vision device. Two ultrasonic sensors are installed on the left and right sides of the head, and the infrared sensor is installed in the middle. The distance information between the snake robot and obstacles can be obtained by using ultrasonic sensors and infrared sensor, and the infrared sensor can effectively

FIGURE 4: Control system of the snake robot.

compensate for the fade zone existing in ultrasonic sensor ranging.

The pose control system is mainly composed of four horizontal and four vertical direction servomotors. The horizontal direction servomotors control serpentine and straight posture of the snake robot. The vertical direction servomotors control concertina and head rising posture. The M-24 digital servomotor is selected. When used as the joint motor, it can rotate 0~300 degrees and provide up to 16 kg-cm of torque, which is 2 times that of the general digital servomotor. Each servomotor has an individual ID. Each servomotor can be independently controlled, and thus the control of snake robot's gait and pose is acquired.

The master controller uses the Vensmile W10 Mini PC, which has the Atom Z3735F Bay Trail processor, 2 GB memory, and 64 GB storage space.

The mobile mechanism is made up of two speed servomotors, ten direct current speed reduction motors, and ten blades wheels. By controlling the speed servomotors, the direct current speed reduction motors are driven to motivate the blades wheels to rotate, and then the purpose of controlling the speed of the snake robot is realized.

The obstacle avoidance module uses Arduino Uno R3 control board, and its main control chip is ATMEGA328P-PU of Atmel Company.

The communication system uses a self-organized LAN (local area network) to complete data and video image transmission function.

The host computer system includes a monitoring screen of the snake robot's locomotion and the real-time display screen of the environment of the simulated coal mine tunnel.

FIGURE 5: Flowchart of the pose control algorithm.

And intelligent identification and location of cracks on the tunnel could also be achieved.

3. Pose Control Algorithm Based on the PSOELM

Pose control algorithm based on particle swarm optimization weight coefficient of extreme learning machine (PSOELM) is proposed, and the algorithm is applied to the robot's surmounting obstacle behavior. The obstacles are divided into cuboids, cubes, or cylinders. The flowchart of the pose control algorithm is shown in Figure 5, in which R-1, R-2, and R-3 refer to the first three rules in the expert rules of robot's pose control. The algorithm is composed of three

steps: firstly, the expert rules of robot's pose control are obtained; secondly, the control model based on extreme learning machine (ELM) is established; thirdly, the particle swarm optimization algorithm is used to optimize the hidden layer matrix weight coefficient of extreme learning machine, so as to obtain the optimized hidden layer matrix, and the optimal control of joint pitch angles of the snake robot is achieved.

3.1. The First Step: Expert Rules of Pose Control. (1) When the sensors detect the height of obstacle is low ($h < 5$ cm), the first joint is raised to 45 degrees and the second joint is raised to 15 degrees by UK1 and UK3, and then UK5 and UK7 cause the third and fourth joints to synchronously move ahead. Here, UK1, UK3, UK5, and UK7, respectively, refer to the output variables of master controller controlling four vertical direction servomotors.

(2) When the sensors detect the height of obstacle is higher ($5 \text{ cm} < h < 8 \text{ cm}$), the first joint is raised to 45 degrees and the second joint is raised to 35 degrees by UK1 and UK3, and then UK5 and UK7 cause the third and fourth joints to synchronously move ahead.

(3) When the sensors detect the height of obstacle is higher ($8 \text{ cm} < h < 12 \text{ cm}$), the first joint is raised to 45 degrees and the second joint is raised to 45 degrees and the third joint is raised to 25 degrees by UK1, UK3, and UK5, and then UK7 lets the fourth joint move ahead.

(4) When the sensors detect the height of obstacle is highest ($h > 12 \text{ cm}$), the robot chooses the obstacle avoidance action to bypass the obstacle.

(5) While surmounting the obstacle, the robot must bend at the joint (waist). When the head joint of the snake is past the apex of the obstacle, it must tilt downwards such that the joint contacts the ground and the angle of the joint in the vertical direction is 0 degrees. Such as the first joint inclines downward to -45 degrees and, at the same time, the joint is associated with the first joint rises. When the first joint is on the ground and the joint angle changes to 0 degrees, the actions of surmounting the obstacle are completed. In summary, the whole process of surmounting an obstacle includes the six related actions, which are raising the head, moving forward, bowing head, bowing the body, raising the tail, and making straightening out.

3.2. The Second Step: The Control Model Based on Extreme Learning Machine (ELM). Snake robot's locomotion of surmounting obstacle refer to that, in the orthogonal joint, the angle of joint in the horizontal direction (namely left and right) keeps 0 degrees, while the angle of joint in the vertical direction (namely up and down) changes according to the cosine curve. Here, according to the equation of serpenoid curve curvature of Hirose, θ_i is defined as the angle between the i th joint and the horizontal direction (namely the pitch angle) and expressed as follows:

$$\theta_i = a_0 \cos\left(\frac{iLk}{n}\right), \quad (1)$$

where a_0 is the initial angle of serpenoid curve. k is the scale factor of robot's pitch angle. L is the total length of snake

robot. n is the number of unit modules. $n - 1$ is the number of joints. i represents i th joint. $i = 1, 2, \dots, n - 1$.

φ_i is defined as the relative rotation angle between the i th joint and the $(i - 1)$ th joint. The relative rotation angle function of the i th joint in dynamic condition can be obtained as follows:

$$\varphi_i(t) = -2a_0 \sin(k_1 l) \sin(\omega t + k_1(s - 2(i - 1)l)), \quad (2)$$

where a_0 is the initial angle of serpenoid curve. k_1 is the scale factor of joint's relative rotation angle. s is the total length of serpenoid curve. $2l$ is the length of unit module of snake robot. i represents i th joint. $i = 1, 2, \dots, n - 1$. Here, $k_1 = 2\pi/(s + c)$, c is the adjustment parameter.

A model of extreme learning machine (ELM) algorithm is a feed-forward neural network with single hidden layer. The basic idea is that all the hidden node parameters (including the weight between the hidden layer and the input layer and the bias of the hidden layer nodes) are randomly generated and are independent of the objective functions and the training sample and do not need iterative adjustment. Only the number of hidden layer nodes needs to be set, and then the weight between the hidden layer and the output layer is analyzed and calculated. The optimal weight coefficient is obtained by using the particle swarm optimization (PSO) algorithm and the final network output acquired is the optimal solution. If the pitch angle is optimized, the algorithm is used to optimize formula (1). If the relative rotation angle is optimized, the algorithm is used to optimize formula (2). The following is mainly aimed at the optimization of pitch angle in the second step.

The system model is the 3-6-1 type network. The network input is three kinds of random permitted solution $\theta_i = [\theta_{i1}, \theta_{i2}, \theta_{i3}]^T$, ($i = 1, 2, 3$), which indicates the first three joint angles of snake robot when the obstacle is three different heights. The hidden layer nodes are 6. The output node of network is 1, which corresponds to the actual angles of the first three joints of the snake robot when surmounting the obstacle. The angles specified by each expert rule in the first step are used as the expected angles. The error between the actual angles and the expected angles of the joints is minimized by the extreme learning algorithm; that is, the error between the network output (actual angles) and the expected output (expected angles) is minimized and $\Delta\theta_i \rightarrow 0$ is accomplished.

The extreme learning algorithm is equivalent to solving the objective function L :

$$\begin{aligned} \min: \quad & L \\ & = \frac{1}{2} \|\beta\|^2 + \frac{1}{2} C \sum_{i=1}^3 \|\Delta\theta_i\|^2 \\ & \quad - \sum_{i=1}^3 \sum_{j=1}^6 \lambda_i (h(\theta_i) \beta_j - y_i - \Delta\theta_i), \end{aligned} \quad (3)$$

where $\theta_i = [\theta_{i1}, \theta_{i2}, \theta_{i3}]^T$, ($i = 1, 2, 3$) are input neurons for extreme learning machine; $Y = [y_1 \ y_2 \ y_3]^T$ is expected output for extreme learning machine; β is weight between

hidden layer and output layer; $\Delta\theta_i$ is pitch angle error of snake robot of network training; C is corresponding penalty factor of $\Delta\theta_i$; λ_i is Lagrange multiplier; $h(\theta_i) = [G(a_1, b_1, \theta_i), G(a_2, b_2, \theta_i), \dots, G(a_L, b_L, \theta_i)]$ is the output vector of hidden layer about θ_i , where $G(a_j, b_j, \theta_i)$ is the output of the j th hidden node through the activation function. a_j is the weight between the j th hidden layer node and the network input. b_j is the bias of the j th hidden layer node. Here, the hyperbolic function is chosen as the activation function, the input of the j th hidden node of the network is gained, as the following formula:

$$q_j = \sum_{j=1}^6 a_j \theta_i + b_j. \quad (4)$$

So, the output vector of the j th hidden node about θ_i is as follows:

$$h_j(\theta_i) = \frac{1 - e^{-pq_j}}{1 + e^{-pq_j}}, \quad (5)$$

where p is the optimal weight coefficient.

Calculate the partial derivative for each variable of the objective function L :

$$\begin{aligned} \frac{\partial L}{\partial \beta_j} &= 0 \rightarrow \\ \beta_j &= \sum_{i=1}^3 \lambda_i h(\theta_i)^T \rightarrow \\ \beta &= H^T \lambda \\ \frac{\partial L}{\partial (\Delta\theta_i)} &= 0 \rightarrow \\ \lambda_i &= C(\Delta\theta_i) \\ \frac{\partial L}{\partial \lambda_i} &= 0 \rightarrow \\ h(\theta_i) \beta - y_i^T + (\Delta\theta_i)^T &= 0, \end{aligned} \quad (6)$$

where H is a 3×6 -dimensional hidden layer output matrix:

$$H = [h(\theta_1) \ h(\theta_2) \ h(\theta_3)]^T, \quad (7)$$

because

$$Y = [y_1 \ y_2 \ y_3]^T. \quad (8)$$

From formula (3) we obtain $\partial L / (\partial \beta_j, \partial (\Delta\theta_i), \partial \lambda_i) > 0$. So, according to the principle of least squares, the following formula can be derived:

$$\left(\frac{I}{C} + HH^T \right) \lambda = Y. \quad (9)$$

Then, the weight β can be deduced as shown in the following formula:

$$\beta = H^T \lambda = H^T \left(\frac{I}{C} + HH^T \right)^{-1} Y. \quad (10)$$

FIGURE 6: (a) Action 1 of surmounting the obstacle. (b) Action 2 of surmounting the obstacle.

Therefore, the output formula (11) of the extreme learning machine is obtained:

$$f(\theta_i) = h(\theta_i) \beta = \sum_{j=1}^6 h_j(\theta_i) H^T \left(\frac{1}{C} + HH^T \right)^{-1} Y. \quad (11)$$

3.3. The Third Step: Particle Swarm Optimization Weight Coefficient p . In formula (11), in order to solve the optimal weight coefficient p of hidden layer matrix, the particle swarm optimization algorithm is used. In the process of solving, the particle is updated by tracking the two optimal values. The first one is the optimal solution which is searched by the particle itself, and the other is the optimal solution searched by the whole swarm so far. The formula used is as follows:

$$\begin{aligned} v_i(t+1) &= \omega v_i(t) + c_1 r_1 (x_{pbest_i} - x_i(t)) \\ &\quad + c_2 r_2 (x_{gbest_i} - x_i(t)), \end{aligned} \quad (12)$$

$$x_i(t+1) = x_i(t) + v_i(t+1),$$

where i represents a particle. x_i indicates the position of the particle at t moment (corresponding to the optimal weight coefficient p of output matrix of the hidden layer at t moment). v_i indicates the velocity of particles at t moment. x_{pbest_i} represents the best position of the particle so far. x_{gbest_i} represents the best position searched by the whole swarm so far. $0 < \omega < 1$ is inertial coefficient. c_1 and c_2 are learning factors. r_1 and r_2 are the random numbers uniformly distributed on interval $(0, 1)$. After calculating $v_i(t+1)$, the position $x_i(t+1)$ of the particle i at next moment can be calculated. That is, position x_i is changed by modifying speed v_i to make $\Delta p_i \rightarrow 0$, achieving optimal weight coefficient p .

In this system, the weight coefficient of hidden layer matrix is optimized by the particle swarm optimization algorithm. It makes that the error between the actual output of extreme learning machine $f(\theta)$ and the expected output Y is minimized, that is, $\Delta \theta_i \rightarrow 0$. The optimal pitch angle of each joint is obtained.

4. Simulation Experiment and Result Analysis

Combined with the snake robot's behavior of surmounting obstacle which are shown in Figures 6(a) and 6(b), the 15

training samples and 5 test samples were randomly selected. Input variables are pitch angles of the first three joints when the obstacle is three different heights, that is, $\theta_i = [\theta_{i1}, \theta_{i2}, \theta_{i3}]^T$, $i = 1, 2, 3$, and input layer nodes are 3. Output variables are the actual pitch angles of the first three joints, that is, $f(\theta) = [f(\theta_1) \ f(\theta_2) \ f(\theta_3)]^T$, and the output node is 1. The computer simulation is carried out in MATLAB.

The effects of the ELM and POSELM control models are evaluated by comparing the following 4 indicators. The performance and generalization ability of the model are evaluated by calculating MSE (mean square error) and determinant coefficient R^2 . The computation speed is evaluated by comparing the runtime of models. The stability of the model is evaluated by analyzing the effect of the number of hidden layer nodes on MRE (mean relative error).

(1) *MSE (Mean Square Error) and Determinant Coefficient R^2 .* Here,

$$\begin{aligned} \text{MSE} &= \frac{1}{n} \sum_{i=1}^n (f(\theta_i) - y_i)^2, \\ R^2 &= \frac{(n \sum_{i=1}^n f(\theta_i) y_i - \sum_{i=1}^n f(\theta_i) \sum_{i=1}^n y_i)^2}{(n \sum_{i=1}^n f(\theta_i)^2 - (\sum_{i=1}^n f(\theta_i))^2) (n \sum_{i=1}^n y_i^2 - (\sum_{i=1}^n y_i)^2)}. \end{aligned} \quad (13)$$

In formula (13), n represents the number of samples. $f(\theta_i)$ is the output of the optimal control model, and y_i is the expected output. The smaller the MSE, the better the performance of the model. The value of R^2 is in the interval of $[0, 1]$. The closer to 1, the better the performance of the model.

PSOELM and ELM are used, respectively, to control the joint pitch angles of snake robot, and the simulation results are shown in Figure 7. The x -axis represents the first three joints of snake robot, and the y -axis indicates the joint pitch angle. During surmounting the obstacle, when the obstacle height $h < 5$ cm, pitch angles of the first three joints are, respectively, controlled by the PSOELM and ELM, and the results are, respectively, given as shown in Figures 7(a) and 7(b). Determinant coefficient R^2 is 1, and mean square error of pitch angles, respectively, reaches 6.4198×10^{-35} and 1.0272×10^{-32} ; when the obstacle height $5 \text{ cm} < h < 8$ cm, pitch

FIGURE 7: (a) Pitch angles optimized by the PSOELM ($h < 5$ cm). (b) Pitch angles controlled by the ELM ($h < 5$ cm). (c) Pitch angles optimized by the PSOELM ($5 \text{ cm} < h < 8$ cm). (d) Pitch angles controlled by the ELM ($5 \text{ cm} < h < 8$ cm). (e) Pitch angles optimized by the PSOELM ($8 \text{ cm} < h < 12$ cm). (f) Pitch angles controlled by the ELM ($8 \text{ cm} < h < 12$ cm).

TABLE 1: MSE comparison between PSOELM and ELM.

Algorithm types	MSE of pitch angle ($h < 5$ cm)	MSE of pitch angle ($5 \text{ cm} < h < 8$ cm)	MSE of pitch angle ($8 \text{ cm} < h < 12$ cm)
PSOELM	6.4198×10^{-35}	4.1087×10^{-33}	1.0272×10^{-33}
ELM	1.0272×10^{-32}	1.3132×10^{-32}	1.2326×10^{-32}

FIGURE 8: Comparison of the runtime of the 10 models (POSELM versus ELM).

FIGURE 9: Hidden layer nodes and mean relative error (MRE) curves (POSELM versus ELM).

angles of the first three joints are, respectively, controlled by the PSOELM and ELM, and the results are, respectively, given as shown in Figures 7(c) and 7(d). Determinant coefficient R^2 is 1, and mean square error of pitch angles, respectively, reaches 4.1087×10^{-33} and 1.3132×10^{-32} ; when the obstacle height $8 \text{ cm} < h < 12 \text{ cm}$, pitch angles of the first three joints are, respectively, controlled by the PSOELM and ELM, and the results are, respectively, given as shown in Figures 7(e) and 7(f). Determinant coefficient R^2 is 1, and mean square error of pitch angles, respectively, reaches 1.0272×10^{-33} and 1.2326×10^{-32} . MSE based on PSOELM and ELM is shown in Table 1. It can be seen from Figure 7 and Table 1, compared with ELM, the smaller mean square error and better control accuracy and generalization ability are acquired by PSOELM. Optimal control of joint pitch angles can be realized.

(2) *The Model Runtime.* Since the training set and the test set are generated randomly, each runtime of the model is different. The above-mentioned 10 identical ELM models and 10 identical PSOELM models are, respectively, selected to test their computation speed. The maximum number of iterations is 3. As shown in Figure 8, the average runtime of the 10 ELM models and the 10 PSOELM models is, respectively, 0.0426 S and 0.0535 S. So the runtime of the ELM and PSOELM models is roughly the same, and it is all around 50 ms. It can be seen that the PSOELM model continues the characteristics of the fast learning of the ELM model;

(3) *The Stability of the Model.* The effect of the number of hidden layer nodes on MRE is analyzed to evaluate the stability of the model. The relationship curves between the hidden layer node j and MRE are shown in Figure 9. As obtained from Figure 9, the ELM model shows great volatility with the change of j , which is caused by random selection of input weight and hidden layer node bias. In the PSOELM model, the fluctuation of MRE is smaller with the change of j , and the smaller hidden layer nodes can ensure that MRE is the smallest. When j is 6, MRE is the smallest, which is 1.3673×10^{-16} . The PSOELM model has the characteristic of fast searching for the optimal and stability, and it has significant advantages in the control of the joint pitch angles of the snake robot.

5. Conclusion

A coal mine rescue snake robot is developed, and, taking the robot as an experimental platform, aiming at the robot's obstacle surmounting behavior, a pose control algorithm based on particle swarm optimization weight coefficient of extreme learning machine (PSOELM) is studied in this paper. The following conclusions are drawn.

(1) The mechanical structure and control system of the coal mine rescue snake robot are designed and made, respectively. The mechanical part is consisted of five unit modules, five connecting plates, four orthogonal joints, and two speed servomotors. The robot has better capability to

surmount the obstacle because of using the self-made blades wheels. The control system is made up of the power source, environmental detection system, pose control system, master controller, mobile mechanism, obstacle avoidance module, communication module, and host computer system. The snake robot can walk on the rugged grounds. By controlling the direction servomotors, serpentine, straight, concertina, and head rising posture of the snake robot can be acquired, and the speed of the snake robot can be changed by controlling the speed servomotors.

(2) The pose control algorithm based on PSOELM is proposed and discussed; compared with simulation results of the algorithm based on ELM, the PSOELM algorithm has the following advantages: the smaller mean square error and better control accuracy and generalization ability are acquired; the characteristics of fast learning of ELM are continued, having the characteristic of fast searching for the optimal and better stability; so optimal control of robot's joint pitch angles is achieved by the PSOELM algorithm.

(3) The PSOELM algorithm is applied to the surmounting obstacle control of the developed snake robot, and it lays the foundation for further implement of the coal mine rescue.

Disclosure

The coal mine rescue snake robot developed won special award in 2016 Chinese Education Robot Contest.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research has been supported by Foundation of Science and Technology Innovation Project of China Coal Technology Engineering Group (no. KJ3013-XBMJ-03) and cultivating fund of Xi'an University of Science And Technology (no. 2013024).

References

- [1] A. Crespi, A. Badertscher, A. Guignard, and A. J. Ijspeert, "AmphiBot I: An amphibious snake-like robot," *Robotics and Autonomous Systems*, vol. 50, no. 4, pp. 163–175, 2005.
- [2] S. Hirose, *Biologically Inspired Robots (Snake-like Locomotors and Manipulators)*, Oxford University Press, Oxford, UK, 1993.
- [3] Y. C. Wang, L. Chen, S. G. Ma et al., "Studies on lateral roiling locomotion of a snake robot," in *Proceedings of the International Conference on Robotics and Automation IEEE*, pp. 5070–5074, Piscataway, NJ, USA, 2004.
- [4] C. Ye, S. Ma, B. Li, and Y. Wang, "Turning and side motion of snake-like robot," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '04)*, pp. 5075–5080, IEEE, May 2004.
- [5] R. L. Hatton and H. Choset, "Generating gaits for snake robots: Annealed chain fitting and keyframe wave extraction," *Autonomous Robots*, vol. 28, no. 3, pp. 271–281, 2010.
- [6] Y. Xie, U. Zhenli, and X. Huigang, "Research on underwater snake-like robots mechanism design and their serpentine swimming performance," *Chinese High Technology Letters*, vol. 26, no. 6, pp. 599–605, 2016.
- [7] J. Ostrowski and J. Burdick, "The geometric mechanics of undulatory robotic locomotion," *International Journal of Robotics Research*, vol. 17, no. 7, pp. 683–701, 1998.
- [8] P. Liljeback, K. Y. Pettersen, and K. Stavdahl, "Modelling and control of obstacle-aided snake robot locomotion based on jam resolution," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation, ICRA '09*, pp. 3807–3814, Japan, May 2009.
- [9] L. Chen, Y.-C. Wang, S.-G. Ma, and B. Li, "Study of lateral locomotion of snake robot," *Jiqiren/Robot*, vol. 25, no. 3, p. 246, 2003.
- [10] W. Wei and C. Sun, "Research on Gait Generation and Control of Snake-like Robot for Bridge Cable Climbing [J]," *China Mechanical Engineering*, vol. 23, no. 10, pp. 1230–1235, 2012.
- [11] X. Guo, S.-G. Ma, B. Li, M.-H. Wang, and Y.-C. Wang, "Velocity tracking control of a snake-like robot with a dynamics and control unified model," *Zidonghua Xuebao/Acta Automatica Sinica*, vol. 41, no. 11, pp. 1847–1856, 2015.
- [12] Q. Cheng, G. Wu, S. Li et al., "Static mechanism of a climbing snake robot navigating obstacle," *Machinery Design & Manufacture*, vol. 3, pp. 37–40, 2016.
- [13] A. Crespi and A. J. Ijspeert, "Online optimization of swimming and crawling in an amphibious snake robot," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 75–87, 2008.
- [14] Z. Lu, S. Ma, B. Li, and Y. Wang, "Snake-like robot controller with cyclic inhibitory CPG model," *Jixie Gongcheng Xuebao/Chinese Journal of Mechanical Engineering*, vol. 42, no. 5, pp. 137–143, 2006.
- [15] X. Lian, Guo. W., X. Lian et al., "CPG model parameters optimization based on genetic algorithm for snake-like robot," *Computure Engineering And Design*, vol. 36, pp. 1859–1864, 2015.
- [16] Q. Gao, Z.-L. Wang, W.-J. Hu, and L.-Y. Zhao, "Research on realization and environmental adaptability of serpentine locomotion for a snake robot," *Dalian Ligong Daxue Xuebao/Journal of Dalian University of Technology*, vol. 55, no. 2, pp. 203–208, 2015.