Research of Localization Method of Coal Mine Snake Detecting Robot

Yun Bai, Feng Xing, Hao-bo Wang, Xin-yue Liu

Abstract—**It is a difficult problem to locate the snake detecting robot in the closed complex environment of underground coal mine. The traditional localization methods usually depend on the estimation of complex ground parameters in coal mine tunnel, which leads to the complexity of establishing the localization model. In response to this compelling problem, we propose the new method to estimate the trajectory curvature and path angle of the robot, and a simple localization model of snake robot based on turning is established. On this basis, the dead reckoning localization method based on the combination of a high-precision 6-axis inertial navigation module MPU6050 and magnetic Hall encoders is proposed, which reduces the difficulty of traditional localization methods. In order to improve the localization accuracy of the robot, an algorithm combines Kalman filter and Deep learning is proposed. Firstly, the Kalman filter algorithm is used to eliminate the white Gaussian noise in the path angle signal of robot. Then, aiming at various drifts from low-frequency stage in the path angle signal, a prediction model of path angle output value based on LSTM (Long Short-Term Memory) deep neural network is established, which can predict the path angle output value for a period in the future, thereby achieving the robot's dead reckoning. The simulation and experiment results prove that the relative localization accuracy of the robot can be improved, in the simulated coal mine tunnel, the maximum relative error of the robot's location is 1.872×10^{-9} cm, and the maximum error of the path angle is 0.0028 rad, the determination coefficient R^2 is 1.000, which reflect that the established LSTM model has higher prediction accuracy, better performance, and stronger generalization ability than RBF (Radial Basis Function) neural network. The method proposed in this paper can provide a new idea for the localization problem of snake robots in the underground unstructured complex environment after the disaster.**

Index Terms—**coal mine, snake detecting robot, relative location, path angle, LSTM deep neural network**

Manuscript received June 12, 2023; revised May 06, 2024.

This work was supported by Doctoral start-up capital project of Xi'an University of Science and Technology (no.6310120503) and Experimental equipment R & D project of Xi'an University of Science and Technology(no.2030114005).

Yun Bai is a senior engineer of the Engineering training center, Xi'an University of Science & Technology, Xi'an, Shanxi, China, 710054, China (corresponding author; E-mail: 494361962@qq.com).

Feng Xing is a graduate student of the School of electrical and control engineering, Xi'an University of Science & Technology, Xi'an, Shanxi, China, 710054, China (Email: 631622407@qq.com).

Haobo Wang is a graduate student of the School of electrical and control engineering, Xi'an University of Science & Technology, Xi'an, Shanxi, China, 710054 (Email: 3173911616@qq.com).

Xinyue Liu is a graduate student of the School of electrical and control engineering, Xi'an University of Science & Technology, Xi'an, Shanxi, China, 710054 (Email: 2282786102@qq.com).

I. Introduction

THE localization of mobile robot is the premise for achieving its autonomous ability and the basic guarantee for the robot to complete predetermined tasks, such as path planning and navigation [1], [2], and [3]. Localization refers to the determination of a robot's location and pose relative to the global coordinate system within a two-dimensional or three-dimensional workspace. Robot localization technology can be divided into relative localization, absolute localization and combined localization [4], [5], and [6].

Relative localization: The principle of relative localization is to calculate the current pose of the robot by measuring the displacement and pose of the robot relative to the initial location. The method includes the dead reckoning localization and Odometry [7], [8], and [9], among which the dead reckoning localization method is most widely used. The advantage of the method is that realizing relative localization does not need to rely on external environmental information, and only needs to measure the displacement and pose of the robot relative to the initial location by using proprioceptive sensors such as gyroscopes and encoders, which has flexibility and good autonomy. The disadvantage of the method is that the accumulative error is implied in the calculation process, and the tracking error is accumulating with the increase of time and distance [10]. Therefore, this method is not suitable for long-distance localization. Odometry is used to calculate the real-time location and pose of the robot according to the measurement information of the relative reference feature. The advantage of the method is that the localization error does not accumulate with time. The disadvantage is that achieving robot localization relies on reference feature information, which also suffers from error accumulation. Patric Jensfelt et al. [11] applied the dead reckoning localization method to the localization of mobile robot, and proposed minimum environment model localization method based on the Kalman filter, and the robustness and accuracy of the robot localization system was improved. Lee Seok Ju et al. [12] used the smartphone carried by the robot for real-time two-dimensional code recognition, by which autonomous localization was realized, and as provided reference for the accurate relative localization of indoor mobile robot. Aiming at localization of the swarm robots, Wang Sunxin [13] proposed an improved relative localization method based on polar coordinate, which included distance calculation, relative direction calculation and coordinate transformation, and the method met the

localization requirements of swarm robots.

Absolute localization: The principle of absolute localization is to determine location and pose of the robot in the global coordinate system according to the relationship among the global coordinate system, the reference point and the pose of robot relative to the reference point [14], [15], and [16]. This method depends on workspace environment information, such as GPS (Global Positioning System) information, navigation landmarks or beacons, and map matching, etc. The advantage of absolute localization is high localization accuracy, and the localization error does not accumulate with time. The disadvantage is that hardware and maintenance costs are high, and this method may fail when the robot is unable to obtain the environmental reference information in advance. Se Stephen et al. [17] used scale-invariant landmarks to locate the mobile robot, and applied Kalman filter to track the 3D landmarks in the constructed 3D environment map, and finally obtained the more accurate location of the robot in the map. Kirmse H et al. [18] proposed a robot localization system based on the principle of trilateral method. The system did not require odometers or other auxiliary sensors, only a small base station was needed to build a map, and the system had low cost and high localization accuracy. Aiming at the localization problem for mobile robots in large-scale ongoing and featureless scenes, Xu Zhen et al. [19] built an artificial landmarks map, and proposed a nonlinear optimization algorithm based on graphs to reduce the uncertainty of the whole map, the average absolute localization accuracy of robot was improved.

Combined localization: Combined localization is the combination of the above two localization methods, complementing each other's strengths and weaknesses. Usually, the method of information fusion is used to combine the two methods to improve the localization accuracy of mobile robots. At present, the combination of multiple sensors such as inertial sensors, visual and image localization, SLAM (Simultaneous Localization and Mapping), and wireless localization are widely used in mobile robot localization [20], [21]. Aiming at non-systematic errors caused by unpredictable external environmental factors. Scheding S et al. [22] proposed integrating information from inertial sensors, laser rangefinders and encoders to correct the localization errors caused by robot wheel slip. Mu Zhou et al. [23] proposed a new ray-assisted generative adversarial model to automatically build wireless maps, which was used for the indoor WLAN (Wireless Local Area Network) intelligent target intrusion sensing and localization, and experimental results showed that this method could locate the target accurately. Matthias Eder et al. [24] presented a robot localization evaluation method based on particle filter and deep learning, which used particle filter to identify temporal patterns in particles and then combined these patterns with weak classifiers from particle sets and sensor perception to enhance the learning of localization estimator, which improved the performance of robot localization.

Due to different localization methods, localization sensors are divided into relative localization sensors and absolute localization sensors [25], [26]. Because the sensor has detection errors, the sensor data should be filtered to improve the localization accuracy of the robot. Sensor data filtering algorithms mainly include Kalman filter algorithm, wavelet denoising algorithm, processing methods based on error model, and intelligent control combined with the first three methods, etc. [8], [27], [28], and [29]. Hao Ming et al. [30] used a neural network to model and evaluate systematic and non-systematic errors of robot odometer. Li Ying et al. [31] applied wavelet neural network to identify the drift error of fiber optic gyro. The wavelet analysis method was adopted to eliminate high-frequency noise, and then adopted the wavelet neural network model was used to identify the drift error of fiber optic gyro. Aiming at the problem of large random errors in existing MEMS (Micro Electro Mechanical System) gyroscopes, Sun Wei et al. [32] proposed a non-stationary time series modeling and prediction method for MEMS gyroscope drift based on wavelet threshold denoising and RBF neural network, which effectively improved the accuracy of MEMS gyroscope inertial navigation system.

This paper mainly studies the localization problem of coal mine snake detecting robot in closed complex environment. our contributions are as follows:

Firstly, in order to avoid the complexity of traditional localization methods, a method of estimating the trajectory curvature and path angle of the robot is proposed to replace the conventional idea of estimating the complex ground parameters, and the simple localization model of the snake detecting robot based on turning motion is established.

Secondly, a dead reckoning localization method based on the combination of a high-precision 6-axis inertial navigation module MPU6050 and magnetic Hall encoders is proposed. The path angle of the robot is obtained through the 6-axis inertial navigation module MPU6050, and the motion displacement of the robot is obtained through the magnetic Hall encoders.

Thirdly, in order to obtain more accurate path angle, an algorithm combining Kalman filter and Deep learning is proposed. The white Gaussian noise contained in the output value of MPU6050 is eliminated by using Kalman filter algorithm. Aiming at sensor drift covering only part of the low-frequency stage, MPU6050 output value prediction model based on LSTM deep neural network [33], [34] is established, which can predict the output value of MPU6050 for a period in the future, and realize relative localization of the snake detecting robot in unknown underground tunnel environment.

The remainder of this paper is structured as follows: In Section 2, localization system modelling is described in detail. In Section 2.1, localization model of snake detecting robot based on dead reckoning is presented. In Section 2.2, MPU6050 output value denoising method based on Kalman filter is explained. In Section 2.3, prediction method of MPU6050 output value with drift based on LSTM deep neural network is proposed. In Section 3, simulation experiment and result analysis are discussed. Finally, Section 4 concludes the paper.

II. Localization System Modelling

The accuracy and stability of robot localization system determines the robot's ability to perform its behaviors. The underground coal mine environment after disaster is a closed and unstructured environment, which is characterized by the inability to receive external radio signals and realize global

satellite localization like GPS. In addition, there are large incentive interference and less available environmental information inside coal mine tunnel internal. Therefore, only autonomous localization can be adopted. In this paper, the dead reckoning localization method based on the combination of a high-precision 6-axis inertial navigation module MPU6050 and magnetic Hall encoders is adopted.

The flow chart of the dead reckoning localization method is shown in Fig. 1. The specific steps are as follows: The coal mine snake detecting robot is composed of five units and four orthogonal joints. The high-precision 6-axis inertial navigation module MPU6050 is configured in the center of the third unit, and the two direct-current reduction motors of the third unit are equipped with magnetic Hall encoders. Among them, the magnetic Hall encoder is used to calculate the velocity and displacement of the robot, path angle of the robot can be obtained through the high-precision 6-axis inertial navigation module MPU6050. Measurement accuracy of the robot's path angle is affected by the errors of MPU6050, so it needs to be filtered in the algorithm. Aiming at sensor drifts, MPU6050 output value prediction model based on LSTM is established, dead reckoning localization method is realized.

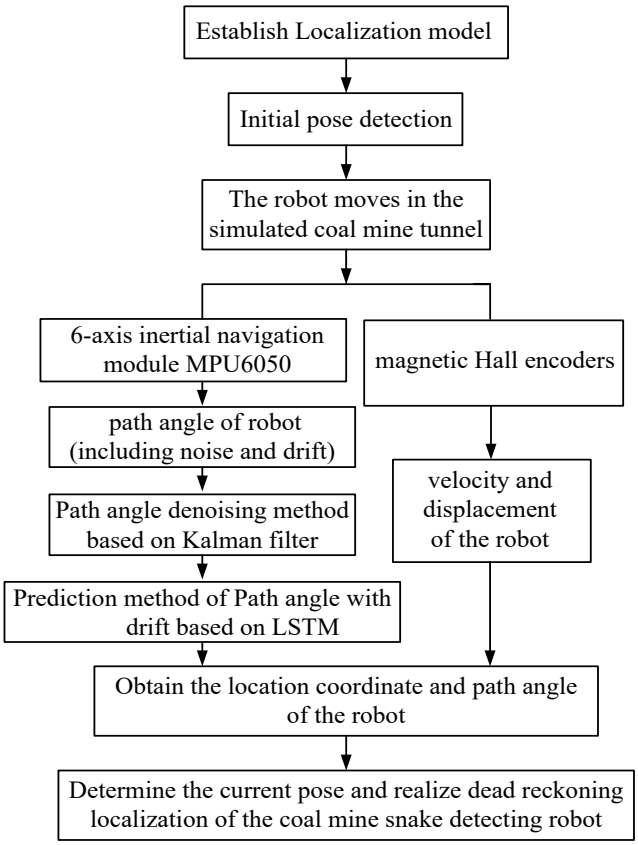

Fig. 1. The flow chart of the dead reckoning localization method

2.1 Localization model of snake detecting robot based on dead reckoning

The high-precision 6-axis inertial navigation module MPU6050 integrates a 3-axis MEMS gyroscope and a 3-axis MEMS accelerometer, due to affected by the accuracy of the MEMS device, the velocity and displacement estimates of the robot may deviate. Therefore, the MPU6050 is not suitable for solving the robot's velocity and displacement. The encoder is often considered as an error-free sensor because of its high measurement accuracy and small error. In order to achieve the function of dead reckoning, the magnetic Hall encoder is used to solve the velocity and displacement of the robot. Two magnetic Hall encoders assembled on two direct-current reduction motors respectively are used to calculate the actual linear motion trajectory of the robot. Setting the circumference of the robot's blades wheel is L_1, the number of encoding disk lines of magnetic Hall encoder is N', and the number of pulses of encoding disk output is M_1 in T_1 time, thus, the linear displacement of the wheel in T_1time is $s_1 = \frac{M_1}{N'} \cdot L_1$, linear velocity is $v_1 = \frac{M_1}{T_1 N'} \cdot L_1$.

The following discussion focuses on dead reckoning of snake detecting robot during turning movement. The localization of the robot is closely related to the ground environment of the tunnel. The uneven ground of the tunnel can cause vibration, impact and other disturbances when the robot moves, resulting in errors of the robot's pose, velocity and location, which may reduce the localization accuracy. To address the challenge, the method of estimating the curvature and path angle of the robot's trajectory is used to replace the idea of estimating the complex ground parameters, and a simple localization model is established.

For clarity, the notations of the main variables used in this paper are listed in Table I.

TABLE I
NOTATIONS OF THE MAIN VARIABLES USED IN THIS PAPER

Notation	Description
$X(k)$	The global pose of the robot at k time
$U(k-1)$	The control input at $k-1$ time
$(x(k), y(k), \theta(k))$	The location coordinate and path angle of the snake detecting robot at k time
$\Delta d(k-1)$	The linear displacement of the robot form $k-1$ time to k time
$\Delta\theta(k-1)$	The change angle of the robot's path form $k-1$ time to k time.
$\omega(k-1)$	The process noise at $k-1$ time
$Z(k)$	Systematic observation vector at k time
$v(k)$	The observation noise at k time
x_i^t	The input vector of input layer in the LSTM model at t time
b_h^{t-1}	The output value of the hidden layer at $t-1$ time
a_n^t , b_n^t	The input and output value of the input gate in the LSTM model at t time
w	The connection weight
f, g , g_1	The activation functions
a_φ^t , b_φ^t	The input and output value of the forget gate at t time
a_c^t , s_c^t	The input and state value of the memory cell at t time
a_ω^t , b_ω^t	The input and output value of the output gate at t time
b_c^t	The final output of the memory module at t time
a_m^t , y_m^t	The input and output vector of the output layer at t time
E	The objective function error
η	The learning rate
δ_i^t	The error signal of node i at t time
ω'	The weight update of the connection weight

The localization model describes the change of the robot state $X(k)$ under the action of control input $U(k-1)$. The pose change process of the snake detecting robot during turning movement is shown in Fig. 2.

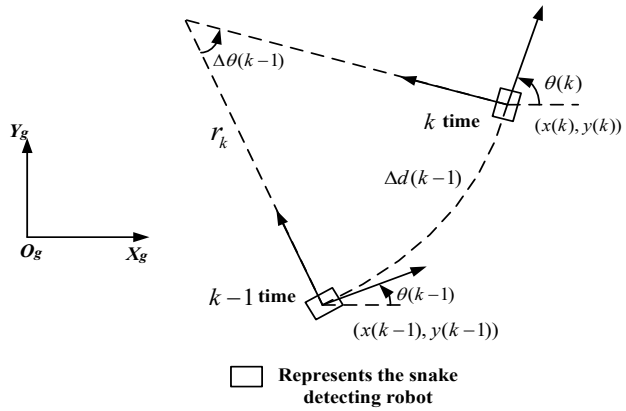

Fig. 2. Localization model of coal mine snake detecting robot

In Fig. 2, $\theta(k)$ is the path angle at k time.

Control input $U(k-1)$ is given by

$$U(k-1) = [\Delta d(k-1) \quad \Delta\theta(k-1)]^T \tag{1}$$

Where, $\Delta d(k-1)$ represents the linear displacement of the robot form $k-1$ time to k time, $\Delta\theta(k-1)$ is the change angle of the robot's path.

Robot state $X(k)$ is given by

$$X(k) = [x(k) \quad y(k) \quad \theta(k)]^T \tag{2}$$

The formula (2) represents the global pose of the robot, where, $(x(k), y(k))$ is the location coordinate which take the center of the third unit of the snack detecting robot as the reference point, and $\theta(k)$ is the path angle of the third unit. The dynamic equation of the robot localization system is established as follows:

$$X(k) = f(X(k-1), U(k-1)) + \omega(k-1) \tag{3}$$

Where, $\omega(k-1)$ is the process noise, which is assumed to be white Gaussian noise with zero mean, satisfying $\omega(k-1) \sim N(0, Q(k-1))$.

When the turning trajectory of the robot is an arc, the localization model of the robot is given by

$$X(k) = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} = \begin{bmatrix} x(k-1) + \frac{\Delta d(k-1)}{\Delta\theta(k-1)}(cos(\theta(k-1) + \Delta\theta(k-1)) - cos(\theta(k-1))) \\ y(k-1) + \frac{\Delta d(k-1)}{\Delta\theta(k-1)}(sin(\theta(k-1) + \Delta\theta(k-1)) - sin(\theta(k-1))) \\ \theta(k-1) + \Delta\theta(k-1) \end{bmatrix} + \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_\theta \end{bmatrix} \tag{4}$$

When the robot moves in a straight line, the path angle does not change, the input control $U(k-1) = \Delta d(k-1)$, and the localization model of the robot is given by

$$X(k) = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} \begin{bmatrix} x(k-1) + \Delta d(k-1)\, cos(\theta(k-1)) \\ y(k-1) + \Delta d(k-1)\, sin(\theta(k-1)) \\ \theta(k-1) \end{bmatrix} + \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_\theta \end{bmatrix} \tag{5}$$

2.2 MPU6050 output value denoising method based on Kalman filter

Dead reckoning localization method is based on the determination of robot localization model. According to the initial pose given by the 6-axis inertial navigation module MPU6050 and magnetic Hall encoder, the current pose of the robot is obtained by solving the displacement and direction of the robot relative to the initial pose. Because the high-precision 6-axis inertial navigation module MPU6050 has the characteristics of MEMS, it is suitable for directional localization, therefore path angle of the robot can be obtained through it. The encoder is often considered as an error-free sensor because of its high measurement accuracy and small error, so the robot's velocity and displacement are solved by magnetic Hall encoder.

As far as MPU6050 is concerned, its errors are mainly manifest as white Gaussian noise, drift (including temperature drift and constant drift), etc. In the frequency domain, white Gaussian noise basically covers the entire frequency range, whereas temperature drift and constant drift only cover some low-frequency ranges. Therefore, the Kalman filter algorithm is proposed in this paper to control the internal noise of MPU6050, mainly white Gaussian noise, which affects the accuracy of the robot's path angle.

Aiming at the state parameter path angle $\theta(k)$ in the robot's localization model, the Kalman filter is used to establish the path angle movement model as follows:

$$\theta(k) = \theta(k-1) + \Delta\theta(k-1) \tag{6}$$

According to Fig. 2, $\Delta\theta(k-1) = \frac{\Delta d(k-1)}{r_k}$, where r_k is the arc radius, so

$$\theta(k) = \theta(k-1) + \frac{\Delta d(k-1)}{r_k} \tag{7}$$

Systematic observation vector Z is given by

$$Z = \theta(k) \tag{8}$$

Let system state vector $X(k) = \theta(k)$, control input $U(k) = \frac{\Delta d(k)}{r_k}$, ω is system Gaussian noise, v is observation noise. The mathematical model is as follows [35], [36]:

$$\begin{cases} X(k) = X(k-1) + U(k-1) + \omega(k-1) \\ Z(k) = X(k) + v(k) \end{cases} \tag{9}$$

The mathematical model state is estimated by using Kalman filter system. In the experiment, the process noise variance of the system is $Q_k = std(\omega).\hat{}\,2$, and observed noise variance is $R_k = std(v).\hat{}\,2$.

2.3 Prediction method of MPU6050 output value with drift based on LSTM deep neural network

As described in Section 2.2, the high-precision 6-axis inertial navigation module MPU6050 is used in the localization system of coal mine snake detecting robot, path angle of the robot can be obtained by using it. Kalman filter algorithm is used to eliminate the influence of white Gaussian noise in the output value of MPU6050. The experimental research shows that the Kalman filter can effectively eliminate white Gaussian noise, but various drifts still exist, among which the constant value drift has the greatest influence on the error of MPU6050. When solving for the pose angle of the snake detecting robot, the constant value drift is treated as a constant and accumulated continuously in each solution, resulting in the accumulation of the robot pose angle error and a decrease in accuracy of robot localization over time. Therefore, aiming at the MPU6050 output value containing drifts, a MPU6050 output prediction algorithm based on LSTM deep neural network is proposed in this paper. This algorithm can predict the output value of MPU6050 for a period in the future, and obtain more accurate path angle of the robot, so improve the accuracy of the robot localization system.

2.3.1 Analysis of LSTM deep neural network Principle

The LSTM model is a new type of Deep learning neural network [37], [38], which is improved based on the RNN (Recurrent Neural Network). In order to solve the gradient explosion or gradient dispersion problem that RNN is easy to appear, the LSTM model introduces the threshold mechanism to control the accumulation speed of information, and at the same time, the model can choose to forget the information accumulated. The LSTM model is the cyclic neural network under this threshold mechanism, which is also called long and short-term memory neural network. This model uses LSTM memory module to replace the hidden layer nodes of RNN, so that it has long-term memory. For the time series data, the LSTM model has more ideal prediction effect, which has been successfully applied to the stock price prediction [39], [40] and other fields, but it is rarely applied to robot localization at present.

As shown in Fig. 3, the LSTM deep neural network is composed of input layer, hidden layer and output layer. For convenience of description, only one hidden layer is given here. Among them, the hidden layer contains memory module, which can store and transmit information for a long time. Each memory module consists of input gate, forget gate, output gate and memory cell, among which, the three gates generate a series of number between 0 and 1 under the action of activation function, which are used to control the opening and closing of the gate. The 0 indicates that the gate is closed and the 1 indicates that the gate is opened. The function of the input gate is to transfer the information of the input layer to the memory module of the hidden layer; The function of the forget gate is to select and retain the historical information in the memory module at the current time; The function of the output gate is to transmit the information form the memory module; The memory cell represents the memory of neuronal state. The design of three gates and memory cell enables the LSTM deep neural network to selectively control the gates of each layer according to the historical data and network error, thereby achieving long-term memory and updating of historical data, extracting data features, and achieving high accuracy in network predictions.

In Fig. 3, x_i^t, x_i^{t-1}, and x_i^{t-n} are the input vectors of input layer at t, $t-1$, and $t-n$ time respectively, (In this paper the input vectors are composed form the output value by MPU6050 at the previous $t-4$ time), b_h^t, b_h^{t-1}, and b_h^{t-n} are the output values of the hidden layer at t, $t-1$, and $t-n$ time respectively, f denotes the activation function of each control gate, g and g_1 are the input and output activation functions of the memory cell, s_c^t, s_c^{t-1}, and s_c^{t-n} denote the state value of memory cell at t, $t-1$, and $t-n$ time respectively, and y_m^t is the output vector of output layer in the LSTM model, (In this paper y_m^t is calculated by the LSTM model and used as the MPU6050 output value at t time). After the model structure is determined, the BPTT (Back Propagation Through Time) algorithm is used for training the model, the model parameters such as network weights, activation functions and learning rates etc. are determined, and finally the predicted values of MPU6050 output are acquired at t time.

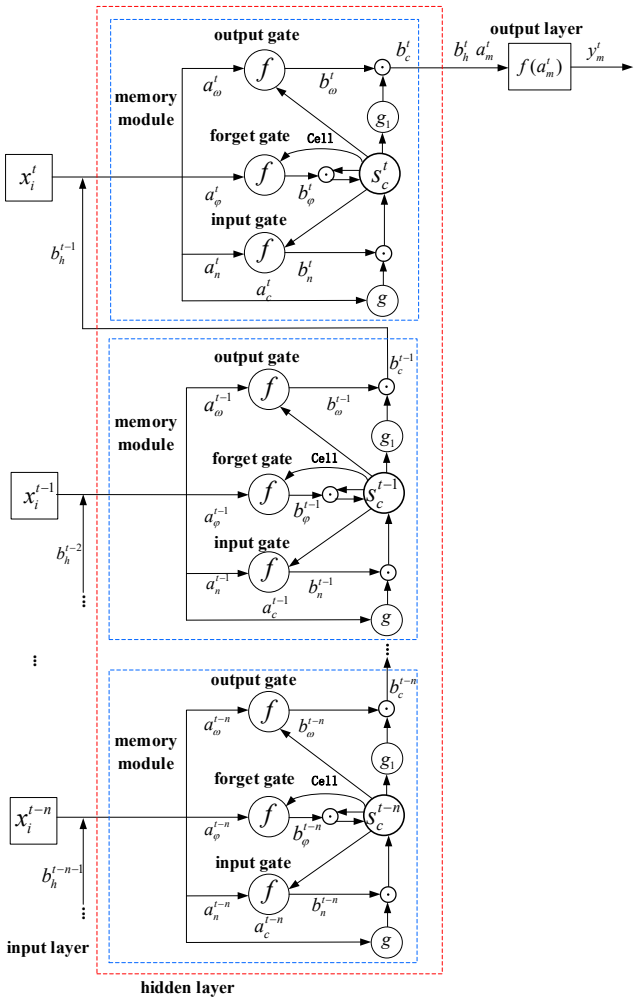

Fig. 3. Structure of LSTM deep neural network model

Training LSTM model with BPTT algorithm includes two processes, which are follows: the information forward propagation and the error back propagation, which will be explained below.

(1) Information forward propagation process

① Input gate

The input value of the input gate is consisted of three items at t time, which are the weighted sum of the input vector of the input layer, the weighted sum of the output value of the hidden layer node at $t-1$ time, and the weighted sum of the state value of the memory cell at $t-1$ time. Its expression is defined by

$$a_n^t = \sum_{i=1}^{I} w_{in} x_i^t + \sum_{h=1}^{H} w_{hn} b_h^{t-1} + \sum_{c=1}^{C} w_{cn} s_c^{t-1} \tag{10}$$

Where, subscript n denotes the input gate, I represents the total number of neurons in the input layer, w_{in} expresses the connection weight between the input layer and the input gate, x_i^t means the input vector of the input layer at t time, H denotes the total number of memory modules in the hidden layer, w_{hn} refers to the connection weight between the hidden layer and the input gate at $t-1$ time, b_h^{t-1} signifies the output value of the hidden layer at $t-1$ time, C denotes the total number of memory cell in the per memory module, w_{cn} is the connection weight between memory cell and input gate at $t-1$ time, s_c^{t-1} indicates the state value of memory cell at $t-1$ time.

The output value of the input gate at t time is expressed as

$$b_n^t = f(a_h^t) \tag{11}$$

② Forget gate

The input of forget gate is also composed by three parts, its expression is represented as

$$a_\varphi^t = \sum_{i=1}^{I} w_{i\varphi} x_i^t + \sum_{h=1}^{H} w_{h\varphi} b_h^{t-1} + \sum_{c=1}^{C} w_{c\varphi} s_c^{t-1} \tag{12}$$

Where, the subscript φ denotes the forget gate, $w_{i\varphi}$ represents the connection weight between the input layer and the forget gate at t time; $w_{h\varphi}$ is the connection weight between the hidden layer and the forget gate at $t-1$ time; and $w_{c\varphi}$ indicates the connection weight between the memory cell and the forget gate at $t-1$ time.

The output value of the forget gate at t time is denoted by

$$b_\varphi^t = f(a_\varphi^t) \tag{13}$$

③ Memory cell

The input value of the memory cell at t time includes the weighted sum of the input vector of the input layer and the weighted sum of the output value of the hidden layer node at $t-1$ time, and we can get

$$a_c^t = \sum_{i=1}^{I} w_{ic} x_i^t + \sum_{h=1}^{H} w_{hc} b_h^{t-1} \tag{14}$$

Where, subscript c denotes memory cell, w_{ic} is the connection weight between the input layer and the memory cell at t time, and w_{hc} indicates the connection weight between the hidden layer and the memory cell at $t-1$ time.

It can be seen from Fig. 3 that the state value of the memory cell at t time is consisted of two parts: the dot product of the forget gate output and the state of the memory cell at $t-1$ time, and the dot product of the output of the input gate and the input of the memory cell excited by function g at t time. the expression is given by

$$s_c^t = b_\varphi^t \odot s_c^{t-1} + b_n^t \odot g(a_c^t) \tag{15}$$

④ Output gate

The input value of the output gate at t time is also composed by three items, i.e., the weighted sum of the input vector of the input layer, the weighted sum of the output value of the hidden layer node at $t-1$ time, and the weighted sum of the state value of the memory cell at t time, and its expression is defined by

$$a_\omega^t = \sum_{i=1}^{I} w_{i\omega} x_i^t + \sum_{h=1}^{H} w_{h\omega} b_h^{t-1} + \sum_{c=1}^{C} w_{c\omega} s_c^t \tag{16}$$

Where, the subscript ω represents the output gate, $w_{i\omega}$ denotes the connection weight between the input layer and the output gate at t time, $w_{h\omega}$ is the connection weight between the hidden layer and the output gate at $t-1$ time, and $w_{c\omega}$ means the connection weight between the memory cell and the output gate at t time.

The output value of the output gate at t time is given by

$$b_\omega^t = f(a_\omega^t) \tag{17}$$

The final output of the memory module at t time is the dot product of the output value of the output gate and the state value of the memory cell excited by function g_1 at t time, and its expression is represented as

$$b_c^t = b_\omega^t \odot g_1(s_c^t) \tag{18}$$

⑤ Output layer

The input vector of the output layer is expressed as

$$a_m^t = \sum_{c=1}^{H} w_{cm} b_c^t \tag{19}$$

Where, the subscript m is the output layer, and w_{cm} denotes the connection weight between the memory cell and the output layer at t time.

The final output of the output layer is described as

$$y_m^t = f(a_m^t) \tag{20}$$

(2) The error back propagation process

The purpose of error back propagation is to adjust the weight according to the error, so that the output of LSTM model is close to the expected output and achieve the expected prediction effect.

Let E denotes the objective function error, according to gradient descent method and the chain rule, the derivative is obtained in turn and the weight is updated, the adjustment and update formula of weight is turned into

$$w_{ij}' = w_{ij} - \eta \nabla E(w_{ij}) \tag{21}$$

Where, η is the learning rate, $\eta > 0$.

Let δ_i^t denotes the error signal of the node i at t time, $\delta_i^t = \frac{\partial E}{\partial a_i^t}$, and then:

$$\nabla E(w_{ij}) = \frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial a_j^t} \frac{\partial a_j^t}{\partial w_{ij}} = \delta_j^t b_j^t \tag{22}$$

$$w_{ij}' = w_{ij} - \eta \delta_j^t b_j^t \tag{23}$$

From formula (23), we can see that when using BPTT algorithm to update the weight, the correction process of connection weight at different time of the same hidden layer is considered, which is the difference between BPTT algorithm and ordinary BP algorithm.

For example, the weight update formula of the connection weight w_{ic} between the input layer and the memory cell at t time is gained as

$$w_{ic}' = w_{ic} - \eta \delta_c^t b_c^t \tag{24}$$

To sum up, LSTM deep neural network is train by using BPTT algorithm, and the steps are as follows:

(1) The network is initialized and all weights are set to small random numbers.

(2) The training samples at t time is inputted to the input layer, and according to expression $\sum_{i=1}^{I} w_{in} x_i^t$, the weighted sum of the input vectors of the input layer is calculated.

(3) We input the weighted sum of the input vector of the input layer, the weighted sum of the output value of the hidden layer node at $t-1$ time, and the weighted sum of the state value of the memory cell at t and $t-1$ time , to the input gate, forget gate, memory cell and output gate respectively, according to formulas (10) to (18), the final output of memory cell at t time is calculated.

(4) The final output of memory cell is inputted to the output layer, according to formulas (19) and (20), the output value

of output layer at t time is calculated.

(5) The error between the output value of the output layer and the expected output is calculated, which is taken as the objective function, and the derivative is obtained in turn according to gradient descent method and the chain rule, and each weight is updated according to formula (23).

(6) We judge whether the maximum training times of the network are reached, and judge whether the network error accuracy meets the requirements. If the maximum number of training times has been reached, the training is over, otherwise skip to step (2) to continue; If the network error accuracy meets the requirements, the training ends, otherwise skip to step (2) to continue.

2.3.2 MPU6050 output prediction model based on LSTM deep neural network

Based on the analysis of the LSTM deep neural network principle, and combined with the characteristics that the MPU6050 output data is time series, MPU6050 output prediction model based on LSTM deep neural network is established. The specific steps are as follows:

(1) As described in section 2.2, the high-frequency noise of MPU6050 output data is denoised by Kalman filter, and the output data of MPU6050 containing only low-frequency drift is obtained. Because these output data have the characteristics of time series, the input and output parameters of the LSTM model can be determined. Let the input $x_i = [x_{i-1}, x_{i-2}, x_{i-3}, x_{i-4}]^T$ denotes the output value of MPU6050 at the first four moments of the current time, namely, there are four nodes in the input layer of LSTM, there is only one node in the output layer of LSTM, namely, the output y^t is the MPU6050 predicted output value at the current time.

(2) Determining the model structure. The number of hidden layer nodes is determined by the empirical formula $m \approx \sqrt{n+l} + a'$, where m represents the number of hidden layer nodes, n is the number of input layer nodes, l denotes the number of output layer nodes, and a' is a constant between 1 and 10. Therefore, it is determined that the number of hidden layer nodes is 8. Since the output time series of MPU6050 is one-dimensional nonlinear numerical data, according to the empirical method or trial-and-error method, when the number of hidden layers is 1, it can be well meet the experimental requirements. Therefore, the network structure of the model is 4-8-1.

(3) According to the time needed for the snake robot to go through the simulated coal mine tunnel and the collected MPU6050 sample data, the training set samples and test set samples of the LSTM model are determined.

(4) Sample normalization. The maximum and minimum value method is adopted to preprocess the input and output data.

Namely:

$$Y = (Y_{max} - Y_{min}) \frac{X - X_{min}}{X_{max} - X_{min}} + Y_{min} \tag{25}$$

The data is normalized to between the maximum value Y_{max} and the minimum value Y_{min} by Formula (25), where, X is the sample data, X_{max} and X_{min} are the maximum value and minimum value of the sample data respectively, and Y is the normalized data.

(5) According to formulas (10) to (23), the training set is used to train the LSTM model, and the best parameters is determined, including the selection of the activation function, the learning rate, the error objective function, and the number of iterations etc. The output value of MPU6050 after the Kalman filter denoise at the current time is regarded as the expected value, and training is stop when the expected error or the maximum number of iterations is reached.

(6) The trained LSTM model is used to predict the testing set, and obtain the predicted data. The final predicted value of the MPU6050 at the current time is obtained by inverse normalization of prediction data.

(7) In order to verify the feasibility and accuracy of the built prediction model and evaluate its prediction performance, this paper mainly uses MAE (Mean Absolute Error), MSE (Mean Square Error), and determination coefficient R^2 to evaluate the predictive effect of the model. The specific formulas (26), (27) and (28) are given by

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f(\theta_i) - y_i| \tag{26}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(f(\theta_i) - y_i)^2 \tag{27}$$

$$R^2 = \frac{(n\sum_{i=1}^{n} f(\theta_i)y_i - \sum_{i=1}^{n} f(\theta_i)\sum_{i=1}^{n} y_i)^2}{(n\sum_{i=1}^{n} f(\theta_i)^2 - (\sum_{i=1}^{n} f(\theta_i))^2)(n\sum_{i=1}^{n} y_i{}^2 - (\sum_{i=1}^{n} y_i)^2)} \tag{28}$$

Where, n denotes the number of samples, $f(\theta_i)$ is the model output and y_i represents the expected output. MAE can accurately reflect the actual prediction error. The smaller the MAE is, the better the model performance is. Similarly, the smaller the MSE is, the better the model performance is. R^2 is taken within the range of [0,1]. The closer the value of R^2 is to 1, the better of the model performance is. The closer the value of R^2 is to 0, the worse of the model performance is.

III. Simulation Experiments and Results Analysis

3.1 Kalman filter simulation experiment

In this paper, path angle of the robot can be obtained through the high-precision 6-axis inertial navigation module MPU6050. In order to remove the white Gaussian noise from the output data of the MPU6050, the Kalman filter algorithm is used. In order to verify the effectiveness of the Kalman filter algorithm, the experiment is carried out in the MATLAB simulation environment.

In the Kalman filter simulation experiment, 10,000 samples are taken for the experiment. As shown in Fig. 4, the abscissa represents the sampling point, the ordinate represents the path angle of the snake robot, and the blue line expresses the path angle measurement value with white Gaussian noise, and the red line expresses the effect after Kalman filtering. The simulation experiment result demonstrates that the influence of white Gaussian noise is largely eliminated in filtered path angle measurement values. Thus, the white Gaussian noise in the output data of the inertial sensor can be effectively suppressed by Kalman filtering.

3.2 Prediction simulation experiment based on LSTM deep neural network

The output data of the MPU6050 denoised by Kalman filter contains various drifts. In order to obtain more accurate path angle of the robot, aiming at the output data of the MPU6050 with drifts, LSTM deep neural network is used to predict the

output data of MPU6050 for a period in the future.

According to the steps of establishing the MPU6050 output prediction model based on the LSTM deep neural network, i.e., the specific steps (1) and (2) in section 2.3.2, the structure of LSTM deep neural network is calculated and determined as 4-8-1.

According to step (3) in section 2.3.2, the MPU6050 sampling period is 20 ms by the experiment. It takes 11.56 s for the robot to go through the semi-oval simulated coal mine tunnel with a perimeter of 291.2 cm. It is determined that 578 sample data are collected, of which, 423 data are taken as training set samples and the remaining 155 data are test set samples. Because of the structure of LSTM deep neural network is 4-8-1, that is, the current measured values are predicted though the measured values at the first four times. Therefore, the 155 measured values used as the test set samples are divided into 5 portions, so the number of predicted values should be 31.

According to step (4) in section 2.3.2, the sample data is normalized.

According to step (5) in section 2.3.2, the Sigmoid is selected as the activation function of each control gate of the LSTM, and the Tanh is choosed as the input and output activation function of the memory unit of the LSTM, the learning rate is 0.01, the error objective function is MSE, the number of iteration times are 100 times.

According to step (6) in section 2.3.2, the predicted data obtained by LSTM are inversely normalized, and then the path angle of the robot moving in the coal mine tunnel can be obtained.

According to step (7) in section 2.3.2, the prediction effect of LSTM deep neural network is tested. Among the traditional feedforward neural networks such as BP (Back Propagation), RBF (Radial Basis Function), ELM (Extreme Learning Machine), GRNN (Generalized Regression Neural Network), and PNN (Probabilistic Neural Network), etc., RBF has excellent performance, unique best approximation characteristics, and there is no local optimal solution. However, for the time series data by MPU6050 output in this paper, the prediction effect of LSTM model is more ideal. Therefore, the prediction effect of LSTM and RBF neural network are compared. The comparison result is shown in Fig. 5, where the abscissa represents the predicted value sequence, and the ordinate represents the path angle of robot. It can be seen form Fig. 5 that compared with RBF, LSTM can obtain better prediction results.

Fig. 6 and Fig. 7 show the prediction error curves of LSTM and RBF, respectively. The abscissa represents the prediction value sequence, and the ordinate represents the norm of the error matrix between the actual output and the expected output of network. When using LSTM for prediction, the maximum error is 0.0028 and the minimum error is 2.173×10^{-5}. When using RBF for prediction, the maximum error is 5.423 and the minimum error is 0.014.

Table II shows the comparison of MAE (Mean Absolute Error), MSE (Mean Square Error) and determination coefficient R^2 when using LSTM, RBF, ELM, and BP to predict the output data of MPU6050. It can be seen from Table II that LSTM has higher prediction accuracy and stronger generalization ability than RBF, ELM, and BP.

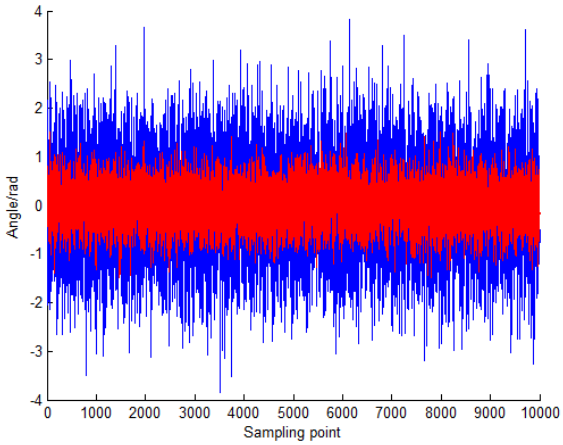

Fig. 4. Kalman filtering simulation result

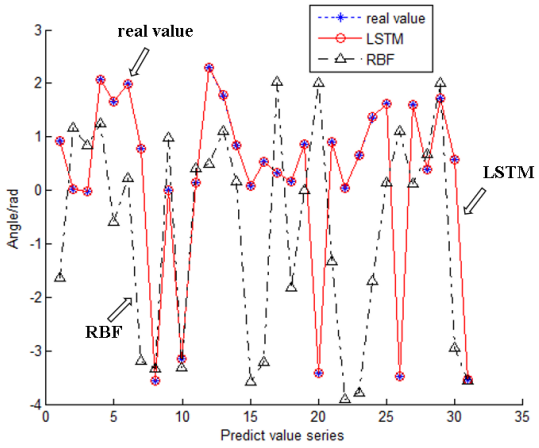

Fig. 5. Comparison of test set samples prediction result

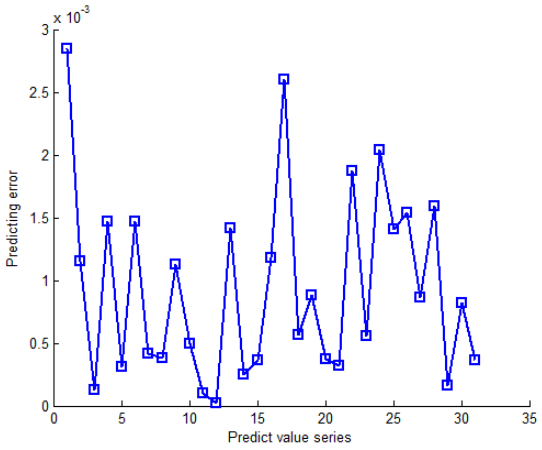

Fig. 6. Prediction error curve of LSTM

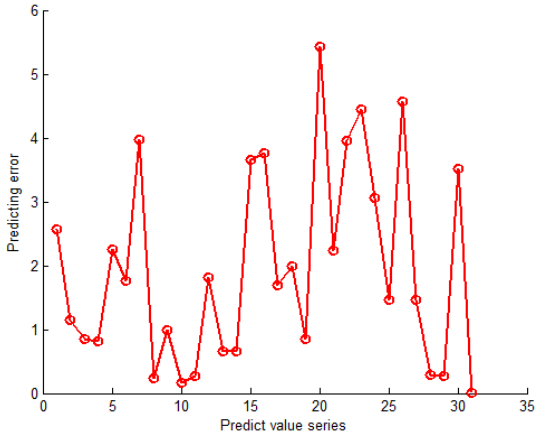

Fig. 7. Prediction error curve of RBF

TABLE II

COMPARISON OF PREDICTION EFFECTS OF LSTM, RBF, ELM, AND BP FOR MPU6050 OUTPUT DATA

Effect Evaluation	MAE	MSE	R^2
LSTM	9.390×10^{-4}	1.4152×10^{-6}	1.000
RBF	1.9621	6.146	0.056
ELM	1.9834	6.312	0.053
BP	2.245	7.032	0.048

The snake robot is put into the simulated coal mine tunnel for the experiment. As shown in Fig. 8, the simulated coal mine tunnel is semi-oval. Fig. 9 shows the change curve interface of the path angle of the robot as it moves in the coal mine tunnel displayed by the upper computer. As shown in Fig. 9, the blue curve represents the change of the path angle of the robot. Correspondingly, the experimental data display interface during robot moving is shown in Fig. 10.

The following simulation results are further obtained: The trajectory of simulated coal mine tunnel is shown in Fig. 11, the trajectory is semi-oval, the long axis is 80 cm, the short axis is 60 cm, and the perimeter is 291.2 cm. The abscissa represents the X-axis coordinate of the tunnel trajectory, and the ordinate represents the Y-axis coordinate of the tunnel trajectory. The comparative experiment between the actual location of the robot and the estimated location is shown in Fig. 12, the abscissa represents the X-axis coordinate of the robot's location, and the ordinate represents the Y-axis coordinate of the robot's location, of which, the black line represents the actual moving trajectory of the robot recorded by the magnetic Hall encoder, while the blue line represents the location trajectory of the robot calculated by using the MPU6050 through the Kalman filter algorithm. Here, the robot's location is obtained by the quadratic integral of the measured acceleration for time in the navigation coordinate system. The red line represents the trajectory of the robot learned through the LSTM deep neural network. The green line represents the trajectory of the robot learned through the RBF neural network. It can be seen from Fig. 12 that the trajectory learned through the LSTM is closer to the actual trajectory of the robot. Fig. 13 shows the location error after LSTM learning, where the abscissa represents the sampling point, and the ordinate represents the relative error of the robot's location. The maximum location error is 1.872×10^{-9} cm, and the minimum location error is 3.299×10^{-12} cm. Fig. 14 shows the location error after RBF learning, where the abscissa represents the sampling point, and the ordinate represents the relative error of the robot's location. The maximum location error is 14.119 cm, and the minimum location error is 1.612×10^{-5} cm. It can be seen that LSTM has higher learning accuracy compared to RBF.

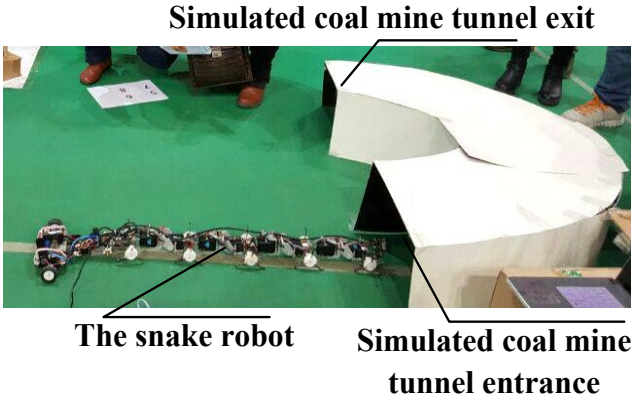

Fig. 8. Snake robot entering simulated coal mine tunnel

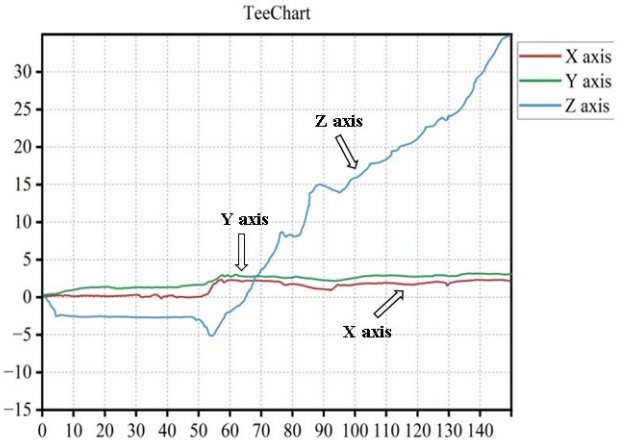

Fig. 9. Display interface of robot path angle change

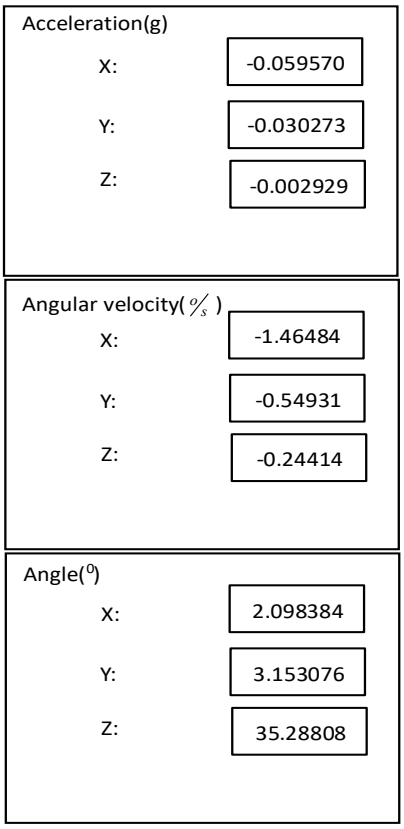

Fig. 10. Experimental data display interface of robot

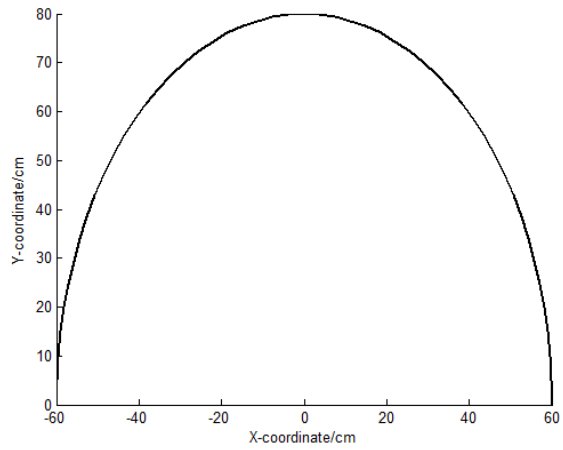

Fig. 11. Trajectory of simulated coal mine tunnel

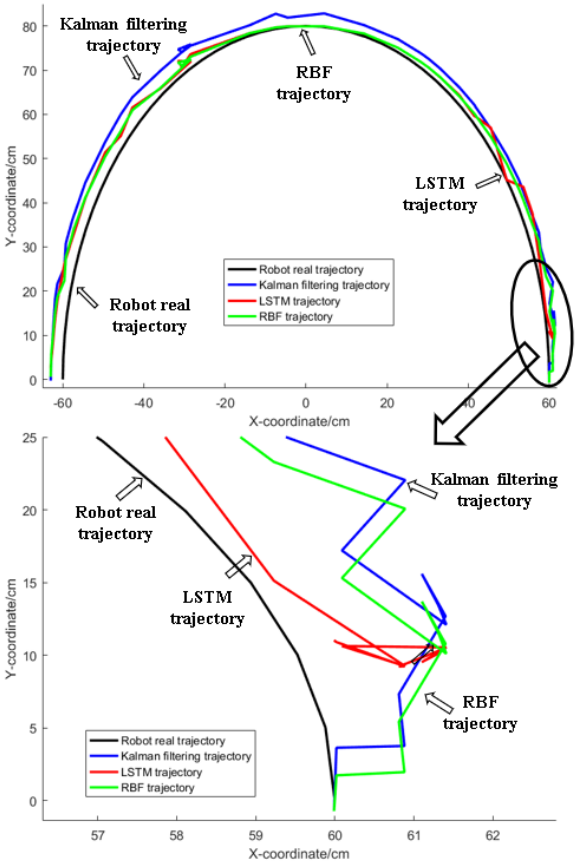

Fig. 12. Simulation comparison of robot location curve

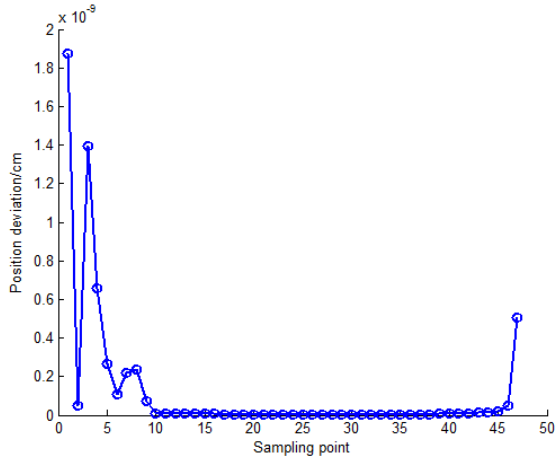

Fig. 13. Location error curve after LSTM learning

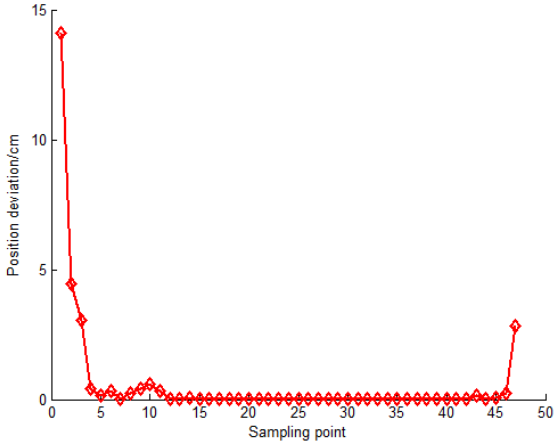

Fig. 14. Location error curve after RBF learning

IV. Conclusion

In this paper, the localization problem of snake detecting robot in the coal mine tunnel is studied.

(1) Aiming at the special environment of uneven ground in underground tunnel of coal mine, the method for estimating the trajectory curvature and path angle of the robot is proposed instead of estimating the complex ground parameters in the coal mine tunnel, the localization model of snake detecting robot based on turning is established, which avoids the complexity of traditional establishing localization model for snake robot. Then, a dead reckoning localization method of coal mine snake detecting robot based on the combination of a high-precision 6-axis inertial navigation module MPU6050 and magnetic Hall encoders is proposed, which reduces the difficulty of conventional localization methods.

(2) The algorithm combining Kalman filter and Deep learning is proposed, and its main novelty lies in obtaining more accurate path angles and improving the localization accuracy of robot, the experimental results show that the maximum relative error of the robot's location is 1.872×10^{-9} cm, and the minimum relative error is 3.299×10^{-12} cm. The maximum error of the path angle is 0.0028 rad, and the minimum error is 2.173×10^{-5} rad. The determination coefficient R^2 is 1.000. The above evaluation metrics indicate that the established LSTM model has higher prediction accuracy, better performance, and stronger generalization ability than RBF model. The dead reckoning problem of snake detection robot is solved in unknown underground coal mine tunnel environment. The localization method proposed in this paper can provide theoretical support for the dead reckoning problem of coal mine snake detecting robot in the underground unstructured complex environment after the disaster, laying the foundation for the snake robot's localization research in other disaster environments.

(3) Because the experiments in this paper are carried out in the simulated coal mine tunnel, it is verified that the method proposed in this paper has advantages in realizing short-distance accurate localization. Due to the internal error of the localization sensors, the track error will accumulate with the increase of time and distance, and the long-distance localization of the snake detection robot will become a challenge. Therefore, achieving the long-distance accurate localization of the robot will be the next research direction.

REFERENCES

[1] Kim YJ, Kang HH, Lee, SS, et al. Distributed Finite Memory Estimation From Relative Measurements for Multiple-Robot Localization in Wireless Sensor Networks[J]. IEEE ACCESS, 2022, (10): 5980-5989.

[2] Tan Xiangquan, Wu Qingwen. Indoor mobile robot localization based on multi-sensor fusion technology [J]. Transducer and Microsystem Technologies, 2021, 40(8): 53-56.

[3] Gao Qi, Bai Jinniu. Research on localization algorithms for mobile robots based on multi-sensor fusion [J]. Automation and Instrumentation, 2023 (03): 295-300.

[4] Chung MA, Lin CW. An Improved Localization of Mobile Robotic System Based on AMCL Algorithm[J]. IEEE Sensors Journal, 2022, (1): 900-908.

[5] Lie Yu, Lei Ding, and Yukang Tian, "Tracking Control for Intelligent Tracing Car based on Novel Path Tracking Strategy," IAENG International Journal of Applied Mathematics, vol. 53, no.2, pp664-669, 2023.

[6] Matthias Eder, Michael Reip, Gerald Steinbauer. Creating a robot localization monitor using particle filter and machine learning approaches [J]. Applied Intelligence, 2021: 1-15.

[7] Wang Weihua. Research on Localization Technology for Mobile Robots[D]. Wuhan: Huazhong University of Science and Technology, 2005.

[8] Cai Lihua, Fang Haifeng, Li Yunwang, et al. Research of self-localization method of mine rescue robot[J]. Industry and Mine Automation, 2015, 41(7): 63-67.

[9] Yang Guang. Research of Mobile Robot Localization Based on Multi-sensor[D]. Dalian: Dalian University of Technology, 2021.

[10] Zhang Qianrong, Li Yi. Indoor positioning method of multi-sensor fusion in NLOS environment [J]. Computer engineering and Design, 2023,44 (03): 732-738.

[11] Patric Jensfelt, Henrik I. Christensen. Pose Tracking Using Laser Scanning and Minimalistic Environmental Models[J]. IEEE Transactions on Robotics and Automation, 2001, 17(2): 34-48.

[12] Lee Seok Ju, Lim Jongil, Tewolde, Girma, et al. Autonomous tour guide robot by using ultrasonic range sensors and QR code recognition in indoor environment[C]. IEEE International Conference on Electro Information Technology, EIT 2014: 410-415.

[13] Wang Sunxin, Li Yuhua, Zhang, Shaohua, et al. Relative localization of swarm robotics based on the polar method[J]. International Journal of Advanced Robotic Systems, 2022, 19(1): 1-13.

[14] GhaemiDizaji Manizheh, Dadkhah Chitra, Leung Henry. Efficient robot localization and SLAM algorithms using Opposition based High Dimensional optimization Algorithm[J]. Engineering Applications of Artificial Intelligence, 2021, (104).

[15] Bertolli, Federico, Fiorini, et al. Visual Slam-Mobile Robot Localization With Environment Mapping [C]. 8th International IFAC Symposium on Robot Control, SYROCO 2006: 286-291.

[16] Bertolli, Federico, Jensfelt, et al. SLAM using visual scan-matching with distinguishable 3D points[C] IEEE International Conference on Intelligent Robots and Systems, 2006: 4042-4047.

[17] Se Stephen, Lowe David, and Little Jim. Mobile robot localization and mapping with uncertainty using scale−invariant visual landmarks[J]. The International Journal of Robotics Research, 2002, 21(8): 735-758.

[18] Kirmse H, Hennig M, Janschek K. Global localization of an indoor mobile robot with a single base station[C]. 4th IFAC Symposium on Mechatronic Systems, MX 2006: 578-583.

[19] Xu Zhen, Guo Shuai, Song Tao, et al. Localization of Mobile Robot Aided for Large-Scale Construction Based on Optimized Artificial Landmark Map in Ongoing Scene[J]. Cmes-Computer Modeling in engineering & Sciences, 2022, 130(3): 1853-1882.

[20] Petros Spachos, Konstantinos N. Plataniotis. BLE Beacons for Indoor Positioning at an Interactive IoT-Based Smart Museum[J]. IEEE Systems Journal, 2020, 14(3): 3483-3493.

[21] Shi Junyi, Zha Fusheng, Sun Lining, et al. A Survey of Visual-Inertial SLAM for Mobile Robots[J]. Robot, 2020,42(6): 734-748.

[22] Scheding S, Dissanayake G, Nebot E M, et al. An experiment in autonomous navigation of an underground mining vehicle[J]. IEEE Transactions on Robotics and Automation, 1999, 15(1): 85-95.

[23] Mu Zhou, Yixin Lin, Nan Zhao, et al. Indoor WLAN Intelligent Target Intrusion Sensing Using Ray-Aided Generative Adversarial Network [J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2020, 4 (1): 61-73.

[24] Matthias Eder, Michael Reip, Gerald Steinbauer. Creating a robot localization monitor using particle filter and machine learning approaches[J]. Applied Intelligence, 2022, 52 (6): 6955-6969.

[25] Martinelli A. Estimating the odometry error of a mobile robot during navigation[C]. In Proceedings of the 1st European Conference on Mobile Robots, Warszawa, Poland: Zturek Research Scientific Inst. Press, 2003: 218-223.

[26] B. Sadeghi Bigham, S. Dolatikalan, A. Khastan. Minimum landmarks for robot localization in orthogonal environments [J]. Evolutionary Intelligence, 2021, 1-4.

[27] Carlos Prados Sesmero, Sergio Villanueva Lorente, Mario Di Castro. Graph SLAM Built over Point Clouds Matching for Robot Localization in Tunnels[J]. Sensors, 2021, 21(16): 5340-5356.

[28] Ballesta Monica, Paya Luis, Cebollada Sergio, et al. A CNN Regression Approach to Mobile Robot Localization Using Omnidirectional Images[J]. Applied Sciences, 2021, 11(16): 7521-7538.

[29] ALATISE M B, HANCKE G P. A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods[J]. IEEE access, 2020, (8): 39830-39846.

[30] Haoming X, Collins J J. Estimating the Odometry Error of a Mobile Robot by Neural Networks[C]. International Conference on Machine Learning and Applications, Miami Beach, Florida, 2009: 378-385.

[31] Li Ying, Chen Xinglin, Song Shenmin. Application of wavelet neural network for identification of drifts errors in fiber optical gyroscope[J]. Optics and Precision Engineering, 2007, 15(5): 773-778.

[32] Sun Wei, Duan Shunli, Wen Jian. Application of Threshold Denoising and RBF Neural Network in the Error Compensating of MEMS Gyro[J]. Chinese Journal of Sensors and Actuators, 2017, 30(1): 115-119.

[33] Juliana Adeola Adisa, Samuel Ojo, Pius Adewale Owolawi, Agnieta Pretorius, and Sunday Olusegun Ojo, "The Effect of Imbalanced Data and Parameter Selection via Genetic Algorithm Long Short-Term Memory (LSTM) for Financial Distress Prediction," IAENG International Journal of Applied Mathematics, vol. 53, no.3, pp796-809, 2023.

[34] Yuan Luo, YongChao Zeng, RunZhe Lv, and WenHao Wang, "Dual-stream VO: Visual Odometry Based on LSTM Dual-Stream Convolutional Neural Network," Engineering Letters, vol. 30, no.3, pp926-934, 2022.

[35] Kalman R E. A New Approach to Linear Filtering and Prediction Problems[J]. Journal of basic Engineering, 1960, 82(1):35-45.

[36] Bucy R S, Reune K D. Digital Synthesis of Nonlinear Filter[J]. Automatica, 1971, 7(3): 287-289.

[37] Greff, Klaus, Srivastava, Rupesh K, et al. LSTM: A Search Space Odyssey[J]. IEEE Transactions on Neural Networks and Learning Systems, 2017, 28(10): 2222-2232.

[38] Yang Yafeng, Liang Kaihuan, Xiao Xuefeng,et al. Accelerating and compressing LSTM based model for online handwritten Chinese character recognition[C]. 16th International Conference on Frontiers in Handwriting Recognition, ICFHR 2018: 110-115.

[39] Chengyang Li. Research on Stock Price Prediction and Quantitative Stock Selection Based on CNN-LSTM [D]. Xi'an: Northwest University, 2021.

[40] Churan Ji. Research on Stock Index Price Forecasting Based on LSTM Neural Network[D]. Hefei: Anhui University of Finance and Economics, 2021.