Proceeding of the 2015 IEEE
International Conference on Information and Automation
Lijiang, China, August 2015

BP Neural Network Based Localization for a Front-Wheel Drive and Differential Steering Mobile Robot*

Shiwei Jia[1,2,3], Quan Qiu[*1,2], Junmin Li[3], You Li[1,2], Yue Cong[1,2]

(1. Beijing Academy of Agriculture and Forest Sciences, Beijing 100097, China;

2. Beijing Research Center of Intelligent Equipment for Agricultural, Beijing 100097, China;

3. Mechanical Engineering and Automation College, XiHua University, Chengdu, 610039, China.)

***Abstract* - This paper presents a new BP-neural-network-based localization algorithm for a wheeled agricultural mobile robot, which is front-wheel drive and differential steering. Training the BP neural network is the first step of the localization algorithm. During this process, the drive pulse number is regarded as the input; the length of the left or right wheel's trajectory is regarded as the output; and Bayesian rule is used to generate the training function. Inducing the displacement of the robot's geometric center is the second step, in which the trajectories of the left and right wheels are assumed as two concentric circular arces. In the contrast experiments with the least square localization algorithm, the new proposed BP-neural-network based algorithm shows high accuracy and feasibility.**

Index Terms - mobile robot, BP neural network, localization

I. INTRODUCTION

Localization is a key part of autonomous navigation in mobile robotics. The existing localization algorithms can be sorted into two main categories: absolute localization and relative localization [1, 2]. Absolute localization requires robots to fulfill the localization task without a pre-determined original start position, so that the local minimum problem can be solved. Absolute localization has subbranches, such as landmark/beacon based localization [3-6], multi-sensor fusion based localization [7], GPS based localization [8], and probabilistic based localization [9]. Relative localization requires a pre-determined original start position, and the current localization of robot is computed by accumulating all the displacements of every sample plot. As landmarks or beacons are difficult to be found in many cases, relative localization draws more attentions from robotics researchers.

Relative localization has two typical research directions: inertial navigation based localization and odometry based localization. Accelerometer, digital compass, and gyroscope are the common sensors for inertial navigation based localization [10]. Accelerometer is sensitive to drifts and suffers from measurement error. Gyroscope also suffers from static drift. Compass can give accurate gesture information of robots, but carry out bad performance if there is magnetic interference in the scenario. Odemetry based localization mainly employs encoder's output to calculate the displacement increment. Also, environmental perception sensors, including laser scanner, vision camera, and sonar, can cooperate with the encoder to do the localization [11]. Similar to other localization methods, odemetry based localization methods also have difficulty on handling errors coming from several common sources, such as slippage, mechanism asymmetry, and senor accuracy limitation. Researchers tried to find proper calibration solutions by building error models for different error sources [11, 12]. But the model building process is complex and a large quantity of properties needs to be determined. To tackle this problem, this paper present a BP neutral network based localization algorithm. BP Neutral network [13, 14] can fulfill the mapping from input to output without knowing the system's inner mechanism. In this paper, we take the encoder data of the two driving wheels as the input and take the displacement of the robot's geometrical center as the output. Thus, the localization task can be carried out directly with encoder data other than building complex estimation models.

II. BP NEURAL NETWORK BASED LOCALIZATION

The BP neural network based localization algorithm presented here for a front-wheel driving robot (Fig.1) will localize the four-wheeled robot with two steps. First, BP neural network will be employed to estimation the travel lengths of the left-front wheel and right-front wheel with the help of impulse numbers coming from encoders fixed on the left-front and right-front wheel driving motors; Second, the displacement of the robot's geometrical center will be computed based on the travel lengths. In this section, the working principle of the two steps will be elaborated intensively.

Fig.1 Front-wheel drive mobile robot

* This work is partially supported by National High Technology Research and Development Program of China Grant #2013AA102406, and National Natural Science Foundation of China Grant #61305105 to Q. Qiu.

A. Estimating Driving Wheels' Travel Lengths with Back Propagation Network (BP)

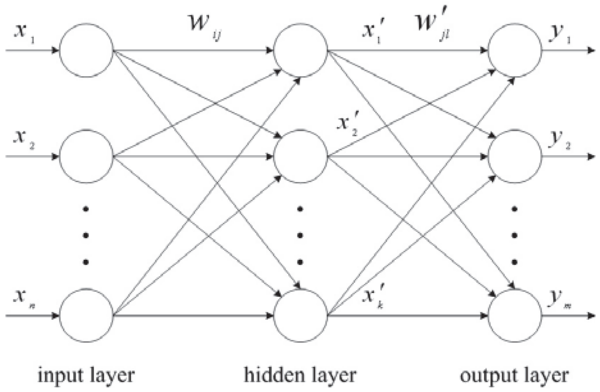

Fig.2 BP topology structure

Back propagation network (BP) is a kind of feedforward neutral network, which is trained by transferring error reversely. BP is the most widely used neutral network currently. It can learn and store a large quantity of input-output mapping relationships. Ordinarily, steepest decent back-propagation is employed as the training rule of BP neutral network, which will get the minimum mean square error for the whole network. As shown in Fig.2, the topology structure of BP contains three layers: input layer, hidden layer, and output layer. If we assume the input layer has n neurons, the hidden layer has k neurons, and the output layer has m neurons, the input vector X, the output vector Y, and the output vector of hidden layer X' can be denoted as

$$\mathrm{X} = (x_1, x_2, \cdots, x_n)^{\mathrm{T}} \tag{1}$$

$$\mathrm{Y} = (y_1, y_2, \cdots, y_m)^{\mathrm{T}} \tag{2}$$

$$\mathrm{X}' = (x'_1, x'_2, \cdots, x'_k)^{\mathrm{T}} \tag{3}$$

If we denote w_{ij} as the connection weight from the ith input neuron to the jth hidden layer neuron, w_{jl} as the connection weight from the jth hidden layer neuron to the lth output layer neuron, $\theta_j (j = 1, 2, \cdots, k)$ as the activation threshold of the jth hidden layer neuron, $\theta_l (l = 1, 2, \cdots, m)$ as the activation threshold of the lth output layer neuron, the mapping functions between two adjacent layers can be written as

$$x'_j = f_1(\sum_{i=1}^{n} w_{ij} x_i + \theta_j) \tag{4}$$

$$y_l = f_2(\sum_{j=1}^{k} w_{jl} x'_j + \theta_l) \tag{5}$$

Where $f_1(\bullet)$ and $f_2(\bullet)$ are the activation functions of hidden layer neurons and output layer neurons, respectively.

According to Kolmogorov's theorem [15], a three-layer BP neural network can approach to arbitrary nonlinear functions with arbitrary accuracies. As a result, we use a three-layer BP neural network to fulfill the transfer function simulation task. In this paper, we want to employ BP neural network to estimate the travel lengths of robot's driving wheels and compute the displacement of robot's geometrical center with the travel lengths. We hope to simulate a transfer function for each driving wheel separately. Because the robot we use in this paper has two driving wheels and carry out the turning actions under the help of speed differential, we need two instances of BP neural network, and each instance has one input neuron and one output neuron. To simplify the network structure, we use 3 hidden layer neurons. Now the structure of a BP neural network for simulating the transfer function from encoder output, here is impulse number, to travel length will look like the instance in Fig.3, where N is the impulse number of encoder and S is the travel length of the corresponding driving wheel.

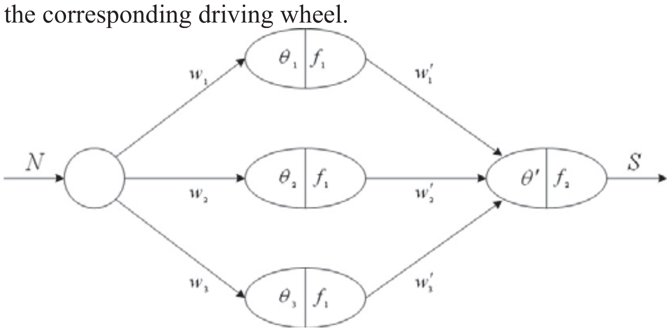

Fig.3 Structure of a BP neural network instance in this paper

For the activation functions, we choose as

$$f_1(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{6}$$

$$f_2(x) = x \tag{7}$$

B. Calculating the Displacement of Robot's Geometrical Center

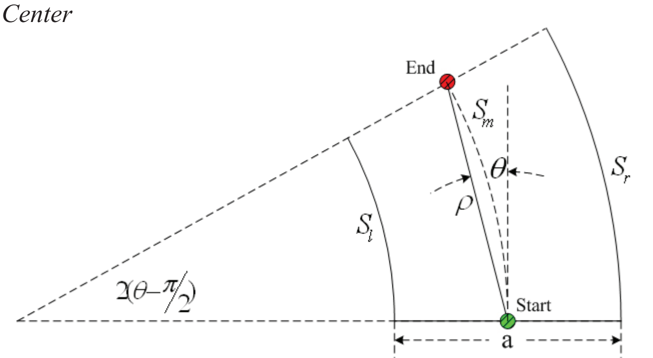

Fig.4 Geometrical relationship of S_m, S_r and S_l.

If we divide the robot's moving trajectory into infinitesimal segments, each segment can be regarded as a part of straight line. Theoretically, the displacement of robot's geometrical center is proportional to encoder's output impulse number. But the travel lengths of different driving wheels will differ from each other because of the error induced by slippage, mechanism, etc. Through experiments, we find that the robot Walle will show a slight left-turn which is visible to human eyes after it is supposed to be moving 2 meters following a straight line. This result means that the travel length of the right-front wheel is longer than the one of the left-front wheel in a segment. And this difference of travel length is mainly caused by mechanism asymmetry of the robot. If we denote the travel length of the right-front wheel and left-front wheel with S_r and S_l respectively, and denote the displacement of the robot's geometrical center with S_m, their geometrical relationship can be described as shown in Fig.4.

And if we express S_m in a polar coordination as $S_m(\rho,\theta)$, where ρ is the distance from the start point to the end point of the tiny segment, θ is the common radian of arc S_r and S_l. $S_m(\rho,\theta)$ can be computed as

$$\rho = a \times \frac{S_r + S_l}{S_r - S_l} \times \sin\left(\frac{S_r - S_l}{2 \times a}\right) \tag{8}$$

$$\theta = \frac{\pi}{2} + \frac{S_r - S_l}{a} \tag{9}$$

in which a is the width of the robot. With (8) and (9), we can compute $S_m(\rho,\theta)$ directly from S_r and S_l. As a result, the localization task will be fulfilled in two steps: BP neural network estimates the right-front wheel's travel length S_r and the left-front wheel's travel length S_l firstly; (8) and (9) induce the displacement of robot's geometrical center $S_m(\rho,\theta)$ secondly.

III. EXPERIMENTS

A. Training the BP Neural Network

There are many methods for training BP neural network. In this paper, we choose to train the network in Matlab by Bayesian training rule [16]. The training sample data are 1000 pairs of impulse number and travel length, which are collected on 500 sample points in the same straight line tracking test. The terrain condition of the line tracking test is shown in Fig.5.

Fig.5 Terrain conditions for collecting training data

In Fig.5, we can see that the terrain is concrete floor with fissure and grass. Slippage and jolt will easily happen when Walle is walking on such terrain, which will introduce strong noise for localization algorithms. During the experiment, the left-front wheel and right-front wheel of Walle will receive same quantity of control impulse at every time plot. Theoretically, if the robot is mechanism symmetry and the terrain is even, the robot should walk following a straight line. But in the experiment, the robot walked following a curve because of uneven terrain and system error.

For each sample point, the impulse number and the left-front/right-front travel length will be recorded. So we will have 2 pairs of impulse number and travel length at each sample point. The impulse is generated by motor driving controller and can be counted by the encoder. The travel length is the distance from a predefined start point to the contact point between the wheel and the terrain.

Before staring training, all the input and output data need to be normalized for the purpose of transferring all data into the same scale. In this way, all inputs and outputs will behaviour in the same level and big differential caused by different units of measurement will be eliminated. The normalization function can be

$$G_{out} = \frac{G_{in} - \min(G_{in})}{\max(G_{in}) - \min(G_{in})} \tag{10}$$

in which G_{in} is the data to be normalized and G_{out} is the normalization output. After normalization, we will obtain GN_r, GN_l, GS_r, and GS_l for each sample point. When the training process is finished, we can use the BP neural network to estimate GS_r' and GS_l'. The estimated S_r' and S_l' can be got through an anti-normalization function as below

$$S_i' = NS_i' \times (\max(NS') - \min(NS')) + \min(NS') \tag{11}$$

We can explain the training process with the left-front wheel as an example. If we assume the impulse number of clock T_i is n_i, and the left-front wheel travel length to be estimated is s_i, then s_i can be computed through four steps:

1) normalize n_i with (10) and get n_i';

2) from input layer to hidden layer, compute three outputs of hidden layer with function f_1,

$$\begin{aligned} x_1' &= f_1\left(n_i' \times w_1 + \theta_1\right), \\ x_2' &= f_1\left(n_i' \times w_2 + \theta_2\right), \\ x_3' &= f_1\left(n_i' \times w_3 + \theta_3\right). \end{aligned} \tag{12}$$

where x_i' is the output of hidden layer, w_i is the connection with between the input neuron and a hidden neuron, θ_i is the activation threshold of the corresponding hidden neuron;

3) from hidden layer to output layer, compute the output of output layer with function f_2,

$$y_i = f_2(x_1' \times w_1' + x_2' \times w_2' + x_3' \times w_3' + \theta') \tag{13}$$

where y_i is the output of output layer neuron;

4) compute s_i with an anti-normalization function similar to (11).

$$\begin{aligned} s_i &= y_i \times \left(\max\left(\Upsilon\right) - \min\left(\Upsilon\right)\right) + \min\left(\Upsilon\right), \\ \Upsilon &= \left\{y_1, y_2, \cdots, y_k\right\} \end{aligned} \tag{14}$$

where $\mathbf{Y}$ is a vector formed by all the outputs in the training process.

B. *Experimental Results of Localization Estimation*

To verify the feasibility and accuracy of the proposed BP neural network (BPNN) localization algorithm, we use least square method (LSM) as a contrast. Both LSM and BPNN are employed to estimate S_r' and S_l'.

TABLE I

LEASE SQUARE ERROR FOR LMS AND BPNN

Methods	$ES_r(\text{mm}^2)$	$ES_l(\text{mm}^2)$	$E\rho(\text{mm}^2)$
LMS	26.1281	26.2007	26.0669
BPNN	1.2589	2.0379	1.4685

The test sample data are collected in a new straight line tracking trial on the same terrain shown in Fig.5. The robot's moving trajectory is less than 3.5 meters, and 100 sample points are selected along it. The corresponding impulse number and the left-front wheel/right-front wheel travel length are recorded. For evaluating the localization precision, the distance from the start point to the robot's geometrical center at each sample point are also recorded. The estimation error ES_r and ES_l, the localization error $E\rho$ are all listed in Table.1 as mean square error. We can infer that the estimation error of BPNN is obviously smaller than the estimation error of LMS.

The estimation error distribution is shown in Fig.5. The upper row is the results of BPNN, and the lower row is for LMS. For each row, the first column is the travel length error of the right-front wheel, the second column is the travel length error of the left-front wheel, and the third column is the displacement error of the robot's geometrical center. From Fig.5, we can see that most estimation error of BPNN is less than 3mm, while most estimation error of LMS is larger than 10mm. The largest estimation error of LMS exceeds 50mm. Then we can infer that the new proposed BP localization algorithm can give good positioning performances. Also, the estimation error of LMS will come down while the travel length of the robot increases.

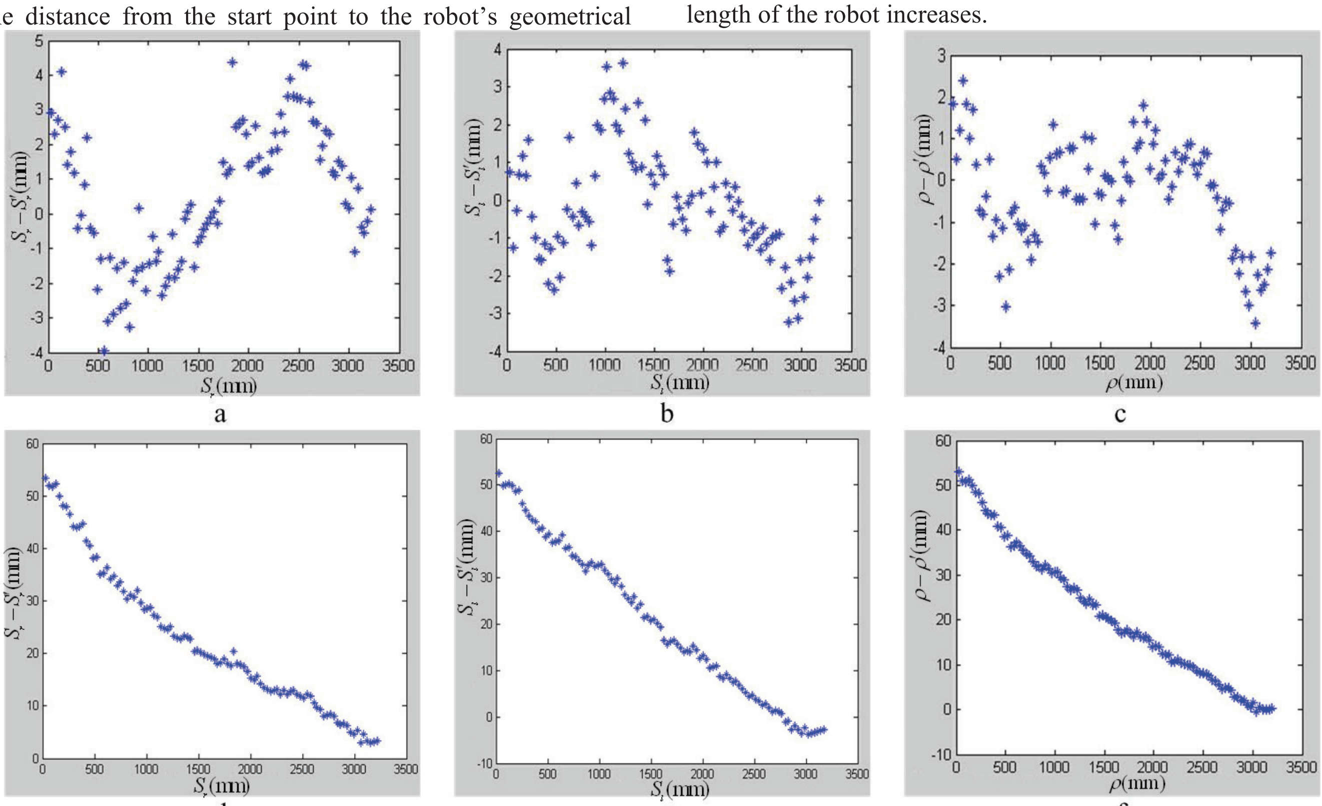

Fig.5 Estimation error distribution of BPNN and LMS

IV. CONCLUSIONS AND FUTURE WORK

For solving the localization problem of a four-wheeled robot, we present a BP neural network based localization algorithm. The algorithm can estimate the displacement of robot's geometrical center with the impulse number obtained from the encoder directly, other than building complex models for slippage, mechanical error, and other system noise. Experiments on a four-wheeled robot named with Walle verified the feasibility and accuracy of the proposed algorithm. But training and experiments in this paper only concentrates on the straight line tracking problem on concrete terrain. In future, we will collect training sample data for different situations, including soil terrain, uneven terrain, slope and arbitrary curve tracking. We hope to build a BPNN based localization framework for various terrain conditions and tracking route.

ACKNOWLEDGMENT

The authors would like to thank Wengang Zheng and Zhijun Meng for their kind help and enlightening suggestions.

REFERENCES

[1] L. Li, T. Ye, M. Tan, and X. J. Chen, "Present state and future development of mobile robot technology research," *Robot*, vol. 24, no. 5, pp. 475-480, 2002. (in Chinese with English abstract)

[2] M. A. Salichs, L. Moreno, "Navigation of mobile robots: open questions," *Robotica*, vol. 18, no. 2, pp. 227-234, 2000.

[3] W. Burgard, D. Fox, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *Journal of Artificial Intelligence Research*, vol. 11, no. 1, pp.391-427, 1999.

[4] L. Kleeman, "Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead-reckoning," *IEEE International Conference on Robotics and Automation*, pp. 2582-2587, Nice, France, May 12-14, 1992.

[5] E. Colle, M. Jallouli, and F. Chavand, "Absolute localization of an autonomous vehicle with beacons," *International Journal of Robotics and Automation*,vol. 8, no. 1, pp.30-38, 1992.

[6] S. Borthwick, M. Stevens, H. F. Durrant-Whyte, "Position estimation and tracking using optical range data," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2172-2177, Tokyo, Japan, July 26-30, 1993.

[7] M. Kam, X. Zhu, and P. Kalata, "Sensor fusion for mobile robot navigation," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 108-119, 1997.

[8] Z. J. Yang, F. Y. Liao, and W. Qing, "Apply of GPS on the technologies of mobile robot navigation and location," Ship Electronic Engineering, vol.25, no. 6, pp. 5-7, 2005. (in Chinese with English abstract)

[9] M. H. Li and B. R. Hong, "Progress of probabilistic localization methods in mobile robots," Robot, vol. 27, no. 4, pp. 380-384, 2005. (in Chinese with English abstract)

[10]B. Barshan and H. F. Durrant-Whyte, "An inertial navigation system for a mobile robot," *IEEE Transaction on Robotics and Automation*, vol. 11, no.3, pp. 328-342, 1995.

[11]K. Chong and L. Kleeman, "Accurate odometry and error modeling for a mobile robot," *IEEE International Conference on Robotics and Automation,* vol. 4, pp. 2783-2788, Albuquerque, USA, 1997.

[12]W. H. Wang, "Research on the localization technologies for mobile robots," *Doctoral Thesis*, Huazhong University of Science & Technology, 2005.

[13]W. S. McCulloch and W. Pitts, "A logical calculus of ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115-133, 1943.

[14]D. E. Rumelhart and J. McClelland, "Parallel distributed processing: explorations in the microstructure of cognition," *Cambridge: MIT Press*, 1986.

[15]V. Ků rková, "Kolmogorov's theorem and multilayer neural networks," *Neural Networks*, vol. 5, no. 3, pp. 501-506, 1992.

[16]R. M. Neal, "Bayesian learning for neural networks," *Lecture Notes in Statistics*, vol.118, 1996.