Fuzzy Neural Network with Sliding-Mode Controller Applications to Mobile Robotic

Lon-Chen Hung* and Hung-Yuan Chung**
Department of Electrical Engineering, National Central University,
Chung-Li, Tao-Yuan, 320, Taiwan, R.O.C
TEL: 886-3-4227151 ext 4475
Fax: 886-3-425-5830

Abstract: - This paper deals with the design of single fuzzy neural network (SFNN) with sliding mode control (SMC) systems by using the approach of sliding mode control. The fuzzy sliding mode control (FSMC) in the system guarantee the state can reach the user-defined surface in finite time and then it slides into the origin along the surface. The key idea is to apply parameters of the membership function of the fuzzy rule base are self-tuning with neural network (NN). They will change according to some tuning scheme. The initial fuzzy membership function can be given by human experts, and it is tuned automatically by the tuning scheme so as to eliminate the chattering effect of the system. Next, we will apply the single fuzzy neural network with sliding mode controller (SFNNSMC) to a nonlinear mobile robotic system. At last, experiment results show the feasibility of the method that used to verify the effectives and robustness.

Key-Words: - fuzzy, sliding, neural, mobile robotic

1 Introduction

In classical control theory, the controller is designed based on the mathematical model of the system. However, the mathematical model of the system is hard to know. So it is not easy to design a model-based controller to control these complex and ill-defined system. Fortunately, human knowledge can do well in dealing with this kind of problem. It is well known that fuzzy logic provides a systematic procedure to transform a knowledge base into a practical control strategy. Particularly, fuzzy control shows its powerful capability in complex systems with ill-defined processes, or systems lack of knowledge of their dynamics. However, the design of fuzzy controllers is time-consuming and experiment-oriented and requires the knowledge acquisition, definition of the controller structure, definition of rules, and other controller parameters. So, one important issue in fuzzy logic systems is how to reduce the number of involved rules and their corresponding computation requirements [1-2].

It is important the fuzzy membership functions are updated iteratively and automatically, because a change in fuzzy membership functions may alter the performance of the fuzzy controller significantly. Several algorithms of tuning of the fuzzy membership functions have been proposed in [3-4]. These algorithms have an important characteristic of global tuning of the fuzzy membership functions and that most of them need off-line preprocessing.

The fuzzy techniques have been widely applied to the systems with uncertainties. Recently, researchers have utilized the fuzzy techniques together with the sliding mode control for many engineering control systems [5-7]. In [5], the sliding mode control schemes with fuzzy system are proposed with the uncertain system function is approximated by the fuzzy system. The fuzzy control rules based on the sliding function and the derivative of the sliding function are constructed in [6]. The complexity of the fuzzy SMC is increased quickly as the number of sliding functions increases. The technology that combines or fuses the neural-network theory with the fuzzy reasoning is being watched with keen interest [10].

In this paper, an SFNNSMC position control is investigated, in which the SFNNSMC is utilized to estimate the bound of uncertainties real-time for the position control and tracking control system. The inputs of the SFNNSMC are the switching surface, and the output of the SFNNSMC is the estimated bound of uncertainties. If the uncertainties are absent, once the switch surface is reached initially, a very small positive estimated value of bounded of uncertainties would be sufficient to keep the trajectory on the switching surface. We also apply it to controlling a mobile robot driven by two independent wheels. The mathematical model of mobile robot system is not completely known. But, to measure some physical parameters, e.g., viscous friction factor and moment of inertia around the center of gravity for the robot, so that the SFNNSMC which is not based on the mathematical model is recommendable for this subject.

The rest of the paper is divided into five sections. In Section 2, the systems are described. In Section 3, the adaptability complexity hybrid fuzzy sliding-mode control is presented. In Section 4, the proposed controller is used to control a mobile robotic system. Finally, we conclude with Section 5.

2 Single Fuzzy Neural Network with Sliding Mode Controller

2.1 Signed Distance Fuzzy Logic Control

In this section, the idea of [8] named the signed distance is used, and the feasibility of the present approach will be demonstrated. The switching line is defined by:

$$s : \dot{x} + c_1 x = 0 \tag{1}$$

First, we introduce a new variable called the signed distance. Let $P(x, \dot{x})$ be the intersection point of the switching line and the line perpendicular to the switching line from an operating point $Q(x_1, \dot{x}_1)$, as illustrated in Fig. 1. Next, *ds* is evaluated. The distance between $P(x, \dot{x})$ and $Q(x_1, \dot{x}_1)$ can be given by the following expression:

$$d_s = \frac{|\dot{x}_1 + c_1 x_1|}{\sqrt{1 + c_1^2}} \tag{2}$$

The signed distance *ds* is defined for an arbitrary point $Q(x_1, \dot{x}_1)$ as follows:

$$\begin{aligned} d_s &= \mathrm{sgn}(s)\frac{|\dot{x} + c_1 x|}{\sqrt{1 + c_1^2}} \\ &= \frac{\dot{x} + c_1 x}{\sqrt{1 + c_1^2}} = \frac{s}{\sqrt{1 + c_1^2}} \end{aligned} \tag{3}$$

where

$$\mathrm{sgn}(s) = \begin{cases} 1 & \text{for } \; s > 0 \\ \text{-}1 & \text{for } \; s < 0 \end{cases} \tag{4}$$

For the switching line is chosen as

$$s = c_1 x_1 + x_2 \tag{5}$$

By taking the time derivative of both sides of (5), we can obtain

$$\begin{aligned} \dot{s} &= c_1 \dot{x}_1 + \dot{x}_2 \\ &= c_1 x_2 + f(\mathbf{x}) + b(\mathbf{x})u + d \end{aligned} \tag{6}$$

Then, multiplying both sides of the above equation by   gives

$$s\dot{s} = sc_1 x_2 + sf(\mathbf{x}) + sb(\mathbf{x})u + sd \tag{7}$$

Here, we assume that $b(\mathbf{x}) > 0$. In (6), it is seen that   increases as   increases and vice versa. Equation (7) provides the information that if s>0, the decreasing *u* will make $s\dot{s}$ decrease and that if s<0, the increasing *u* will make $s\dot{s}$ decrease.

Hence, the fuzzy rule table can be established on a one-dimensional space of   as shown in Table 1 instead of a two-dimensional space of x and $\dot{x}$. The control action can be determined by d_s only. Hence, we can easily add or modify rules for fine control. For implementation, the membership function of the input is triangular-shaped which are in the form of $\mu(x) = \max[1 - \frac{|x - m|}{\sigma}, 0]$, where *m* is the center of membership function and σ is the width. The membership function of the output is singleton. The fuzzy rule base is as Fig. 2. The membership function of the fuzzy input and output is shown in Fig 3.

2.2 The Neural-network-based Fuzzy Logic Controller

We apply neural network to fuzzy logic controller. The neural network based fuzzy logic controller (NN-FLC) is a multilayer network. It combines the merits of fuzzy controller and neural network. It integrates the basic elements and functions of a traditional fuzzy logic controller into a connected structure that possesses the learning ability. The architecture of NN-FLC is show in Fig. 4. The NN-FLC contains five layers .The first layer is input layer and the fifth layer is output layer. In the second layer, nodes compute the degree of compatibility from the first layer, and connect to relative nodes of next layer. Nodes at the third layer are rule nodes which indicate the connection with fourth layer by fired rules. Nodes at the fourth layer are term nodes which is similar to the second layer, representing the membership function of the consequence label.

The links between every layer represent the weight. The second and the fifth layer are full connected .The arrow of link indicate the propagation direction of the signal. There are two kind of function required in each node. One is summation of all inputs from the node of the previous layer. The function can be expressed as follows:

$$Node_{input} = I(u_1^{(l)}, u_2^{(l)}, \cdots, u_n^{(l)}; w_1^{(l)}, w_2^{(l)}, \cdots, w_n^{(l)}) \tag{8}$$

where u_i^l represents the input signal from previous layer, w_i^l represents the *i*-th link weight of the *l*-th layer, and *n* represent the number of the connected nodes. The other function is transfer function which denotes the output of this node. The function can be written as

$$Node_{output} = O(I) \tag{9}$$

The details of this network are described as follows:

Layer 1:

This layer just transmits the value of the input signal to the next layer. So the function can be written as

$$I = u_n^1 \text{ and } O = I \tag{10}$$

We note that the link weight $w_i^1 = 1$.

Layer 2:

In this layer, we use triangular membership function to fuzzify the input signals that propagate from the previous layer.

The function is

$$I = \max(1 - \left| \frac{u_i^2 - m_{ij}}{\sigma_{ij}} \right|, 0) \text{ and } O = I \tag{11}$$

Where m_{ij} and σ_{ij} are the convex vertex and the width of the triangular-shape function of the *j*-th term of the *i*-th input linguistic variable, hence the link weight w_{ij}^2 can be represented as m_{ij}.

Layer 3:

Nodes at this layer perform the fuzzy **AND** operation and indicate where the output connected by matching rule. The function can be denoted

$$I = \min(u_1^{(3)}, u_2^{(3)}, \cdots, u_n^{(3)}) \text{ and } O = I \tag{12}$$

We note that the link weight $w_i^3 = 1$.

Layer 4:

Nodes at this layer sum up the contributions which have the same consequence from Layer 3 and perform fuzzy **OR** operation. The function can be denoted as

$$I = \sum_{i=1}^{n} u_i^4 \text{ and } O = I \tag{13}$$

and the link weight $w_i^4 = 1$.

Layer 5:

This layer executes the defuzzification. One selects the COA (center of area) method. The function can be denoted as

$$I = \sum_{i=1}^{n} (m_i) u_i^5 \text{ and } O = \frac{I}{\sum u_i^5} \tag{14}$$

Where m_i is the value of each singleton fuzzy output function of the *i*-th term of the output linguistic variable, hence the link weight can be represented as $w_i^5 = m_i$.

2.3 The Tuning Algorithm

The back-propagation training algorithm is applied to this fuzzy system. We usually check the learning performance by the performance index(cost function)and minimize the error between desired and current output. The function can be written as

$$E = \frac{1}{2} \sum (y(t) \text{-} y_d(t))^2 \tag{15}$$

In general, we minimize the function by the gradient descent method which is used to update the weight of the neural network.

That is, the modified value can be represented as:

$$\Delta w \propto -\frac{\partial E}{\partial w} \tag{16}$$

$$w(t+1) = w(t) + \eta(-\frac{\partial E}{\partial w}) \tag{17}$$

where η is the learning rate.

For simplifying the tuning problem, we derive the tuning algorithm to Layer five only. In Layer five, the adaptive rule of the value m_i is derived as

$$\frac{\partial E}{\partial m_i} = \frac{\partial E}{\partial O}\frac{\partial O}{\partial m_i} = \frac{\partial E}{\partial O}\frac{\partial O}{\partial I}\frac{\partial I}{\partial m_i} \tag{18}$$

From (15)

$$\frac{\partial E}{\partial O} = -[y_d(t) - y(t)] \tag{19}$$

$$\frac{\partial O}{\partial I} = \frac{1}{\sum u} \tag{20}$$

$$\frac{\partial I}{\partial m_i} = u_i \tag{21}$$

Then the membership function can be written as:

$$m_i(t+1) = m_i(t) + \eta(y_d(t) - y(t)) \frac{u_i}{\sum u_i} \tag{22}$$

In measuring the chattering of the system, β is defined as follows:

$$\beta = \sum_{i=0}^{30} \alpha_i \tag{23}$$

$$\alpha = \begin{cases} 0 & \text{if} \quad S(kT)S(k-1)T > 0 \\ 1 & \text{if} \quad S(kT)S(k-1)T < 0 \end{cases} \tag{24}$$

where *T* is the sampling time. Because y_d is unknown, we replace (y_d - y) by (0-β), and then (22) can be represented as

$$m_i(t+1) = m_i(t) - \eta\beta \frac{u_i}{\sum u_i} \tag{25}$$

3 Model of a mobile robot

Let the mobile robot with two independent driving wheels be rigid moving on the plane. It is assumed that the absolute coordinate system O-XY is fixed on the plane as shown in Fig. 6. Then, the dynamic property of the mobile robot is given by the following equation of motion [11]:

$$I_\upsilon \ddot{\phi} = D_r l - D_l l \tag{26}$$

$$M \dot{\upsilon} = D_r - D_l \tag{27}$$

For the right and left wheels, the dynamic property of the driving system becomes

$$I_\omega \ddot{\theta}_i + c\dot{\theta}_i = ku_i - rD_i,\ \ i = r, l \qquad (28)$$

where each parameter and variable are defined by

(1) I_υ : moment of inertia around the e.g. of robot;
(2) M : mass of robot;
(3) D_r, D_l : left and right driving forces;
(4) l : distance between left or right wheel and the e.g. of robot;
(5) ϕ : azimuth of robot;
(6) υ : velocity of robot;
(7) I_ω : moment of inertia of wheel;
(8) c : viscous friction factor;
(9) k : driving gain factor;
(10) r : radius of wheel;
(11) θ_i : rotational angle of wheel; and
(12) u_i : driving input.

The geometrical relationships among variables ϕ , υ , θ_i are given by

$$r\dot{\theta}_r = \upsilon l + l\dot{\phi} \qquad (29)$$

$$r\dot{\theta}_l = \upsilon l - l\dot{\phi} \qquad (30)$$

From these equations, defining the state variable for the robot as $x = [\upsilon\ \phi\ \dot{\phi}]^T$.

4 Experiment of the Mobil Robot System

The velocity and azimuth of the robot are controlled by manipulating the torques for the left and the right wheels. That is, the control system considered here is of multi-input/multi-output. In the SFSMNN with e_υ , e_ϕ , $\dot{e}_\upsilon$, and $\dot{e}_\phi$ as inputs, and with u_r and u_l as reasoning outputs.

From the property of a mobile robot with two independent driving wheels, the following relations are considered:

$$u_r = u_\upsilon + u_\phi \qquad (31)$$

$$u_l = u_\upsilon - u_\phi \qquad (32)$$

where u_υ is the torque required for controlling the robot's velocity by using the measurement of the robot's velocity and u_ϕ is the torque required for controlling the robot's azimuth by using the measurement of the robot's azimuth. If u_υ is generated from the SFSMNN1 with the velocity error e_υ and its rate as the inputs and u_ϕ is generated from the SFSMNN2 with the azimuth error e_ϕ and its rate as inputs, then the learning controller for the velocity and azimuth of a robot. In this realization, the membership function in the antecedent for each SFNNSMC is fine-tuned by the connection weights ω_1, ω_2, ω_3, and ω_4.

The experimental mobile robot system used in this paper is shown in Fig. 7. It consists of a vehicle with two driving front wheels mounted on the same axis. The motion and orientation are achieved by independent actuators, e.g., each front wheel is driven by a dc motor. The displacement output of each motor was by a crude incremental encoder on the motor shaft. No pre-filtering was used to filter the noisy feedback signals. Velocity signals of the servo were obtained by differentiating the displacement outputs. The tracking controller was implemented in a PC 486. A 12-bit resolution D/A card and a 16-bit encoder care were used to transfer the control and feedback signals. Fig. 8 and Fig. 10 shows the response of angle and the state trajectory.

5 Conclusion

In this paper, the SFSMNN control system has been proposed to solve the position and tracking problem for a wheeled mobile base has been addressed. In order to cope with the unavoidable presence of uncertainties in the dynamical model and to guarantee the implementability of the controller, the proposed control law has been designed using single fuzzy neural network with sliding mode control techniques. The asymptotic boundedness of the tracking errors has been theoretically proved. It is shown that the proposed method is feasible and effective.

References:

[1] C.L. Tan, A self-tuning fuzzy logic controller based on neural networks structure, *M.S thesis, Department of Electrical Engineering ,National Central University,* 1995.

[2] C.T. Lin, and C.S. George Lee, Neural Fuzzy System: A Neuro-Fuzzy Synergism to Intelligent Systems, *Prentice-Hall Interantional,Inc.,*1996.

[3] Erbatur, K.,Kaynak, O., Use of adaptive fuzzy systems in parameter tuning of sliding-mode controllers, *Mechatronics, IEEE/ASME Transactions on* , Vol, 6, No. 4 , 2001, pp., 474 -482.

[4] D. Park, A. Kandel, and G. Langholz, Genetic-based new fuzzy reasoning models with

application to fuzzy control, *IEEE Trans. Syst., Man, Cybern.*, Vol. 24, 1994, pp. 39-47.

[5] Palm, R., Robust Control by Fuzzy Sliding Mode, *Automatic*, Vol. 30, No.9, 1994, pp. 1429-1437.

[6] Y. R. Hwang and M. Tomizuka, Fuzzy smoothing algorithms for variable structure systems, *IEEE Trans. Fuzzy Syst.*, Vol. 2, 1994,pp. 227–284.

[7] Ji-Chang Lo, Ya-Hui Kuo, Decoupled Fuzzy Sliding-Mode Control, *IEEE Trans. Fuzzy Syst*, Vol. 6. No. 3., 1998, pp. 426-435.

[8] Byung-Jae Choi, Seong-Woo Kwak, Byung Kook Kim, Design of a single-input fuzzy logic controller and its properties, *Fuzzy Sets and Systems* , Vol. 106, 1999, pp. 299-308.

[9] Zhang, J., Morris, A.J., Fuzzy neural networks for nonlinear systems modelling, *Control Theory and Applications, IEE Proceedings*, Vol. 142 , No. 6, 1995, pp. 551-561.

[10] Watanabe, K., Jun Tang; Nakamura, M., Koga, S., Fukuda, T., A fuzzy-Gaussian neural network and its application to mobile robot control, *Control Systems Technology*, *IEEE Transactions on* , Vol. 4 , No. 2, 1996, pp. 193-199.

[11] Zhang, Qiuju, Shippen, James, Jones, Barrie, Robust backstepping and neural network control of a low-quality nonholonomic mobile robot, *International Journal of Machine Tools and Manufacture*, Vol. 39, No. 7, 1999, pp. 1117-1134.

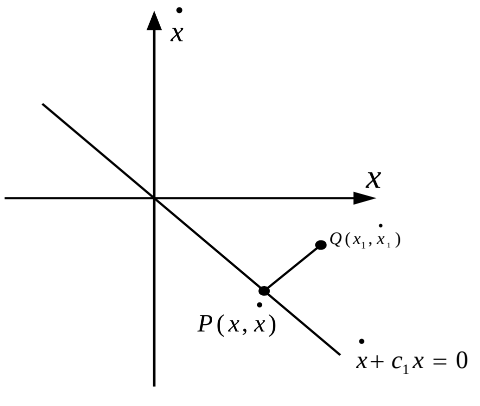

Fig. 1. Derivation of a signed distance.

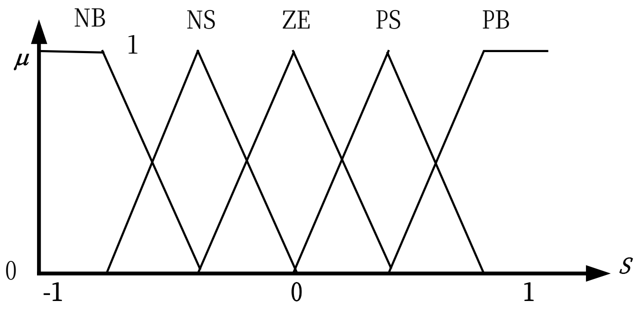

Fig 2. Fuzzy variable of triangular type.

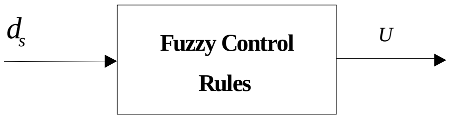

Fig. 3. The block diagram of the SDFLC.

S	NB	NS	ZE	PS	PB
u	PB	PS	ZE	BS	NB

Table.1 The fuzzy rule table

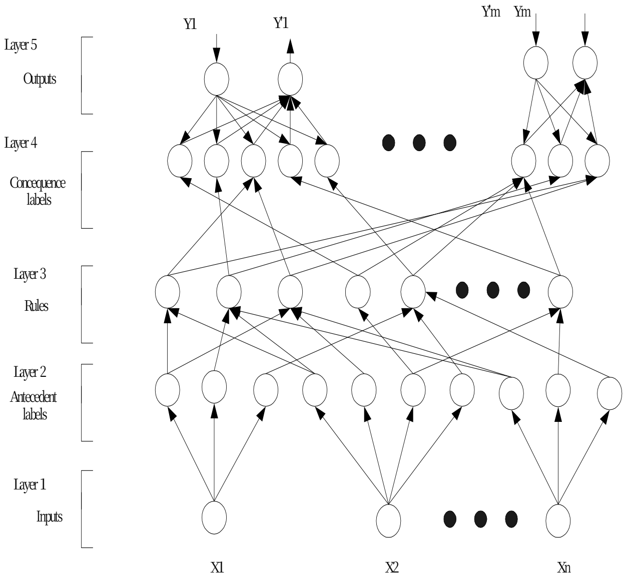

Fig. 4. The fuzzy adaptive learning control network

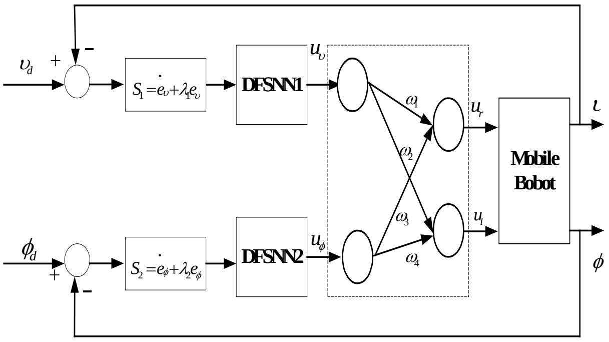

Fig. 5. The Structure of the SFNNSMC System

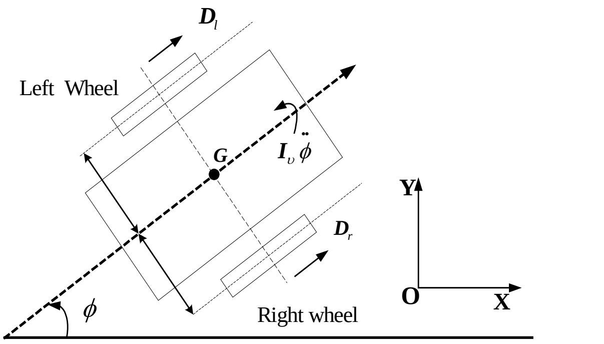

Fig. 6 A mobile with two independent drive wheels.

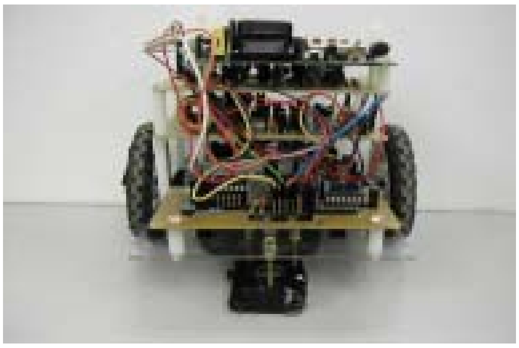

Fig. 7. The photo of the experimental mobile robotic

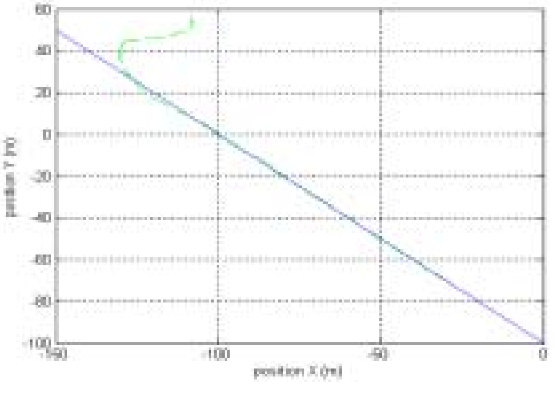

Fig. 8. The mobile robotic tracking line (y=x+100)

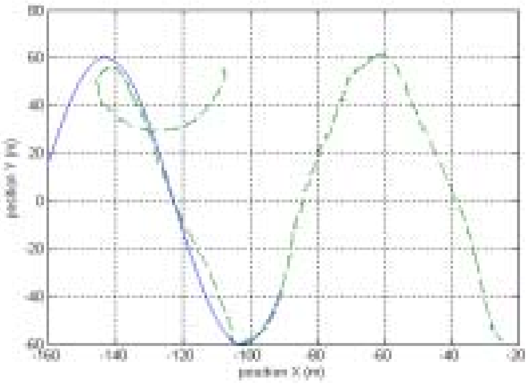

Fig. 9. The mobile robotic tracking line (y = -60sin(x/13+π))

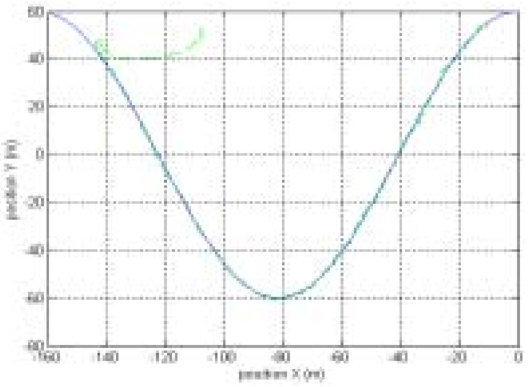

Fig. 10. The mobile robotic tracking line (y = -60cos(x/26+π))