The Chaotic Robot Prediction by Neuro Fuzzy Algorithm

Mana Tarjoman, Shaghayegh Zarei

Abstract— **In this paper an application of the adaptive neuro-fuzzy inference system has been introduced to predict the behavior of a chaotic robot. The chaotic mobile robot implies a mobile robot with a controller that ensures chaotic motions. Chaotic motion is characterized by the topological transitivity and the sensitive dependence on initial conditions. We have used the controller such that the total dynamics of the mobile robot is represented by the Arnold equation, which is known to show the chaotic behavior of non-compressive perfect fluid. Then we have used the adaptive neuro fuzzy inference system for predicting of this chaotic mobile robot.**

We propose to predict the behavior of the chaotic mobile robot by using an adaptive neuro-fuzzy inference system. This system is functionally similar to fuzzy inference systems which based on hybrid learning rule and also are more quick and accurate than methods used neural networks or Kalman filter.

Key words: ANFIS-chaotic mobile robot- hybrid learning rule

I. INTRODUCTION

Chaos characterizes one of mysterious rich behaviors of nonlinear dynamical systems. Many research efforts have been paid to establish the mathematical theory behind chaos. This paper introduces a chaotic mobile robot that the chaotic behavior is achieved by designing a controller which ensures chaotic motion. A mobile robot with such characteristics may find its applications as a patrol robot or a cleaning robot in a closed room, floor, or building. The sensitive dependence on initial condition also yields a favorable nature as a patrol robot since the scanning trajectory becomes highly unpredictable.

A. *Chaotic mobile robot with the Arnold equation*

As the mathematical model of mobile robots, we assume a two wheeled mobile robot as shown in Fig. 1. Let the linear velocity of the robot v [m/s] and the angular velocity ω [rad/s] be the inputs to the system. The state equation of the mobile robot is written as follows [1] :

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} \qquad (1)$$

Where (x [m], y [m]) is the position of the robot and θ [rad] is the angle of the robot.

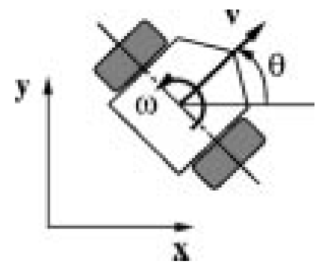

Fig. 1. Mobile robot

In order to generate chaotic motions of the mobile robot, we employ the Arnold equation, which is written as follows:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} A\sin x_3 + C\cos x_2 \\ B\sin x_1 + A\cos x_3 \\ C\sin x_2 + B\cos x_1 \end{pmatrix} \qquad (2)$$

Where A, B, and C are constants.

B. *Integration of the Arnold equation*

In order to integrate the Arnold equation into the controller of the mobile robot, we define and use the following state variables:

$$\begin{cases} \dot{x}_1 = D\dot{y} + C\cos x_2 \\ \dot{x}_2 = D\dot{x} + B\sin x_1 \\ x_3 = \theta \end{cases} \qquad (3)$$

Where B, C and D are constants. Substituting (1) into (3), we obtain a state equation on x1, x2 and x3 as follows:

$$\begin{cases} \dot{x}_1 = Dv + C\cos x_2 \\ \dot{x}_2 = Dv + B\sin x_1 \\ x_3 = \omega \end{cases} \qquad (4)$$

We now design the inputs as follows:

$$\begin{cases} v = \dfrac{A}{D} \\ \omega = C\sin x_2 + B\cos x_1 \end{cases} \qquad (5)$$

M. Tarjoman is with the Abhar Islamic Azad University, Zanjan, Iran (e-mail: mana_tarjoman@ yahoo.com).

Sh. Zarei is with the Tehran Islamic Azad University (central branch), Tehran, Iran (e-mail: shaghayeghzarei@yahoo.com).

Consequently, the state equation of the mobile robot becomes:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} A\sin x_3 + C\cos x_2 \\ B\sin x_1 + A\cos x_3 \\ C\sin x_2 + B\cos x_1 \\ v\cos x_3 \\ v\sin x_3 \end{pmatrix} \qquad (6)$$

Equation (6) includes the Arnold equation. The Arnold equation behaves chaotically or not, depending upon the initial states. We choose the initial states of the Arnold equation such that the trajectory should behave chaotically. The whole states evolve in a 5-D space according to (6), which includes a 3-D subspace of the Arnold flow. Fig. 2 shows an example of motion of the mobile robot with the introduced controller, obtained by numerical simulation.

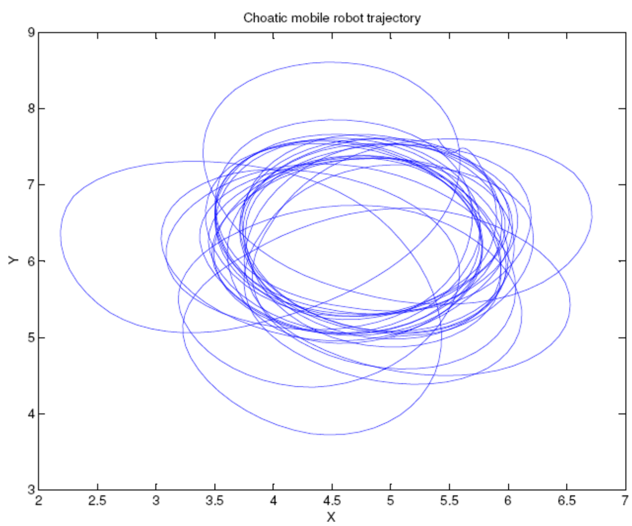

Fig 2. Chaotic mobile robot trajectory

Here we propose to predict the behavior of a chaotic robot by using an adaptive neuro-fuzzy inference system. This system is functionally similar to fuzzy inference systems which based on hybrid learning rule and also are more quick and accurate than methods used neural networks or Kalman filter.

Adaptive neural fuzzy inference system (ANFIS) is an idea by combining the fuzzy inference system with neural network. The fuzzy inference system is used widely in fuzzy control, it can number rules by leading into a new ideal of membership function to deal with structural knowledge. ANFIS fully makes use of the excellent characteristics of neural network and fuzzy inference system. ANFIS can approach all nonlinear system with less training data and quicker weakening speed and higher precision.

In the following sections, the structure of the neuro-fuzzy inference system and prediction of the chaotic robot trajectory by using the ANFIS algorithm will be explained, respectively. Finally, the result of simulation and conclusion can be seen.

II. Adaptive Neuro-Fuzzy Inference system

In this section, we describe a class of adaptive network that are functionally equivalent to fuzzy inference systems. The propose architecture is referred to as ANFIS [2], which stands for adaptive network-based fuzzy inference system. We describe how to decompose the parameter set to facilitate the hybrid learning rule for ANFIS architecture representing both the Sugeno and Tsukamoto fuzzy models.

A. *ANFIS architecture*

For simplicity, we assume that the fuzzy inference system under consideration has two input x and y and output z. For a first-order Sugeno fuzzy model, a common rule set with two fuzzy if-then rules is the following [2]:

Rule 1: If x is A_1 and y is B_1, then $f_1=p_1x+q_1y+r_1$,
Rule 2: If x is A_2 and y is B_2, then $f_2=p_2x+q_2y+r_2$.

Figure 3 illustrate the reasoning mechanism for this Sugeno model; the corresponding equivalent ANFIS architecture is shown in figure 4, where nodes of the same layer have similar functions, as described next

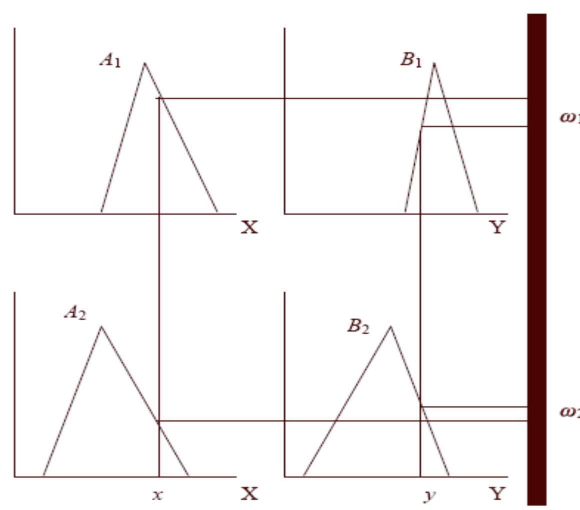

Fig. 3 Sugeno fuzzy model, with two inputs and two fuzzy if-then rules has been shown.

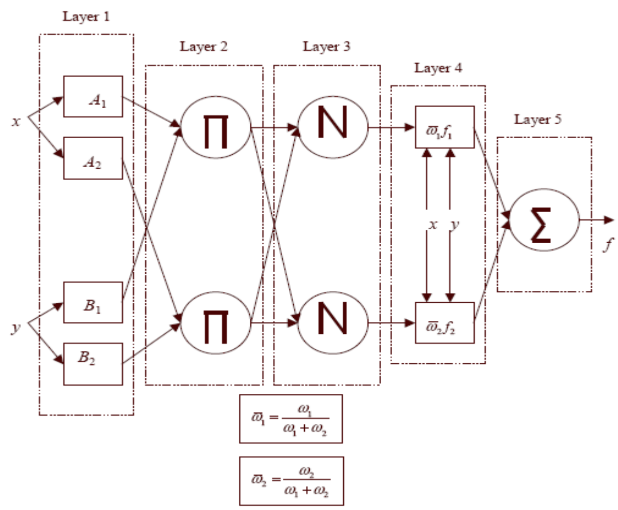

Fig. 4 Structure of ANFIS algorithm

Layer 1. Every node I in this layer is an adaptive node with a node function

$$O_{1,i} = \mu_{A_i}(x)\text{, for } i\text{=1, 2} \qquad (7) \qquad \text{or}$$

$$O_{1,i} = \mu_{B_{i-2}}(y)\text{, for } i\text{=3, 4} \qquad (8)$$

Where x (or y) is the input to node i and A_i (or B_{i-2}) is a linguistic label (such as "small" or "large") associated with this node. In other words, i is the membership grade of a fuzzy set A (=A_1, A_2, B_1 or B_2) and it specifies the degree to which the given input x (or y) satisfies the quantifier A. Here the membership function for A can be any appropriate parameterized membership function, such as the generalized bell function:

$$\mu_A(x) = \frac{1}{1+\left|\frac{x-c_i}{a_i}\right|^{2b}} \qquad (9)$$

Where $\{a_i, b_i, c_i\}$ is the parameter set. As the values of these parameters change, the bell-shaped function varies accordingly, thus exhibiting various forms of membership for fuzzy set A. Parameters in this layer are referred to as premise parameters.

Layer 2. Every node in this layer is a fixed node labeled Π, whose output is the product of all the incoming signals:

$$o_{2,i} = w_i = \mu_{A_i}(x)\mu_{B_i}(y)\text{, i=1,2} \qquad (10)$$

Each node output represents the firing strength of a rule. In general, any other T-norm operators that perform fuzzy AND can be used as the node function in this layer.

Layer 3. Every node in this layer is a fixed node labeled N. The i-th node calculates the ratio of the i-th rule's firing strength to the sum of all rule's firing strengths.

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1+w_2}\text{, i=1,2} \qquad (11)$$

For convenience, outputs of this layer are called normalized firing strengths.

Layer 4. Every node I in this layer is an adaptive node with a node function

$$O_{4,i} = \overline{w}_i\ f_i = \overline{w}_i\left(p_i x + q_i y + r_i\right) \qquad (12)$$

Where ϖ_i is a normalized firing strength from layer 3 and $\{p_i, q_i\ r_i\}$ is the parameter set of this node. Parameters in this layer are referred to as consequent parameters.

Layer 5. The single node in this layer is a fixed node labeled Σ, which computes the overall output as the summation of all incoming signals.

$$O_{5,i} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \qquad (13)$$

B. Hybrid Learning Algorithm

From ANFIS architecture shown in the figure 4, we observe that the values of the premise parameters are fixed, the overall output can be expressed as a linear combination of the consequent parameters. In symbols, the output f in the figure 4 can be rewritten as

$$\begin{aligned} f &= \frac{w_1}{w_1+w_2} f_1 + \frac{w_2}{w_2+w_2} f_2 = \\ &\overline{w}_1(p_1x+q_1y+r_1)+\overline{w}_2(p_2x+q_2y+r_2)= \\ &(\overline{w}_1x)p_1+(\overline{w}_1y)q_1+(\overline{w}_1)r_1+(\overline{w}_2x)p_2+(\overline{w}_2y)q_2+(\overline{w}_2)r_2 \end{aligned} \qquad (14)$$

Which is linear in the consequent parameters p_1,q_1,r_1,p_2,q_2 and r_2.

The learning algorithm for ANFIS is a hybrid algorithm which is a combination between gradient descent and least-squares method. More specifically, in the forward pass of the hybrid learning algorithm, node outputs go forward until layer 4 and the consequent parameters are identified by the least-squares method. In the backward pass, the error signals propagate backward and the premise parameters are updated by gradient descendent. Table1 summarizes the activities in each pass.

The consequent parameters are identified optimal under the condition that the premise parameters are fixed. Accordingly, the hybrid approach converges much faster since it reduced the search space dimensions of the original pure back propagation method.

TABLE 1: LEARNING PARAMETERS OF THE ANFIS ALGORITHM

	Forward pass	Backward pass
Premise parameters	Fixed	Gradient descent
Consequent parameters	Least-Squares estimator	Fixed
Signals	Node Outputs	Error signals

III. PREDICTION THE BEHAVIOR OF THE CHAOTIC ROBOT USING ANFIS ALGORITHM

In this part the application of the ANFIS algorithm in predicting the behavior of the chaotic robot and its next values is considered.

The goal of the task is to use past values of the dynamic up to time t to predict the value at some point in the future $t+P$. The standard method for this type of prediction is to create a mapping from D points of the dynamic spaced D apart that is, $[x(t-(D-1)\Delta),..,x(t-\Delta),x(t)]$, to a predicted future value $x(t+P)$. In our simulation, the values D=4 and $\Delta=P$=6 were used.

We have extracted 1000 input-output data pairs of the following format from the robot trajectory and used ANFIS algorithm for prediction of next values:

$$[x(t-18), x(t-12), x(t-6), x(t); x(t+6)]$$

The first 500 pairs were used as training data set for ANFIS, while the remaining 500 pairs were the checking data set for validating the identified ANFIS. For predicting of the chaotic robot behavior, we have considered it separately in two axis X and Y and used the ANFIS algorithm at each of these axes [3]. The number of membership functions assigned to each input of the ANFIS was set to two that have been selected bell shaped, so the number of rules is 16.

In next section, the results of simulation consisting of error diagram, main trajectory and ANFIS prediction of the chaotic robot will be reviewed [4].

IV. Simulation Results

In this part the results of simulation in X direction and then in Y direction are shown.

A. *Result of simulation in X direction:*

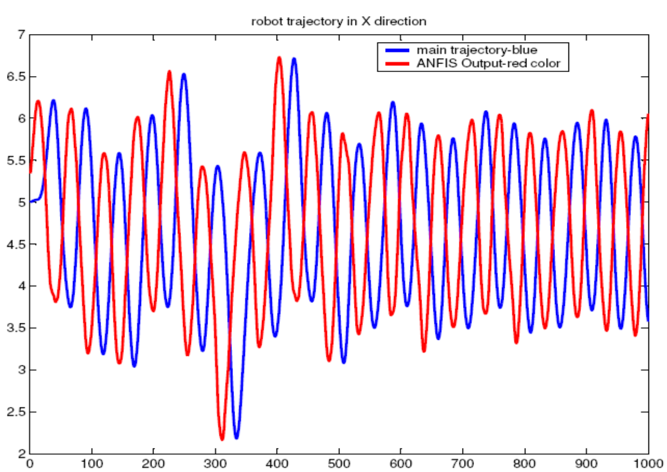

Fig. 5 chaotic robot trajectory in X direction
(blue : main trajectory, red : ANFIS output)

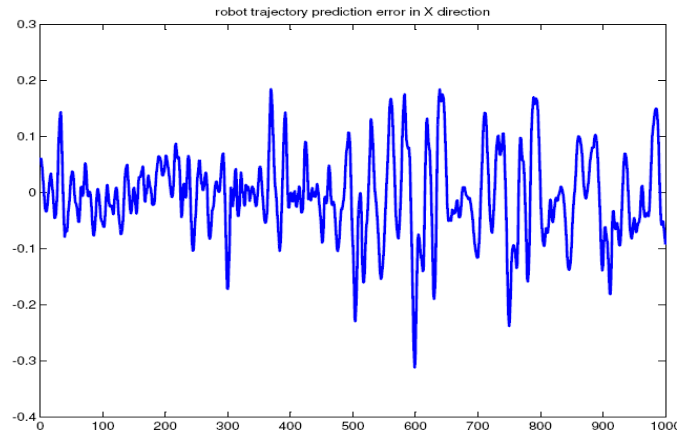

Fig. 6 chaotic robot trajectory prediction error in X direction

B. *Result of simulation in Y direction:*

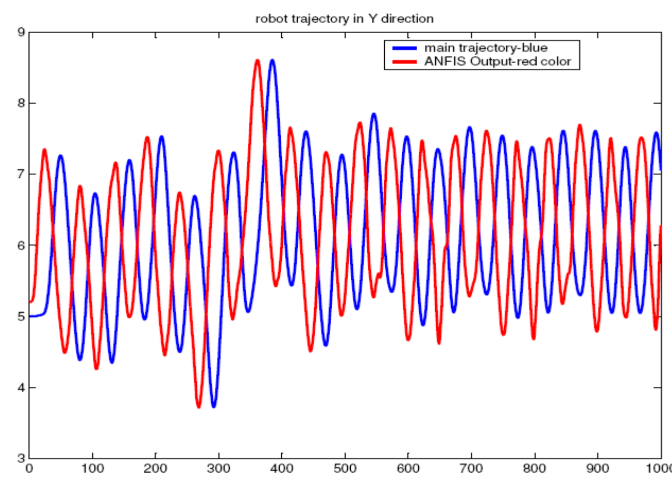

Fig. 7 chaotic robot trajectory in Y direction
(blue : main trajectory, red : ANFIS output)

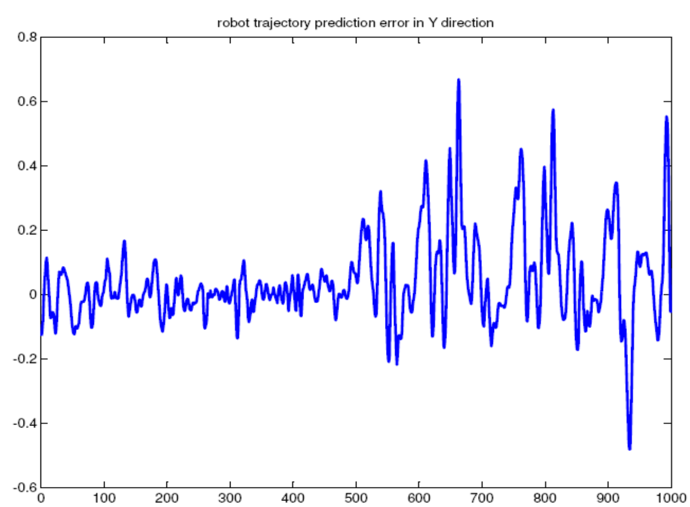

Fig. 8 chaotic robot trajectory prediction error in Y direction

V. CONCLUSION

In this paper, the prediction of the behavior of a chaotic robot by using an adaptive neuro-fuzzy inference system accomplished. We extracted 1000 input-output data pairs. The first 500 pairs were used as training data set for ANFIS, while the remaining 500 pairs were the checking data set for validating the identified ANFIS. For predicting the chaotic robot behavior, we have considered it separately in two axes X and Y and used the ANFIS algorithm at each of these axes.

The result of ANFIS and prediction error of the considered robot shows that this algorithm is more accurate in compare with methods used neural networks or Kalman filter for predicting [5].

REFERENCES

[1] Yoshihiko Nakamura and Akinori Sekiguchi, The Chaotic Mobile Robot, IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. 17, NO. 6, DECEMBER 2001

[2] Jang S. R., Sun C. T. and Mitzutani E., Neurofuzzy and Soft Computing, Prentice Hall, 1998. New York.

[3] Cuevas E., Zaldivar D. and Rojas R., Intelligent Tracking, Technical Report B-13- 03, Freie Universität Berlin, November, 2003.

[4] Fuzzy logic Toolbox, Mathworks, 1999, New York.

[5] Tae-Wan, Kim, Mackey-Glass time series prediction using RNN and ANFIS, Fuzzy & Intelligent System, intelligent Multimedia Lab., Dept. of Computer & Communication Engineering, , June 1, 2007, POSTECH, South Korea.