INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 7, Number 1, Pages 66–86

NUMERICAL DYNAMIC MODELING AND DATA DRIVEN CONTROL VIA LEAST SQUARE TECHNIQUES AND HEBBIAN LEARNING ALGORITHM

XIAODONG LIU AND YANLI ZHANG

Abstract. The modelling and controlling for complex dynamic systems which are too complicated to establish conventionally mathematical mechanism models require new methodology that can utilize the existing knowledge, human experience and historical data. Fuzzy cognitive maps (FCMs) are a very convenient, simple, and powerful tool for simulation and analysis of dynamic systems. Since human experts are subjective and can handle only relatively simple FCMs, there is an urgent need to develop methods for automated generation of FCM models using historical data. In this paper, a novel FCM, which is automatically generated from data and can be applied to on-line control, is developed by improving its constitution, introducing Least Square methods and using Hebbian Learning techniques. As an illustrative example, the simulations results of truck backer-upper control problem quantifies the performance of the proposed constructions of FCM and emphasizes its effectiveness and advantageous characteristics of the learning techniques and control ability.

Key Words. Least Square Learning, Fuzzy cognitive map, Takagi_Sugeno model, Complex dynamic system, Hebbian learning algorithm

1. Introduction

Many conventional methods were used, successfully, to model and control systems but their contribution is often limited in the representation, analysis and solution of the systems with well established mathematically analyzable models. Unfortunately, it is impossible or costly to build the mathematical analyzable model for most of complex nonlinear systems. Currently the advanced digital technology has made the digitized data easy to capture and cheap to store. So there is a great demand for the development of intelligence and data based, autonomous systems that can be achieved taking advantage of human like reasoning and the available data from the systems and their circumstance. Human reasoning process for any procedure includes uncertain descriptions and can have subtle variations in relation to time and space. For such situations, Fuzzy Cognitive Maps(FCMs)[6] seems to be capable to deal with. Fuzzy cognitive maps (FCMs) is a soft computing technique for modeling complex systems, which follows an approach similar to human reasoning and the human decision-making process. FCMs can successfully represent knowledge and human experience, introducing concepts to represent the essential

Received by the editors January 1, 2009 and, in revised form, March 22, 2009.
2000 *Mathematics Subject Classification.* 06D72, 03B52.
This work is supported by the Natural Science Foundations of China under grant 60575039 60875032.

elements and the cause and effect relationships among the concepts to model the behavior of any system.

In [7], Kosko pointed out that it is very difficult to build FCMs for large scale intelligent systems just relying on human experts who can observe and know the operation of the systems. The large amount of temporal knowledge discovered from the database is important information which objectively and truthfully reflects the nature of dynamic complex systems. Applying this temporal knowledge to construct FCMs has two significant benefits: One is that large amount of knowledge discovered from huge databases can be integrated into a model which not only can be systematically studied by many powerful mathematical tools, but also can be very expediently understood and utilized by users. The other is that large-scale intelligent systems, which are beyond human expert's capability to observe and understand, can be represented and control via constructing large dimensional data and human knowledge based FCMs. Therefore, the investigation of constructing FCMs by the temporal data from the complex dynamic systems is very important and is one main objective of this paper.

Kosko proposed a new model by using simple Differential Hebbian Learning law (DHL) in 1994, but he used this model to learning FCMs without any applications[2]. This learning process modified weights of edges existing in a FCM in order to find the desired connection weights. In general, when the values of corresponding concept changes, the value of the related edges for that nodes will be modified too. In 2002, Vazquez introduced a new extension to DHL algorithm presented by Kosko. He used a new idea to update edge values in a new formula [22]. This method was applied only to FCMs with binary concept values, which significantly restricts its application areas. Another method of learning FCM based on the first approach(Hebbian algorithm), was introduced in 2003 by Papageorgiou et al. He developed another extension to Hebbian algorithm, called Nonlinear Hebbian Learning (NHL)[23]. The main idea behind this method is to update weights associated only with edges that are initially suggested by experts. The NHL algorithm requires human intervention before the learning process starts, which is a substantial disadvantage. Active Hebbian Algorithm (AHL) introduced by Papageorgiu et al. in 2004[13]. Nevertheless it still requires some initial human intervention. In the recent method, experts not only determined the desired set of concepts, initial structure and the interconnections of the FCM structure, but also identified the sequence of activation concepts[13]. Another category in learning connection weights of the FCM is application of genetic algorithms or evolutionary algorithms. Koulouriotis et al. applied the Genetic Strategy (GS) to learn FCM structure in 2001[24]. In mentioned model, they focused on the development of an GS-based procedure that determines the values of the cause-effect relationships (causality). Parsopoulos et al. also published other related papers in 2003. They tried to apply Particle Swarm Optimization (PSO) method, which belongs to the class of Swarm Intelligence algorithms, to learn FCM structure[11]. Khan and Chong worked on learning initial state vector of FCM in 2003. They performed a goal-oriented analysis of FCM and their learning method did not aim to compute the connection weights, and their model focused on finding initial state vector for FCMs [26]. In 2005, Stach et al. applied real-coded genetic algorithm (RCGA) to develop FCMs model from a set of historical data in 2005[25]. In most case, the performance of genetic programming depends crucially on the choice of representation and on the choice of fitness function and must search in the huge hypothesis space, hence they have the issue of the potential convergence and heavy calculation burden.

The previous researches of FCMs such as the forementioned ones were often undertaken by the assumption that the connection weights of an FCM are invariant for a given system. However a real world application system may have such high degree of nonlinearity that it is difficult or impossible to properly represent its complex dynamic behavior using the invariant weights. In this paper, the time-varying connection weights of a FCM are introduced via the aggregation of the local invariant connection weights in the IF-THEN rules, like Takagi-Sugeno (T-S) fuzzy models. Thus the FCMs with time-varying connection weights can more properly and accurately represent complex dynamic systems than the conventional FCMs.

The learning algorithms for constructing FCMs in the existing literatures such as the forementioned ones often need iterative calculation which leads to a heavy calculation burden, as well as convergence problem and iterative stopping criteria have to be taken into consideration. This paper introduces the Least Square technique in the procedures of learning the local connection weights of the FCMs based on the historical data, briefly called LS-FCMs. Then the local LS-FCMs are integrated into a FCM for the whole domain via the fuzzy IF-THEN rules, briefly named TS-LS-FCM. TS-LS-FCMs not only can overcome the above mentioned issues in the previous algorithms, but also greatly improve their capability to describe the dynamic nonlinear systems.

A TS-LS-FCM, which is learned from the historical data, not only can describe and predict the dynamic behavior of a given nonlinear system very well, but also can be applied to control the real system as follows. First the learned TS-LS-FCM is connected to the physical system via adding some new nodes and edges. Here the adding nodes may represent the observed states and the input variables of the physical system, and the adding nodes are connected to the relative nodes in the learned TS-LS-FCM by adding some edges. Then the physical system is controlled to a desired goal by just learning the weights of the adding edges connected to the input variables using Hebbian Learning techniques without any modification to the learned TS-LS-FCM.

This proposed TS-LS-FCM technique is a promising approach for objectively extracting information from temporal data and at the same time improving the effectiveness of the FCM operation mode and thus it broadens the applicability of FCMs modelling for complex systems. As an illustrative example, the truck backer-upper control problem demonstrates that the proposed methodology is more convenient and effective than the previous approaches of FCMs.

The remainder of the paper is organized as follows. Section 2 briefly describes the formulation of FCMs. Section 3 presents proposed the Least Square Learning FCM methods based on historical data and the TS-LS-FCM. Section 4 introduces the control-scheme of dynamic systems by the learned TS-LS-FCM via the Hebbian Learning. Section 5 applies the proposed framework to the truck backer-upper control problem to demonstrate its learning capability of the TS-LS-FCM and the control ability of the proposed control algorithm. Finally, Section 6 provides the conclusion.

2. Fuzzy Cognitive Maps, Research Objective and Methodology

This section first presents a historical overview of FCMs along with background information concerning both the underlying model and the ensuing learning methods. Then we concisely outline this research objective and methodology.

2.1. Fuzzy Cognitive Maps.

2.1.1. Background. Fuzzy Cognitive Maps have their roots in graph theory. Axelord for the first time used cognitive maps as a formal way to represent social scientific knowledge and to model decision-making in social and political systems. The cognitive map model is represented by a simple graph, which consists of nodes and edges. The nodes represent concepts relevant to a given domain and the casual relationships between them are depicted by directed edges. Each edge is associated with only two values, $+1$ (or simply "+") and -1(or simply "−"). A positive edge from a node A to a node B reflects positive influence on B exerted by A. It means that an increase in the value of the node A will lead to an increase in the value of the node B. A negative edge from the node A to the node B means that increasing value of A leads to decreasing value of B.

"Fuzzy Cognitive Maps" was coined by Kosko[6] in order to describe a cognitive map model with a significant characteristic: Causal relationships between nodes are fuzzed. Instead of only using signs to indicated positive or negative causality. Each edge is associated with a number that determines the degree of considered casual relation.

FCMs describe the behavior of a system in terms of concepts: each concept represents a state, variable or a character of the system. Values of concepts (nodes) change over time, and take values in the interval [0,1]. The causal links between concepts are represented by directed weighted edges that illustrate how much one concept influences the interconnected concepts. The cause and effect interconnection between two concepts C_j and C_i is described with the weight w_{ji}, taking value in the range -1 to 1. System with n concepts can be represented by a $n \times n$ connection matrix. An example FCM model signed and weighted arcs is shown in Fig.1. There are three possible types of causal relationships between concepts:

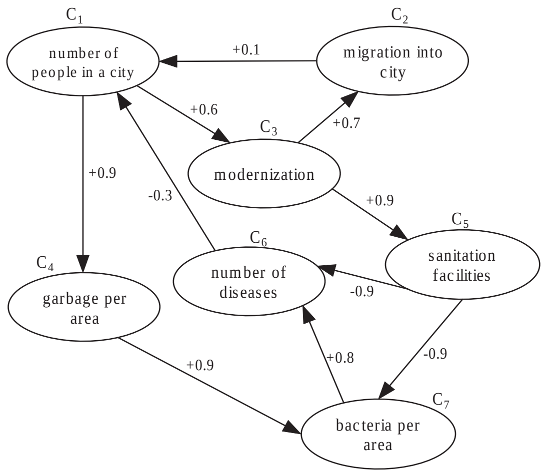
- $w_{ji} > 0$: which indicates positive causality between concepts C_j and C_i. That is, an increase (decrease) in the value of C_j leads to an increase(decrease) in the value of C_i.
- $w_{ji} < 0$: which indicates negative causality between concepts C_j and C_i. That is, an increase (decrease) in the value of C_j leads to a decrease(increase) in the value of C_i.
- $w_{ji} = 0$: which indicates no relationship between C_j and C_i.

Behind the graphical representation of an FCM there is a mathematical formulation which describes the FCM. Values of concepts are fuzzy and arise from the transformation of the real values of the corresponding variables for each concept, and there are fuzzy values for the weights of the interconnections among concepts, then the FCM is free to interact, at every step of interaction every concept has a new value that is calculated according to the following FCM transformation function:

$$A_j^t = f\left(\sum_{i=1, i\neq j}^{N} A_i^{t-1} w_{ij} + A_j^{t-1}\right) \tag{1}$$

Where, A_j^t is the value of concept C_j at step t, which means the activation degrees of concept C_j at step t, A_i^{t-1} is the value of concept C_i at step $t-1$, and w_{ij} is the weight of the interconnection from concept C_i to concept C_j, and f is a squashing function that squashes the result of the multiplication in the interval [0,1]. Continuous Squashing Functions usually used [23][30] are shown as follows:

1) $f = \tanh(x)$ maps the nodes values in [-1,1]



	C_1	C_2	C_3	C_4	C_5	C_6	C_7
C_1	0	0	0.6	0.9	0	0	0
C_2	0.1	0	0	0	0	0	0
C_3	0	0.7	0	0	0.9	0	0
C_4	0	0	0	0	0	0	0.9
C_5	0	0	0	0	0	-0.9	-0.9
C_6	-0.3	0	0	0	0	0	0
C_7	0	0	0	0	0	0.8	0

Fig. 1 An FCM for a public health study

2) $f = \frac{1}{1+e^{-cx}}$ converts the nodes values in [0,1]. It is also called sigmoid function.

The second function is the most common function which is used in FCM model. Where c is a parameter used to determine proper shape of the function.

Eq.(1) describes a functional model of FCM, which is used to perform simulations of the system dynamical behaviors. Simulation consists of computing the states of the system, which is described by a state vector, over a number of successive iterations. The state vector specifies current values of all concepts(nodes) in a particular iteration. Value of a given node is defined by the result of taking all the causal event weights pointing into this concept and multiplying each weight by the value of the concept that causes the event, then the sigmoid function is applied to the result of the calculations and it is transformed to the interval between 0 and 1.

The FCM model was applied to many different areas to express dynamic behavior of a set of related concepts. For example, the public city health issue, heat changer model[15], supervisor model for heat exchanger performance[16], assessment of human reliability[18],determination of brain tumor grade[19], modelling of IT project[20], modelling of supply chain[21], E-business models and so on.

2.1.2. Related work. In general, two approaches to development of FCMs are used: manual and computational. Most, if not all, of the reported models were developed manually by domain experts based on expert knowledge in the area of application. The experts design and implement adequate model manually based on their mental understanding of the model domain. However, the main difficulty is to accurately establish weights of the defined relationships. The manual procedures for the development of FCM have a number of drawbacks. They require an expert, who has knowledge of the model domain, and at the same time knowledge about the FCM formalism. Also, manual methods imply subjectivity of the developed model and problems with unbiased assessing of its accuracy.

These problems led to the development of computational methods for learning FCM connection matrix i.e. causal relationships(edges), and their strength (weights) based on historical data. In this way, the expert knowledge is substituted by a set of historical data and a computational procedure that is able to automatically compute the connection matrix.

At present, weight learning based on the Hebbian algorithm, which is a kind of artificial neural network learning algorithms, has been presented[8]. The Hebbian algorithm changes the weights gradually toward reducing the differences between the state vectors predicted by FCM and actual state vectors. Also Papageorgiou and Groumpos presented a two-step weight learning methodology where differential evolution algorithm decides the ending point of the learning and a nonlinear Hebbian learning algorithm executes the weight learning during the learning interval. Besides this Parsopoulos, Papageorgiou, Grumpos, and Vrahatis (2003)performed a research on the weight learning using particle swarm method, a heuristic algorithm for seeking optimal solutions. Recently, in 2005, Stach, Kurgan, Pedrycz et al. applied genetic learning method to learning FCM connection matrix. Aforementioned methodologies used for constructing FCM model from historical data have the following limitations that can be overcome by employing the TS-LS-FCM methodology presented in this paper.

- i Most of previous learning algorithms need iterative calculation which leads to a heavy calculation load , as well as convergence problem and iterative stopping criteria have to be taken into consideration.
- ii The range of A_j^t calculated by the transformation function in Eq.(1) is too limited to represent the real world applications. For example, if for any i, $w_{ij} \geq 0$, then A_j^t is always larger than 0.5. It is clear that most real systems do not observe this rule. Thus the applicability of the existing FCMs is limited.
- iii The FCM model with an invariant connection matrix is difficult to work effectively in the high degree nonlinearity and wide operating range of some real world application systems. Therefore the FCMs with time-varying connection weights have to be developed.

2.2. Research Objective and Methodology. The first goal of this research is to develop a new FCM technique to overcome the aforementioned three issues. Utilization of the Least Square methodology can significantly overcome the most weakness of the existing FCMs, namely the heavy calculation burden, convergence and iterative stopping criteria. In the proposed LS-FCMs, the squashing function function is improved by adding a tuning coefficient to make it more applicable. Thus the forementioned issue ii) can be overcome. We decompose the domains of concepts to several local regions and construct a local LS-FCM for each region, then by the integration of local connection weights and the tuning coefficients in the fuzzy IF-THEN rules via the T-S fuzzy model, then a TS-LS- FCM with time-varying connection weights and the tuning coefficients is constructed.

The second objective of the research is to provide on-line control using the TS-LS-FCM learned from the historical data. For a desired goal, apply the learned TS-LS-FCM and Hebbian learning to find the control law.

We note that the proposed method is a natural continuation of the research performed in the domain of learning and applying FCMs. It draws conclusions from the methods proposed in the past, and provides substantial advancement. Next section provides detailed description of the proposed method.

3. Least Squares Learning FCMs

The proposed LS-FCM extracts the knowledge(FCM model) by learning the connection weights and the tuning coefficients from historical data. The whole scheme is outlined as the following points:

1. **Normalization of data**: In order to keep the comparability among concepts, the original data $x = (x_1, x_2, ..., x_n) \in R^n$ is formalized to $\overline{x} = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_n)$, $\overline{x}_i \in [0, 1]$, where

$$\overline{x}_i = \frac{1}{1 + e^{-ct_i(x_i - m_i)}}, \quad i = 1, 2, ..., n. \tag{2}$$

2. **FCM operation function**: A tuning coefficient w_{0j} is added into the aforementioned FCM operation function Eq.(1).

$$A_j^t = f\left(\sum_{i=1, i\neq j}^{n} A_i^{t-1} w_{ij} + w_{0j} + A_j^{t-1}\right), \quad f = \frac{1}{1 + e^{-cx}} \tag{3}$$

3. **Learning connection weights and the tuning coefficients**: For a FCM with concepts C_j, $j = 1, 2, ..., n$, let $C_{j_1}, C_{j_2}, .., C_{j_{l_j}}$ are all concepts connected to C_j, $\{j_1, j_2, ..., j_{l_j}\} \subseteq \{1, 2, ..., n\}$. Let $X_t = (x_{j_1 t}, x_{j_2 t}, ..., x_{j_{l_j} t})'$ and x_{jt} be the historical data of the values of concepts connected to C_{j_k} and C_j at time t, $k = 1, 2, ..., l_j, t = 1, 2, ..., T$. The connection weights $w_{j_1 j}, w_{j_2 j}, ..., w_{j_{l_j} j}$ and w_{0j} are learned via Least Square technique according to the data$< X_t, x_{jt} >$.

4. **Domain partition**: The domain of system states and variables is parted into a proper number of regions respecting with a given error limitation of the prediction of the LS-FCM in each region.

5. **Fuzzy integration**: The connection weights and the tuning coefficients of the LS-FCMs learned by the above procedures 1, 2, 3 in the local regions are merged into a single fuzzy cognitive map TS-LS-FCM with time-varying connection weights and the tuning coefficients in the whole domain.

In what follows, we present, illustrate and analyze each of the procedures 1 to 5 in detail.

3.1. Normalization of Data. In the framework of FCMs, values of concepts(nodes) change with time, and have to take them in the interval [0,1]. But the values of the concepts in real systems derived from different measurements may take a wide range, sometimes take negative. So, we need normalize dynamic data within different range into the interval [0,1]. Ranges of the values of different states and variables of a physical system may be greatly different. In this situation, a simplified method to normalize the real value to [0,1] and avoid weakening comparability among concepts is necessarily. Eq.(2) transforming data to the range [0,1] is adopted in this paper. We achieve two objectives by using the formula, it normalizes the range of all features of the original data into [0,1] and makes each node having the similar range size in order to strengthen comparability among nodes. In Eq.(2), m_i is the middle value of ith concept domain, $t_i > 0$ is the proportion where $t_i = \frac{S_{max}}{S_i}$, S_{max} = the max span of all concepts in system, S_i is the span of ith concept in system. $c > 0$ is the parameter used to tune the mapped domain in [0,1].

For example, let an FCM have two concepts named x, y, $x \in [-1, 3], y \in [-100, 100]$. If they are normalized in [0,1] by $f = \frac{1}{1+e^{-cx}}$ and $c = 0.1$, then $x \to [0.4750, 0.5744]$, $y \to [0, 1]$. But if Eq.(2) is applied, then $m_x = \frac{-1+3}{2} = 1, t_x = \frac{100-(-100)}{3-(-1)} = 50, m_y = 0, t_y = \frac{100-(-100)}{100-(-100)} = 1$ and $x \to [0, 1]$, $y \to [0, 1]$.

3.2. FCM Operation Function. Most of previous FCMs, at each step, the value A_j of a concept is calculated according to Eq.(1) $A_j^t = f\left(\sum_{i=1, i\neq j}^{n} A_i^{t-1} w_{ij} + A_j^{t-1}\right)$ and sigmoid function $f = \frac{1}{1+e^{-cx}}$, where $A_j^t \in [0, 1]$ is the value of concept C_j at step t, $A_i^{t-1} \in [0,1]$ is the value of concept C_i at step $t-1$, and w_{ij} is the weight of the interconnection from concept C_i to concept C_j, and $c > 0$ is a parameter used to determine proper shape of the function. Let's study sigmoid function $y = \frac{1}{1+e^{-cx}}$. It is clear that if $x > 0$ and $c > 0$, then y always larger than 0.5. This implies that if for any i, $w_{ij} > 0$, then A_j^t is always larger than 0.5 for any A_j^{t-1}. If $A_j^{t-1} = 0$ and for any i, $w_{ij} = 0$, then $A_j^t = 0.5$ for any time t. The applicability of conventional FCMs is greatly limited by this unpractical constrain. In this paper, a novel FCM operation function(Eq.(3)) is presented. Where $c > 0$ is a parameter used to determine proper shape of the function. $w_{0j} \in [-1,1]$ is the tuning coefficient of concept C_j. If for any i, $w_{ij} > 0$, $w_{0j} < 0$, and $\sum\limits_{i=1, i\neq j}^{n} (A_i^{t-1} w_{ij} + A_j^{t-1}) > -w_{0j}$, then A_j^t is larger than 0.5. If for any i, $\sum\limits_{i=1, i\neq j}^{n} (A_i^{t-1} w_{ij} + A_j^{t-1}) < -w_{0j}$, then A_j^t is less than 0.5. Assume that for any i, $w_{ij} = 0$. This implies that the value of concept C_j is not influenced by any other concepts, i.e.,$A_j^t = f(w_{0j} + A_j^{t-1})$. In this situation , the changing of node j is only determined by w_{0j}. The roles of the parameters c, w_0 in Eq.(3) are shown in Fig.2. Thus the FCMs based on operation function Eq.(3) are more flexible and applicable.

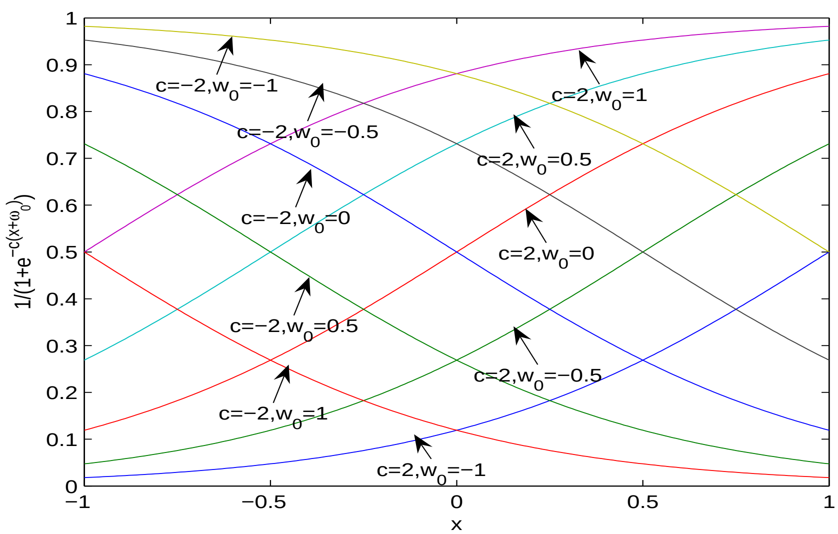


Fig. 2 The operation functions $f(x) = \frac{1}{1+e^{-c(x+w_0)}}$ for different parameters c and w_0

3.3. Learning Connection Weights and the Tuning Coefficients. For every concept C_j, the connection weights $w_{j_1 j}, w_{j_2 j}, ..., w_{j_{l_j} j}$ and w_{0j} are learned via Least Square technique according to the data $< X_t, x_{jt} >$, Where $C_{j_1}, C_{j_2}, .., C_{j_{l_j}}$ are all concepts connected to C_j, $\{j_1, j_2, ..., j_{l_j}\} \subseteq \{1, 2, ..., n\}$, and $X_t = (x_{j_1 t}, x_{j_2 t}, ..., x_{j_{l_j} t})'$ and x_{jt} be the values of concepts connected to C_j and C_j at time t, $t = 1, 2, ..., T$. In general, T is much larger than l_j. For the sake of simplicity, let $j_1 = 1, j_2 = 2, ..., j_{l_j} = N$,$w_{0j} = w_0, w_{j_1 j} = w_1, w_{j_2 j} = w_2, ..., w_{j_{l_j} j} = w_N$ and x_{jt}=y_t. Thus by

Eq.(3), we have

$$y_t = \frac{1}{1 + e^{-c(\sum_{k=1}^{N} w_k x_{k(t-1)} + y_{t-1} + w_0)}}, \quad t = 1, 2, ..., T. \tag{4}$$

Further

$$\sum_{k=0}^{N} w_k x_{k(t-1)} = -c^{-1} ln(y_t^{-1} - 1) - y_{t-1}, \quad x_{0t} = 1, \quad t = 1, 2, ..., T. \tag{5}$$

Let $-c^{-1}ln(y_t^{-1} - 1) - y_{t-1} \triangleq c_t, t = 1, 2, ..., T$. Then the best weights $w_i, i = 0, 1, 2, ..., N$ fitting the historical data $< X_t, y_t >$ are obtained by solving the following Least Square problem,

$$\min_{w_0, w_1, \ldots, w_N} \sum_{t=1}^{T} (c_t - \sum_{k=0}^{N} w_k x_{k(t-1)})^2. \tag{6}$$

Let

$$\frac{\partial}{\partial w_k} (\sum_{t=1}^{T} (c_t - \sum_{j=0}^{N} w_j x_{j(t-1)})^2) = 0, \quad k = 0, 1, ..., N. \tag{7}$$

One has the following normal equation

$$AA'w = Ac, \tag{8}$$

where $A = (x_{ji})_{(N+1)\times T}$, $w = (w_0, w_1, ..., w_N)'$, $c = (c_1, c_2, ..., c_T)'$. By solving the normal linear equation Eq.(8), one can obtain all weights which describe the degree of cause and effect of all concepts connected to the concept C_j, $j = 1, 2, ..., n$. Thus all connection weights w_{ij} and all tuning coefficients w_{0j} can be learned by applying the above Least Square algorithm to each concept. The FCMs constructed by this algorithm are called LS-FCMs.

It is clear that LS-FCM algorithm, which is the solutions of n simple linear equations, does not require iterative calculation and greatly alleviate computation cost. By introducing the tuning coefficients w_{0j}, LS-FCMs greatly improve the prediction accuracy of the FCMs learned by Hebbian learning in which the range of output of each concept is limited either $[0, 0.5]$ (all weights of the concepts connected to this concept are negative) or $[0.5, 1]$ (all weights of the concepts connected to this concept are positive). Further more LS-FCMs avoid the complicated issues such as the convergence, iterative stopping condition and learning rate which the other FCM learning algorithms have to deal with. All the aforementioned advantages of LS-FCMs will be demonstrated by the examples in Section V.

3.4. Domain Partition. The range of the states and variables of a real world application may be too large to be represented by a LS-FCM model. Therefore, the whole domain of the system should be parted into some local regions, then a LS-FCM model is constructed for each region. Since the complexity of relationship among the states and variables may greatly different in different regions, hence the range could not be uniformly parted. In this paper the domain is parted according to the error limitation of the LS-FCMs constructed in the local regions. Each local region gradually enlarges until the error between the LS-FCM and the physical system exceeds the error limitation.

3.5. Fuzzy Integration. In 1985, Takagi and Sugeno proposed T-S fuzzy model to describe nonlinear systems using a set of conditional fuzzy rules or local models, with each one being valid in a particular operating region determined by the corresponding specification of antecedent membership functions. A conditional part or antecedent defines each operating region. The consequent part of each rule is an analytical expression describing the corresponding local model. These collections of local models together with a merging procedure are the basic ingredients of Takagi-Sugeno (T-S) model. In general the aggregation procedure consists of a convex combination of the corresponding local models defined in the corresponding local region. Standard T-S systems provide a formal method for modelling vagueness of local regions associated to the input domain and an output merging procedure of the corresponding fuzzy rules.

After the construction of a sub LS-FCM in each local region, a TS-LS-FCM with time-vary connection weights and tuning coefficients is constructed based on the Takagi- Sugeno (T-S) fuzzy model as follows. Consider the following fuzzy IF-THEN rules:

$$\text{(9)} \quad \textit{Rule } l: \ \textit{IF } x_{1t} \textit{ is } \widetilde{A_1^l} \textit{ and } x_{2t} \textit{ is } \widetilde{A_2^l}, ..., \textit{ and } x_{pt} \textit{ is } \widetilde{A_p^l} \textit{ THEN } W = W^l.$$

Where $l = 1, ..., r$, r is the number of local regions, W^l is a matrix with the connection weights and the tuning coefficients of the LS-FCM in lth local region. $X_t = (x_{1t}, x_{2t}, ..., x_{pt})$, x_{it}, $i = 1, 2, ..., p$, are the state variables of system at time t, namely the premise variables and measurable. $\widetilde{A_1^l}, \widetilde{A_2^l}, ..., \widetilde{A_p^l}$ are the fuzzy sets in lth local region. Further more the time-varying matrix with the connection weights and the tuning coefficients are formulated as follows:

$$W(X_t) = \sum_{l=1}^{r} \lambda^l(X_t) W^l. \tag{10}$$

Where $\lambda^l(X_t) = \frac{\mu^l(X_t)}{\sum_{l=1}^{r} \mu^l(X_t)}$. $\mu^l(X_t) = \mu_1^l(x_{1t}) \times \mu_2^l(x_{2t}) \times ... \times \mu_p^l(x_{pt})$, $\mu_i^l(x_{it})$ is the membership degree of x_{it} in the fuzzy set $\widetilde{A_i^l}$. We always assume that $\sum_{l=1}^{r} \mu^l(X_t) \neq 0, \forall t$. It is obvious that $\sum_{l=1}^{r} \lambda^l(X_t) = 1$. The FCM with time-varying connection weight matrix $W(X_t)$ shown as (10) is called TS-LS-FCM.

Let $w_{uv}(X_t)$ be the connection weight from concept C_u to concept C_v and $w_{0v}(X_t)$ be the tuning coefficient of concept C_v at time t. Then by (10) and (3), we have

$$A_j^t = f\left(\sum_{i=1, i \neq j}^{n} A_i^{t-1} w_{ij}(X_{t-1}) + w_{0j}(X_{t-1}) + A_j^{t-1} \right) \tag{11}$$

$$w_{uv}(X_t) = \sum_{l=1}^{r} \frac{\prod_{j=1}^{n} (\mu_j^l(x_{jt}))}{\sum_{l=1}^{r} (\prod_{j=1}^{n} \mu_j^l(x_{jt}))} w_{uv}^l = \sum_{l=1}^{r} \frac{\mu^l(X_t)}{\sum_{l=1}^{r} \mu^l(X_t)} w_{uv}^l \tag{12}$$

$$w_{0v}(X_t) = \sum_{l=1}^{r} \frac{\prod_{j=1}^{n}(\mu_j^l(x_{jt}))}{\sum_{l=1}^{r}(\prod_{j=1}^{n}\mu_j^l(x_{jt}))} w_{0v}^l = \sum_{l=1}^{r} \frac{\mu^l(X_t)}{\sum_{l=1}^{r}\mu^l(X_t)} w_{0v}^l \tag{13}$$

4. Control of Dynamic System by TS-LS-FCM

Let D^{FCM} be the TS-LS-FCM learned according to the historical data of a physical system and A^{FCM} be the set of the states and input variables of the physical system. D^{FCM} is connected to the physical system by adding new edges to connect A^{FCM} (see Figure 3). Each state in A^{FCM} is connected to the concept in D^{FCM} representing the state and each concept in D^{FCM} is connected to the input variables in A^{FCM} which have relationship with the concept. Thus a new FCM included D^{FCM} and A^{FCM} is formed, briefly denoted as $D^{FCM} \leftrightharpoons A^{FCM}$. The values of current states of the physical system are transferred to D^{FCM} through the edges from the states to the concepts in D^{FCM}. To implement to the control of the physical system, first the weights of the edges from the concepts in D^{FCM} to the input variables in A^{FCM} are learned by Hebbian Learning algorithm according to the desired goal while all other weights in $D^{FCM} \leftrightharpoons A^{FCM}$ remain unchange. Then the values of the input variables of the physical system at time t are obtained via the values of the concepts representing the input variables at time t in the new fuzzy cognitive map $D^{FCM} \leftrightharpoons A^{FCM}$ in which the values of concepts in D^{FCM} are calculated by (11) and the values of concepts representing the input variables in A^{FCM} are calculated by (1). So for a desired goal, the control law is obtained by the weights connecting to the concepts representing the input variables which are timely learned by the Hebbian algorithm while other connection weights in $D^{FCM} \leftrightharpoons A^{FCM}$ remain unchanged.

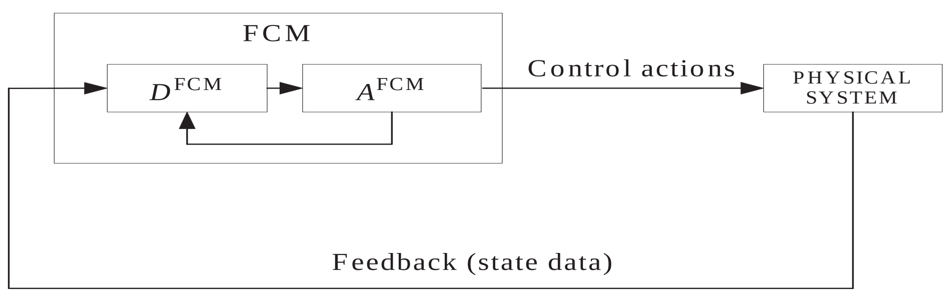


Fig.3 The connection of the learned TS-LS-FCM to the physical system

A generic description of the proposed Control algorithm via the learned TS-LS-FCM

Let $C_1, C_2, ..., C_n$ be the concepts in D^{FCM} and $C_{output_1}, C_{output_2}, ..., C_{output_k}$ be the concepts in $D^{FCM} \leftrightharpoons A^{FCM}$ representing input variables of the physical system. Let the desired goal be $A_{d_1} = \alpha_1, A_{d_2} = \alpha_2, ..., A_{d_{l_d}} = \alpha_{l_d}$, Where A_{d_u} is the value of concept C_{d_u}, $\{d_1, d_2, ..., d_{l_d}\} \subseteq \{1, 2, ..., n\}$ and $\alpha_1, \alpha_2, ..., \alpha_{l_d} \in [0, 1]$. For each concept C_{output_i}, $i = 1, 2, ..., k$, let $C_{i_1}, C_{i_2}, ..., C_{i_{l_i}}$ be all concepts in D^{FCM} connected to C_{output_i}, Where $\{i_1, i_2, ..., i_{l_i}\} \subseteq \{d_1, d_2, ..., d_{l_d}\}$. Let each $C_{input_i}, i = 1, 2, ..., q$ be the concept in D^{FCM} representing an observable state of the physical system, Where $\{input_1, input_2, ..., input_q\} \subseteq \{1, 2, ..., n\}$.

Table 1 TS-LS-FCM Control Algorithm via Hebbian Learning

Step 1: Arbitrarily set initial weights $w_{i_u output_i}$, $i = 1, 2, ..., k, u = i_1, i_2, ..., i_{l_i}$, which are the weights of the edges from the concepts in D^{FCM} to the input variables in A^{FCM}.

Step 2: $Temp\_error_{d_u} = \alpha_{d_u} - A^t_{d_u}$, $u = 1, 2, ..., l_d$.

step 3:

- Update the weights $w_{i_u output_i}$, $i = 1, 2, ..., k, u = i_1, i_2, ..., i_{l_i}$, which are the weights of the edges from the concepts in D^{FCM} to the input variables in A^{FCM}.
  - * For $i = 1, 2, ..., k$,
    $w^{t+1}_{i_u output_i} = w^t_{i_u output_i} + \triangle w_{i_u output_i}$,
    $\triangle w_{i_u output_i} = \eta * Temp\_Error_{i_u} * (1 - Temp\_Error_{i_u}) * A^t_{i_u}$,
    where η is the learning rate, for $u = i_1, i_2, ..., i_{l_i}$.
  - * For $i = 1, 2, ..., k$,
    $$A^{t+1}_{output_i} \longleftarrow f\left(\sum_{u=1}^{l_i} w^{t+1}_{i_u output_i} A^t_{i_u} + A^t_{output_i}\right);$$
  - * Implement controlling actions by updating the input variables of the physical system with the values corresponding to $A^{t+1}_{output_i}$, $i = 1, 2, ..., k$ at time $t + 1$.
  - * For $i \in \{input_1, input_2, ..., input_q, \}$, A^t_i is updated by the value of the concept C_i corresponding to the current value of the state represented by it in the physical system;
  - * for $i = 1, 2, ..., n$,
    $$A^{t+1}_i \longleftarrow f\left(\sum_{i=1, i\neq j}^{n} w_{ji} A^t_j + A^t_i\right).$$
- Until the termination condition is met, Goto step 2.

---

5. Experiments Study

Control of backing a trailer-truck to loading dock is a difficult problem since the system is non-linear and unstable. Conventional control algorithms require large amount of efforts in system analysis and design. Several researches have suggested using intelligent controllers to solve this problem. Those researches have mainly focused on implementing neural networks and the methods for network optimization [4]-[10]. The trailer-truck model is a vital element in optimizing the neural networks. Nquyen and Widrow [10] used neural emulator to emulate the dynamic of trailer-truck. However, their algorithm required thousands of backups to train the emulator that is not feasible in real applications. Neural networks is usually called a “black-box” for it is difficult to verify what the network has learned so that the users cannot make a decision, simultaneity, it is not suitable for expressing rule-based knowledge, such limit its application. In [27, 28] the authors present a partially fuzzy approach and its analysis of stability. It is true that this approach is very efficient, but what about the cases when we do not have the luxury of exact knowledge that is required for the non-fuzzy part of the controller (e.g. when the knowledge is too expensive or almost impossible to collect). In our research, the TS-LS-FCM, which models the dynamic of the trailer-truck system, is just learned by the historical data of the system. The truck can be controlled to any goal in the domain via the learned TS-LS-FCM. The following experiments show that TS-LS-FCM not only can accurately model a complex dynamic system based on the observed data of the system, but also can be applied to control the system.

5.1. Statement of Truck Backer-Upper Control Problem. The basic model of truck and loading zone are shown in Fig. 4. The truck corresponds to the cab part of the neural truck in the Nguyen-Widrow [10] neural truck backer-upper system. The truck position is exactly determined by the three state variables ϕ, x, and y, where ϕ is the angle of the truck with respect to the horizontal line. The coordinate pair (x,y) specifies the rear center position of the truck in the plane. The desired driving direction is achieved by turning the truck's wheels (first pair). Their declination from the truck's symmetrical axis (i.e. turning angle) is represented by angel θ. Only reverse driving is considered. The truck moves backward by a fixed unit distance every stage. For simplicity, we assume enough clearance between the truck and the loading dock such that y does not have to be considered as an input. If x and ϕ have arrived at the final state, y can reach any state needless changing θ. The task here is to design a control system, whose inputs are $\phi \in [-90°, 270°]$, $x \in [0, 20]$, and output is $\theta \in [-40°, 40°]$, such that the final states will be $(x_f, \phi_f) = (10, 90°)$. The controller should produce the appropriate steering angle θ at every stage to make the truck back up to the loading dock from any initial position (x_0,ϕ_0). In the existing literature [29] there can be found some the following approximate kinematic that are valid for the truck backer-upper procedure.

$$x(t+1) = x(t) + \cos[\phi(t) + \theta(t)] + \sin[\theta(t)]\sin[\phi(t)] \tag{14}$$

$$y(t+1) = y(t) + \sin[\phi(t) + \theta(t)] - \sin[\theta(t)]\cos[\phi(t)] \tag{15}$$

$$\phi(t+1) = \phi(t) - \sin^{-1}[\frac{2\sin(\theta(t))}{b}] \tag{16}$$

where b is the length of the truck. We assumed b =4 in the simulations of this paper.

Remark: In practice, the mathematical mechanism model like Eq.(14)-(16) is unnecessary for the construction of the TS-LS-FCM and the construction just requires some samples of the states x, y and ϕ corresponding to the turning angle θ (i.e., sample $(\theta_t, x_t, y_t, \phi_t)$ for the input θ_t and outputs x_t, y_t, ϕ_t of the truck system at time t). Here Eq.(14)-(16) just take the role of a physical truck to produce the necessary data for us to construct its TS-LS-FCM and to verify the capability of TS-LS-FCM to model and control a dynamic system. For the proposed methodology, the TS-LS-FCM can be well-learned by the data from the physical systems without any mathematical mechanism model. Observe, from Eq.(14)-(16), that even this simplified dynamic model of the truck is nonlinear.

5.2. Construction of $TS-LS-FCM$ Based on Data to Model the Truck Backer-Upper System. In this section, the LS-FCM learning algorithm described by (6)-(8) is applied to construct a LS-FCM shown as Fig. 5 in each local region and the fuzzy integration described by (11)-(13) is applied to integrate the local LS-FCM to be TS-LS-FCM named D^{FCM} to describe the dynamic characteristic of truck backer-upper system in the whole range.

5.2.1. The graph structure of the local LS-FCM. Each local LS-FCM is shown as Fig. 5 which is established according to our experiences for the relationships among the steering angle θ and the angle ϕ, the truck position x and y.

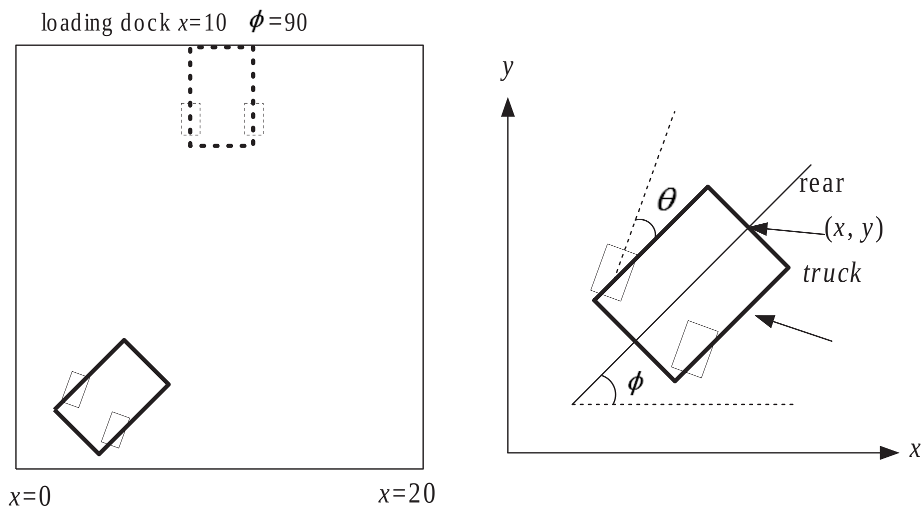


Fig. 4 Diagram of simulated truck and loading zone

By our experience, we know that $\Delta\phi$, the changing of ϕ (shown as Fig. 4) with the steering angle θ is independent of the current ϕ and the truck position. Thus there is just one edge w_{35} connected to the concept C_5 which represents $\Delta\phi$. Since both Δx and Δy are dependent on θ and ϕ, hence θ and ϕ are connected to them. Similarly, the connections between other concepts can be established as Fig. 5.

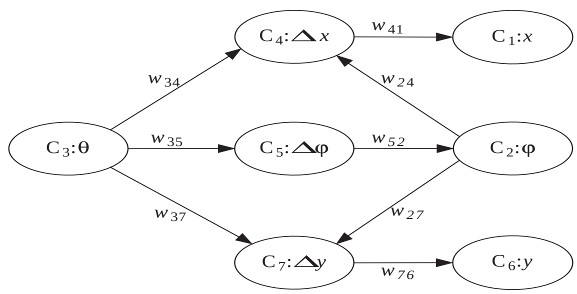


Fig. 5 The graph structure of the local LS-FCM for the truck backer-upper system

5.2.2. Domain partition and learning local LS-FCMs. In this truck backer-upper system, let the range of x, ϕ, θ be [0, 20], $[-90°, 270°]$, $[-40°, 40°]$ respectively. For the construction of each local LS-FCM, intervals [0, 20], $[-90°, 270°]$, $[-40°, 40°]$ are normalized into [0,1] by formula (2). Let $D = \{X_t \mid t = 1, 2, ..., T\}$ be the set of historical data of the truck system, Where $X_t = (x_{1t}, x_{2t}, ..., x_{7t})'$ and x_{jt} be the value of concept C_j corresponding to the value of the states x, y, ϕ or the input variable θ of the truck at time t. By Fig. 5., one can observe that as long as the values of concepts $C_3 : \theta$ and $C_2 : \phi$ are given all values of the concepts Δx, Δy and $\Delta\phi$ are determined. Therefore, let θ and ϕ be the premise variables for the fuzzy rules shown as (9). Under the condition that in every region the local LS-FCM learned by (6)-(8) according to the samples in D full into the region has the maximum errors for Δx and $\Delta\phi$ less than 0.1 and 0.2° respectively, the domain of

θ and ϕ, $[-40°, 40°] \times [-90°, 270°]$ is parted into 25 local regions with different sizes shown as Fig.6. The following Table 1 shows that the number of partition regions for different error limitations of Δx and $\Delta\phi$. Table 3 in the Appendix shows that all connection weights and the tuning coefficients of the local LS-FCMs learned by the learning algorithm described by (6)-(8) in the 25 regions shown as Fig.6. Where the samples in the set of historical data of the truck system D are evenly drawn from the domain.

Table 1 Number of partition regions for different error limitations

Error limitation of $\Delta x(m)$	0.05	0.1	0.2	0.4
Error of limitation $\Delta\phi(^o)$	0.1	0.2	0.8	2
Number of regions	50	25	16	9

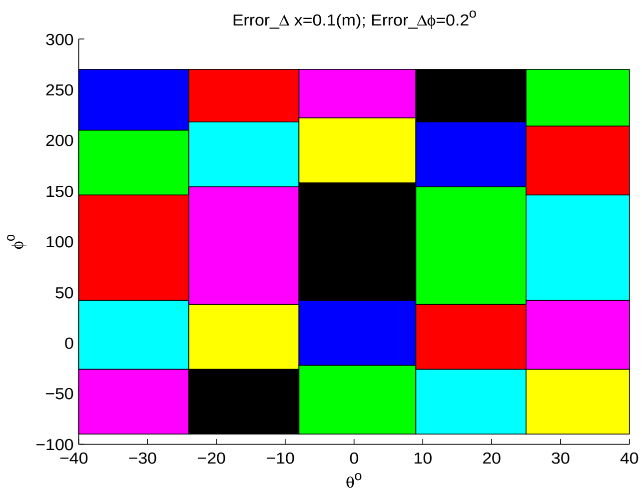


Fig. 6 The partition of the domain of θ and ϕ.

5.2.3. Fuzzy integration. Let x be a premise variable in the fuzzy IF-THEN rules shown as (9). The projection of the ith region on x is $[a_i, b_i]$ and the width of the projection is $W_i = b_i - a_i$. The triangle shape membership function $\widetilde{A_i}$ on x is shown as Fig.7, in which the width of the triangle is $2W_i$. Then we have the TS-LS-FCM with time varying connection weights and the tuning coefficients for the Truck Backer-Upper System by (11)-(13) in the domain $x \in [0, 20]$, $\phi \in [-90°, 270°]$, $\theta \in [-40°, 40°]$. Thus the TS-LS-FCM learned from the historical data is formed by the integration of the local LS-FCMs shown as Fig.5, the domain parted as Fig. 6 and the local LS-FCMs with the local connection weights and the tuning coefficients shown as Table 3 in the Appendix via the fuzzy sets on the premise variables with the membership functions shown as Fig. 7.

5.3. Modeling Dynamic Behavior of the Truck Backer-Upper Using the Learned TS-LS-FCM. In order to evaluate the capability of TS-LS-FCMs to model dynamic behavior of the nonlinear systems, we respectively input a sequence of steering angles θ_i, $i = 1, 2, ..., T$ to the about learned TS-LS-FCM and the kinematic equations Eq.(14,15,16) of the truck at any initial state (x_0, y_0, ϕ_0). The TS-LS-FCM and the kinematic equations produce very similar trajectories for any sequence of steering angles from any initial states in the domain. As an example, Fig. 8 shows the trajectories of TS-LS-FCM and the kinematic equations (labeled

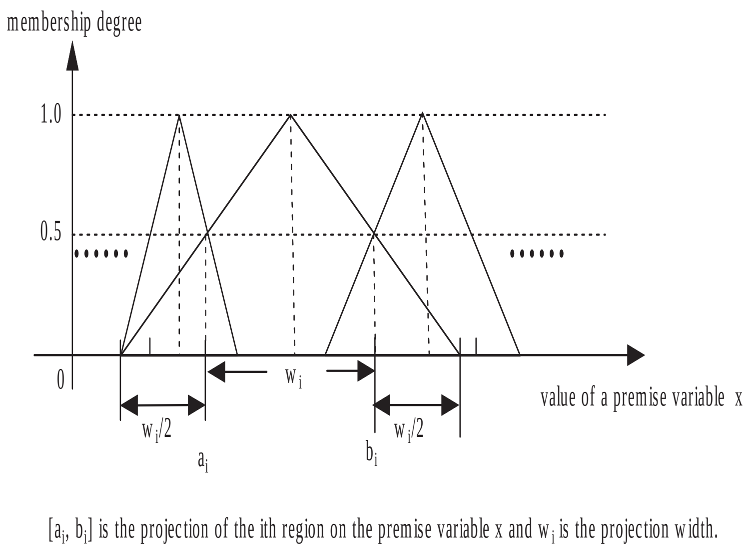


Fig. 7 The triangle membership function in each dimension of each region

"truck") for the sequence of steering angles θ_i: 20°, 20°, 20°, 20°, 20°, 20°, 20°, 20°, 20°, 20°, 20°, 20°, 20°, −20°, −20°, −20°, −20°, −20°, −20°, −20°, −20°, −20°, −20°, −20°, −20°, −20° starting form the initial state $(10, 0, 180°)$. Although the errors between them are gradually accumulated as the increase of the steps, Fig. 8 shows that the TS-LS-FCM can follows the truck very well. The following experiments will show that the TS-LS-FCM can accurately control the truck to any desired goal.

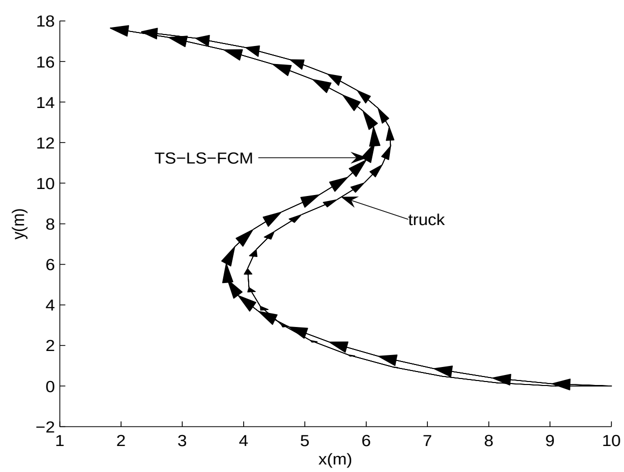


Fig. 8 The trajectories of TS-LS-FCM and the truck

5.4. Truck Back-Upper Online Controlling via the Learned TS-LS-FCM from the Historical Data. Let D^{FCM} be the TS-LS-FCM learned according to the historical data of the truck, Where the local LS-FCM is shown as Fig.5, the domain is parted as Fig. 6 and the local LS-FCMs with the local connection weights and the tuning coefficients shown as Table 3 in the Appendix are integrated into the TS-LS-FCM via the fuzzy sets on the premise variables with the membership functions shown as Fig. 7. Let A^{FCM} be the set of the observable states x, y, ϕ and input variable θ of the truck. D^{FCM} is connected to the truck shown as Fig. 9 by adding a edge from concept $C_3 : \theta$ to the input variable $\theta \in A^{FCM}$ to implement the control of the truck, and adding edges from the observable states $x, y, \phi \in A^{FCM}$ to the concepts $C_1 : x, C_6 : y, C_2 : \phi$ respectively to transfer the current states of the truck. The concepts $C_1 : x, C_6 : y, C_2 : \phi$ in D^{FCM} which determined the input variable $\theta \in A^{FCM}$ at each time are connected to the concept $C_3 : \theta \in D^{FCM}$ with the connection weights w_{13}, w_{23}, w_{63} respectively. Thus a new fuzzy cognitive map $D^{FCM} \leftrightharpoons A^{FCM}$ included D^{FCM} and A^{FCM} is formed (see Fig. 9). To implement to the control of the physical system, first w_{13}, w_{23}, w_{63} the connection weights from the concepts to the steering angle θ are learned by Hebbian Learning algorithm shown as Table 1 according to the desired goal. While the weights in D^{FCM} remain the form as (11)-(13). Then the values of the input variable θ of the truck at time t is obtained via the values of the concepts representing it in the new fuzzy cognitive map $D^{FCM} \leftrightharpoons A^{FCM}$ in which the values of concepts in D^{FCM} are calculated by (11) and the values of concepts representing θ is calculated by (1). Finally the current value of θ is applied to control the truck. The current states of the truck are observed and update the values of the corresponding concepts in D^{FCM}. The control law of next time can be obtained by repeat the above procedure. The experiment study shows that the truck can controlled to any desired goal in the domain by the TS-LS-FCM via the control algorithm described in Table 1.

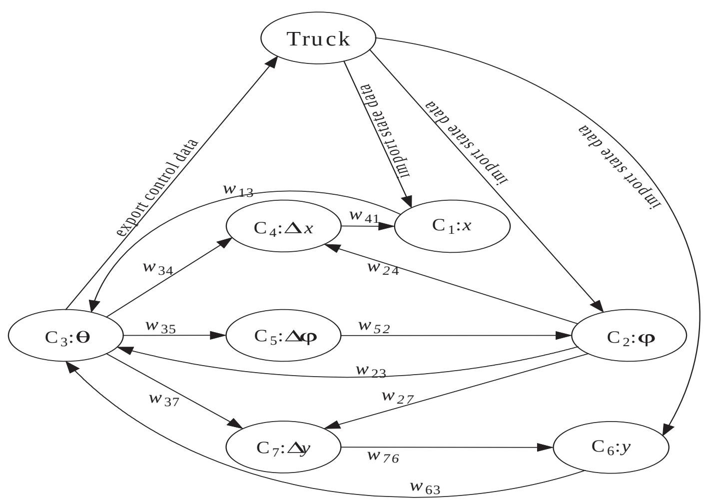


Fig. 9 The connection of the learned TS-LS-FCM to the truck.

As examples, Fig.10 shows the trajectories of the truck controlled starting from the following six randomly selected initial states:$(x_0, \phi_0) = (3, -30^o)$, $(8, 90^o)$, $(10, 220^o)$, $(12, -90^o)$,$(13, 10^o)$, $(16, 250^o)$ to the desired goal $1(10, 90^o)$, and from three randomly selected initial states: $(10, 220^o), (17, 30^o), (20, 200^o)$ to the desired goal 2 $(14, 90^o)$. The terminal states of the truck shown in Fig. 10 from the nine initial states are $[9.9153, 88.5747]$, $[9.8967, 88.2261]$, $[9.8924, 88.1445]$, $[10.0754, 91.5612]$, $[10.0939, 91.8968]$, $[10.096, 91.9351]$ for the desired goal 1 and $[13.9627, 89.0033]$,

$[14.0588, 91.9681]$, $[14.0506, 91.7341]$ for the desired goal 2 respectively. The errors between the terminal state of the truck and the desired goal can be less than any small positive number, if the procedure could be taken for infinity steps. It is very clear that the control of the truck by the TS-LS-FCM to any desired goal from any initial state don't require any sample data from the initial state to the goal which is required for the neural network control algorithm.

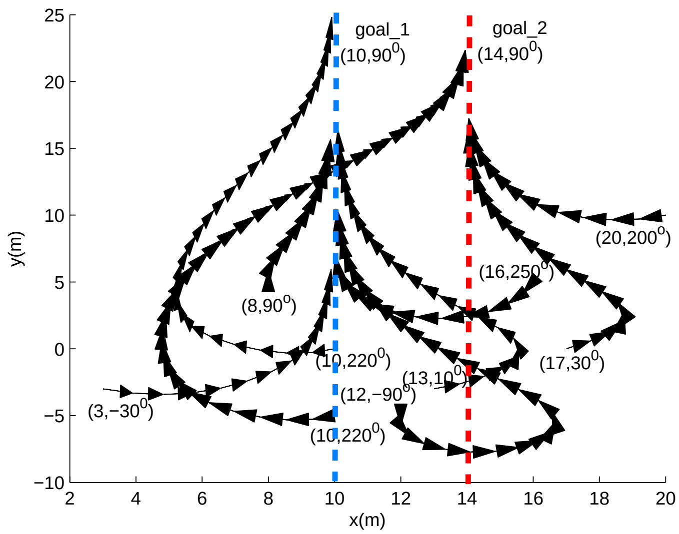


Fig. 10 Truck Backer-Upper trajectories controlled by the learned TS-LS-FCM

Although the errors between state values of TS-LS-FCM and the truck may by quit large for many steps due to gradually accumulation of the errors in each step, shown as Fig. 8, the dynamic errors in the control procedure for the control algorithm described as Table 1 are usually very small and the TS-LS-FCM can accurately control the truck to any desired goal. As an example, Table 2 shows the errors between the states of TS-LS-FCM and the states of the truck in the procedure to control the truck from initial state $(x_0, \phi_0) = (1, 0^o)$ to desired goal $(10, 90^o)$. Fig. 11 shows trajectories of the TS-LS-FCM and the truck.

Table 2 Dynamic errors between the TS-LS-FCM and the truck in the controlling procedure from initial state $(x_0, \phi_0) = (1, 0^o)$ to desired goal $(10, 90^o)$

t	1	2	3	4	5	6	7	8	9
x^{TRUCK}(m)	1.95	2.90	3.80	4.66	5.44	6.14	6.75	7.28	7.74
φ^{TRUCK}(^{o})	8.68	17.30	26.19	35.09	43.60	51.17	57.30	62.37	65.88
x^{FCM}-x^{TRUCK}(m)	0.04	0.02	0.02	0.04	0.05	0.06	0.05	0.03	0.03
φ^{FCM}-φ^{TRUCK}(^{o})	-0.03	-0.02	-0.04	-0.04	-0.03	-0.01	-0.02	-0.04	-0.06
$\theta^{streeing\ angle}$(^{O})	-17.57	-17.43	-18.01	-18.03	-17.21	-15.28	-12.34	-10.18	-7.03

10	11	12	13	14	15	16	17	18	19
8.15	8.50	8.83	9.10	9.33	9.53	9.67	9.77	9.84	9.89
69.17	71.00	73.78	76.83	78.55	81.76	84.31	86.10	87.34	88.19
0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
-0.06	-0.04	-0.06	-0.06	-0.04	-0.06	-0.05	-0.04	-0.03	-0.02
-6.59	-3.68	-5.56	-6.11	-3.43	-6.41	-5.11	-3.60	-2.47	-1.69

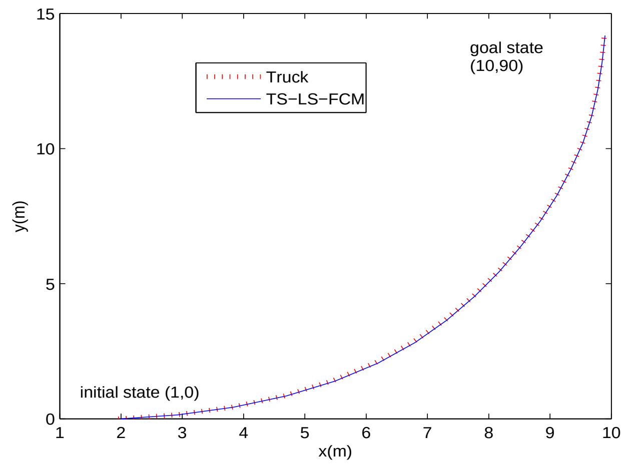


Fig. 11 The trajectories of TS-LS-FCM and the truck.

6. Conclusions

For large and complex dynamic systems that are common in the process industry, it is extremely difficult to describe the entire system by a precise mathematical model. Thus, it is more attractive and useful to represent it, in a graphical abstract way showing the causal relationships between states-concepts. This study shows clearly that the complex dynamic nonlinear systems can be modeled and controlled very well just by the applications of the TS-LS-FCM learned from the historical data observed from the system.

References

[1] J. Chang, J. H. Lin and T. D. Chiueh, Neural Network for Truck Backer-upper Control System, Industrial Automation and Control, International IEEE/IAS Conference on, May 1995: 328-334.
[2] J. A. Dickerson, B. Kosko, Fuzzy virtual worlds, Artif. Intel. Expert, 1994, 7: 25-31.
[3] R. E. Jenkins, B. P. Yuhas, A Simplified Neural Network Solution Through Problem Decomposition: The Case of the Truck Backer-Upper, IEEE Transactions on Neural Network, July 1993, 4(4): 718-720.
[4] H. lchihashi, M. Tokunaga, Neuro-Fuzzy Optimal Control of Backing up a Trailer Truck, Proc. of IEEE International Conference on Neural Network, 1993, 1: 306-311.
[5] H. Kinjo, B. Wang and T. Yamamoto, Backward Movement Control of Trailer-Truck System Using Neuro-controller Evolved by Genetic Algorithm, Industrial Electronics Society, 2000, IECON 2000, 26th Annual Conference of the IEEE, Vol. 1, 2000.
[6] B. Kosko, Fuzzy cognitive maps, Internat. J. Man-Mach. Studies, 1986, 24: 65-75.
[7] B. Kosko, Fuzzy Engineering, Englewood Cliffs, NJ: Prentice Hall, 1999.
[8] B. Kosko, Differential Hebbian Learning, Proceedings American Institute of Physics, Neural Networks for Computing, April, 1986: 277-282.
[9] D. E. Koulouriotis, I. E. Diakoulakis, D. M. Emiris, E. N. Antonidakis, I. A. Kaliakatsos, Efficiently modelling and controlling complex dynamic systems using evolutionary fuzzy cognitive maps (Invited Paper), Internat. J. Comput. Cognition 1, 2003, 2: 41-65.
[10] D. Nguyen and B. Widrow, The truck backer-upper: An example of self-learning in neural network, IEEE Contr. Syst. Mag., 1990, 10(3): 18-23.
[11] K. E. Parsopoulos, E. I. Papageorgiou, P. P. Groumpos, M. N. Vrahatis, A first study of fuzzy cognitive maps learning using particle swarm optimization, Proc. IEEE 2003 Congr. on Evolutionary Computation, 2003: 1440-1447.

[12] E. I. Papageorgiou, C. D. Stylios, P. P. Groumposa, Unsupervised learning techniques for fine-tuning fuzzy cognitive map causal links, International Journal of Human-Computer Studies 2006, 64: 727-743.

[13] E. I. Papageorgiou, C. D. Stylios, P. P. Groumpos, Active Hebbian learning algorithm to train fuzzy cognitive maps, International Journal of Approximate Reasoning, 2004, 37:219-249.

[14] S. Renals, R. Rohwer, A study of net work dynamics, J. Statist. Phys., 1990, 58: 825-848.

[15] C. D. Stylios, P. P. Groumpos, Fuzzy Cognitive Maps: a model for intelligent supervisory control systems, Computers in Industry, 1999, 39: 229-238

[16] C. D. Stylios and P. P. Groumpos, Modeling complex systems using fuzzy cognitive maps, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2004, 34(1): 155-162.

[17] A. K. Tsadiras, K. Margaritis, An experimental study of the dynamics of the certainty neuron fuzzy cognitive maps, Neurocomputing, 1999, 24: 95-116.

[18] Massimo Bertolini, Assessment of human reliability factors: A fuzzy cognitive maps approach, International Journal of Industrial Ergonomics, 2007,37: 405–413.

[19] E.I. Papageorgiou, P.P. Spyridonos, D. Th. Glotsos, C.D. Stylios, P. Ravazoula d, G.N. Nikiforidis, P.P. Groumpos, Brain tumor characterization using the soft computing technique of fuzzy cognitive maps, Applied Soft Computing, 2008, 8: 820–828.

[20] Luis Rodriguez-Repiso, Rossitza Setchi, Jose L. Salmeron, Modelling IT projects success with Fuzzy Cognitive Maps, Expert Systems with Applications, 2007, 32: 543–559.

[21] Moon-Chan Kim, Chang Ouk Kim, Seong Rok Hong, Ick-Hyun Kwon, Forward–backward analysis of RFID-enabled supply chain using fuzzy cognitive map and genetic algorithm, Expert Systems with Applications, 2007.

[22] T. Kottas, Y. Boutalis, and M. Christodoulou, A new method for weight updating in Fuzzy cognitive maps using system feedback, in Proc. 2nd ICINCO, Barcelona, Spain, Sep. 13–17, 2005: 202–209.

[23] E.I. Papageorgiou, Peter P. Groumpos, A new hybrid method using evolutionary algorithms to train Fuzzy Cognitive Maps, Applied Soft Computing, 2005,5: 409–431.

[24] D.E. Koulouriotis, I.E. Diakoulakis, D.M. Emiris, Learning fuzzy cognitive maps using evolution strategies: a novel schema for modeling and simulating high-level behavior, in: IEEE Congr. On Evolutionary Computation (CEC2001), 2001: 364–371.

[25] Wojciech Stach, Lukasz Kurgan, Witold Pedrycz, Marek reformat genetic learning of fuzzy cognitive maps, Fuzzy Sets Syst. 2005, 153: 371–401.

[26] M. Khan, A. Chong, Fuzzy cognitive map analysis with genetic algorithm, in: Proc. 1st Indian Internat. Conf. on Artificial Intelligence (IICAI-03), 2003.

[27] K. Tanaka, T. Taniguchi, H.O. Wang, Fuzzy controller and observer design for backing control of a trailer-truck, Eng. Appl. Artif. Intell. 1997, 10(5): 441–452.

[28] K. Tanaka, M. Sano, A robust stabilization problem of fuzzy control system and its application to backing up control of a truck-trailer, IEEE Transactions on Fuzzy Syst. 1994, 2(2): 119–134.

[29] Lixin Wang, M.Mendel, Generating fuzzy rules by learning from examples, IEEE Transactions on Systems, Man, and Cybernetics, 1992, 22(6): 1414–1427.

[30] Peter P. Groumpos, Chrysostomos D. Stylios, Modelling supervisory control systems using fuzzy cognitive maps, Chaos, Solitons and Fractals, 2000, 11: 329–336.

[31] Baoyan Zhu, Qingling Zhang and Xuefeng Zhang, Decentralized Robust Guaranteed cost control for uncertain T-S fuzzy interconnected systems with time delay, International Journal of Information and Systems Sciences,2005, 1(1):73-88.

[32] Mustapha Jazar and Steve Younan, Numerical Study of an Alternative Method of Fictitious Domain Technique Applied to Outside Domain Wave,International Journal of Information and Systems Sciences,2005, 1(2):150-161.

[33] Q. Liu, M. Fu and Z. Deng, Adaptive inverse control for nonlinear system,International Journal of Information and Systems Sciences,2005, 1(2):253-263.

[34] Y. Ma, Q. Zhang and X. Zhang, Decentralized output feedback robust control for a class of uncertain nonlinear circle-linked large-scale composite systems,Journal of Information and Systems Sciences,2006, 2(1):20-30.

[35] X. Liu, Y. Zhang and Y. Ren, Dynamics systems description and control based on date, Journal of Information and Systems Sciences, 2008, 4(3):335-347

Appendix:

Table 3 Weights of the local LS-FCMs shown as Fig.5 in all regions shown as Fig.6

regions	w_{24}	w_{34}	w_{35}	w_{27}	w_{37}	w_0^4	w_0^5	w_0^7
[-90,-26]$\times$[$-40, -24$]	0.42	0.034	-0.18	0.26	-0.055	-0.59	-0.18	-0.60
[-26,42]$\times$[$-40, -24$]	-0.05	0.064	-0.18	0.35	0.01	-0.48	-0.18	-0.63
[42,146]$\times$[$-40, -24$]	-0.29	-0.00	-0.18	-0.02	0.06	-0.38	-0.18	-0.49
[146,214]$\times$[$-40, -24$]	-0.00	-0.06	-0.18	-0.36	0	-0.56	-0.18	-0.27
[214,270]$\times$[$-40, -24$]	0.44	-0.03	-0.18	-0.25	-0.057	-0.89	-0.18	-0.34
[-90,-26]$\times$[$-24, -8$]	0.47	0.01	-0.14	0.30	-0.02	-0.60	-0.19	-0.61
[-26,38]$\times$[$-24, -8$]	-0.04	0.02	-0.14	0.40	0.00	-0.48	-0.19	-0.64
[38,154]$\times$[$-24, -8$]	-0.31	-0.00	-0.14	-0.03	0.02	-0.36	-0.19	-0.47
[154,218]$\times$[$-24, -8$]	0.042	-0.02	-0.14	-0.42	-0.00	-0.59	-0.19	-0.22
[218,270]$\times$[$-24, -8$]	0.51	-0.01	-0.14	-0.28	-0.02	-0.96	-0.19	-0.33
[-90,-22]$\times$[$-8, 9$]	0.47	-0.00	-0.13	0.32	0.00	-0.59	-0.20	-0.63
[-22,42]$\times$[$-8, 9$]	-0.07	-0.00	-0.13	0.41	-0.00	-0.45	-0.20	-0.64
[42,158]$\times$[$-8, 9$]	-0.33	0.00	-0.13	-0.06	-0.00	-0.36	-0.20	-0.45
[158,222]$\times$[$-8, 9$]	0.08	0.00	-0.13	-0.44	0.00	-0.63	-0.20	-0.21
[222,270]$\times$[$-8, 9$]	0.55	0.00	-0.13	-0.27	0.00	-1	-0.20	-0.35
[-90,-26]$\times$[9, 25]	0.47	-0.01	-0.14	0.30	0.02	-0.58	-0.19	-0.64
[-26,38]$\times$[9, 25]	-0.04	-0.03	-0.14	0.40	-0.00	-0.45	-0.19	-0.64
[38,154]$\times$[9, 25]	-0.32	0.00	-0.14	-0.03	-0.02	-0.37	-0.19	-0.45
[154,218]$\times$[9, 25]	0.04	0.03	-0.14	-0.42	0.00	-0.62	-0.19	-0.23
[218,270]$\times$[9, 25]	0.51	0.01	-0.14	-0.27	0.02	-0.97	-0.19	-0.36
[-90,-26]$\times$[25, 40]	0.41	-0.04	-0.18	0.26	0.06	-0.55	-0.16	-0.65
[-26,42]$\times$[25, 40]	-0.05	-0.07	-0.18	0.35	-0.01	-0.41	-0.16	-0.62
[42,146]$\times$[25, 40]	-0.29	0.00	-0.18	-0.02	-0.06	-0.38	-0.16	-0.43
[146,214]$\times$[25, 40]	-0.00	0.07	-0.18	-0.36	0	-0.61	-0.16	-0.27
[214,270]$\times$[25, 40]	0.43	0.03	-0.18	-0.25	0.06	-0.93	-0.16	-0.40

Research Center of Information and Control, Dalian University of Technology, Dalian 116024, P.R. China
E-mail: xdliuros@hotmail.com

Software College, Shenyang Normal University, Shenyang 110034, P. R. China
E-mail: zhangyl911@live.cn