Article ID: 1000 - 8152(2001)06 - 0851 - 06

Linear Multi-step Fuzzy Logical Systems *

SUN Duoqing and HUO Wei

(The Seventh Research Division, Beijing University of Aeronautics and Astronautics · Beijing, 100083, P.R.China)

Abstract: A linear multi-step fuzzy logical system for identifying unknown dynamic systems described by ordinary differential equations is presented. The fuzzy system is constructed according to the linear multi-step approximation method for solving ordinary differential equations. The new fuzzy logic system can predict state variables of the unknown system and approximate unknown functions in the system differential equation. The prediction and approximation precisions of the fuzzy system are analyzed theoretically. The capabilities of the proposed fuzzy system is verified by the simulation result.

Key words: linear multi-step method; fuzzy logic system; accuracy analysis

Document code: A

线性多步法模糊逻辑系统

孙多青　霍　伟

(北京航空航天大学第七研究室·北京,100083)

摘要: 根据求解常微分方程的线性多步逼近方法构造了线性多步法模糊逻辑系统以辨识由常微分方程所描述的未知动态系统.这种新的模糊逻辑系统在预测状态的同时也能够逼近系统微分方程中的未知函数.文中从理论上对其精度作了分析,并用仿真验证了所得的结果.

关键词: 线性多步法; 模糊逻辑系统; 精度分析

1 Introduction

In many practical applications it is required to identify the unknown nonlinear dynamic system described by ordinary differential equation (ODE) as follows

$$\dot{x} = f(x(t)),\ x(0) = x_0, \tag{1}$$

where $f(x(t))$ is an unknown function. Suppose that[1]

1) State trajectory $x(t) \in D \subset \mathbb{R}^n, t \in [0, T]$, and D is a bounded closed domain.

2) $f(x)$ is a continuous function on D and satisfies the Lipschitz condition; i.e., there exists a positive constant c such that $\| f(x) - f(y) \| \leqslant c \| x - y \|$. $\forall x, y \in D$.

3) State vector $x(t)$ is measurable.

Under the above assumptions, a Runge-Kutta neural network (RKNN) to identify the unknown dynamic system (1) has been constructed in [1] based on the Runge-Kutta method for solving initial-value problems of ODEs. However, the Runge-Kutta method is one of the one-step approximate methods for solving ODEs; in order to improve the accuracy of approximate solutions, the number of times to evaluate function $f(\cdot)$ must be increased in each step. Thus the computational burden is large. Another method to improve accuracy of approximate solutions is that the calculation of state x_{n+1} (Its definition will be given in Section 2) not only depends on x_n, but is also related to the computed $x_{n-1}, x_{n-2}, \cdots$ directly, and function $f(\cdot)$ is only calculated once in each step. This kind of methods is called multi-step method. Denote $f_n = f(x_n)$, if the solving formula of a multi-step method consists of a linear combination of $f_n, f_{n-1}, \cdots$, this multi-step method is called linear multi-step method. In addition, it is well known that fuzzy identifiers have advantage over neural network identifiers in many respects. So the purpose of this paper is to construct an adaptive fuzzy logic system (FLS)—linear multi-step method fuzzy logic system (LMFLS). This fuzzy system can not only predict the future state values

* Foundation item: supported by Doctor Degree Programs Foundation of China (1999000602).
Received date: 2000 - 07 - 05; Revised date: 2001 - 04 - 09.

by training its parameters based on the previously measured states, but also approximate the unknown function $f(x)$ describing the dynamic system.

2 Introduction to linear multi-step method

The main content about the linear multi-step method is summarized as follows[3,4].

Introduce a sequence of points $t_n = t_{n-1} + h = t_0 + nh, n = 0,1,\cdots$; where h is the step size in the sequence. Define $x(t_n)$ to be the exact solution $x(t)$ of (1) at point t_n and x_n to be an approximation of $x(t_n)$, a general form of the linear k-step method is

$$\sum_{j=0}^{k} \alpha_j x_{n+j} = \sum_{j=0}^{k} \beta_j f_{n+j},\ n = 0,1,\cdots, \tag{2}$$

where α_j and β_j are real constants, $\alpha_k \neq 0$ (usually set $\alpha_k = 1$), $|\alpha_0| + |\beta_0| > 0$. In order to compute x_{n+k} from (2), k starting values must be known. These values can be obtained by (1) and other means, e.g., with the aid of a one-step method. After the starting values are obtained, $x_k, x_{k+1}, \cdots$ can be calculated in order. A linear k-step method is called an implicit method if $\beta_k \neq 0$; and it is called an explicit method if $\beta_k = 0$. The above method becomes a one-step method when $k = 1$ and it is a multi-step method if $k \geqslant 2$. Denote

$$\begin{cases} c_0 = \alpha_0 + \alpha_1 + \cdots + \alpha_k, \\ c_1 = \alpha_1 + 2\alpha_2 + \cdots + k\alpha_k - (\beta_0 + \beta_1 + \cdots + \beta_k), \\ \quad\vdots \\ c_q = \dfrac{1}{q!}(\alpha_1 + 2^q\alpha_2 + \cdots + k^q\alpha_k) - \\ \qquad \dfrac{1}{(q-1)!}(\beta_1 + 2^{q-1}\beta_2 + \cdots + k^{q-1}\beta_k), \\ \qquad q = 2,3,\cdots. \end{cases} \tag{3}$$

Definition 1 A linear multi-step method is called a method of order p if $c_0 = c_1 = \cdots = c_p = 0$ and $c_{p+1} \neq 0$.

Definition 2 The local truncation error of a linear multi-step method at x_{n+k} is defined as

$$T_{n+k} = \sum_{j=0}^{k} \alpha_j x_{n+j} - \sum_{j=0}^{k} \beta_j f_{n+j}.$$

The first term of the Taylor expansion of T_{n+k} in powers of h is called main local truncation error or main term of a local truncation error.

It can be proven that the main term of a local truncation error is $c_{p+1} h^{p+1} x^{(p+1)}(t_n)$, and c_{p+1} is called error constant.

In order that the result obtained by approximate formula (2) is a reasonable approximation of equation (1), (2) should satisfy the following consistence condition.

Definition 3 The linear multi-step method is said to be consistent (with respect to OED (1)), if for each f satisfy assumption 2) its local truncation error T_{n+k} satisfies

$$\lim_{h\to 0} \frac{1}{h} T_{n+k} = 0,\ t = t_0 + nh.$$

It can be shown that a consistent linear multi-step method has at least order 1. Conversely, it is obvious that a linear multi-step method of order q ($q \geqslant 1$) is consistent.

Let $e_n = x_n - x(t_n)$ denote the global truncation error of a linear multi-step method. For a given initial value x_0 and additional initial values $x_1, x_2, \cdots, x_{k-1}$, we expect

$$\| e_i \| \leqslant \eta(h),\ i = 0,1,\cdots,k-1, \lim_{h\to 0} \eta(h) = 0.$$

Definition 4 The linear multi-step method is said to be convergent if

$$\lim_{h\to 0} x_n = x(t),\ t = t_0 + nh,$$

for all $0 \leqslant t \leqslant T$, all functions f satisfy assumption 2), and all initial values x_i ($i = 0,1,\cdots,k-1$) satisfy the above condition.

Theorem 1 A convergent linear multi-step method is also consistent.

Definition 5 For the linear multi-step method (2), a polynomial $\rho(\lambda) = \sum_{j=0}^{k} \alpha_j \lambda^j$ is introduced. If all the roots of $\rho(\lambda)$ are on unit circumference or in unit disk, and the roots on unit circumference are all simple roots, then it is said that method (2) satisfies the root condition.

Theorem 2 Suppose that a linear multi-step method (2) is consistent, then the method is convergent if and only if the root condition holds.

Definition 6 The linear multi-step method (2) is said to be stable for initial-value problems if for all f satisfying assumption 2) there exist constants C and h_0 such that when $0 < h \leqslant h_0$, any two solutions $\bar{x}_n$ and $\tilde{x}_n$ of (2) satisfy

$$\max_{0\leqslant t_n\leqslant T} \| \bar{x}_n - \tilde{x}_n \| \leqslant C \max_{0\leqslant j\leqslant k-1} \| \bar{x}_j - \tilde{x}_j \|,$$

where $\bar{x}_j$ and $\tilde{x}_j$ ($j = 0,1,\cdots,k-1$) are initial values of

$\bar{x}_n$ and $\tilde{x}_n$, respectively.

Theorem 3 A linear multi-step method (2) is stable for initial-value problems if and only if the root condition holds.

In summary, for a linear multi-step method of order $q(q \geq 1)$ the root condition is a necessary and sufficient condition that it is convergent and stable.

3 Linear multi-step fuzzy logic systems

With the above preliminary a linear multi-step fuzzy logic system can be constructed, and its structure expression is

$$\begin{aligned}
&y(i+k) = \\
&-\alpha_0 y(i) - \alpha_1 y(i+1) - \\
&\cdots - \alpha_{k-1} y(i+k-1) + \\
&h[\beta_0 F_f(y(i)) + \beta_1 F_f(y(i+1)) + \cdots + \\
&\beta_{k-1} F_f(y(i+k-1))],
\end{aligned} \tag{4}$$

where $k \geq 2, y(i) \equiv y(ih)$; α_i and $\beta_i (i = 1, 2, \cdots, k-1)$ satisfy the conditions for consistence, convergence and stability; fuzzy logical system F_f is an approximation of f. The structure of LMFLS can also be illustrated by Fig.1:

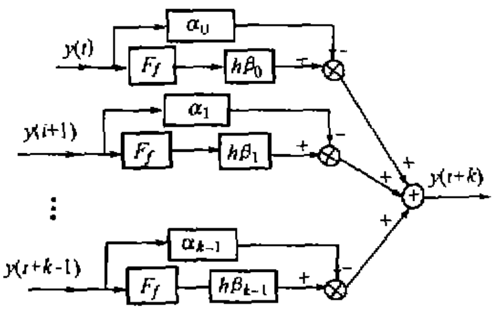


Fig. 1 The structure of a linear multi-step fuzzy logical system

Theorem 4 (Approximation and Convergence Theorem) For an approximate solution x_{i+k} of ODE (1) obtained by a linear multi-step method

$$\begin{aligned}
&x_{i+k} = \\
&-\alpha_0 x_i - \alpha_1 x_{i+1} - \cdots - \alpha_{k-1} x_{i+k-1} + \\
&h[\beta_0 f(x_i) + \beta_1 f(x_{i+1}) + \cdots + \beta_{k-1} f(x_{i+k-1})],
\end{aligned} \tag{5}$$

there exists an LMFLS described by (4) such that its outputs $y(i+k)$ satisfy

$$\lim_{h\to 0} \| y(i+k) - x_{i+k} \| = 0, \ i = 0, 1, 2, \cdots.$$

If the linear multi-step method (5) satisfies the root condition and its order ≥ 1, then the LMFLS is convergent.

Proof Among the k starting values of (4), except x_0 obtained by (1), the others are obtained by a one-step method FLS (e.g., $y(i+1) = y(i) + hF_f(y(i))$). Since the proof that starting values $y(i+1)$ can approximate $x_{i+1} (i = 0, 1, \cdots, k-2)$ is similar to the proof that $y(i+k)$ obtained by LMFLS can approximate $x_{i+k} (i = 0, 1, \cdots)$, the starting values of (5) can be regarded as those of (4).

For all $\varepsilon > 0$, since f is continuous on the bounded closed region D, from [5], for $\varepsilon_0 \leq \dfrac{\sqrt{\varepsilon}}{k \sum_{j=0}^{k-1} |\beta_j|}$, there exists a FLS $\hat{f}$ such that

$$\| \hat{f}(x(t)) - f(x(t)) \| < \varepsilon_0, \ x(t) \in D.$$

Set $F_f = \hat{f}$, and construct an LMFLS described by (4). For $i = 0$, take $\delta = \sqrt{\varepsilon}$, when $h < \delta$, one has

$$\begin{aligned}
&\| y(k) - x_k \| = \\
&\| h [\beta_0 (F_f(x_0) - f(x_0)) + \beta_1 (F_f(x_1) - f(x_1)) + \\
&\cdots + \beta_{k-1} (F_f(x_{k-1}) - f(x_{k-1}))] \| \leq \\
&h[|\beta_0| \cdot \| F_f(x_0) - f(x_0) \| + |\beta_1| \cdot \| F_f(x_1) - f(x_1) \| + \\
&\cdots + |\beta_{k-1}| \cdot \| F_f(x_{k-1}) - f(x_{k-1}) \|] < \\
&hk\varepsilon_0 \sum_{j=0}^{k-1} |\beta_j| \leq \varepsilon.
\end{aligned}$$

The conclusion holds for $i = 0$.

Suppose that the conclusion holds for $i \leq n-1$, i.e., for $\varepsilon_1 \leq \dfrac{\varepsilon}{k \sum_{j=0}^{k-1} |\alpha_j|}$, when h is sufficient small, $\| y(i+k) - x_{i+k} \| < \varepsilon_1, i = 0, 1, \cdots, n-1$. Then, for $i = n$, one has

$$\begin{aligned}
&\| y(n+k) - x_{n+k} \| = \\
&\| \alpha_0 (y(n) - x_n) + \alpha_1 (y(n+1) - x_{n+1}) + \cdots + \\
&\alpha_{k-1} (y(n+k-1) - x_{n+k-1}) + \\
&h[\beta_0 (F_f(y(n)) - f(x_n) + \\
&\beta_1 (F_f(y(n+1)) - f(x_{n+1})) + \cdots + \\
&\beta_{k-1} (F_f(y(n+k-1)) - f(x_{n+k-1})] \| < \\
&k\varepsilon_1 \sum_{j=0}^{k-1} |\alpha_j| \leq \varepsilon (h \to 0).
\end{aligned}$$

This implies that the conclusion holds for $i = n$. The first conclusion is proved through induction.

If (5) satisfies the root condition and its order $p \geq 1$, then $\lim_{h\to 0} x_{i+k} = x(t), t = t_{i+k}$. Also,

$$\| y(i+k) - x(t) \| =$$
$$\| y(i+k) - x_{i+k} + x_{i+k} - x(t) \| \leqslant$$
$$\| y(i+k) - x_{i+k} \| + \| x_{i+k} - x(t) \| .$$

Therefore, $\lim_{h \to 0} \| y(i+k) - x(t) \| = 0, t = t_{i+k}$, i.e., $y(i+k)$ is convergent. This ends the proof of Theorem 4.

This theorem shows that when a system model is unknown, as long as h is sufficiently small, the iterative sequence produced by an LMFLS will tend to the approximate solution obtained by a linear multi-step method for the accurate model, and the LMFLS constructed based on a convergent linear multi-step method is also convergent.

In the following, we quantitatively analyze the approximate accuracy that the FLS approximates unknown function $f(x)$ in (1) while an LMFLS predicts state variables of (1). Here only LMFLS's of order 2 are considered.

Theorem 5 Suppose that $f(x)$ and $F_f(x)$ defined on D have continuous partial derivatives of order 3. The state trajectory of system (1) is predicted by an LMFLS of order 2:

$$y(i+k) =$$
$$-\alpha_0 x(i) - \alpha_1 x(i+1) - \cdots - \alpha_{k-1} x(i+k-1) +$$
$$h[\beta_0 F_f(x(i)) + \beta_1 F_f(x(i+1)) + \cdots +$$
$$\beta_{k-1} F_f(x(i+k-1))].$$

If $\| y(i+k) - x(i+k) \| \approx O(h^3)$, then the accuracy of F_f approximating f is

$$\| f(x(i) + rhf(x(i))) - F_f(x(i) + rhf(x(i))) \| \approx O(h^2),$$

where

$$r = \frac{\beta_1 + 2\beta_2 + \cdots + (k-1)\beta_{k-1}}{\beta_0 + \beta_1 + \cdots + \beta_{k-1}},$$
$$\beta_0 + \beta_1 + \cdots + \beta_{k-1} \neq 0, \ k \geqslant 2.$$

Proof Denote the Jacobi matrices of f and F_f by J_f and J_F, respectively, the state of System (1) at time $(i+k)h$ can be expressed as

$$x(i+k) =$$
$$x(i) + \dot{x}\,|_{x(i)} \cdot kh + \frac{1}{2}\ddot{x}\,|_{x(i)}(kh)^2 + O(h^3) =$$
$$x(i) + khf(x(i)) + \frac{(kh)^2}{2} J_f\,|_{x(i)} f(x(i)) + O(h^3).$$

The output of the LMFLS at time $(i+k)h$ can be written in the following form

$$y(i+k) =$$
$$-\alpha_0 x(i) - \alpha_1[x(i) + hf(x(i)) +$$
$$\frac{h^2}{2} J_f\,|_{x(i)} f(x(i)) + O(h^3)] - \cdots -$$
$$\alpha_{k-1}[x(i) + (k-1)hf(x(i)) +$$
$$\frac{(k-1)^2 h^2}{2} J_f\,|_{x(i)} f(x(i)) + O(h^3)] +$$
$$h\{\beta_0 F_f(x(i)) + \beta_1[F_f(x(i)) + hJ_F\,|_{x(i)} f(x(i))] +$$
$$\cdots + \beta_{k-1}[F_f(x(i)) + (k-1)hJ_F\,|_{x(i)} f(x(i))]\} + O(h^3).$$

Denote

$$s_1 = \alpha_1 + 2\alpha_2 + \cdots + k\alpha_k,$$
$$s_2 = \frac{1}{2}(\alpha_1 + 2^2\alpha_2 + \cdots + k^2\alpha_k),$$

where $\alpha_k = 1$. By the definition of order,

$$\beta_0 + \beta_1 + \cdots + \beta_{k-1} = s_1,$$
$$\beta_1 + 2\beta_2 + \cdots + k\beta_{k-1} = s_2.$$

Thus

$$\| x(i+k) - y(i+k) \| =$$
$$hs_1 \| f(x(i)) - F_f(x(i)) + rh\, J_F\,|_{x(i)} f(x(i)) -$$
$$rhJ_F\,|_{x(i)} f(x(i)) \| + O(h^3).$$

Hence

$$\| f(x(i) + rhf(x(i))) - F_f(x(i) + rhf(x(i))) \| \approx O(h^2).$$

This completes proof of the theorem.

The above theorem implies that if $\| \frac{\partial f}{\partial x} \|$ and $\| \frac{\partial F_f}{\partial x} \|$ do not vary rapidly and h is sufficiently small, while an LMFLS predicts state variables, the FLS in the LMFLS can also approximate the unknown function on the rightside of (1), but its approximate accuracy is lower than that of the LMFLS predicting state variables.

In Theorem 5, the condition $\beta_0 + \beta_1 + \cdots + \beta_{k-1} \neq 0$, $k \geqslant 2$ is not strict, since there exist many linear multi-step methods which satisfy this condition and are consistent, convergent and stable. For example, the linear two-step method of order 2 given by the famous Adams formula

$$x_{n+2} = x_{n+1} + \frac{h}{2}(3f_{n+1} - f_n)$$

does satisfy $\beta_0 + \beta_1 \neq 0$. In fact, it can be proved that providing

$$\alpha_0 > -1, \ \alpha_1 < 3, \ -3 < \alpha_2 < 1,$$
$$\beta_0 + \beta_1 < 0 \text{ or } 0 < \beta_0 + \beta_1 < 5,$$

linear two-step methods of order 2 satisfy $\beta_0 + \beta_1 \neq 0$ and are consistent, convergent and stable.

4 Parameter learning algorithms of LMFLS's

Suppose that an LMFLS has been constructed and at each time step the sampling data

$$x(i;x_0) \equiv x(ih;x_0),\ i = 0,\cdots,\left[\frac{T}{h}\right] = L + k - 1$$

are obtained. Our task is to develop a learning algorithm to adjust parameters of the LMFLS based on the data, such that the constructed LMFLS can approximate the solution of ODE (1). With the singleton fuzzifier, the product rule of inference, the centre-average defuzzifier and the Gaussian membership functions, the output of FLS is

$$F_f(x) \equiv F_f(x;\theta) =$$
$$(F_{f_1}(x;\theta), F_{f_2}(x;\theta),\cdots,F_{f_n}(x;\theta))^{\mathrm{T}},$$

where

$$\theta = (\theta_1,\theta_2,\cdots,\theta_n)^{\mathrm{T}},$$
$$\theta_j = (\theta_j^1,\theta_j^2,\cdots,\theta_j^M)^{\mathrm{T}},\ j = 1,2,\cdots,n;$$
$$F_{f_j} = \theta_j^{\mathrm{T}} p(x),$$
$$p(x) = (p_1(x),p_2(x),\cdots,p_M(x))^{\mathrm{T}},$$
$$p_l(x) = \frac{\prod_{i=1}^{n}\exp\left[\left(\frac{x_i-\bar{x}_i^l}{\sigma_i^l}\right)^2\right]}{\sum_{l=1}^{M}\prod_{i=1}^{n}\exp\left[-\left(\frac{x_i-\bar{x}_i^l}{\sigma_i^l}\right)^2\right]},$$
$$l = 1,2,\cdots,M.$$

Thus (4) can be written as

$$y(i+k) \equiv$$
$$(y_1(i+k),y_2(i+k),\cdots,y_n(i+k))^{\mathrm{T}} =$$
$$-\alpha_0 y(i) - \alpha_1 y(i+1) - \cdots - \alpha_{k-1}y(i+k-1) +$$
$$h(\beta_0(\theta_1^{\mathrm{T}}p(y(i)),\theta_2^{\mathrm{T}}p(y(i)),\cdots,\theta_n^{\mathrm{T}}p(y(i)))+\cdots+$$
$$\beta_{k-1}(\theta_1^{\mathrm{T}}p(y(i+k-1)),\theta_2^{\mathrm{T}}p(y(i+k-1)),$$
$$\cdots,\theta_n^{\mathrm{T}}p(y(i+k-1)))^{\mathrm{T}}. \qquad (6)$$

In the following, we will give two types of parameter learning algorithms for LMFLS's. The first one is a back propagation (BP) algorithm. Define the error at time step i to be

$$e = \frac{1}{2}\|y(i+k) - x(i+k)\|^2. \qquad (7)$$

The following learning algorithm can be used to adjust parameter θ_j:

$$\theta_j(m+1) = \theta_j(m) - \alpha\frac{\partial e}{\partial\theta_j}\bigg|_m =$$
$$\theta_j(m) - \alpha h(y_j(i+k) - x_j(i+k))\cdot$$
$$\left[\beta_0\frac{\partial F_{f_j}(y(i);\theta_j)}{\partial\theta_j}+\cdots+\beta_{k-1}\frac{\partial F_{f_j}(y(i+k-1);\theta_j)}{\partial\theta_j}\right]\Bigg|_m,$$

where $j = 1,2,\cdots,M$; $m = 0,1,2,\cdots$; $0 < \alpha < 1$ is a learning constant, and

$$\frac{\partial F_{f_j}(y(i+s);\theta_j)}{\partial\theta_j} = (p_1(y(i+s)),\cdots,p_M(y(i+s)))^{\mathrm{T}}, \qquad (8)$$

where $s = 0,1,\cdots,k-1$.

The second one is a least-square learning algorithm.

Denote the total error between outputs of the LMFLS and the training trajectories by

$$e = \frac{1}{2}\sum_{i=0}^{L-1}\|y(i+k) - x(i+k)\|^2. \qquad (9)$$

From (6),

$$y(i+k)+\alpha_0x(i)+\alpha_1x(i+1)+\cdots+\alpha_{k-1}x(i+k-1) =$$
$$(\theta_1^{\mathrm{T}}h[\beta_0p(x(i)) + \cdots + \beta_{k-1}p(x(i+k-1))],\cdots,$$
$$\theta_n^{\mathrm{T}}h[\beta_0p(x(i)) + \cdots + \beta_{k-1}p(x(i+k-1))]).$$

Define

$$A(x(i),x(i+1),\cdots,x(i+k-1)) = h(A_j)_{n\times 1},$$
$$A_j = \beta_0p(x(i)) + \cdots + \beta_{k-1}p(x(i+k-1)),$$
$$j = 1,2,\cdots,n.$$

Thus,

$$y(i+k) = A^{\mathrm{T}}(x(i),x(i+1),\cdots,x(i+k-1))\theta -$$
$$\alpha_0x(i)-\alpha_1x(i+1)-\cdots-\alpha_{k-1}x(i+k-1).$$

Substituting it into (9), it can be seen that the minimization of (9) is equivalent to finding the least square solution of the following equations:

$$\begin{bmatrix} A^{\mathrm{T}}(x(0),x(1),\cdots,x(k-1)) \\ \vdots \\ A^{\mathrm{T}}(x(L-1),x(L),\cdots,x(j-2)) \end{bmatrix}\begin{bmatrix}\theta_1\\ \vdots \\ \theta_n\end{bmatrix} =$$
$$\begin{bmatrix} \alpha_0x(0) + \cdots + \alpha_{k-1}x(k-1) + x(k) \\ \vdots \\ \alpha_0x(L-1) + \cdots + \alpha_{j-1}x(j-1) + x(j)\end{bmatrix}, \qquad (10)$$

where $j = L + k - 1$.

Now we have already provided two parameter learning algorithms for LMFLS's: the BP learning algorithm and the least square algorithm. The former can adjust parameters of LMFLS's on-line, but easily plunges into a local extreme value. The latter can only adjust parameters of LMFLS's off-line, but the global optimum of (9) may be reached by choosing a proper method for solving (10).

5 Example and simulation

Consider the following system

$\ddot{y} = -\sqrt{1-\dot{y}^2},\ y(0) = 0,\ \dot{y}(0) = 1,\ 0 \leqslant t \leqslant 16.$

Assume that the structure of this system is unknown but its position y and velocity $\dot{y}$ at each time step ih can be measured. Let

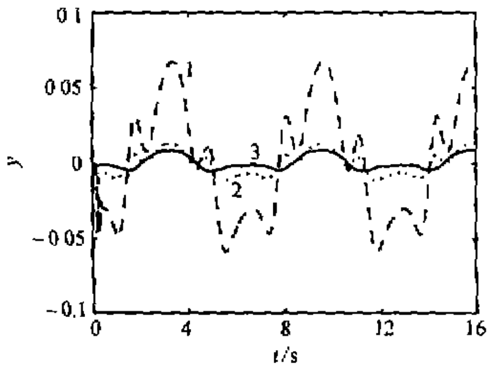
$x_1 = \dot{y},\ x_2 = y,\ x(t) = (x_1, x_2)^{\mathrm{T}},$

$f(x(t)) = (f_1(x(t)), f_2(x(t)))^{\mathrm{T}} = (-\sqrt{1-x_1^2},\ x_1)^{\mathrm{T}},$

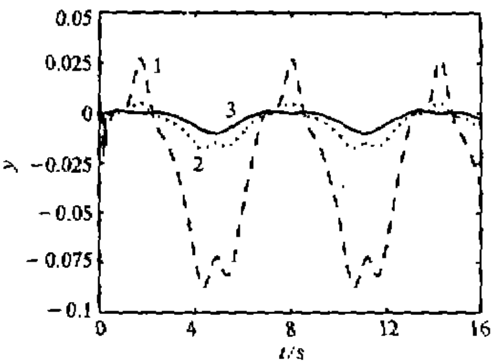
then the system becomes

$$\dot{x} = f(x(t)),\ x(0) = (1,0)^{\mathrm{T}}.$$

Take step sizes $h = 0.05,\ 0.01,\ 0.005$, respectively. Using the obtained states as training data, we apply the BP algorithm to adjust the parameters of an LMFLS until $t = 8$. Thus we obtain an LMFLS that can approximate the given system. And then we test the prediction capacity of the obtained LMFLS from $t = 0$ to $t = 16$. The simulation results are given in Fig.2.



(a) The curve of errors between y and corresponding outputs of the LMFLS



(b) The curve of errors between $\dot{y}$ and corresponding outputs of the LMFLS

Fig. 2 The simulation results

In Fig. 2 curves 1, 2 and 3 denote the error curves with $h = 0.05,\ 0.01$ and 0.005, respectively. The mean value of absolute errors for tracking y are 0.0335, 0.0068, 0.0038, respectively; The mean value of absolute errors for tracking $\dot{y}$ are 0.0322, 0.0064, 0.0028, respectively. From Fig. 2, we see that the shorter the step size h is, the smaller the state tracking errors of the LMFLS are.

6 Conclusion

In this paper a new LMFLS is presented based on the linear multi-step methods. It is proved that the LMFLS can predict states of an unknown system and simultaneously approximate the unknown function determining the system. Since multi-step methods make full use of previous information to reduce errors, but one-step methods need to increase compute quantity in each step for this purpose. Generally, because achieving the same accuracy the computational quantity of a multi-step method is less than that of a one-step method, the proposed LMFLS has lower computing burden.

References

[1] Wang Yijen and Lin Chinteng. Runge-Kutta neural network for identification of dynamical systems in high accuracy [J]. IEEE Transactions on Neural Networks, 1998, 9(2): 249 – 307

[2] Wang L X. Adaptive Fuzzy Systems and Control-Design and Stability Analysis [M]. Beijing: National Defence Industry Press, 1995, 38 – 39

[3] Hu Jianwei and Tang Huaimin. Numerical Methods in Differential Equations [M]. Beijing: Science Press, 1999, 26 – 36

[4] Guan Zhi and Lu Jinfu. Basic Numerical Analysis [M]. Beijing: Higher Education Press, 1998, 445 – 474

[5] Bart Kosko. Fuzzy Engineering [M]. Xi'an: Xi'an Jiaotong University Press, 1999, 31 – 70

本文作者简介

孙多青 1964年生.1984年毕业于河北师范学院数学系并获学士学位.现为北京航空航天大学博士研究生、河北职业技术师范学院讲师.主要研究方向:智能控制.

霍 伟 1951年生.北京航空航天大学教授,博士生导师.主要研究方向为非线性动力学系统控制及智能控制.