

控制与决策

Control and Decision

基于多维泰勒网的超前d步预测模型

李晨龙, 严洪森

引用本文:

李晨龙, 严洪森. 基于多维泰勒网的超前d步预测模型[J]. 控制与决策, 2021, 36(2): 345–354.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2019.0722>

您可能感兴趣的其他文章

Articles you may be interested in

基于偏态深度分类的高炉硅含量及波动预测

Prediction of blast furnace silicon content and fluctuation based on skewness depth classification

控制与决策. 2021, 36(2): 491–497 <https://doi.org/10.13195/j.kzyjc.2019.1116>

基于Stacking的高炉铁水质量区间预测

Prediction of blast furnace hot metal quality interval based on stacking

控制与决策. 2021, 36(2): 335–344 <https://doi.org/10.13195/j.kzyjc.2019.0786>

抗遮挡与尺度自适应的改进KCF跟踪算法

Improved KCF tracking algorithm based on anti-occlusion and scale transformation

控制与决策. 2021, 36(2): 457–462 <https://doi.org/10.13195/j.kzyjc.2019.0394>

基于仿生算法改进粒子滤波的SLAM算法精度预测

Accuracy prediction of SLAM algorithm based on bionic algorithm to improve particle filter

控制与决策. 2021, 36(1): 166–172 <https://doi.org/10.13195/j.kzyjc.2019.0555>

基于改进堆叠自动编码器的循环冷却水系统工艺介质温度预测控制方法

Predictive control method of process medium temperature in circulating cooling water system based on improved stacked auto encoders

控制与决策. 2020, 35(12): 2835–2844 <https://doi.org/10.13195/j.kzyjc.2019.0694>

基于多维泰勒网的超前 d 步预测模型

李晨龙, 严洪森[†]

(1. 东南大学 自动化学院, 南京 210096; 2. 东南大学 复杂工程系统测量与控制教育部重点实验室, 南京 210096)

摘要: 针对单入单出 (SISO) 与多入多出 (MIMO) 非线性时滞系统构建预测模型准确性问题, 分别提出基于多维泰勒网 (MTN) 的预测模型构建方案. 首先, 分别依靠非递推技术与递推技术来设计非递推 d 步与递推 d 步超前 MTN 预测模型, 给出二者表达式, 二者皆可对未来 d 步范围进行预测, 并有效弥补时滞带来的影响; 然后, 利用阻尼递推最小二乘 (DRLS) 算法, 带有动量因子的 BP 算法, Levenberg Marquardt (L-M) 算法和扩展卡尔曼滤波 (EKF) 算法分别作为 MTN 预测模型的学习算法进行实时在线学习; 最后, 引入两个仿真例子来验证所建立预测模型的准确性和实时性, 并与神经网络预测模型作对比. 实验结果表明, 相比较神经网络预测模型, 所提出的两种在线构建预测模型方案具有更好的准确性与实时性. 同时, 4 种不同的学习算法对 MTN 预测模型的准确度影响不大.

关键词: 多维泰勒网; 预测模型; 递推 d 步超前预测; 非递推 d 步超前预测; 非线性时滞系统

中图分类号: TM351

文献标志码: A

DOI: 10.13195/j.kzyjc.2019.0722

开放科学 (资源服务) 标识码 (OSID):



引用格式: 李晨龙, 严洪森. 基于多维泰勒网的超前 d 步预测模型 [J]. 控制与决策, 2021, 36(2): 345-354.

d -step-ahead predictive model based on multi-dimensional Taylor network

LI Chen-long, YAN Hong-sen[†]

(1. School of Automation, Southeast University, Nanjing 210096, China; 2. MOE Key Laboratory of Measurement and Control of Complex Systems of Engineering, Southeast University, Nanjing 210096, China)

Abstract: For the accuracy of the predictive models for single input single output (SISO) and multiple input multiple output (MIMO) nonlinear time-delay systems, the schemes of constructing predictive models based on the multi-dimensional Taylor network (MTN) are proposed. First, the non-recursive direct d -step-ahead MTN predictive model and the recursive d -step-ahead MTN predictive model are designed and the expressions are given respectively relying on non-recursive and recursive technology, which can predict the range of d steps in the future and compensate the influence of time-delay. Then the damped recursive least squares (DRLS), the BP algorithm with momentum factors, the Levenberg Marquardt (L-M) and the extended Kalman filter (EKF) learning algorithms are used as the learning of the MTN predictive model to conduct on-line learning, respectively. Finally, two experimental examples are given to certify the accuracy and real-time performance of the predictive model, which are compared with those of the neural network predictive models. Results from our experiments show that the two kinds of predictive models constructed using the proposed schemes have a better accuracy and real-time performance than the neural network predictive models. Meanwhile, four kinds of different learning algorithms have little effect on the accuracy of the MTN predictive models.

Keywords: multi-dimensional Taylor network; predictive model; recursive d -step-ahead prediction; non-recursive d -step-ahead prediction; nonlinear time-delay system

0 引言

随着经济社会的发展, 大量的问题离不开对变量未来值的预测, 因此建立一个有效的预测模型显得尤为重要. 实际的系统中因为具有非线性、时滞等特性而难以建模和控制^[1]. 以往线性系统中, 因为可以得

到其精确的数学模型, 所以利用传统的建模方法可以达到要求, 比如递推最小二乘法、最大似然法、阶跃响应法^[2-3]. 而对于非线性系统建模问题, 由于很难获得系统精确的数学模型, 以往对于线性系统的方法很难达到理想的效果, 给建模带来困难^[4]. 基于此, 学者

收稿日期: 2019-05-25; 修回日期: 2019-10-10.

基金项目: 国家自然科学基金项目 (61673112, 60934008); 中央高校基本科研业务费专项资金项目 (2242017K10003, 2242014K10031); 江苏省高校优势学科建设工程项目; 江苏省研究生培养科研创新工程项目 (KYCX18_0100).

责任编辑: 张维海.

[†]通讯作者. E-mail: hsyang@seu.edu.cn.

们提出了不同的解决方案,比如文献[5]利用 Takagi-Sugeno 模糊模型逼近非线性系统;文献[6]针对非线性系统,利用 Hammerstein 模型提出一种迭代的建模方法;文献[7]针对非线性离散动态系统,提出基于神经网络的建模方法.然而, Takagi-Sugeno 模糊模型的构建是通过聚类得到的,而聚类点通常在斜轴,而非平行轴,从而导致分解误差的产生^[8]; Hammerstein 模型未知参数向量的维度过大,导致计算复杂度增加^[9];神经网络由于在训练过程中花费时间长,易陷入极小值点,且算法复杂度较高,从而不满足实时构建预测模型的要求^[10].

对于时滞问题,使用预测的思想可以有效抵消时滞的影响,预测可以对未来有限范围进行过程输出预测,这对于线性系统有很好的效果^[11],而对于含有非线性等因素的工程领域,由于难以得到精确的数学模型,基于线性系统的预测模型构建方法显得力不从心.神经网络因其具有强大的逼近性能,常用来作为预测模型来抵消时滞的影响,比如文献[12]针对 SISO 非线性系统,提出基于神经网络的一步超前预测方案;文献[13]针对非线性时滞系统,提出基于神经网络的递推 d 步超前预测模型;文献[14]针对非线性系统,将自组织递推 RBF 神经网络用于预测模型,提出基于自组织递推 RBF 神经网络的预测控制方案.然而,神经网络存在训练过程中花费时间长,易陷入极小值点等问题^[10].

针对以上问题,严洪森教授于2010年10月提出多维泰勒网的建模思想,之后被应用在非线性的时间序列预测上^[15-17].多维泰勒网具有强大的逼近性能,且结构简单,因其拥有较低的算法复杂度而满足实时性能的要求.

基于此,本文首先针对 SISO 与 MIMO 非线性时滞系统,基于非递推技术与递推技术,分别设计非递推 d 步和递推 d 步超前 MTN 预测模型,给出二者表达式,二者皆可对未来 d 步范围进行预测,均能有效弥补时滞带来的影响;然后,利用 DRLS 算法、带有动量因子的 BP 算法、L-M 算法和 EKF 算法分别作为 MTN 预测模型的学习算法来实施在线建模;最后,引入两个例子来验证本文所建预测模型的准确性与实时性,同时与神经网络预测模型作对比,给出二者所建预测模型的准确性以及实时性分析.

基于此,本文的主要贡献为:

1) 针对非线性时滞系统,借助 MTN 的强大逼近性能,提出基于 MTN 的构建预测模型方案.

2) 针对 SISO 离散非线性时滞系统,给出非递推 d

步超前 MTN 预测模型的表达式;针对 MIMO 离散非线性时滞系统,给出递推 d 步超前 MTN 预测模型的表达式.

3) DRLS 算法、带有动量因子的 BP 算法、L-M 算法和 EKF 算法分别作为 MTN 预测模型的学习算法,4 种不同的学习算法对预测模型的准确度影响不大,突出了利用 MTN 构建预测模型的优越性.

1 系统描述

本文主要考虑两种系统,分别是 SISO 和 MIMO 离散非线性时滞系统.

1.1 SISO 离散非线性时滞系统

SISO 离散非线性时滞系统

$$y(k) = g[y(k-1), y(k-2), \dots, y(k-n_y), \\ u(k-d), \dots, u(k-n_u-d)]. \quad (1)$$

其中: $g[\cdot]$ 为非线性函数, $u(k)$ 和 $y(k)$ 分别是系统的输入和输出, n_y 和 n_u 分别代表 $\{y(k)\}$ 和 $\{u(k)\}$ 的阶次, d 为系统的输入时滞.

1.2 MIMO 离散非线性时滞系统

MIMO 离散非线性时滞系统

$$\mathbf{y}(k) = \mathbf{g}[\mathbf{y}(k-1), \mathbf{y}(k-2), \dots, \mathbf{y}(k-n_y), \\ \mathbf{u}(k-d), \dots, \mathbf{u}(k-n_u-d)]. \quad (2)$$

其中: $\mathbf{g}(\cdot) = [g_1(\cdot), \dots, g_q(\cdot)]^T \in R^q$ 为未知的非线性向量函数, $\mathbf{y}(k) = [y_1(k), \dots, y_q(k)]^T \in R^q$, $\mathbf{u}(k) = [u_1(k), \dots, u_p(k)]^T \in R^p$ 分别为系统(2)的 q 维输出和 p 维输入, n_{y_j} ($j = 1, 2, \dots, q$), n_{u_i} ($i = 1, 2, \dots, p$) 分别为第 j 个输出分量和第 i 个输入分量的阶次,且 $n_y = \{n_{y_1}, n_{y_2}, \dots, n_{y_q}\}$, $n_u = \{n_{u_1}, n_{u_2}, \dots, n_{u_p}\}$, d 为系统的输入时滞.

在 k 时刻,系统(1)的第 j 个输出分量可以表示为

$$y_j(k) = g_j(y_1(k-1), \dots, y_1(k-n_{y_1}), \dots, \\ y_q(k-1), \dots, y_q(k-n_{y_q}), \\ u_1(k-d), u_1(k-d-1), \dots, \\ u_1(k-d-n_{u_1}), \dots, u_p(k-d), \\ u_p(k-d-1), \dots, u_p(k-d-n_{u_p})). \quad (3)$$

2 多维泰勒网模型

2.1 SISO 多维泰勒网

多维泰勒网本质上是一个多项式类型的非线性自回归滑动平均模型.根据多元泰勒公式的原理,如果函数在某点邻域处为 m 阶可导,则该函数在该点展开式为变量幂级数不大于 m 次的形式.因此,针对 SISO 非线性系统,基于多维泰勒网模型,可以将 n 维

系统的一般动力学方程^[16]表示为

$$\hat{y}(k+1) = \sum_{\hat{p}=1}^{\hat{N}(\hat{t}, \hat{m})} \hat{w}_{\hat{p}}(k) \prod_{\hat{q}=1}^{\hat{t}} \hat{z}_{\hat{q}}^{\hat{\lambda}(\hat{p}, \hat{q})}(k). \quad (4)$$

其中: $\hat{N}(\hat{t}, \hat{m})$ 表示 \hat{t} 元函数 $\hat{f}(\cdot)$ 展开成 \hat{m} 次逼近多项式乘积项的总项数; $\hat{w}_{\hat{p}}(k)$ 表示第 \hat{p} 个乘积项的权重系数; $\hat{\lambda}(\hat{p}, \hat{q})$ 表示第 \hat{p} 个乘积项中变量 $\hat{z}_{\hat{q}}(k)$ 的次数, 且 $\sum_{\hat{q}=1}^{\hat{t}} \hat{\lambda}(\hat{p}, \hat{q}) \leq \hat{m}$.

SISIO 非线性系统的 MTN 结构如图 1 所示.

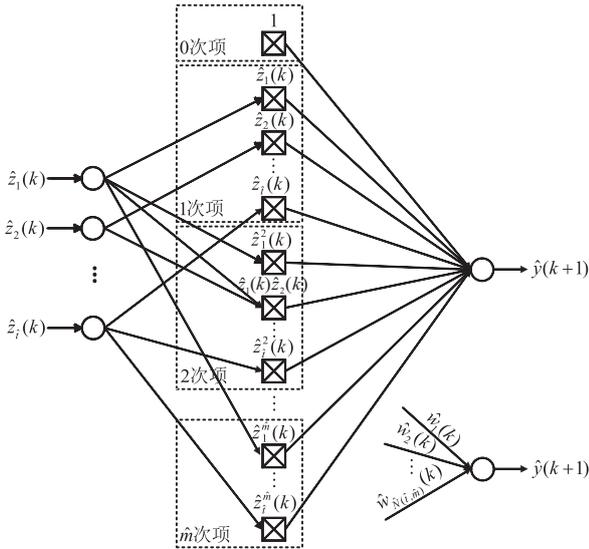


图 1 SISIO 的 MTN 结构

2.2 MIMO 多维泰勒网

针对 MIMO 非线性系统, 基于多维泰勒网, 第 j 个输出分量 $\hat{y}_j(k+1)$ ^[16] 可写成如下形式:

$$\hat{y}_j(k+1) = \sum_{\hat{r}=1}^{\hat{N}(\hat{t}, \hat{m})} \hat{w}_{j, \hat{r}}(k) \prod_{\hat{s}=1}^{\hat{t}} \hat{z}_{\hat{s}}^{\hat{\lambda}(\hat{r}, \hat{s})}(k). \quad (5)$$

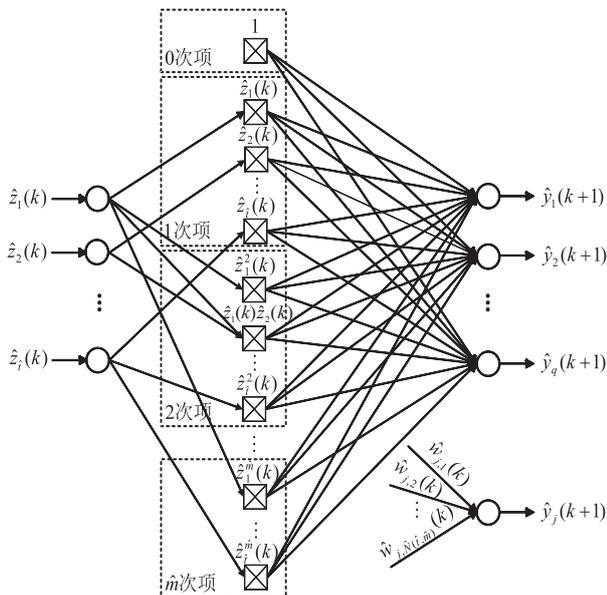


图 2 MIMO 的 MTN 结构

其中: $\hat{N}(\hat{t}, \hat{m})$ 表示 \hat{t} 元函数 $\hat{f}_j(\cdot)$ 展开成 \hat{m} 次逼近多项式乘积项的总项数; 权重向量为 $\hat{w}_j(k) = [\hat{w}_{j,1}(k), \dots, \hat{w}_{j, \hat{N}(\hat{t}, \hat{m})}(k)]^T$; $\hat{w}_{j, \hat{r}}(k)$ 表示第 \hat{r} 个乘积项的权重系数; $\hat{\lambda}(\hat{r}, \hat{s})$ 表示第 \hat{r} 个乘积项中变量 $\hat{z}_{\hat{s}}(k)$ 的次数, 且 $\sum_{\hat{s}=1}^{\hat{t}} \hat{\lambda}(\hat{r}, \hat{s}) \leq \hat{m}$.

MIMO 非线性系统的 MTN 结构如图 2 所示.

2.3 多维泰勒网逼近性能

引理 1^[18] 任何定义于一个闭区间的连续函数可以用多项式函数任意准确地逼近.

引理 2^[17] 对于定义于一个闭区间的连续函数 $f(x_1, x_2, \dots, x_n)$, 可以用 $\sum_{t=1}^{N(n, m)} w_t \prod_{i=1}^n x_i^{\lambda_{t, i}}$ 逼近. 其中 $N(n, m)$ 为逼近展开式中乘积项的总项数, w_t 为逼近展开式中第 t 个乘积项之前的权值, $\lambda_{t, i}$ 为展开式中第 t 个乘积项中变量 x_i 的幂次.

正如图 1 和图 2 所示, MTN 采用前向单中间层结构, 包括输入层、中间层和输出层. 同时, 基于引理 1 和引理 2, 只要 $N(n, m)$ 足够大, 它能够以足够的精度逼近任意模型^[18].

3 MTN 预测模型

本文给出两种构建 MTN 预测模型方案, 分别为非递推 d 步与递推 d 步超前预测模型, 二者皆可预测未来 d 步范围的输出. 同时需要结合适当的算法作为 MTN 预测模型的学习算法. 采用 d 步超前预测模型可有效弥补时滞带来的影响.

3.1 非递推 d 步超前 MTN 预测模型

因为 MTN 具有强大的逼近性能, 所以式 (1) 中的非线性函数 $g[\cdot]$ 可以由 MTN 以及一个适当的算法任意逼近^[16]. 映射关系记为 $\hat{g}[\cdot]$, 对于系统 (1) 的 MTN 模型可以表示成如下形式:

$$\hat{y}(k) = \hat{g}[y(k-1), y(k-2), \dots, y(k-n_y), u(k-d), \dots, u(k-n_u-d), \hat{w}(k)]. \quad (6)$$

其中: $\hat{y}(k)$ 为 MTN 输出; $\hat{w}(k)$ 为 MTN 预测模型的权重系数; $u(k), y(k), n_y, n_u$ 和 d 与式 (1) 含义相同.

本节采用非递推 d 步超前预测模型, 其不依靠递推技术, 被用来预测未来的输出^[13]. 此方法已经有效地应用于线性系统, 但下面的引理 3 给出了非递推 d 步预测模型应用于非线性系统的可能.

引理 3^[13] 设 $Y(k) = \{y(k), \dots, y(k-n_y+1)\}$, $U(k) = \{u(k), \dots, u(k-d-n_u+1)\}$ 为采样点 k 时刻过去的输入和输出序列. 设 $\phi^d(\cdot)$ 为连续的映射, 预测输出可以写为

$$\hat{y}(k+d) = \phi^d[Y(k), U(k), \beta_d]. \quad (7)$$

式(7)能够满足

$$Q[y(k+d) - \hat{y}(k+d)] \rightarrow \beta_d(\min). \quad (8)$$

其中: $Q(\cdot)$ 为确定性标准, $u(k)$ 和 $y(k)$ 分别是系统的输入和输出, n_y 和 n_u 分别代表 $\{y(k)\}$ 和 $\{u(k)\}$ 的阶次, d 为系统的输入时滞.

MTN 被用作非递推 d 步超前预测模型. 与非递推预测模型的原理相反^[13], d 步超前非线性预测模型易于推导, 则如式(6)所示的 MTN 预测模型可以改写为

$$\hat{y}(k+d) = \hat{g}[y(k), y(k-1), \dots, y(k-n_y), u(k), \dots, u(k-n_u), \hat{w}(k)]. \quad (9)$$

其中: $\hat{g}[\cdot]$ 为非线性函数, $\hat{y}(k+d)$ 为模型输出, $u(k)$ 和 $y(k)$ 分别为系统的输入和输出, $\hat{w}(k)$ 为 MTN 预测模型的权值系数, n_y 和 n_u 分别为 $\{y(k)\}$ 和 $\{u(k)\}$ 的阶次, d 为系统时滞.

设 $\hat{t} = n_y + n_u + 2$, $\hat{v}(k) = [v_1(k), \dots, v_{n_y+1}(k), v_{n_y+2}(k), v_i(k)]^T = [y_P(k), \dots, y_P(k-n_y), u_P(k), u_P(k-1), \dots, u_P(k-n_u)]^T$, $\hat{w}(k) = (\hat{w}_1(k), \hat{w}_2(k), \dots, \hat{w}_{\hat{N}(\hat{t}, \hat{m})}(k))^T$.

MTN 预测模型(9)可以表示为如下形式^[16]:

$$\hat{y}(k+d) = \sum_{\hat{p}=1}^{\hat{N}(\hat{t}, \hat{m})} \hat{w}_{\hat{p}}(k) \prod_{\hat{q}=1}^{\hat{t}} \hat{v}_{\hat{q}}^{\hat{\lambda}(\hat{p}, \hat{q})}(k). \quad (10)$$

其中: $\hat{N}(\hat{t}, \hat{m})$ 表示 \hat{t} 元函数 $\hat{g}[\cdot]$ 展开成 \hat{m} 次逼近多项式乘积项的总项数; $\hat{w}_{\hat{p}}(k)$ 表示第 \hat{p} ($\hat{p} = 1, 2, \dots, \hat{N}(\hat{t}, \hat{m})$) 个乘积项的权重系数; $\hat{\lambda}(\hat{p}, \hat{q})$ 表示第 \hat{p} 个乘积项中变量 $\hat{v}_{\hat{q}}(k)$ 的次数, 且 $\sum_{\hat{q}=1}^{\hat{t}} \hat{\lambda}(\hat{p}, \hat{q}) \leq \hat{m}$. 非递推 d 步 MTN 预测模型结构如图3所示.

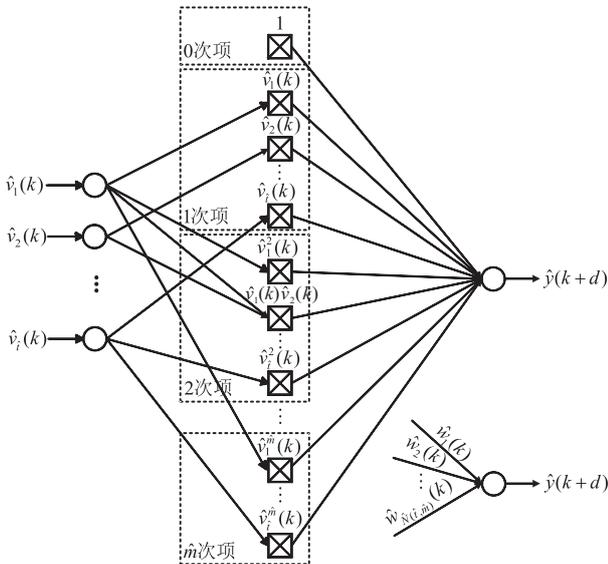


图3 非递推 d 步 MTN 预测模型结构

3.2 递推 d 步 MTN 预测模型

对于非线性时滞系统, 基于递推技术^[13,19], 本节提出基于 MTN 的递推 d 步超前预测模型来弥补时滞的影响.

针对如式(2)所示的 MIMO 非线性时滞系统, MTN 可以任意精度逼近 $g(\cdot)$, $\hat{g}(\cdot)$ 为映射关系, MTN 模型可以写为

$$\hat{y}(k) = \hat{g}[y(k-1), y(k-2), \dots, y(k-n_y), u(k-d), \dots, u(k-n_u-d), \hat{w}(k)]. \quad (11)$$

其中: $\hat{y}(k)$ 为 MTN 预测模型输出, 且 $\hat{y}(k) = [\hat{y}_1(k), \dots, \hat{y}_q(k)]^T \in R^q$; $\hat{w}(k)$ 为 MTN 权系数向量; $\hat{g}(\cdot) = [\hat{g}_1(\cdot), \dots, \hat{g}_q(\cdot)]^T \in R^q$; $u(k)$ 、 $y(k)$ 、 n_y 、 n_u 和 d 与式(2)有相同的含义.

然后, 系统(11)在第 k 时刻的第 j 次输出分量为

$$\begin{aligned} \hat{y}_j(k) = & \hat{g}_j(y_1(k-1), \dots, y_1(k-n_{y_1}), \dots, \\ & y_q(k-1), \dots, y_q(k-n_{y_q}), \\ & u_1(k-d), u_1(k-d-1), \dots, \\ & u_1(k-d-n_{u_1}), \dots, u_p(k-d), \\ & u_p(k-d-1), \dots, u_p(k-d-n_{u_p})). \end{aligned} \quad (12)$$

MIMO 系统(11)的一步超前 MTN 预测模型为

$$\begin{aligned} \hat{y}(k+1) = & \hat{g}[\hat{y}(k), y(k-1), \dots, y(k-n_y+1), \\ & u(k-d+1), \dots, u(k-d-n_u+1), \hat{w}(k)]. \end{aligned} \quad (13)$$

其中 $\hat{y}(k)$ 、 $\hat{w}(k)$ 、 $u(k)$ 、 $y(k)$ 、 n_y 、 n_u 和 d 与式(11)中的含义相同.

系统(13)在 k 时刻的第 j 次输出分量为

$$\begin{aligned} \hat{y}_j(k+1) = & \hat{g}_j(y_1(k), \dots, y_1(k-n_{y_1}+1), \dots, y_q(k), \dots, \\ & y_q(k-n_{y_q}+1), u_1(k-d+1), u_1(k-d), \dots, \\ & u_1(k-d-n_{u_1}+1), \dots, u_p(k-d+1), \\ & u_p(k-d), \dots, u_p(k-d-n_{u_p}+1)). \end{aligned} \quad (14)$$

为了方便而不失一般性, 设

$$\hat{t}_u = \sum_{i=1}^p (n_{u_i} + 1), \hat{t}_y = \sum_{j=1}^q (n_{y_j} + 1), \hat{t} = \hat{t}_u + \hat{t}_y,$$

$$\begin{aligned} \hat{z}(k) = & [\hat{z}_1(k), \dots, \hat{z}_{\hat{t}_y}(k), \hat{z}_{\hat{t}_y+1}(k), \dots, \hat{z}_{\hat{t}}(k)]^T = \\ & [y_1(k), \dots, y_q(k-n_{y_q}+1), u_1(k-d+1), \\ & \dots, u_p(k-n_{u_p}-d+1)]^T. \end{aligned}$$

MTN 预测模型的第 j 次输出分量 $\hat{y}_j(k)$ 的权值

向量为

$$\hat{\boldsymbol{w}}_j(k) = [\hat{w}_{j,1}(k), \dots, \hat{w}_{j,\hat{N}(\hat{t},\hat{m})}(k)]^T.$$

相应的MTN模型(14)可以改写为^[16]

$$\hat{y}_j(k+1) = \sum_{\hat{r}=1}^{\hat{N}(\hat{t},\hat{m})} \hat{w}_{j,\hat{r}}(k) \prod_{\hat{s}=1}^{\hat{t}} \hat{z}_{\hat{s}}^{\hat{\lambda}(\hat{r},\hat{s})}(k). \quad (15)$$

根据MTN模型以及递推技术,可以得到递推 d 步超前MTN预测模型. 令 $k = k + 1$,重复以上的迭代过程,代入式(13),在 $d - 1$ 步迭代后可以获得递推 d 步超前MTN预测模型.

$$\begin{aligned} \hat{\boldsymbol{y}}(k+d) = & \hat{\boldsymbol{g}}[\hat{\boldsymbol{y}}(k+d-1), \hat{\boldsymbol{y}}(k+d-2), \dots, \\ & \hat{\boldsymbol{y}}(k+d-n_y), \boldsymbol{u}(k), \dots, \boldsymbol{u}(k-n_u), \hat{\boldsymbol{w}}(k)], \end{aligned} \quad (16)$$

其中 $\hat{\boldsymbol{g}}(\cdot)$ 、 $\boldsymbol{y}(k)$ 、 $\boldsymbol{u}(k)$ 、 q 、 p 、 n_{y_j} 、 n_{u_i} 和 d 与式(11)中的含义相同. 递推 d 步超前MTN预测模型结构如图4所示.

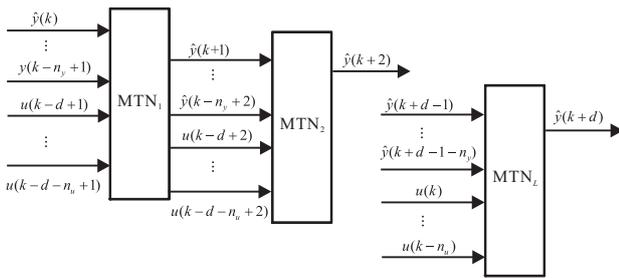


图4 递推 d 步超前MTN预测模型结构

系统(16)在 $k + d - 1$ 时刻的第 j 次输出分量为

$$\begin{aligned} \hat{y}_j(k+d) = & \hat{g}_j[\hat{y}_1(k+d-1), \dots, \hat{y}_1(k+d-n_{y_1}), \dots, \\ & \hat{y}_q(k+d-1), \dots, \hat{y}_q(k+d-n_{y_q}), u_1(k), \dots, \\ & u_1(k-n_{u_1}), \dots, u_p(k), \dots, u_p(k-n_{u_p})]. \end{aligned} \quad (17)$$

为了方便而不失一般性,设

$$\hat{t}_u = \sum_{i=1}^p (n_{u_i} + 1), \hat{t}_y = \sum_{j=1}^q (n_{y_j} + 1), \hat{t} = \hat{t}_u + \hat{t}_y,$$

$$\begin{aligned} \hat{\boldsymbol{z}}(k) = & [\hat{z}_1(k), \dots, \hat{z}_{\hat{t}_y}(k), \hat{z}_{\hat{t}_y+1}(k), \dots, \hat{z}_{\hat{t}}(k)]^T = \\ & [\hat{y}_1(k+d-1), \dots, \hat{y}_q(k+d-n_{y_q}), \\ & u_1(k), \dots, u_p(k-n_{u_p})]^T. \end{aligned}$$

定义MTN的第 j 次输出分量 $\hat{y}_j(k)$ 的权值向量为

$$\hat{\boldsymbol{w}}_j(k) = [\hat{w}_{j,1}(k), \dots, \hat{w}_{j,\hat{N}(\hat{t},\hat{m})}(k)]^T.$$

相应的MTN模型(17)可以改写为^[16]

$$\hat{y}_j(k+d) = \sum_{\hat{r}=1}^{\hat{N}(\hat{t},\hat{m})} \hat{w}_{j,\hat{r}}(k) \prod_{\hat{s}=1}^{\hat{t}} \hat{z}_{\hat{s}}^{\hat{\lambda}(\hat{r},\hat{s})}(k). \quad (18)$$

递推技术可以应用到远程预测^[19],MTN具有非线性映射能力,可作为预测模型.

如图4所示,其实只用了一个MTN. 但是,对于远程预测情况,需要 $L(L \geq 0)$ 个迭代过程,且需考虑每次的迭代误差. 所以,有必要修正MTN预测模型的误差.

总体而言,非递推形式的建模方案更适用于小时滞过程,其反应速率快;而递推形式的建模方案不但适用于小时滞情况,更适用于大时滞情况,但相对于非递推形式的建模方案反应速率相对较慢^[19]. 同时需要说明的是,本章只推导了针对SISO非线性时滞系统的非递推直接 d 步超前MTN预测模型和针对MIMO非线性时滞系统的递推 d 步超前MTN预测模型,而对于SISO非线性时滞系统的递推 d 步超前MTN预测模型和对于MIMO非线性时滞系统的非递推 d 步超前MTN预测模型的推导显然成立.

3.3 MTN预测模型学习算法

结合MTN预测模型,并选取一个适当的算法可以逼近任何一个非线性系统^[16],本文利用几种常用的算法作为MTN预测模型的学习算法,其中包括DRLS算法^[4,20-21],带有动量因子的BP算法^[22-23],L-M算法^[24-25]和EKF算法^[26-27].

3.3.1 DRLS学习算法

最小二乘(recursive least squares, RLS)算法常常用来训练神经网络的权值,但是由于协方差矩阵的减小,该参数容易发生突变. 为了增加算法的稳定性,在RLS算法的基础上引入阻尼参数,即阻尼递推最小二乘(damped recursive least squares, DRLS)算法^[4]. 因此本文采用DRLS算法作为预测模型的学习算法之一.

1) RLS算法. RLS算法常被应用到系统的参数辨识^[20]. 通过递归的方式更新参数进行系统辨识,以适应实际系统和环境的变化.

定义目标函数^[20]如下:

$$J(\hat{\boldsymbol{w}}(k)) = \sum_{i=0}^k \sigma^{k-i} e^2(\hat{\boldsymbol{w}}(i)). \quad (19)$$

其中: $e(\boldsymbol{w}(i)) = e(i)$, $e(i) = y(i) - \hat{y}(i)$ 代表预测误差;RLS用来调节MTN权值系数; $y(i)$ 为系统第 i 时刻的系统输出; $\hat{y}(i)$ 为MTN模型在第 i 时刻的预测输出; $\hat{\boldsymbol{w}}(k)$ 为在 k 时刻的权值系数; σ 为遗忘因子,在RLS算法的基础上用来估计新老数据的变化,通常 $0.9 < \sigma < 1$,遗忘因子选择的不同会带来不同的结果,小的遗忘因子会带来快速的遗忘速度,也就是说记忆是短暂的.

设

$$\hat{\mathbf{w}}(k) = (\hat{w}_1(k), \hat{w}_2(k), \dots, \hat{w}_{\hat{N}(\hat{t}, \hat{m})}(k))^T,$$

$$\boldsymbol{\beta}(k) = \left(\prod_{\hat{q}=1}^{\hat{t}} \hat{v}_{\hat{q}}^{\hat{\lambda}(1, \hat{q})}(k), \prod_{\hat{q}=1}^{\hat{t}} \hat{v}_{\hat{q}}^{\hat{\lambda}(2, \hat{q})}(k), \dots, \prod_{\hat{q}=1}^{\hat{t}} \hat{v}_{\hat{q}}^{\hat{\lambda}(\hat{N}(\hat{t}, \hat{m}), \hat{q})}(k) \right)^T.$$

RLS算法权值更新公式

$$\hat{\mathbf{w}}(k+1) = \hat{\mathbf{w}}(k) + \mathbf{K}(k+1)e(k). \quad (20)$$

其中

$$e(k) = y(k) - \hat{\mathbf{w}}^T(k)\boldsymbol{\beta}(k),$$

$$\mathbf{K}(k+1) = \frac{\mathbf{P}(k)\boldsymbol{\beta}(k)}{\sigma + \boldsymbol{\beta}^T(k)\mathbf{P}(k)\boldsymbol{\beta}(k)},$$

$$\mathbf{P}(k+1) = \sigma^{-1}[\mathbf{I} - \mathbf{K}(k+1)\boldsymbol{\beta}^T(k)]\mathbf{P}(k);$$

$\hat{\mathbf{w}}(0) = \mathbf{0}$, $\mathbf{P}(0) = \delta\mathbf{I}$, δ 为一个很大的正数,取值范围为100~1000, \mathbf{I} 为单位矩阵.

2) DRLS算法. 为了克服RLS算法的问题,增加算法的稳定性,在RLS算法的基础上引入阻尼参数,即DRLS算法^[4,21].

设目标函数^[4]为

$$J(\hat{\mathbf{w}}(k)) = \sum_{i=0}^k \sigma^{k-i} e^2(\mathbf{w}(i)) + \xi(\mathbf{w}(i+1) - \mathbf{w}(i)), \quad (21)$$

其中 ξ 为阻尼系数,且 $\xi > 0$. ξ 的大小用来描述自变量的增量和使目标函数极小化之间的相对重要性. 当 $\xi = 0$ 时, DRLS算法变成RLS算法. 这里的 σ , $\hat{\mathbf{w}}(i)$ 和 $\hat{\mathbf{w}}(k)$ 与式(19)表示的含义相同.

DRLS算法权值更新公式表示如下:

$$\hat{\mathbf{w}}(k+1) = \hat{\mathbf{w}}(k) + \psi\xi\mathbf{P}(k+1)(\mathbf{w}(k) - \mathbf{w}(k-1)) + \mathbf{P}(k+1)\boldsymbol{\beta}^T e(k). \quad (22)$$

其中: $\mathbf{P}(k+1) = [\xi\mathbf{I} + \sigma(\mathbf{P}^{-1}(k) - \xi\mathbf{I}) + \boldsymbol{\beta}^T\boldsymbol{\beta}]^{-1}$, ψ 为估计误差系数, \mathbf{I} 为单位矩阵, $e(k)$ 与式(20)表示的含义相同.

3.3.2 带有动量因子的BP算法

BP算法是基于误差驱动的算法,解决了多层神经网络的学习问题. 它有修复不完整信息和抑制噪声样本的特点^[22]. 但是,该算法易陷入局部极小值,且训练过程缓慢. 为了解决以上问题,在BP算法的基础上加入动量因子来提高算法的收敛速率^[23].

权值可以通过以下目标函数进行训练:

$$J(k) = \frac{1}{2}e^2(k). \quad (23)$$

其中: $e(k)$ 为预测模型的误差, $e(k) = \mathbf{y}(k) - \hat{\mathbf{y}}(k)$, $\mathbf{y}(k)$ 为系统的输出, $\hat{\mathbf{y}}(k)$ 为MTN预测模型的输出.

BP算法的权值更新公式如下:

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \mu \frac{\partial J}{\partial \mathbf{w}_i}. \quad (24)$$

为了解决易陷入极小值的问题,在BP算法的基础上加入动量因子来提高算法的收敛速率. 权值更新公式如下:

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \mu \frac{\partial J}{\partial \mathbf{w}_i} + \alpha \Delta \mathbf{w}_i. \quad (25)$$

其中: μ 为学习因子, α 为动量因子.

3.3.3 L-M算法

L-M算法是高斯-牛顿(Gauss-Newton)法的改进形式. 其既拥有Gauss-Newton法的局部特性,又拥有梯度法的全局特性^[24-25].

权值可以通过以下目标函数进行训练:

$$J(k) = \frac{1}{2}e^2(k) = \frac{1}{2}[y(k) - \hat{y}(k)]^2. \quad (26)$$

其中: $y(k)$ 为系统的输出, $\hat{y}(k)$ 为MTN预测模型的输出, $e(k) = y(k) - \hat{y}(k)$ 为系统输出与预测模型输出之间的误差. L-M算法来源于牛顿算法与Gauss-Newton算法,其权值更新公式如下:

$$\mathbf{w}_{i+1} = \mathbf{w}_i - \left[\left(\frac{\partial \hat{f}}{\partial \mathbf{w}_i} \right) \left(\frac{\partial \hat{f}}{\partial \mathbf{w}_i} \right)^T + \varsigma \mathbf{I} \right]^{-1} \frac{\partial J}{\partial \mathbf{w}_i}, \quad (27)$$

其中 ς 为常数且 $\varsigma \geq 0$, 可以从0到一个很大的数. 如果 ς 为0,则L-M算法变为Gauss-Newton算法,如果 ς 很大, L-M算法接近一个最陡的下降搜索,此时L-M算法与梯度算法相似.

3.3.4 EKF学习算法

将卡尔曼滤波应用于线性动态系统中,在白噪声的驱动下进行滤波器的最优估计. 然而,对于非线性系统,首先,需要对状态方程进行线性化;其次,需要利用扩展的卡尔曼滤波进行估计,即扩展卡尔曼滤波(extended Kalman filter, EKF)的过程^[26-27]. 因此, EKF作为MTN预测模型的学习算法,有

$$\mathbf{w}_{k+1} = f(\mathbf{w}_k) + v_k. \quad (28)$$

其中: v_k 为白噪声, $f(\cdot)$ 为非线性函数.

由EKF算法,可以得到 \mathbf{w}_{k+1} 的估计值 $\hat{\mathbf{w}}_{k+1}$. 如果式(28)足够光滑,则可以围绕状态估计 $\hat{\mathbf{w}}_{k+1}$ 进行泰勒级数展开,即

$$f(\mathbf{w}_k) = f(\hat{\mathbf{w}}_k) + F_k(\mathbf{w}_k - \hat{\mathbf{w}}_k) + A. \quad (29)$$

其中: A 为高阶项,

$$F_k = \left. \frac{\partial f(\mathbf{w})}{\partial \mathbf{w}} \right|_{\mathbf{w}=\hat{\mathbf{w}}_k}. \quad (30)$$

忽略高阶项,式(29)可由下式逼近:

$$\mathbf{w}_{k+1} = F_k \mathbf{w}_k + v_k + f(\hat{\mathbf{w}}_k) - F_k \hat{\mathbf{w}}_k. \quad (31)$$

设

$$\mathbf{H}(k) = \left(\prod_{\hat{s}=1}^{\hat{t}} \hat{z}_{\hat{s}}^{\hat{\lambda}(1,\hat{s})}(k), \prod_{\hat{s}=1}^{\hat{t}} \hat{z}_{\hat{s}}^{\hat{\lambda}(2,\hat{s})}(k), \dots, \prod_{\hat{s}=1}^{\hat{t}} \hat{z}_{\hat{s}}^{\hat{\lambda}(\hat{N}(\hat{t},\hat{m}),\hat{s})}(k) \right)^T \quad (32)$$

采用如下公式来更新 \hat{w}_k :

$$\hat{w}_k = \hat{w}_{k-1} + \mathbf{K}_k e_k, \quad (33)$$

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{H}_k (\mathbf{R} + \mathbf{H}_k^T \mathbf{P}_k \mathbf{H}_k)^{-1}, \quad (34)$$

$$\mathbf{P}_{k+1} = \mathbf{F}_k (\mathbf{P}_k - \mathbf{K}_k \mathbf{H}_k^T \mathbf{P}_k)^{-1} \mathbf{F}_k^T + \mathbf{Q}. \quad (35)$$

其中: $e_k = \mathbf{y}(k) - \hat{\mathbf{y}}(k)$, \mathbf{P} 为状态估计误差的协方差矩阵, \mathbf{R} 为观测噪声方差矩阵, \mathbf{Q} 为过程噪声方差矩阵, \mathbf{K} 为卡尔曼滤波增益.

4 实验仿真

非线性时滞系统广泛存在于工业领域,比如化工领域^[28],机械领域^[29]和生态领域^[30]等.像化工领域中的pH中和过程则属于SISO非线性时滞系统;连续搅拌反应釜(continuous stirred-tank reactor, CSTR)则可以看作MIMO非线性时滞系统.如何克服非线性、时滞等特性进行实时在线建模至关重要.本文以两个数值仿真例子为例,分别为SISO非线性时滞系统和MIMO非线性时滞系统来验证两种构建预测模型的准确性与实时性,同时引入神经网络预测模型作对比.

4.1 SISO非线性时滞系统

考虑如下SISO非线性时滞系统:

$$y(k) = (y(k-1)y(k-2) + u(k-d) - 0.2u(k-d) \sin(y(k-2) - y(k-1))) / (1 + y^2(k-1) + y^2(k-2)),$$

其中 $d = 5$.

在本例中,给出两种构建预测模型的方案,分别为递推 d 步超前MTN预测模型(以下简称递推MTN预测模型)和非递推 d 步超前MTN预测模型(以下简称非递推MTN预测模型),L-M算法作为递推MTN预测模型的学习算法,DRLS算法作为非递推MTN预测模型的学习算法,同时引入BP神经网络预测模型作对比.递推MTN预测模型采用4-15-1的结构,即4个输入节点,展开到2次.L-M算法阻尼系数 $\varsigma = 1$.非递推MTN预测模型采用4-35-1的结构,即4个输入节点,展开到3次.DRLS算法作为其学习算法,其初始权值设计如下: $\hat{w}(0) = 0, \mathbf{P}(0) = \delta \mathbf{I}, \delta = 150, \sigma = 0.95, \psi = 0.01$,阻尼系数为 $\xi = 0.01$.BP神经网络预测模型采用4-50-1的结构,即4个输入节点,50个隐层节点,BP算法作为BP神经网络的学习

算法,其学习率为0.5.

构建预测模型的结果如图5所示.其中: y 代表实际输出, y_{RMTN} 代表递推MTN预测模型, y_{NRMTN} 代表非递推MTN预测模型, y_{BP} 代表BP神经网络预测模型.表1为3种方案的均方误差.

需要说明的是针对多维泰勒网的两种建模方案在线初值选取为随机值,BP神经网络预测模型在线初值选取为随机值.

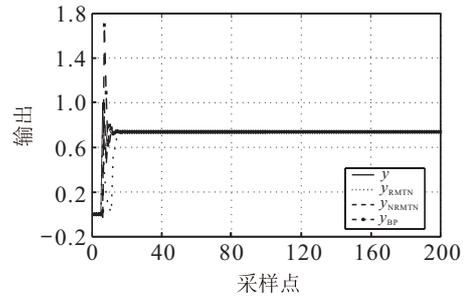


图5 预测模型构建结果

表1 均方误差

预测模型构建方案	均方误差
递推MTN预测模型	0.0107
非递推MTN预测模型	0.0070
BP神经网络预测模型	0.0121

由图5和表1可以看到,本文所提出的两种建模方案结合适当的学习算法能够准确地对SISO非线性时滞系统构建预测模型.从MTN结构选择来看,当输入节点为4个时,无论展开成2次还是3次,结合两种不同的学习算法进行在线建模,对预测模型的准确度影响不大,而算法参数的选取需满足参数的选取条件,同时根据不同背景进行不同的选取.虽然在初始点处稍有不准确,这与MTN初始权值的选取有关,本文采用随机选取的原则是造成稍有不准确的原因,但是,构建预测模型的准确性优于BP神经网络.

使用相同的计算机,相同的时间来做实时性分析是不可能的,因此本文引用浮点数来估计方案的计算复杂度,即实时性^[31-32].一个加号和一个乘号都叫作一个浮点,加号和乘号的总数作为总的浮点数,即浮点数的总和代表着计算复杂度.非递推MTN预测模型、递推MTN预测模型与BP神经网络预测模型3种方案的复杂度分析如表2所示.

表2 均方误差

预测模型构建方案	结构	加法	乘法	浮点数总和
递推MTN预测模型	4-15-1	14	25	39
非递推MTN预测模型	4-35-1	34	93	127
BP神经网络预测模型	4-50-1	149	350	499

由表2可以看出,BP神经网络预测模型的浮点数远远大于所提出方案,也就是说本文方案计算量

远远小于BP神经网络预测模型,即所提出方案具有很好的实时性.需要说明的是,MTN预测模型的结构由输入层、中间层和输出层3部分组成,在输入层和输出层节点确定的情形下,该模型结构主要体现为中间层节点数目,它们可由输入层节点及其展开次数确定.MTN的SISO形式预测模型的具体结构见图1.在该例中,对于递推MTN模型,采用4个输入节点,展开到2次,由图1可知中间层节点数为15个;对于非递推MTN模型,采用4个输入节点,展开到3次,对应的中间层节点数为35个.而BP神经网络预测模型结构选取基于多次实验,由此看到,MTN在拥有较小中间节点数量的情况下具有更好的准确性.

基于以上分析,本文所提出MTN的两种预测模型建模方案从准确性分析角度优于BP神经网络预测模型,即使从均方误差上来看三者差别不大,但从复杂度分析角度来看,MTN的建模方案拥有更好的实时性能.

4.2 MIMO非线性时滞系统

考虑如下的MIMO非线性时滞系统:

$$y_1(k) = \frac{2y_1(k-1)u_1^2(k-1)}{1+y_1^2(k-1)} + 0.3u_1(k-1-d) + 0.7u_1(k-2) + 0.2u_2(k-d) + a(k),$$

$$y_2(k) = \frac{2y_2(k-1)u_2^2(k-1)}{1+y_2^2(k-1)} + 0.3u_2(k-1-d) + 0.5u_2(k-2) + 0.2u_1(k-d) + b(k),$$

其中: $a(k) = \sin(0.5u_2(k-d))$, $b(k) = 0.96(1 - 0.5e^{-0.2y_1(k-1)})$, $d = 3$.

在本例中,给出两种构建预测模型的方案,分别为递推MTN预测模型和非递推MTN预测模型,EKF算法作为递推MTN预测模型的学习算法,带有动量因子的BP算法作为非递推MTN预测模型的学习算法,同时引入RBF神经网络预测模型作对比.递推MTN预测模型采用6-28-2的结构,即6个输入节点,展开到2次.EKF算法作为MTN预测模型的学习算法,初值设置如下: $P_1(0) = 0.95I$, $P_2(0) = 0.9I$.非递推MTN预测模型采用6-28-2的结构,即6个输入节点,展开到2次,带有动量因子的BP算法作为其学习算法.带有动量因子的BP算法的学习参数设置如下:学习因子 $\mu = (0.05, 0.05)^T$,动量因子 $\alpha = (0.5, 0.5)^T$.RBF预测模型采取6-10-2的结构,即6个输入节点,10个隐层节点,2个输出.BP算法作为RBF模型的学习算法,BP算法的学习率为0.3.RBF的在线初值经过离线学习得到,选取 $w = (0.35, 0.30, -0.03, 0.49, 0.41, 1.20, 0.50, 0.09, 1.47,$

$0.08; 0.58, -0.17, -0.75, 0.47, 0.73, -0.64, 0.52, -0.76, 0.30, 0.68)^T$.

构建预测模型的结果如图6所示.其中 y_1 和 y_2 代表实际输出, y_{1RMTN} 和 y_{2RMTN} 代表递推MTN预测模型, y_{1NRMTN} 和 y_{2NRMTN} 代表非递推MTN预测模型, y_{1RBF} 和 y_{2RBF} 代表RBF神经网络预测模型.表3为3种方案的均方误差.

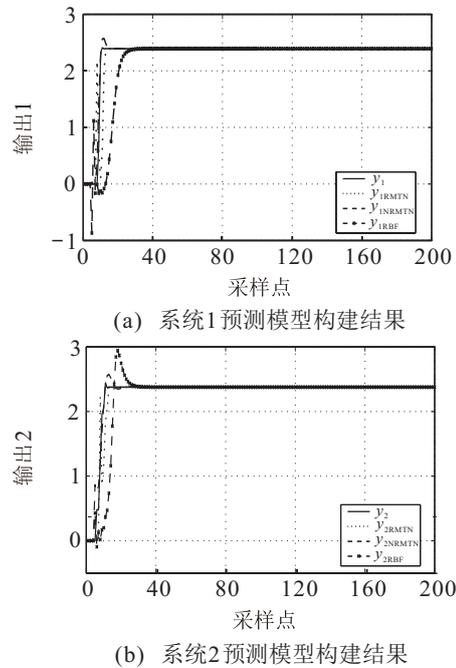


图6 预测模型构建结果

表3 均方误差

预测模型构建方案	均方误差	
	系统1	系统2
递推MTN预测模型	0.0659	0.0192
非递推MTN预测模型	0.0554	0.0272
RBF神经网络预测模型	0.2714	0.1621

需要说明的是,针对多维泰勒网的两种建模方案在线初值选取为随机值,RBF神经网络预测模型经过离线学习后得到在线学习的初值,也就是说RBF神经网络在线初值的选取优于多维泰勒网的在线初值,有利于初始状态的准确性.而相关算法参数的选取需满足参数的选取条件,同时根据不同背景进行不同的选取.

由图6和表3可以看出,本文所提出的两种基于MTN的建模方案分别结合两种不同的学习算法能够有效地对MIMO非线性时滞系统构建预测模型.同时,两种不同的学习算法对预测模型的准确度影响不大.虽然在初始点处稍有不准确,这与MTN初始权值的选取有关,本文采用随机选取的原则是造成稍有不准确的原因,但是,构建预测模型的准确性优于经过离线求取初值的RBF神经网络.

递推、非递推MTN预测模型与RBF神经网络预测模型的复杂度分析如表4所示。

表4 复杂度分析

预测模型构建方案	结构	加法	乘法	浮点数总和
递推MTN预测模型	6-28-2	54	98	152
非递推MTN预测模型	6-28-2	54	98	152
RBF神经网络预测模型	6-10-2	208	120	328

由表4可以看出,RBF神经网络预测模型的浮点数远远大于所提出方案,也就是说计算量远远小于RBF神经网络预测模型,即所提出方案具有很好的实时性能^[31-32]。需要说明的是,针对MTN预测模型,采用6个输入节点,展开到2次,由图2可知中间层节点数为28个。RBF神经网络预测模型的结构选取基于多次实验。同时看到,在RBF神经网络预测模型隐层节点数量选择较少的情况下,却拥有较高的算法复杂度,即实时性较差。

基于以上分析,从均方误差上来看,本文所提出MTN的两种预测模型建模方案的准确性明显优于RBF神经网络预测模型,同时从复杂度分析角度来看,MTN的建模方案拥有更好的实时性能。

5 结论

本文针对非线性时滞系统构建预测模型准确性问题,提出了基于MTN的预测模型构建方案。

1) 基于非递推技术与递推技术,分别设计非递推 d 步和递推 d 步超前MTN预测模型,给出了二者表达式,二者皆可对未来 d 步范围进行预测,可有效弥补时滞带来的影响。

2) 4种不同的学习算法分别作为MTN预测模型的学习算法进行实时在线学习。

3) 实验结果表明,基于MTN模型,无论应用在SISO还是MIMO非线性时滞系统,本文所提出的两种方案都能够准确地对非线性时滞系统实时构建预测模型。同时,4种不同的学习算法对MTN预测模型的准确度影响不大。此外,所建预测模型的准确性与实时性优于神经网络。

4) MTN预测模型的准确性与实时性可为解决后续非线性时滞系统的控制问题做铺垫。

参考文献(References)

[1] 樊兆峰, 马小平, 邵晓根. 非线性系统RBF神经网络多步预测控制[J]. 控制与决策, 2014, 29(7): 1274-1278.
(Fan Z F, Ma X P, Shao X G. RBF neural network multi-step predictive control for nonlinear systems[J]. Control and Decision, 2014, 29(7): 1274-1278.)

[2] Yang Z J, Iemura H, Kanae S, et al. Identification

of continuous-time systems with multiple unknown time delays by global nonlinear least-squares and instrumental variable methods[J]. Automatica, 2007, 43(7): 1257-1264.

- [3] Wang Q G, Guo X, Zhang Y. Direct identification of continuous time delay systems from step responses[J]. Journal of Process Control, 2001, 11(5): 531-542.
- [4] 刘金琨, 沈晓蓉, 赵龙. 系统辨识理论及Matlab仿真[M]. 北京: 电子工业出版社, 2013.
(Liu J K, Shen X R, Zhao L. System identification theory and Matlab simulation[M]. Beijing: Publishing House of Electronics Industry, 2013.)
- [5] Wang H O, Tanaka K, Griffin M. Parallel distributed compensation of nonlinear systems by Takagi-Sugeno fuzzy model[C]. Proceedings of 1995 IEEE International Conference on Fuzzy Systems. Yokohama, 1995: 531-538.
- [6] Narendra K S, Gallmax P G. An iterative method for the identification of nonlinear systems using a Hammerstein model[J]. IEEE Transactions on Automatic Control, 1966, 11(3): 546-550.
- [7] Chen S, Billings S A. Neural networks for nonlinear dynamic system modelling and identification[J]. International Journal of Control, 1992, 56(2): 319-346.
- [8] Abonyi J, Babuska R, Szeifert F. Modified gath-geva fuzzy clustering for identification of Takagi-Sugeno fuzzy models[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2002, 32(5): 612-621.
- [9] Ding F. Hierarchical multi-innovation stochastic gradient algorithm for Hammerstein nonlinear system modeling[J]. Applied Mathematical Modelling, 2013, 37(4): 1694-1704.
- [10] Rummelhart D E, Hinton G E, Williams R J. Learning representations by back propagating errors[J]. Nature, 1986, 323(6088): 533-536.
- [11] Sokoart P O M, Mayne D Q. Min-max feedback model predictive control for constrained linear systems[J]. IEEE Transactions on Automatic Control, 1998, 43(8): 1136-1142.
- [12] Tan Y H, van Cauwenberghe A. Nonlinear one-step-ahead control using neural networks, control strategy and stability design[J]. Automatica, 1996, 32(12): 1701-1706.
- [13] Tan Y H, van Cauwenberghe A. Neural-network-based d-step-ahead predictors for nonlinear systems with time delay[J]. Engineering Applications of Artificial Intelligence, 1999, 12(1): 21-35.
- [14] Han H G, Zhang L, Ying Hou, et al. Nonlinear model predictive control based on a self-organizing recurrent neural network[J]. IEEE Transactions on Neural

- Networks and Learning Systems, 2016, 27(2): 402-415.
- [15] 严洪森. 多维泰勒网优化控制[R]. 南京: 东南大学自动化学院制造系统控制与优化研究所, 2010.
(Yan H S. MTN(multi-dimensional Taylor network) optimal control[R]. Nanjing: Research Institute of Control and Optimization of Manufacturing Systems, School of Automation, Southeast University, 2010.)
- [16] 周博, 严洪森. 基于小波和多维泰勒网动力学模型的金融时间序列预测[J]. 系统工程理论与实践, 2013, 33(10): 2654-2662.
(Zhou B, Yan H S. Financial time series forecasting based on wavelet and multi-dimensional Taylor network dynamics model[J]. Systems Engineering — Theory & Practice, 2013, 33(10): 2654-2662.)
- [17] 林屹, 严洪森, 周博. 基于多维泰勒网的非线性时间序列预测方法及其应用[J]. 控制与决策, 2014, 29(5): 795-801.
(Lin Y, Yan H S, Zhou B. Nonlinear time series prediction method based on multi-dimensional Taylor network and its applications[J]. Control and Decision, 2014, 29(5): 795-801.)
- [18] Klambauer G. Mathematical analysis[M]. New York: Marcel Dekker INC, 1975: 236-237.
- [19] Yu D L, Gomm J B. Implementation of neural network predictive control to a multivariable chemical reactor[J]. Control Engineering Practice, 2003, 11: 1315-1323.
- [20] Cowan C F N, Grant P M. Adaptive filters[M]. New Jersey: Prentice-Hall, Inc., Englewood Cliffs, 1985.
- [21] Meiron J. Damped least-squares method for automatic lens design[J]. Journal of the Optical Society of America, 1965, 35(9): 1105-1109.
- [22] Chen J, Fossorier M P C. Density evolution for two improved BP-Based decoding algorithms of LDPC codes[J]. IEEE Communications Letters, 2002, 6(5): 208-210.
- [23] Yu C C, Liu B D. A backpropagation algorithm with adaptive learning rate and momentum coefficient[C]. Proceedings of the 2002 International Joint Conference on Neural Networks. Honolulu, 2002: 1218-1223.
- [24] Lera G, Pinzolas M. Neighborhood based Levenberg-Marquardt algorithm for neural network training[J]. IEEE Transactions on Neural Networks, 2002, 13(5): 1200-1203.
- [25] Cui M, Yang K, Xu X L, et al. A modified Levenberg-Marquardt algorithm for simultaneous estimation of multi-parameters of boundary heat flux by solving transient nonlinear inverse heat conduction problems[J]. International Journal of Heat and Mass Transfer, 2016, 97: 908-916.
- [26] Moore T, Stouch D. A generalized extended Kalman filter implementation for the robot operating system[J]. Intelligent Autonomous Systems, 2016, 13: 335-348.
- [27] Liu S, Wei G L, Song Y, et al. Extended Kalman filtering for stochastic nonlinear systems with randomly occurring cyber attacks[J]. Neurocomputing, 2016, 207(26): 708-716.
- [28] Liu L, Yin S, Zhang L, et al. Improved results on asymptotic stabilization for stochastic nonlinear time-delay systems with application to a chemical reactor system[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2017, 47(1): 195-204.
- [29] Xing L, Wen C, Liu Z, et al. Event-triggered adaptive control for a class of uncertain nonlinear systems[J]. IEEE Transactions on Automatic Control, 2017, 62(4): 2071-2076.
- [30] Chakraborty K. Ecological complexity and feedback control in a prey-predator system with Holling type III functional response[J]. Complexity, 2016, 21(5): 346-360.
- [31] 丁锋. 系统辨识算法的复杂性、收敛性及计算效率研究[J]. 控制与决策, 2016, 31(10): 1729-1741.
(Ding F. Complexity, convergence and computational efficiency for system identification algorithms[J]. Control and Decision, 2016, 31(10): 1729-1741.)
- [32] Golub G H, Van Loan C F. Matrix computations[M]. Baltimore: JHU Press, 2012.

作者简介

李晨龙(1988—), 男, 博士生, 从事系统辨识、预测控制的研究, E-mail: lichenlong007@126.com.

严洪森(1957—), 男, 教授, 博士生导师, 从事知识化制造、多维泰勒网优化控制等研究, E-mail: hsyang@seu.edu.cn.

(责任编辑: 孙艺红)