

基于滑膜控制和 SAFNN 的鲁棒混沌同步

边潇俊, 林 达, 谢玉姣

(四川理工学院自动化与电子信息学院, 四川 自贡 643000)

摘 要:针对带有干扰的混沌系统,进行了基于自组织自适应模糊神经网络(SAFNN)的同步控制器的设计,采用基于聚类划分的模糊神经网络,推导出模糊神经网络参数在线更新算法,并设计了神经网络结构自适应算法,从隶属度函数的生长、模糊规则的分裂和模糊规则的修剪三个方面动态调整模糊神经网络结构,摒弃了常规的试错法来调整系统的结构和参数,以便更好地满足对计算负荷和控制性能的要求,实现两者之间的平衡;最后结合滑模控制器和鲁棒控制器,对带有干扰的混沌系统进行同步控制仿真,仿真结果表明该方法有效。

关键词:自适应控制;模糊神经网络(FNN);滑模控制;模糊规则;混沌同步

中图分类号:TP183

文献标志码:A

引 言

由于大多数实际混沌系统具有内在的未知非线性或参数不确定性,很难进行精确的数学模型描述,传统的控制方法已无法对其进行有效的控制,因此,出现了以模糊神经网络为基础的智能控制技术^[1]。

模糊神经网络(FNN)具有模糊逻辑和神经网络的优点,并能克服各自的缺点,其已成为当前研究的热点。FNN拥有神经网络的低层次学习和计算能力以及来自模糊理论的高层次人类知识表达,可以很容易地解决传统的模糊推理中存在的两个问题:(1)隶属度函数的设计;(2)建模过程的自适应性。文献[2]就基于模糊神经网络算法解决了机器人路径规划问题。

在以往的FNN中,网络的结构、参数以及模糊规则在缺少先验知识的时候很难获得,总是要通过大量的试验才能得到比较满意的结果。另外,模糊规则的数目同样影响系统的辨识能力和精度,规则数目过多,计算过

程就会变得非常复杂,不适合运用到实际运用中;规则数目过少,网络学习效果可能又满足不了预期的要求。为了解决FNN的结构难以确定、需要大量的试错法的问题,专家们提出了大量方法。

文献[3]设计了一种基于粒子群优化算法和聚类算法的神经网络优化算法;文献[4]提出了一种自组织RBF神经网络,动态地解决了神经网络的结构问题;文献[5]提出了一种基于模糊系统的自适应控制,解决了未知混沌系统的控制问题;文献[6]结合滑模控制,提出了一种模糊神经网络控制器,取得了较好的效果;文献[7]提出了一种基于鲁棒控制和滑模控制的模糊神经网络控制器,解决了网络的结构自适应问题;文献[8]提出了一种自组织自适应的模糊神经网络控制,很好地解决了非线性系统的控制问题。但是,它们都没有考虑噪声的影响,在实际应用中,干扰总是存在的而且无法避免的。

本文提出了一种基于自组织自适应模糊神经网络(SAFNN)的混沌同步方法。首先基于滑模控制方法,证

收稿日期:2015-07-06

基金项目:四川省省属高校科研创新团队(TD15024)

作者简介:边潇俊(1991-),男,湖北钟祥人,硕士生,主要从事智能控制方面的研究,(E-mail)397897320@.com;

林 达(1974-),男,山东日照人,副教授,博士,主要从事混沌控制方面的研究,(E-mail)linda_740609@aliyun.com

明了系统基于 Lyapunov 稳定性理论的稳定性,然后采用基于聚类划分的模糊神经网络,选取高斯函数作为模糊系统隶属度函数,设计了网络结构自适应算法,从隶属度函数的生长、模糊规则的分裂和模糊规则的修剪三个方面动态调整模糊神经网络结构,并且结合控制方案导出模糊神经网络参数在线学习算法,最后对带有噪声的混沌系统进行同步仿真,得到了良好的效果。

1 滑模控制器的设计

考虑下面一种 n 阶混沌系统:

$$\dot{x}^{(n)} = f(x) + u \quad (1)$$

其中, $x = [x, \dot{x}, \dots, x^{(n-1)}]^T$ 是系统的状态向量, $f(x)$ 是一个连续非线性函数, u 是系统的控制输入。系统的目标是设计一种控制律使得状态变量按照设定的轨迹运动,定义系统的跟踪误差:

$$e = x - x_c \quad (2)$$

其中, $x_c = [x_c, \dot{x}_c, \dots, x_c^{(n-1)}]^T$ 是参考轨迹。定义滑模面:

$$s = e^{(n-1)} + k_n e^{(n-2)} + \dots + k_2 e + k_1 \int_0^t e d\tau \quad (3)$$

其中, $k_i (i = 1, 2, \dots, n)$ 是正的常数。理想的滑模控制器设置:

$$u_{sc} = u_{eq} + u_{ht} \quad (4)$$

其中, u_{eq} 是等效控制器, u_{ht} 是切换控制器。 u_{eq} 设置为:

$$u_{eq} = -f(x) + \dot{x}_c^{(n)} - K^T E \quad (5)$$

其中, $E = [e, \dot{e}, \dots, e^{(n-1)}]^T$, $K = [k_1, k_2, \dots, k_n]^T$, 切换控制器 u_{ht} 的设计是为了保证系统的稳定, 设置为:

$$u_{ht} = -\rho \operatorname{sgn}(s) \quad (6)$$

其中, $\operatorname{sgn}(\cdot)$ 是符号函数。式(4)~式(6)带入(1)式, 得:

$$e^{(n)} + k_n e^{(n-1)} + \dots + k_2 \dot{e} + k_1 e = -\rho \operatorname{sgn}(s) = \dot{s} \quad (7)$$

滑模控制的作用是让状态上滑模面和状态上了滑模面后补偿干扰,使系统稳定。引入 Lyapunov 函数:

$$V_1 = \frac{1}{2} s^2 \quad (8)$$

通过式(8)两边对时间求微分,并联合式(7),有:

$$\dot{V}_1 = s \dot{s} = -\rho s \leq 0 \quad (9)$$

这样,滑模控制器的设计就保证了系统在 Lyapunov 条件下稳定^[9]。

2 SAFNN 的设计

常见的 FNN 具有 4 层神经元:输入层、隶属度函数层、规则层、输出层,如图 1 所示。该网络中的所有节点都与模糊系统的隶属度函数以及推理过程对应:

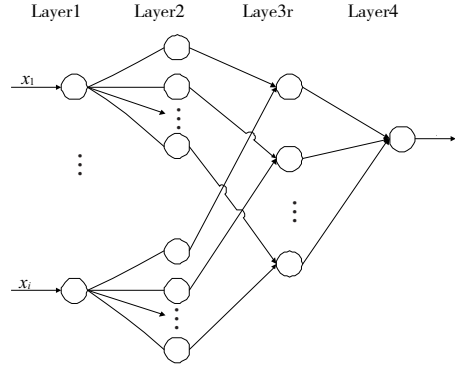


图 1 神经网络结构图

Layer 1: 这一层的各个节点与输入向量的各分量 x_i 连接,作用是把输入值 $x = [x_1, x_2, \dots, x_n]^T$ 传递到下一层。其输入输出分别为:

$$net_i^1 = x_i^1 \quad (10)$$

$$y_i^1 = f_i^1(net_i^1) = net_i^1 (i = 1, 2, \dots, L) \quad (11)$$

其中, x_i^1 表示第 1 层的第 i 个节点的输入, L 为输入变量的个数。

Layer 2: 每个节点表示一个隶属度函数,本文采用高斯函数作为该网络的隶属度函数,对于第 i 个输入变量,其在第 j 个节点的输入输出为:

$$net_{ij}^2 = -\frac{(x_i^2 - m_{ij}^2)^2}{(\sigma_{ij}^2)^2} \quad (12)$$

$$y_{ij}^2 = f_{ij}^2(net_{ij}^2) = \exp(net_{ij}^2) (j = 1, 2, \dots, M) \quad (13)$$

其中, m_{ij}^2 是 x_i^2 的平均值, σ_{ij}^2 是 x_i^2 的标准差, M 是对每个输入变量 x_i^2 对应的隶属度函数的个数。

Layer 3: 采用乘积推理机进行模糊推理,从而产生模糊规则,对于其第 k 条规则,有:

$$net_k^3 = x_1^3 x_2^3 \cdots x_i^3 (i = 1, 2, \dots, L) \quad (14)$$

$$y_k^3 = f_k^3(net_k^3) = \frac{net_k^3}{\sum_k net_k^3} (k = 1, 2, \dots, N) \quad (15)$$

其中, x_i^3 表示第 3 层上第 k 个节点的输入, L 为系统输入变量的数目, N 为模糊规则的总数目。

Layer 4: 对第 3 层的输出进行线性相加,从而得到整个网络的输出,其关系如下:

$$net_o^4 = \sum_k \omega_k^4 x_k^4 (k = 1, 2, \dots, N) \quad (16)$$

$$y_o^4 = f_o^4(net_o^4) = net_o^4 \quad (17)$$

其中, ω_k^4 是第3层节点与第4层节点的连接权值, x_k^4 表示第4层网络的第 k 个输入值, y_o^4 表示整个网络的输出。

但是这种网络的结构和参数必须经过多次的调试才有可能满足系统的需求, 会浪费大量时间。为了避免这种情况发生, 本文采用:

(1) 在第2层网络中, 使用基于聚类划分的网络结构, 当一个新的输入信号距离已有的聚类中心很远的情况下, 就增加一个新的节点。

首先, 设置一个触发强度为 β_k , 满足:

$$\beta_k = y_k^3 (k = 1, 2, \dots, N) \quad (18)$$

其中, N 为现有的模糊规则数目。于是, 对于每一个新的输入 x_i^1 , 就有 N 个触发强度 β_k 与之对应, 当 x_i^1 离所有的聚类中心越远, 就表示触发强度 β_k 越小, 当输入对应的所有的 β_k 均小于一定程度时, 就在第二层网络中新增一个节点, 从而增加隶属度函数的个数。设置 x_i^1 对应的所有触发强度 β_k 中最大的一项为 β_{\max} :

$$\beta_{\max} = \max_{1 \leq k \leq N} \beta_k \quad (19)$$

然后, 规定门限值为 G_{th} , 且 $G_{th} \in (0, 1)$, 当 $\beta_{\max} \leq G_{th}$ 时, 就表示当前的输入变量 x_i^1 远离现有的聚类中心, 处在现有的隶属度函数的边缘位置。于是, 就建立一个新的节点。新节点的隶属度函数的参数和权值设置如下:

$$m_i^{new} = x_i^1 \quad (20)$$

$$\sigma_i^{new} = 1 \quad (21)$$

$$\omega^{new} = 0 \quad (22)$$

其中, x_i^1 表示新的输入变量。

(2) 在第3层网络中进行模糊规则的分裂和修剪, 其中分裂规则的目的是提高系统精度, 修剪规则的目的是防止系统规则过多造成系统过于复杂。

$$\frac{|\Delta\omega_k^4|}{\sum_{k=1}^N |\Delta\omega_k^4|} \geq S_{th} (k = 1, 2, \dots, N) \quad (23)$$

其中, S_{th} 是满足分裂规则条件的门限值, $\Delta\omega_k^4$ 第3层节点和第4层节点连接权值 ω_k^4 的变化量。当式(23)表示的条件满足时, 就分裂规则。当权值的变化量过大时,

表明该网络在逼近理想状态时的精度不够高, 为了解决这个问题, 就把该节点分裂成两个, 从而分裂规则。其中, 新建的节点与第4层网络连接权值及第2层神经元与之连接的参数定义如下:

$$\omega_k^4 = (1 - \lambda)\omega_k^4 \quad (24)$$

$$m_k = m_k \quad (25)$$

$$\sigma_k = \sigma_k \quad (26)$$

其中, α 是一个 ($0 \sim 1$) 的常数。另一个节点的权值则变为:

$$\omega_k^4 = \lambda\omega_k^4 \quad (27)$$

通过分裂第3层网络的节点数目来分裂模糊规则, 使得权值的变化量不至于过大而造成系统精度降低。当系统的参数均达到最优时, 变化量 $\Delta\omega_k^4$ 就会降为0。这样, 模糊规则就不会再进行分裂, 系统的结构就趋于稳定。

但是当分裂的规则过多时, 系统的结构和计算量会越来越复杂, 这时, 就需要删除无用规则, 以简化系统的结构。本文通过判断权值的大小来决定是否删除规则, 当第3层网络的节点与第4层网络的节点连接权值 ω_k^4 过小时, 表明这条规则对于系统的输出影响甚微, 就可以删除这条规则。

3 参数自调整

在实际应用中, 由于混沌系统的 $f(x)$ 是未知的, 而且有可能存在外界干扰, 那么式(1)可以写为:

$$x^{(n)} = F(x) + u \quad (28)$$

其中, $F(x) = f(x) + d$, d 为外界干扰。

本文中设计的控制器是由一个运算控制器和一个鲁棒控制器组成:

$$u = u_{cc} + u_{rc} \quad (29)$$

其中, 运算控制器设置为:

$$u_{cc} = -\hat{F}(X) + x_c^{(n)} + K^T E \quad (30)$$

其中, $\hat{F}(X)$ 为 SAFNN 辨识器的在线预测 $F(x)$ 的结果, $E = [e, \dot{e}, \dots, e^{(n-1)}]^T$ 是 SAFNN 辨识器的输入向量, δ 是一个正的常数。在滑模控制方法中, 系统误差会随着时间的推移逐渐减小至0, 状态到滑模面 s 的距离就是系统的跟踪误差, 当状态完全在滑模面上时, 就表明系统跟踪误差为0。一般把系统的误差函数设置为代价函数, 本文把到达滑模面的条件作为代价函数。本文

中,参数的在线学习方法采用梯度下降法使得 ss 快速收敛到滑模面 s 上。

联立(7)式、(29)式、(30)式,可得:

$$e^{(n)} + k_n e^{(n-1)} + \dots + k_2 \dot{e} + k_1 e = \hat{F}(X) - F(x) - u_{rc} = \dot{s} \quad (31)$$

两边同乘以 s ,有:

$$ss = (\hat{F}(X) - F(x) - u_{rc})s \quad (32)$$

采用梯度下降法调整系统参数,第 3 层网络与第 4 层网络之间的权值变化率:

$$\Delta\omega_k = -\eta_\omega \frac{\partial ss}{\partial(\omega_k^4)} = -\eta_\omega \frac{\partial ss}{\partial(\hat{F})} \frac{\partial(\hat{F})}{\partial(\omega_k^4)} = -\eta_\omega * s * x_k^4 \quad (33)$$

定义一个参数 ξ 为:

$$\xi = \frac{\partial ss}{\partial \hat{F}} \frac{\partial \hat{F}}{\partial(\text{net}_o^4)} \frac{\partial(\text{net}_o^4)}{\partial(y_k^3)} \frac{\partial(y_k^3)}{\partial(\text{net}_k^3)} \frac{\partial(\text{net}_k^3)}{\partial(y_{ij}^2)} \frac{\partial(y_{ij}^2)}{\partial(\text{net}_{ij}^2)} = s * \omega_k^4 * y_k^3 \quad (34)$$

网络的参数平均值 m_{ij} 与方差 σ_{ij} 的修正量 Δm_{ij} 与 $\Delta\sigma_{ij}$ 为:

$$\Delta m_{ij} = -\eta_m \frac{\partial ss}{\partial(m_{ij})} = -\eta_m \frac{\partial ss}{\partial(\hat{F})} \frac{\partial(\hat{F})}{\partial(\text{net}_o^4)} \frac{\partial(\text{net}_o^4)}{\partial(y_k^3)} \frac{\partial(\text{net}_k^3)}{\partial(y_{ij}^2)} * \frac{\partial(y_{ij}^2)}{\partial(\text{net}_{ij}^2)} \frac{\partial(\text{net}_{ij}^2)}{\partial(m_{ij}^2)} = -\eta_m \xi \frac{2(x_i^2 - m_{ij}^2)}{(\sigma_{ij}^2)^2} \quad (35)$$

$$\Delta\sigma_{ij} = -\eta_\sigma \frac{\partial ss}{\partial(\sigma_{ij})} = -\eta_\sigma \frac{\partial ss}{\partial(\hat{F})} \frac{\partial(\hat{F})}{\partial(\text{net}_o^4)} \frac{\partial(\text{net}_o^4)}{\partial(y_k^3)} \frac{\partial(\text{net}_k^3)}{\partial(y_{ij}^2)} * \frac{\partial(y_{ij}^2)}{\partial(\text{net}_{ij}^2)} \frac{\partial(\text{net}_{ij}^2)}{\partial(\sigma_{ij}^2)} = -\eta_\sigma \xi \frac{2(x_i^2 - m_{ij}^2)^2}{(\sigma_{ij}^2)^3} \quad (36)$$

为了加快收敛速度,在加权系数修正公式中增加了一个惯性项,使得加权系数的变化更加平稳。修正后的参数为:

$$\omega_k^4(n+1) = \omega_k^4(n) + \Delta\omega_k^4(n) + \alpha[\omega_k^4(n) - \omega_k^4(n-1)] \quad (37)$$

$$m_{ij}^4(n+1) = m_{ij}^4(n) + \Delta m_{ij}^4(n) + \alpha[m_{ij}^4(n) - m_{ij}^4(n-1)] \quad (38)$$

$$\sigma_{ij}^4(n+1) = \sigma_{ij}^4(n) + \Delta\sigma_{ij}^4(n) + \alpha[\sigma_{ij}^4(n) - \sigma_{ij}^4(n-1)] \quad (39)$$

其中, α 为惯性系数, $\alpha \in (0 \sim 1)$ 。

整个系统的结构图如图 2 所示:

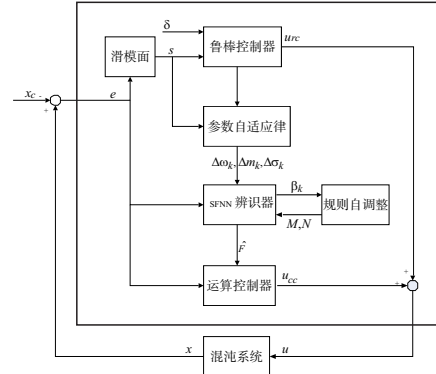


图 2 系统结构图

4 鲁棒控制器的设计

神经网络已经被证明具有任意精度下的非线性函数逼近能力,定义逼近误差为:

$$F(X) - \hat{F}(X) = \varepsilon \quad (40)$$

联立(33)式和(40)式,得:

$$\dot{s} = -\varepsilon - u_{rc} \quad (41)$$

如果 ε 存在,设计一个鲁棒控制器,能够使控制器满足规定的 L_2 跟踪性能:

$$\int_0^T s^2(t) dt \leq s^2(0) + \delta^2 \int_0^T \varepsilon^2(t) dt \quad (42)$$

其中, δ 是一个规定的衰减常数,如果系统开始的初始条件为 $s(0) = 0$, 那么,(42)式可写为:

$$\sup_{\varepsilon \in L_2[0, T]} \frac{\|s\|}{\|\varepsilon\|} \leq \delta \quad (43)$$

其中 $\|s\|^2 = \int_0^T s^2(t) dt$, $\|\varepsilon\|^2 = \int_0^T \varepsilon^2(t) dt$, 当 $\delta = \infty$ 时,表明系统就是在没有外界干扰的情况下有最小的跟踪误差,下面定义一个 Lyapunov 函数来证明系统的稳定性:

$$V_2 = \frac{1}{2}s^2 \quad (44)$$

对(44)式两边进行时间求微分,有:

$$\dot{V}_2 = s\dot{s} = s(-\varepsilon - u_{rc}) \quad (45)$$

本文鲁棒控制器设计为:

$$u_{rc} = \frac{\delta^2 + 1}{2\delta^2}s \quad (46)$$

(45)式可写为:

$$\dot{V}_2 = s(-\varepsilon - \frac{\delta^2 + 1}{2\delta^2}s) =$$

$$\begin{aligned}
 & -s\varepsilon - \frac{s^2}{2} - \frac{s^2}{2\delta^2} = -\frac{s^2}{2} - \frac{1}{2}(2s\varepsilon + \frac{s^2}{\delta^2}) = \\
 & -\frac{s^2}{2} - \frac{1}{2}(\frac{s^2}{\delta^2} + 2s\varepsilon + \varepsilon^2\delta^2) + \frac{1}{2}\varepsilon^2\delta^2 = \\
 & -\frac{s^2}{2} - \frac{1}{2}(\frac{s}{\delta} + \varepsilon\delta)^2 + \frac{1}{2}\varepsilon^2\delta^2 \leq \\
 & -\frac{s^2}{2} + \frac{1}{2}\varepsilon^2\delta^2 \tag{47}
 \end{aligned}$$

假设 $\varepsilon \in L_2[0, T], \forall T \in (0, \infty)$, 对(47)式在 $t \in (0, \infty)$ 的范围内求积分, 有:

$$V_2(T) - V_2(0) \leq -\frac{1}{2}\int_0^T s^2 dt + \frac{1}{2}\delta^2\int_0^T \varepsilon^2 dt \tag{48}$$

因为 $V_2(T) \geq 0$, 则(48)式可写为:

$$\frac{1}{2}\int_0^T s^2 dt \leq V_2(0) + \frac{1}{2}\delta^2\int_0^T \varepsilon^2 dt \tag{49}$$

不等式(49)式满足不等式(42)式的条件。因为 $V_2(0)$ 是有限值, 当逼近误差 $\varepsilon \in L_2$ 时, 那么 $\int_0^T \varepsilon^2(t) dt < \infty$, 可以推断出 $\lim_{t \rightarrow \infty} |s| = 0$, 由此证明了本文所设计的控制器能够在外界有干扰的情况下获得 L_2 跟踪性能。

5 仿真结果

本文采用一个二阶 Duffing 方程进行同步仿真, Duffing 方程是一种典型的混沌系统:

$$\ddot{x} = f(x) + u \tag{50}$$

其中, $f(x) = -px - p_1x - p_2x^3 + q\cos(\omega t)$ 是系统动力学方程, t 是时间变量, ω 是频率, u 是系统的控制输入, p, p_1, p_2, q 是 Duffing 系统的参数。当 u 为零的时候, Duffing 方程的混沌现象如图 3 所示, 其中 $p = 0.4, p_1 = -1.1, p_2 = 1.0, \omega = 1.8, q = 1.95$ 。

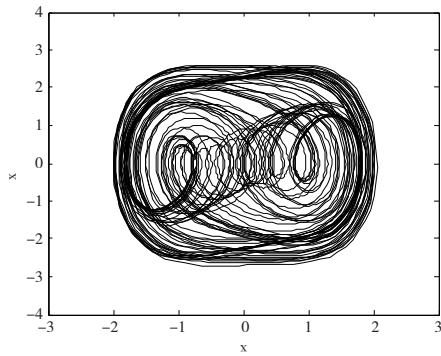


图3 Duffing 方程相图

本文的各参数设置为: $k_1 = 1, k_2 = 2, \eta_\omega = \eta_m = \eta_\sigma = 0.1, \alpha = 0.01, \lambda = 0.7, \delta = 0.5, G_{th} = 0.5,$

$S_{th} = 0.1$, 系统的初始状态为 $(-2, 1)$, 仿真结果如图 4 ~ 图 9 所示。其中, 图 4 表示状态 x 的同步过程以及跟踪误差 $e1, x_c$ 是 x 的响应, 图 5 表示状态 y 的同步过程以及跟踪误差 $e2$, 其中, $y = \dot{x}, y_c$ 是 y 的响应, 图 6 ~ 图 7 分别表示未使用变结构的控制器对应的 $x1, y1$ 的同步过程及误差, 图 8 表示模糊规则数目的变化情况, 图 9 表示施加的高斯噪声。

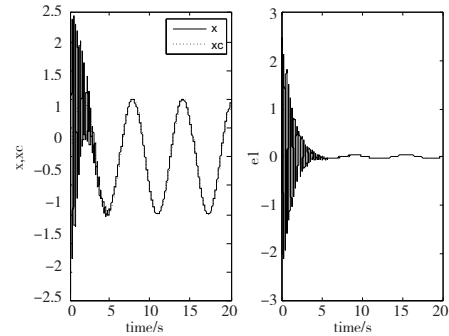


图4 状态 x 的同步和跟踪误差

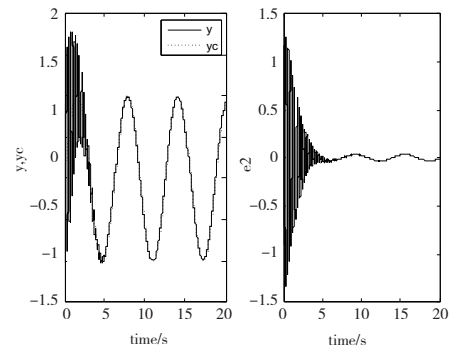


图5 状态 y 的同步和跟踪误差

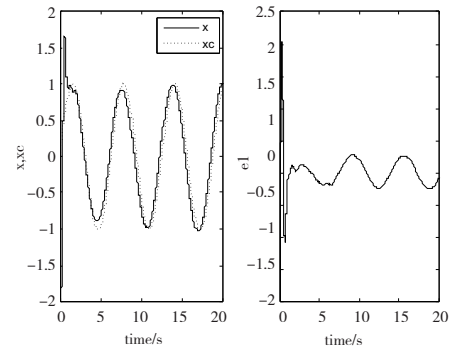


图6 状态 x1 的同步和跟踪误差

从仿真结果可知, 使用本文所设计的 SAFNN 控制器对带有高斯噪声的混沌系统进行同步仿真的效果要明显优于未使用变结构的 FNN 控制器, 其误差更小、鲁棒性更强、同步效果也更优, 体现了该方法的优越性。

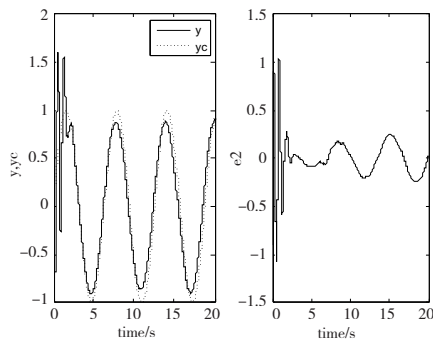
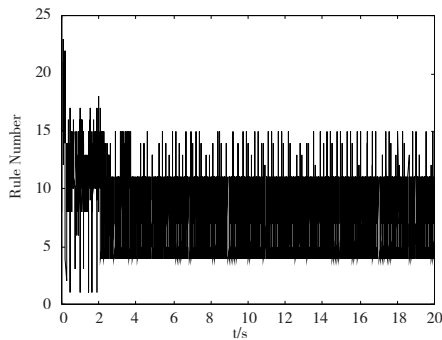
图 7 状态 y_1 的同步和跟踪误差

图 8 模糊规则数

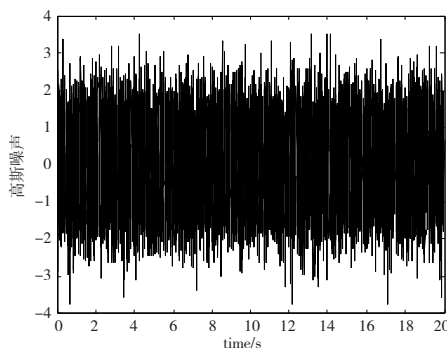


图 9 高斯噪声

6 结束语

本文以一个二阶混沌系统作为研究对象,利用模糊神经网络(FNN)对该混沌系统进行建模,首先设计一个滑模控制器以保证系统基于 Lyapunov 理论的稳定性,然后设计了神经网络结构自适应算法,从隶属度函数的生长、模糊规则的分裂和模糊规则的修剪等三个方面动态调整模糊神经网络结构,并推导出模糊神经网络参数在线更新算法,完成了该网络的自组织自适应性,满足了对计算负荷和控制性能的要求,实现两者之间的平衡。然后结合鲁棒控制器,对带有高斯噪声的 Duffing 混沌系统进行了同步仿真试验,仿真结果

表明该方法具有一定的优越性。

参考文献:

- [1] 杨东升,赵琰,刘鑫蕊.混沌系统模糊建模、同步与控制[M].北京:科学出版社,2012.
- [2] 姚毅,陈光建,贾金玲.基于模糊神经网络算法的机器人路径规划研究[J].四川理工学院学报:自然科学版,2014,27(6):30-33.
- [3] Lin C J,Chen C H,Lin C T.A hybrid of cooperative Particle Swarm Optimization and cultural algorithm for Neural Fuzzy Networks and its prediction applications [J]. IEEE TRANSACTIONS ON SYSTEMS,2009,39(1): 55-68.
- [4] Han H G,Qiao J F.A self-organizing Fuzzy-Neural Network based on a Growing-and-Pruning algorithm[J]. IEEE Transactions on Fuzzy Systems,2010,18(6):1129-1143.
- [5] Chen L,Chen G R,Lee Y W.Fuzzy modeling and adaptive control of uncertain chaotic systems[J].Information Sciences,1999,121(1-2):27-37.
- [6] Mohseni S A,Mohseni S A.Fuzzy neural networks controller for a chaotic nonlinear gyro using sliding-mode surfaces [C]//Proceeding of 2008 IEEE International Symposium on Industrial Electronics,Cambridge,June 30-July 2,2008:1150-1153.
- [7] Lin P Z,Hsu C F,Lee T T,et al.Robust fuzzy-neural sliding-mode controller design via network structure adaptation[J].IET Control Theory,2008,2(12):1054-1065.
- [8] Hus C F.Self-Organizing adaptive Fuzzy Neural Control for a class of Nonlinear Systems[J].IEEE TRANSACTIONS ON NEURAL NETWORKS,2007,18(4):1232-1241.
- [9] Hassan K,著.非线性系统[M].3版.朱义胜,译.北京:电子工业出版社,2005.
- [10] Kalpana M,Balasubramaniam P,Ratnavelu K.Direct delay decomposition approach to synchronization of chaotic fuzzy cellular neural networks with discrete, unbounded distributed delays and Markovian jumping

- parameters[J]. Applied Mathematics and Computation, 2015, 254: 291-304.
- [11] Lin Da, Liu Hongjun, Song Hong, et al. Fuzzy neural control of uncertain chaotic systems with backlash nonlinearity[J]. International Journal of Machine Learning and Cybernetics, 2014, 5(5): 721-728.
- [12] Lin Da, Wang Xingyuan, Zhang Fuchen, et al. Mutual synchronization behavior for Chaotic Systems via limited capacity communication channels[J]. Complexity, 2015, DOI:10.1002/cplx.21692.

Robust Chaotic Synchronization Based on Sliding-mode Control and SAFNN

BIAN Xiaojun, LIN Da, XIE Yujiao

(School of Automation and Electronic Information, Sichuan University of Science & Engineering, Zigong 643000, China)

Abstract: A synchronous controller design based on self-organizing adaptive fuzzy neural network (SAFNN) for chaotic systems with disturbance is proposed. The fuzzy neural network based on clustering division is used, the parameters online updating algorithm is deduced, and the structure adaptive algorithm of neural network is designed. The structure of fuzzy neural network is dynamically adjusted from the growing of membership functions, the splitting of fuzzy rules and the pruning of fuzzy rules, which can avoid the conventional trial-and-error method for adjusting the network structure and parameters. SAFNN can make the balance between computational load and the control performance so as to preferably meet the requirements. Then combined sliding-mode controller and robust controller, synchronous control simulations are conducted on chaotic systems with disturbance, and simulation results show that SAFNN is effective.

Key words: adaptive control; fuzzy neural network (FNN); sliding-mode control; fuzzy rules; chaotic synchronization