

# 一种无约束多步递归神经网络预测控制器

李会军<sup>1</sup>, 肖兵<sup>2</sup>

(1. 中国矿业大学信息与电气工程学院, 江苏徐州 221116; 2. 华北油田采油工艺研究院, 河北任丘 062252)

**摘要:** 本文针对具有强非线性、多工作点特性的控制系统, 提出了一种基于递归BP神经网络的多步预测模型; 通过分析预测模型的内在数学关系, 选择了二次型函数作为预测控制器的目标函数, 并给出了目标函数关于控制序列的雅可比矩阵和赫森矩阵的计算方法; 最后使用Newton-Rhapon算法设计出了滚动优化控制策略, 构建了一个非线性多步预测控制器. 仿真结果表明, 文中提出的多步预测控制器具有较好的控制效果.

**关键词:** 模型预测控制; 神经网络; 非线性自回归滑动平均模型(NARMAX); 优化

**中图分类号:** TP273 **文献标识码:** A

## Multistep recurrent neural network model predictive controller without constraints

LI Hui-jun<sup>1</sup>, XIAO Bing<sup>2</sup>

(1. School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou Jiangsu 221116, China;

2. Oil Production Technology Institute, Huabei Oilfield Company, Renqiu Hebei 062252, China)

**Abstract:** This paper brings forward a multistep predictive model based on the recurrent backpropagation (BP) neural network for the control systems with strong nonlinearity and multiple set-points. By analyzing the internal mathematical relation of the predictive model, we select a quadratic function as the objective function for the multistep predictive controller. For this objective function, we compute the Jacobian matrix and Hessian matrix of the control sequence, and design the receding horizon optimization strategy using Newton-Rhapon algorithm, thus, constituting a nonlinear multistep model predictive controller. Simulation results show desirable performances of the model predictive controller.

**Key words:** model predictive control; neural network; NARMAX; optimization

### 1 引言(Introduction)

模型预测控制(model predictive control, MPC)是一种基于模型的控制策略, 它首先利用系统过去时刻的输入输出信息建立预测模型, 然后利用预测模型对系统未来的输出做出预测, 通过长时域的优化获得局部最优控制序列, 从而实现对被控对象的有效控制. 模型预测控制一般由预测模型、滚动优化策略和反馈校正3部分构成, 根据使用的预测模型不同, 可分为线性预测控制方法和非线性预测控制方法<sup>[1-2]</sup>.

实际中应用的大部分模型预测控制器(动态矩阵控制、模型算法控制、广义预测控制)都是基于系统的线性预测模型. 这是因为大多数控制目标都是让被控对象在某一设定值附近稳定运行(比如锅炉汽包水位控制), 而不是在若干个工作点之间反复切换. 因此, 具有弱非线性的被控对象可以在设定值的较小邻域内被简化为线性系统. 但是, 对于具有强非线性、扰动频繁的系统(比如pH值控制)或者带有时变

特性且工作点跨越较大的系统(比如合成氨控制), 线性化的预测模型通常无法正确描述被控对象的动态特性, 甚至会导致系统不稳定. 如果要将预测控制方法应用于这些场合, 就需要建立合适的非线性预测模型、设计合理的预测控制律.

随着生物计算技术的发展, 人工神经网络为非线性系统建模提供了有力工具. 人工神经网络由众多简单神经元组成, 能表征极为复杂的动态行为特性, 是一种很好的非线性建模工具. 在各种各样结构的神经网络中, BP神经网络是使用最为成熟的一种静态神经网络<sup>[3-4]</sup>. 因此, 本文将使用递归BP神经网络构造一个非线性多步预测模型, 并以此为基础设计一种非线性无约束多步预测控制器.

### 2 预测模型(Predictive model)

#### 2.1 一步预测模型(One-step predictive model)

1982年, Billings提出的非线性自回归滑动平均模型(nonlinear autoregressive moving average with ex-

ogenous inputs, NARMAX)是一种优秀的非线性输入输出模型<sup>[5]</sup>. 对于一个单输入单输出(single input single output, SISO)非线性系统, 其NARMAX模型的数学表达式为

$$y(k+1) = f[y(k), \dots, y(k-n_y+1), u(k), \dots, u(k-n_u+1), e(k), \dots, e(k-n_e)] + e(k), \quad (1)$$

其中:  $u$ 是非线性系统的输入值;  $y$ 是非线性系统的输出值;  $e$ 是零均值高斯白噪声;  $f$ 是非线性输入输出映射关系. 如果忽略噪声对系统的影响, 可以将式(1)简化为

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

如果把式(2)所示的NARMAX模型中的历史输入信息 $u(k), u(k-1), \dots, u(k-n_u+1)$ 和历史输出信息 $y(k), y(k-1), \dots, y(k-n_y+1)$ 作为一个BP神经网络的输入向量, 把系统当前时刻的输出信息 $y(k+1)$ 作为该BP神经网络的输出向量, 利用反向传播学习算法进行网络训练, 就可以建立一个基于BP神经网络的一步预测模型<sup>[6-7]</sup>. 其中, 输入阶数 $n_u$ 和输出阶数 $n_y$ 可通过模型阶次辨识方法估计得出. 得到的一步预测模型示意图如图1所示.

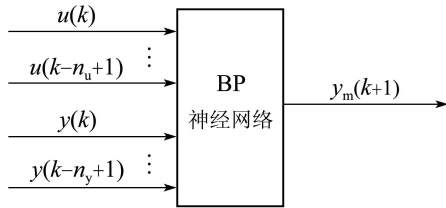


图 1 一步预测模型  
Fig. 1 One-step predictive model

图1中的基于BP神经网络的一步预测模型的输入输出映射关系可描述为

$$y_m(k+1) = F_{NN}[\mathbf{P}(k)], \quad (3)$$

其中:  $y_m(k+1)$ 是系统第 $k+1$ 时刻的预测输出;  $F_{NN}$ 表示一步预测模型的输入输出映射关系;  $\mathbf{P}(k)$ 表示一步预测模型的输入向量, 其结构形式为

$$\begin{aligned} \mathbf{P}(k) &= [\mathbf{u}^T(k) \ \mathbf{y}^T(k)]^T = [p_1 \ p_2 \ \dots \ p_R]^T, \quad (4) \\ \mathbf{u}(k) &= [u(k) \ u(k-1) \ \dots \ u(k-n_u+1)]^T, \quad (5) \\ \mathbf{y}(k) &= [y(k) \ y(k-1) \ \dots \ y(k-n_y+1)]^T, \quad (6) \\ R &= n_u + n_y. \quad (7) \end{aligned}$$

理论研究表明, 一个隐含层神经元使用Tan-Sigmoid传输函数、输出层神经元使用纯线性传输函数的两层BP神经网络, 只要隐含层中具有数量足够多的神经元, 经过训练后可以任意精度逼近绝大多数函数关系<sup>[8-10]</sup>. 因此, 可以选择结构如图2所示的两

层BP神经网络作为一步预测模型.

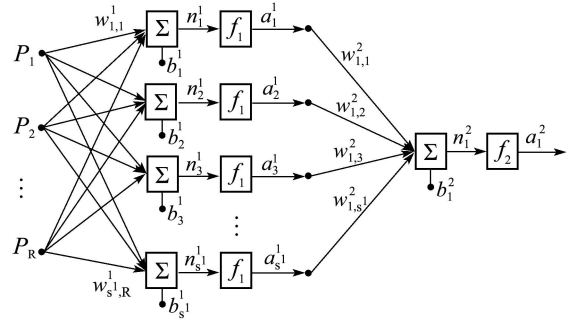


图 2 一步预测模型结构图

Fig. 2 The structure of one-step predictive model

图2中:  $R$ 表示输入个数;  $w_{i,j}^1$ 表示隐藏层中第 $i$ 个神经元和输入向量中第 $j$ 个神经元之间的连接权值;  $S^1$ 表示隐藏层中神经元的个数;  $b_i^1$ 表示隐藏层中第 $i$ 个神经元的偏置值;  $n_i^1$ 表示隐藏层中第 $i$ 个神经元的净输入;  $f_1$ 表示隐藏层神经元的传输函数, 在此使用Tan-Sigmoid函数;  $a_i^1$ 表示隐藏层中第 $i$ 个神经元的输出值;  $w_{1,i}^2$ 表示输出神经元和隐藏层中第 $i$ 个神经元之间的连接权值;  $b_1^2$ 表示输出神经元的偏置值;  $n_1^2$ 表示输出神经元的净输入;  $f_2$ 表示输出神经元的传输函数, 在此使用纯线性传输函数;  $a_1^2$ 表示输出神经元的输出值.

根据图2, 可将式(3)中的输入输出映射关系完全展开, 得到具体的数学表达式为

$$y_m(k+1) = a_1^2(k) = \sum_{i=1}^{S^1} w_{1,i}^2 f_1[n_i^1(k)] + b_1^2, \quad (8)$$

$$n_i^1(k) = \sum_{j=1}^R p_j w_{i,j}^1 + b_i^1. \quad (9)$$

### 2.2 多步预测模型(Multi-step predictive model)

式(8)和式(9)给出了一步预测模型的数学关系式, 如果递归调用该一步预测模型, 用一步预测模型上一时刻的预测输出构成下一时刻的预测输入, 理论上可以预测系统未来任意时刻的输出信息. 因此, 可递归调用图2中的两层BP神经网络构造一个多步预测模型, 模型结构如图3所示.

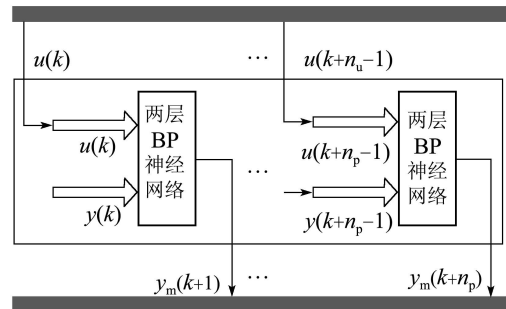


图 3 多步预测模型

Fig. 3 Multi-step predictive model

在该预测模型中,选用的预测时域和控制时域大小相同,用 $n_p$ 表示.根据式(5)和式(6),从第 $k$ 时刻开始,第 $s$ ( $s > 0$ )次递归调用一步预测模型时,一步预测模型的输入向量形式为

$$\mathbf{u}(k+s) = [u(k+s), u(k+s+1) \cdots, u(k+s-n_u+1)], \quad (10)$$

$$\mathbf{y}(k+s) = [y(k+s), y(k+s+1) \cdots, y(k+s-n_y+1)]. \quad (11)$$

从第 $k$ 时刻开始,第 $s$ 次递归调用图2中的一步预测模型时,一步预测模型的输入输出关系表示为

$$y_m(k+s) = \sum_{i=1}^{s^1} w_{1,i}^2 f_1 [n_i^1(k+s-1)] + b_1^2, \quad (12)$$

$$\begin{aligned} n_i^1(k+s-1) = & \sum_{j=1}^{\min(s, n_u)} w_{i,j}^1 u(k+s-j) + \\ & \sum_{j=s+1}^{n_u} w_{i,j}^1 u(k+s-j) + \\ & \sum_{j=1}^{\min(s-1, n_y)} w_{i,j+n_u}^1 y_m(k+s-j) + \\ & \sum_{j=s}^{n_y} w_{i,j+n_u}^1 y(k+s-j) + b_i^1. \end{aligned} \quad (13)$$

当 $s$ 的值遍历 $s = 1, 2, \dots, n_p$ 中的任意一个元素时,即可得到一个从 $k$ 时刻开始 $n_p$ 维的预测输出向量 $\mathbf{y}_m(k+1)$ ,向量的结构形式为

$$\mathbf{y}_m(k+1) = [y_m(k+1) \cdots y_m(k+n_p)]^T. \quad (14)$$

在多步预测控制器中,在线滚动优化的目的就是选择一个合适的 $n_p$ 维输入向量 $\mathbf{u}_m(k)$ ,使得预测输出向量 $\mathbf{u}_m(k+1)$ 尽可能逼近参考输出向量 $\mathbf{y}_r(k+1)$ . $\mathbf{u}_m(k)$ 和 $\mathbf{y}_r(k+1)$ 的向量结构为

$$\mathbf{u}_m(k) = [u(k) \cdots u(k+n_p-1)]^T, \quad (15)$$

$$\mathbf{y}_r(k+1) = [y_r(k+1) \cdots y_r(k+n_p)]^T. \quad (16)$$

### 3 滚动优化策略(Receding horizon optimization strategy)

不考虑系统控制量和输出量的物理约束限制,为了设计滚动优化策略,可选用最常用的二次型函数作为目标函数:

$$\begin{aligned} \min F = & \sum_{i=1}^{n_p} [y_m(k+i) - y_r(k+i)]^2 + \\ & \sum_{i=0}^{n_p-1} \lambda_i [u(k+i) - u(k+i-1)]^2, \end{aligned} \quad (17)$$

其中: $\lambda_i$ 是局部最优控制序列中各个时刻控制量增量的系数,可以根据实际情况进行选择,一般均设置

为1.

滚动优化的目的就是选择一组局部最优的控制作用序列 $\mathbf{u}_m(k)$ ,使得式(17)中的二次型目标函数的函数值最小.因此,可以使用数值优化中的迭代寻优的方法寻找局部控制作用序列.要使用迭代寻优的方法,首先需要保证目标函数对控制作用序列的雅可比矩阵(一阶偏导数)和赫森矩阵(二阶偏导数)是可解的.

目标函数 $F$ 对控制作用序列 $\mathbf{u}_m(k)$ 的雅可比矩阵和赫森矩阵的数学表达式为

$$\mathbf{J} = \frac{\partial F}{\partial \mathbf{u}_m(k)} = \left[ \frac{\partial F}{\partial u(k)} \cdots \frac{\partial F}{\partial u(k+n_p-1)} \right]^T, \quad (18)$$

$$\mathbf{H} = \frac{\partial^2 F}{\partial \mathbf{u}_m(k)^2} = \begin{bmatrix} \frac{\partial^2 F}{\partial u(k)^2} & \cdots & \frac{\partial^2 F}{\partial u(k) \partial u(k+n_p-1)} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 F}{\partial u(k+n_p-1) \partial u(k)} & \cdots & \frac{\partial^2 F}{\partial u(k+n_p-1)^2} \end{bmatrix}. \quad (19)$$

计算出雅可比矩阵和赫森矩阵之后,即可使用经典的Newton-Rhapson算法,求解出目标函数的局部最优控制序列.假设 $\mathbf{u}_m^i(k)$ 表示第 $i$ 次迭代时的控制作用序列, $\mathbf{J}^i$ 表示第 $i$ 次迭代时的雅可比矩阵, $\mathbf{H}^i$ 表示第 $i$ 次迭代时的赫森矩阵,则Newton-Rhapson算法的迭代规则为

$$\mathbf{u}_m^{i+1}(k) = \mathbf{u}_m^i(k) + [\mathbf{H}^i]^{-1}[\mathbf{J}^i]. \quad (20)$$

从式(20)可以看出,在每一步迭代过程中,都要首先计算赫森矩阵的逆矩阵.如果赫森矩阵的维数较高,将需要大量时间进行求逆运算,有可能满足不了预测控制器的实时性要求.为了避免求解赫森矩阵的逆矩阵,可把式(20)进行如下变换:

$$\mathbf{H}^i [\mathbf{u}_m^{i+1}(k) - \mathbf{u}_m^i(k)] = \mathbf{J}^i. \quad (21)$$

为了表示方便,将上式中的符号简化为

$$\begin{cases} \mathbf{H}^i = \mathbf{A}, \\ \mathbf{J}^i = \mathbf{b}, \\ \mathbf{u}_m^{i+1}(k) - \mathbf{u}_m^i(k) = \mathbf{x}. \end{cases} \quad (22)$$

通过符号替换,式(21)可以简化表示为

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad (23)$$

式(23)是一个标准的线性方程组,可以使用LU分解或QR分解快速求得 $\mathbf{x}$ ,然后使用式(24)计算出第 $i+1$ 次迭代的控制作用序列:

$$\mathbf{u}_m^{i+1}(k) = \mathbf{u}_m^i(k) + \mathbf{x}. \quad (24)$$

预测控制律的优化方法确定之后, 控制器设计的关键的问题就是如何求解目标函数对控制作用序列的雅可比矩阵和赫森矩阵. 在本节的后续部分, 将给出雅可比矩阵和赫森矩阵的详细计算过程.

### 3.1 雅可比矩阵(Jacobian matrix)

不失一般性, 给出雅可比矩阵中第  $r$  行第 1 列元素的计算过程如下<sup>[11]</sup>:

$$\begin{aligned} \frac{\partial F}{\partial u(k+r-1)} = & 2 \sum_{i=1}^{n_p} y_m(k+i) \frac{\partial y_m(k+i)}{\partial u(k+r-1)} - \\ & 2 \sum_{i=1}^{n_p} y_r(k+i) \frac{\partial y_m(k+i)}{\partial u(k+r-1)} + \\ & 2 \sum_{i=0}^{n_p-1} \lambda_i u(k+i) \frac{\partial u(k+i)}{\partial u(k+r-1)} - \\ & 2 \sum_{i=0}^{n_p-1} \lambda_i u(k+i-1) \frac{\partial u(k+i)}{\partial u(k+r-1)} - \\ & 2 \sum_{i=0}^{n_p-1} \lambda_i u(k+i) \frac{\partial u(k+i-1)}{\partial u(k+r-1)} + \\ & 2 \sum_{i=0}^{n_p-1} \lambda_i u(k+i-1) \frac{\partial u(k+i-1)}{\partial u(k+r-1)}. \end{aligned} \quad (25)$$

为了表示方便, 定义Delta函数:

$$\delta(x, y) = \begin{cases} 1, & x = y, \\ 0, & x \neq y. \end{cases} \quad (26)$$

将式(26)中的Delta函数代入式(25)进行化简, 可得如下结果:

$$\begin{aligned} \frac{\partial F}{\partial u(k+r-1)} = & 2 \sum_{i=1}^{n_p} y_m(k+i) \frac{\partial y_m(k+i)}{\partial u(k+r-1)} - \\ & 2 \sum_{i=1}^{n_p} y_r(k+i) \frac{\partial y_m(k+i)}{\partial u(k+r-1)} + \\ & 2 \sum_{i=0}^{n_p-1} \lambda_i u(k+i) \delta(i, r-1) - \\ & 2 \sum_{i=0}^{n_p-1} \lambda_i u(k+i) \delta(i, r) - \\ & 2 \sum_{i=0}^{n_p-1} \lambda_i u(k+i-1) \delta(i, r-1) + \\ & 2 \sum_{i=0}^{n_p-1} \lambda_i u(k+i-1) \delta(i, r). \end{aligned} \quad (27)$$

由式(8)可得

$$\begin{aligned} \frac{\partial y_m(k+i)}{\partial u(k+r-1)} = & \sum_{p=1}^{s1} w_{1,p}^2 \times \frac{\partial f_1[n_p^1(k+i-1)]}{\partial n_p^1(k+i-1)} \times \frac{\partial n_p^1(k+i-1)}{\partial u(k+r-1)}. \end{aligned} \quad (28)$$

由式(9)可得

$$\begin{aligned} \frac{\partial n_p^1(k+i-1)}{\partial u(k+r-1)} = & \sum_{q=1}^{\min(i, n_u)} w_{p,q}^1 \frac{\partial u(k+i-q)}{\partial u(k+r-1)} + \\ & \sum_{q=i+1}^{n_u} w_{p,q}^1 \frac{\partial u(k+i-q)}{\partial u(k+r-1)} + \\ & \sum_{q=1}^{\min(i-1, n_y)} w_{p,q+n_u}^1 \frac{\partial y_m(k+i-q)}{\partial u(k+r-1)} + \\ & \sum_{q=i}^{n_y} w_{p,q+n_u}^1 \frac{\partial y(k+i-q)}{\partial u(k+r-1)}. \end{aligned} \quad (29)$$

将式(26)中的Delta函数代入式(29)并化简, 可得

$$\begin{aligned} \frac{\partial n_p^1(k+i-1)}{\partial u(k+r-1)} = & \sum_{q=i+1}^{n_u} w_{p,q}^1 \frac{\partial u(k+i-q)}{\partial u(k+r-1)} + \\ & \sum_{q=1}^{\min(i-1, n_y)} w_{p,q+n_u}^1 \frac{\partial y_m(k+i-q)}{\partial u(k+r-1)} = \\ & \sum_{q=i+1}^{n_u} w_{p,q}^1 \delta(i-q, r-1) + \\ & \sum_{q=1}^{\min(i-1, n_y)} w_{p,q+n_u}^1 \frac{\partial y_m(k+i-q)}{\partial u(k+r-1)}. \end{aligned} \quad (30)$$

根据多步预测模型的结构, 要使

$$\frac{\partial y_m(k+i-q)}{\partial u(k+r-1)}$$

不为零, 必须满足下面的不等式:

$$i - q - (r - 1) \geq 1. \quad (31)$$

对不等式(31)进行化简, 可得

$$q \leq i - r. \quad (32)$$

因为  $r \geq 1$ , 必然有不等式(33)成立:

$$i - r \leq i - 1. \quad (33)$$

因此, 式(30)可进一步化简为

$$\begin{aligned} \frac{\partial n_p^1(k+i-1)}{\partial u(k+r-1)} = & \sum_{q=i+1}^{n_u} w_{p,q}^1 \delta(i-q, r-1) + \\ & \sum_{q=1}^{\min(i-r, n_y)} w_{p,q+n_u}^1 \frac{\partial y_m(k+i-q)}{\partial u(k+r-1)}. \end{aligned} \quad (34)$$

使用式(28)和式(34), 可以计算出雅可比矩阵中的任意一个元素, 从而构造出完整的雅可比矩阵.

### 3.2 赫森矩阵(Hessian matrix)

不失一般性, 给出赫森矩阵中第  $r$  行、第  $c$  列的元

素的计算过程如下:

$$\begin{aligned} \frac{\partial^2 J}{\partial u(k+r-1)\partial u(k+c-1)} = & 2 \sum_{i=1}^{n_p} \frac{\partial y_m(k+i)}{\partial u(k+c-1)} \frac{\partial y_m(k+i)}{\partial u(k+r-1)} + \\ & 2 \sum_{i=1}^{n_p} y_m(k+i) \frac{\partial^2 y_m(k+i)}{\partial u(k+r-1)\partial u(k+c-1)} - \\ & 2 \sum_{i=1}^{n_p} y_r(k+i) \frac{\partial^2 y_m(k+i)}{\partial u(k+r-1)\partial u(k+c-1)} + \\ & 2 \sum_{i=0}^{n_p-1} \lambda_i \frac{\partial u(k+i)}{\partial u(k+c-1)} [\delta(i, r-1) - \delta(i, r)] - \\ & 2 \sum_{i=0}^{n_p-1} \lambda_i \frac{\partial u(k+i-1)}{\partial u(k+c-1)} [\delta(i, r-1) - \delta(i, r)]. \end{aligned} \quad (35)$$

将式(26)中的Delta函数代入上式并化简, 可得

$$\begin{aligned} \frac{\partial^2 F}{\partial u(k+r-1)\partial u(k+c-1)} = & 2 \sum_{i=1}^{n_p} \frac{\partial y_m(k+i)}{\partial u(k+c-1)} \frac{\partial y_m(k+i)}{\partial u(k+r-1)} + \\ & 2 \sum_{i=1}^{n_p} y_m(k+i) \frac{\partial^2 y_m(k+i)}{\partial u(k+r-1)\partial u(k+c-1)} - \\ & 2 \sum_{i=1}^{n_p} y_r(k+i) \frac{\partial^2 y_m(k+i)}{\partial u(k+r-1)\partial u(k+c-1)} + \\ & 2 \sum_{i=0}^{n_p-1} \lambda_i \delta(i, c-1) [\delta(i, r-1) - \delta(i, r)] - \\ & 2 \sum_{i=0}^{n_p-1} \lambda_i \delta(i, c) [\delta(i, r-1) - \delta(i, r)]. \end{aligned} \quad (36)$$

在式(36)中,  $\frac{\partial y_m(k+i)}{\partial u(k+c-1)}$  和  $\frac{\partial y_m(k+i)}{\partial u(k+r-1)}$  这两部分的值可通过式(28)和式(34)计算出来. 赫森矩阵中第*r*行、第*c*列元素值的计算关键在于如何求解  $\frac{\partial^2 y_m(k+i)}{\partial u(k+r-1)\partial u(k+c-1)}$  的值, 式(37)给出了该部分的具体计算方法:

$$\begin{aligned} \frac{\partial^2 y_m(k+i)}{\partial u(k+r-1)\partial u(k+c-1)} = & \frac{\sum_{p=1}^{S^1} w_{1,p}^2 \frac{\partial^2 f_1[n_p^1(k+i-1)]}{\partial u(k+r-1)\partial u(k+c-1)}}{\sum_{p=1}^{S^1} w_{1,p}^2 \frac{\partial f_1[n_p^1(k+i-1)]}{\partial n_p^1(k+i-1)}} \times \\ & \frac{\partial^2 n_p^1(k+i-1)}{\partial u(k+r-1)\partial u(k+c-1)} + \\ & \frac{\sum_{p=1}^{S^1} w_{1,p}^2 \frac{\partial^2 f_1[n_p^1(k+i-1)]}{\partial n_p^1(k+i-1)^2}}{\frac{\partial n_p^1(k+i-1)}{\partial u(k+r-1)} \frac{\partial n_p^1(k+i-1)}{\partial u(k+c-1)}}. \end{aligned} \quad (37)$$

式(37)中:  $\frac{\partial f_1[n_p^1(k+i-1)]}{\partial n_p^1(k+i-1)}$  和  $\frac{\partial^2 f_1[n_p^1(k+i-1)]}{\partial n_p^1(k+i-1)^2}$

的值可通过隐藏层传输函数对净输入的一阶偏导数和二阶偏导数直接求出;

$$\frac{\partial n_p^1(k+i-1)}{\partial u(k+r-1)} \text{ 和 } \frac{\partial n_p^1(k+i-1)}{\partial u(k+c-1)}$$

的值可通过式(34)求出, 唯  $\frac{\partial^2 n_p^1(k+i-1)}{\partial u(k+r-1)\partial u(k+c-1)}$  需要进一步计算. 可以从式(34)出发, 推导该部分的具体计算过程为

$$\frac{\partial^2 n_p^1(k+i-1)}{\partial u(k+r-1)\partial u(k+c-1)} = \sum_{q=1}^{\min(i-r, n_y)} w_{p,q+n_u}^1 \frac{\partial^2 y_m(k+i-q)}{\partial u(k+r-1)\partial u(k+c-1)}, \quad (38)$$

和式(31)–(34)的讨论过程类似, 式(38)可继续化简为

$$\frac{\partial^2 n_p^1(k+i-1)}{\partial u(k+r-1)\partial u(k+c-1)} = \sum_{q=1}^{Q_{\max}} w_{p,q+n_u}^1 \frac{\partial^2 y_m(k+i-q)}{\partial u(k+r-1)\partial u(k+c-1)}, \quad (39)$$

其中  $Q_{\max} = \min(i-r, i-c, n_y)$ . 利用式(36)–(39)即可计算出赫森矩阵中的任何一个元素, 从而构造完整的赫森矩阵.

计算出雅可比矩阵和赫森矩阵之后, 即可利用 Newton–Rhapson 算法, 通过迭代寻优计算第*k*时刻的最优控制作用序列.

#### 4 仿真分析(Simulation)

为了检验前面两节设计的非线性多步预测模型的预测性能和多步预测控制器的控制效果, 使用式(40)所示的非线性系统进行计算机仿真分析:

$$y(k) = u^3(k-1) + 2u(k-2) + u(k-1)u(k-2) + \sin[y(k-1)]. \quad (40)$$

为系统加载一个伪随机序列作激励信号见图4.

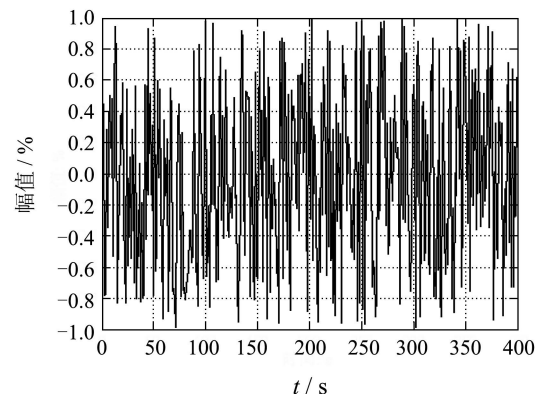


图4 随机激励信号序列

Fig. 4 Random excitation signal series

同时, 采集系统的输出信号如图5所示.

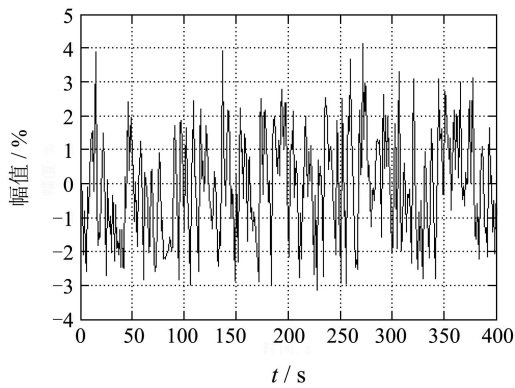


图 5 系统输出信号序列

Fig. 5 Output series of the nonlinear system

根据非线性系统方程可知, 系统的输入阶数 $n_u$ 为2, 输出阶数 $n_y$ 为1. 因此, 可以构造一个3输入1输出的两层BP神经网络作为一步预测模型, 然后递归调用该预测模型构成多步预测模型. 递归调用的次数取决于预测时域的大小.

假设一步预测模型的隐藏层中包含有20个神经元, 使用Levenberg-Marquardt算法训练该神经网络, 然后递归调用该网络3次, 构造一个3步预测模型. 为3步预测模型和实际系统同时加上正弦信号, 预测输出和实际输出比较曲线如图6所示.

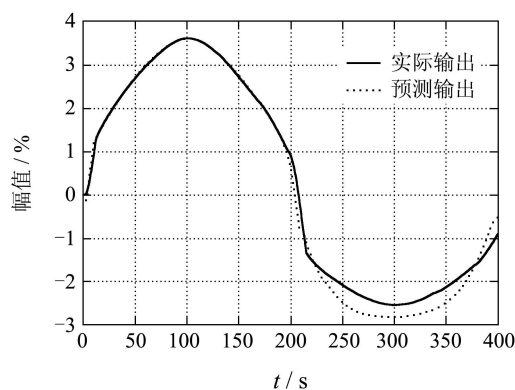


图 6 预测输出与实际输出比较

Fig. 6 The comparison of predictive output and real output

得到非线性系统的3步预测模型之后, 利用第3节中设计的滚动优化控制律, 构造一个无约束的三步预测控制器, 并将该控制器应用于式(40)所示的非线性系统中. 假设该系统在四个不同的工作点之间进行切换, 控制器对于不同的工作点的跟踪效果图如图7所示.

由仿真结果可以看出, 当系统在多个工作点之间切换时, 设计的非线性三步预测控制器能够较好地跟踪工作点的变化, 系统的最大超调量和调节时间都比较理想.

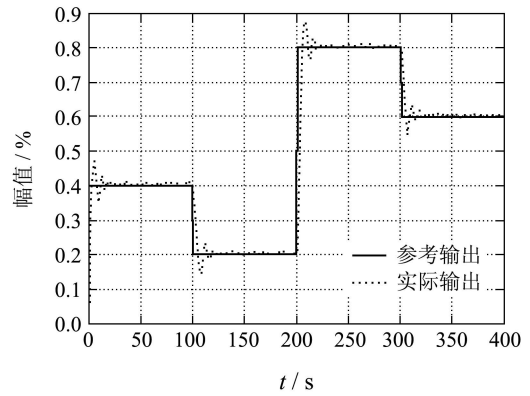


图 7 三步预测控制器的控制效果

Fig. 7 The effect of three-step predictive controller

在制定非线性系统的控制策略时, 滑模控制是人们经常采用的一种控制手段. 滑模控制也叫变结构控制, 这种控制策略的不同之处在于其本身的结构并不固定, 而是随着系统状态(例如偏差或偏差的各阶导数)的变化不断变化, 迫使系统按照预定的状态轨迹运行. 随着研究工作的深入, 人们将滑模控制和预测控制结合起来, 提出了一种滑模预测控制方案, 用于改善非线性系统的控制效果<sup>[12-13]</sup>. 为了进一步检验神经网络预测控制器的性能, 将其与滑模预测控制器的控制效果进行比较, 如图8所示.

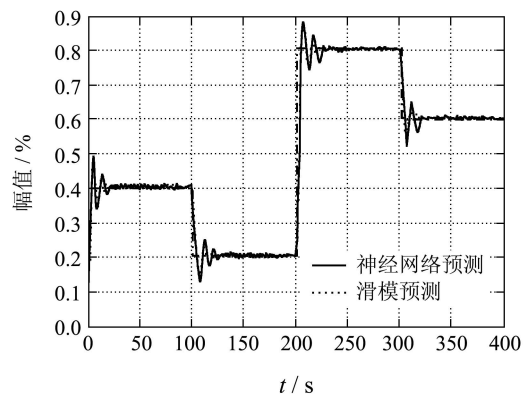


图 8 控制效果比较

Fig. 8 The comparison of difference controllers

从图8所示的比较结果可以看出, 神经网络预测控制器和滑模预测控制器都能较好地跟踪非线性系统的工作点变化. 相比较而言, 使用神经网络预测控制器构成的控制系统, 调节时间较短, 超调量较小, 无论快速性还是还是平稳性都优于使用滑模预测控制器构成的控制系统.

### 5 结论(Conclusion)

本文从非线性系统NARMAX模型的数学表达式出发, 首先使用两层BP神经网络为非线性系统设计了一个一步预测模型, 然后通过递归调用该一步预测模型, 得到了多步预测模型. 递归调用一步预测模型的次数, 取决于多步预测模型的预测时域大小.

通过分析多步预测模型的内在机理, 选择了二次型函数作为预测控制器的目标函数, 给出了目标函数对最优控制序列的雅可比矩阵和赫森矩阵的计算方法, 最后使用Newton-Rhapson算法设计了一种迭代优化控制策略, 构成了完整的无约束非线性多步预测控制器. 仿真结果表明, 该预测控制器能够跟踪非线性系统的多个工作点, 并且具有较好的控制效果.

### 参考文献(References):

- [1] 席裕庚. 预测控制[M]. 北京: 国防工业出版社, 1993.  
(XI Yugeng. *Predictive Control*[M]. Beijing: National Defence Industry Press, 1993.)
- [2] 舒迪前. 预测控制系统及应用[M]. 北京: 机械工业出版社, 1996.  
(SHU Diqian. *Predictive Control System and Application*[M]. Beijing: China Machine Press, 1996.)
- [3] 戴文战, 娄海川, 杨爱萍. 非线性系统神经网络预测控制研究进展[J]. 控制理论与应用, 2009, 26(5): 521 – 530.  
(DAI Wenzhan, LOU Haichuan, YANG Aiping. Overview of neural network predictive control in nonlinear system[J]. *Control Theory & Applications*, 2009, 26(5): 521 – 530.)
- [4] HORNIK K M, STINCHCOMBE M, WHITE H. Multilayer feed-forward networks are universal approximators[J]. *Neural Networks*, 1989, 2(5): 359 – 366.
- [5] CHEN S, BILLINGS S A. Representations of non-linear systems: the NARMAX model[J]. *International Journal of Control*, 1989, 49(3): 1013 – 1032.
- [6] AKESSON B M, TOIVONEN H T. A neural network model predictive controller[J]. *Journal of Process Control*, 2006, 16(9): 937 – 946.
- [7] RUANO A E, CRISPIM E M, CONCEICAO E Z E, et al. Prediction of building's temperature using neural networks models[J]. *Energy and Buildings*, 2006, 38(6): 682 – 694.
- [8] KISHOR N, SINGH S P. Simulated response of NN based identification and predictive control of hydro plant[J]. *Expert Systems with Applications*, 2007, 32(1): 233 – 244.
- [9] YU D W, YU D L. Multi-rate model predictive control of a chemical reactor based on three neural models[J]. *Biochemical Engineering Journal*, 2007, 37(1): 86 – 97.
- [10] AL SEYAB R K, CAO Y. Differential recurrent neural network based predictive control[J]. *Computers and Chemical Engineering*, 2008, 32(7): 1533 – 1545.
- [11] NORIEGA J R, WANG H. A direct adaptive neural-network control for unknown nonlinear systems and its application[J]. *IEEE Transactions on Neural Networks*, 1998, 1(9): 27 – 34.
- [12] 李庆良, 雷虎民, 邵雷, 等. 基于即时学习的MIMO系统滑模预测控制方法[J]. 控制理论与应用, 2011, 28(8): 1159 – 1163.  
(LI Qingliang, LEI Humin, SHAO Lei, et al. Sliding mode predictive control for MIMO systems via lazy learning[J]. *Control Theory & Applications*, 2011, 28(8): 1159 – 1163.)
- [13] PLAZA D, DE KEYSER R, BONILLA J. Model Predictive and sliding mode control of a Boost converter[C] // *International Symposium on Power Electronics, Electrical Drives, Automation and Motion*. New York: IEEE, 2008: 37 – 42.

### 作者简介:

李会军 (1980—), 男, 博士, 讲师, 目前研究方向为计算机控制、非线性模型预测控制, E-mail: plutoli@163.com;

肖兵 (1965—), 女, 工程师, 目前研究方向为石化企业的综合自动化系统集成, E-mail: cyy\_xiaob@petrochina.com.cn.