

神经网络预测控制局部优化初值确定方法

樊兆峰^{1,2}, 马小平^{1†}, 邵晓根²

(1. 中国矿业大学 信息与电气工程学院, 江苏 徐州 221008;

2. 徐州工程学院 信电学院, 江苏 徐州 221111)

摘要: 为解决局部优化算法初值选取不当造成神经网络预测控制性能下降的问题, 本文提出了一种动态确定初值的方法. 在每次优化时通过逆网络将初值选在输出误差最小点, 通过修正目标性能函数中的权重因子来确保初值与当前控制量之间存在极值, 并在理论上进行了证明. 以BP神经网络预测控制为例, 采用牛顿拉夫逊算法实现滚动优化, 对所提方法进行了仿真实验, 结果表明能够解决初值问题, 提高控制系统的可靠性.

关键词: 神经网络; 模型预测控制; 优化; 初值问题

中图分类号: TP273 文献标识码: A

Method to determine initial value of local optimization for neural network predictive control

FAN Zhao-feng^{1,2}, MA Xiao-ping^{1†}, SHAO Xiao-gen²

(1. School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou Jiangsu 221008, China;

2. Department of College of Information and Electrical Engineering, Xuzhou Institute of Technology, Xuzhou Jiangsu 221111, China)

Abstract: To deal with the performance degradation caused by improper initial values in neural network local optimization predictive control, we propose a method to dynamically determine the initial values. In each optimization cycle the minimum output error point is selected by calculating the inverse neural network. The existence of the minimal value of the objective function between this point and the current control point can be ensured and proved through modifying the weighting factor. Finally, a simulation experiment is carried out to verify the proposed method using a back propagation (BP) neural network as the predictive model, and the Newton-Raphson algorithm is employed as the receding horizon optimization strategy. The results show that the initial value problem can be solved to improve the reliability of the control system.

Key words: neural networks; model predictive control; optimization; initial value problems

1 引言(Introduction)

模型预测控制(MPC)来源于工业实践, 是基于预测模型的有限时域优化控制. 近年来虽然MPC获得了广泛应用, 但从应用对象来看, 主要还限于线性或准线性过程, 造成这一现象主要是由于在工业过程中非线性机理建模要耗费很大代价, 而且很难得到准确的模型^[1]. 由于神经网络能够充分逼近复杂的非线性映射关系, 具有学习与适应不确定系统的动态特性和较强的鲁棒性和容错性的特点, 使其成为对非线性系统建立预测模型和优化控制的关键技术之一, 并形成了各种基于神经网络的预测控制算法^[2].

由于神经网络模型自身的非线性导致滚动优化时要求解复杂的非线性方程组才能获得最优解, 因此研究者大多通过局部优化数值算法来获得次优解, 如文

献[3]对神经网络一步预测控制分别采用了梯度下降法(GD), 牛顿拉夫逊(N-R)法及Levenberg-Marquardt(L-M)法进行优化, 并对三种控制器性能做了比较. 文献[4]采用Quasi-Newton(Q-N)方法优化GPC控制器. 文献[5]采用L-M和Q-N算法优化控制量, 避免了控制器参数的频繁调节, 有效提高了抗扰能力. 文献[3-5]都以上一时刻的控制量为初值来优化多步控制时域的控制量. 对神经网络的多步预测问题, 文献[6-7]分别采用L-M, N-R算法进行优化取得了较好的控制效果, 但是文中并没有指出初值的确定方法.

对于具有较强非线性的神经网络预测控制而言, 目标性能函数一般不是控制量的单峰函数, 会出现多个极值. 由局部优化^[8-9]原理知, 对于多峰函数的局部优化结果受初值的影响较大.

本文按照文献[3-5]的初值确定方法利用局部优化进行了大量仿真研究, 结果发现可能存在控制量不变或输出误差较大的控制性能下降问题(详见后文), 使得控制系统的可靠性难以保证, 究其原因都是由于初值问题造成的. 通过查阅文献发现很少有研究者针对此问题提出有效的解决办法, 为克服这一困难, 受目标性能函数构造思想的启发, 提出一种动态确定初值的方法.

为提高论文的可读性, 首先以BP神经网络局部优化预测控制为例, 说明系统控制原理.

2 BP神经网络预测控制(BP neural network predictive control)

预测控制一般建立在3项基本原理基础上: 预测模型、滚动优化、反馈校正^[10]. BP神经网络局部优化预测控制一般采用BP神经网络作为预测模型, 利用局部优化策略实现滚动优化.

2.1 预测模型(Predictive model)

考虑如下单输入单输出(SISO)非线性被控系统:

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

其中: u 是控制输入值; y 是非线性系统的输出值; f 是系统的非线性映射关系; n_y, n_u 分别为输出时间序列和控制时间序列的阶次, 其值可以通过模型阶次辨识的方法得到^[11-14].

如果将模型(1)中的历史(k 时刻及之前时刻)输出信息 $y(k), \dots, y(k-n_y+1)$ 和历史输入信息 $u(k), \dots, u(k-n_u+1)$ 作为BP神经网络的输入, 模型下一时刻的输出 $y(k+1)$ 作为BP神经网络的输出, 通过输入输出样本数据训练神经网络, 使其能够以足够的精度逼近非线性映射关系 f , 则BP网络即可成为非线性系统的预测模型.

一步预测模型如图1所示, 其中 $\mathbf{p}(k)$ 是输入向量, 维数为 $R \times 1$, $R = n_y + n_u$, 结构形式为 $[y(k), \dots, y(k-n_y+1), u(k), \dots, u(k-n_u+1)]^T$.

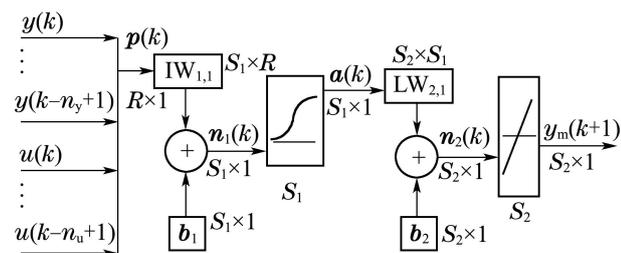


图1 BP神经网络一步预测模型

Fig. 1 One-step predictive model using BP neural network

S_1 是隐层神经元的个数, S_2 是线性输出神经元的个数, 对SISO系统而言有: $S_2 = 1$.

IW 是输入权值矩阵, 维数为 $S_1 \times R$, LW 是输出

权值矩阵, 维数为 $S_2 \times S_1$, \mathbf{b}_1 是输入偏置向量, 维数为 $S_1 \times 1$, \mathbf{b}_2 是输出偏置向量, 维数为 $S_2 \times 1$.

$\mathbf{n}_1(k)$ 是隐层神经元的输入向量, 维数为 $S_1 \times 1$.

$\mathbf{n}_2(k)$ 是线性输出神经元的输入向量, 维数为 $S_2 \times 1$.

$\mathbf{a}(k)$ 是隐层神经元的输出, 维数为 $S_1 \times 1$.

$y_m(k+1)$ 是BP神经网络的预测输出.

根据图1所示, 取隐层神经元映射关系为Logsig函数, 则可得以下关系式:

$$\mathbf{n}_1(k) = IW \times \mathbf{p}(k) + \mathbf{b}_1, \quad (2)$$

$$\mathbf{a}(k) = \text{Logsig}(\mathbf{n}_1(k)) = (1 + \exp(-\mathbf{n}_1(k)))^{-1}, \quad (3)$$

$$y_m(k+1) = LW \times \mathbf{a}(k) + \mathbf{b}_2. \quad (4)$$

其中“ \cdot^{-1} ”表示向量点乘 -1 次方. 由式(2)-(4)可得BP网络输入输出映射关系的数学表达式为

$$y_m(k+1) = LW \times (1 + \exp(-IW \times \mathbf{p}(k) - \mathbf{b}_1))^{-1} + \mathbf{b}_2. \quad (5)$$

将系统历史信息 \mathbf{p} 代入式(5)可计算模型的下一步输出 $y_m(k+1)$, 由于经过训练后BP网络能够充分逼近非线性映射关系 f , 因此可用 $y_m(k+1)$ 代替被控对象的下一步输出 $y(k+1)$, 从而实现一步预测.

在一步预测模型的基础上, 用一步预测得到的输出值 $y_m(k+1)$ 和控制输入值 $u(k+1)$ 更新输入向量 $\mathbf{p}(k+1)$ 为 $[y_m(k+1) \dots y(k-n_y+2) \ u(k+1) \dots u(k-n_u+2)]^T$, 通过代入式(5)递归调用一步预测模型即可预测再下一时刻的输出值 $y_m(k+2)$, 如此不断更新输入向量, 递归调用一步预测模型, 理论上便可预测未来任何时刻的输出值.

2.2 滚动优化(Receding horizon optimization)

设系统预测的步数为 d , 构造目标性能函数如下:

$$J = \sum_{j=1}^d [y_m(k+j) - y_r(k+j)]^2 + \lambda \sum_{j=0}^{d-1} [u(k+j) - u(k+j-1)]^2, \quad (6)$$

其中 $y_m(k+j), y_r(k+j)$ 分别为第 $k+j$ 时刻预测输出与参考输出, 参考输出为控制系统设定值. 令

$$\mathbf{u} = [u(k) \ u(k+1) \ \dots \ u(k+d-1)]^T. \quad (7)$$

由式(5)-(6)可知 J 只是式(7)所示变量 \mathbf{u} 的函数. 滚动优化的任务就是选择 d 步最优控制量 \mathbf{u}^* , 使得目标性能函数 J 取最小值. 在每一步控制时只取 \mathbf{u}^* 的第1个元素 $u(k)^*$ 实现一步控制, 在下一个采样周期重复这一过程即实现所谓滚动优化.

显然如果通过 $\frac{dJ}{d\mathbf{u}} = 0$ 直接求 \mathbf{u}^* 要解复杂非线性方程组, 所以许多研究者通过局部优化数值迭代方法来获得次优解.

由于局部数值迭代优化方法对初值的依赖具有相同的特点, 因此本文选用经典的N-R算法来进行优化, 采用两步预测一步控制, 即 $d = 2$.

N-R算法具有二阶收敛速度, 若搜寻 $\frac{dJ}{d\mathbf{u}} = 0$ 的极小值点, 其迭代公式如下^[7,15]:

$$\mathbf{u}^{i+1} = \mathbf{u}^i - [\mathbf{H}^i]^{-1} J_a^i, \quad (8)$$

其中 J_a^i , \mathbf{H}^i 分别是第 i 次迭代的Jacbian矩阵与Hessian矩阵, 下面分别来求, 对两步预测有

$$J_a = \frac{\partial J}{\partial \mathbf{u}} = \left[\frac{\partial J}{\partial u(k)} \quad \frac{\partial J}{\partial u(k+1)} \right]^T, \quad (9)$$

$$\mathbf{H} = \frac{\partial^2 J}{\partial \mathbf{u}^2} = \begin{bmatrix} \frac{\partial^2 J}{\partial u(k)^2} & \frac{\partial^2 J}{\partial u(k)\partial u(k+1)} \\ \frac{\partial^2 J}{\partial u(k+1)\partial u(k)} & \frac{\partial^2 J}{\partial u(k+1)^2} \end{bmatrix}. \quad (10)$$

由式(2)–(6)可得

$$\begin{aligned} \frac{\partial J}{\partial u(k)} &= 2(y_m(k+1) - y_r(k+1)) \frac{\partial y_m(k+1)}{\partial u(k)} + \\ &2(y_m(k+2) - y_r(k+2)) \frac{\partial y_m(k+2)}{\partial u(k)} + \\ &2\lambda(2u(k) - u(k-1) - u(k+1)), \end{aligned} \quad (11a)$$

$$\begin{aligned} \frac{\partial y_m(k+1)}{\partial u(k)} &= LW \times ((\mathbf{a}(k) - \mathbf{a}(k))^2) \cdot * \\ &IW(:, n_y+1)), \end{aligned} \quad (11b)$$

$$\begin{aligned} \frac{\partial y_m(k+2)}{\partial u(k)} &= LW \times ((\mathbf{a}(k+1) - \mathbf{a}(k+1))^2) \cdot * \\ &(IW(:, 1) \frac{\partial y_m(k+1)}{\partial u(k)} + \\ &IW(:, n_y+2))). \end{aligned} \quad (11c)$$

上式中: $IW(:, n_y+1)$ 表示矩阵 IW 的第 n_y+1 列, 符号“ $\cdot*$ ”表示向量的点乘, 符号“ \cdot^2 ”表示向量的点乘方.

$$\begin{aligned} \frac{\partial J}{\partial u(k+1)} &= 2(y_m(k+2) - y_r(k+2)) \frac{\partial y_m(k+2)}{\partial u(k+1)} + \\ &2\lambda(u(k+1) - u(k)), \end{aligned} \quad (12a)$$

$$\begin{aligned} \frac{\partial y_m(k+2)}{\partial u(k+1)} &= LW \times ((\mathbf{a}(k+1) - \mathbf{a}(k+1))^2) \cdot * \\ &IW(:, n_y+1)). \end{aligned} \quad (12b)$$

对给定的控制量 \mathbf{u} , 按式(2)–(3)(11)–(12)可完整计算出Jacbian矩阵:

$$\begin{aligned} \frac{\partial^2 J}{\partial u(k)^2} &= 2\left(\frac{\partial y_m(k+1)}{\partial u(k)}\right)^2 + 2(y_m(k+1) - \\ &y_r(k+1)) \frac{\partial^2 y_m(k+1)}{\partial u(k)^2} + \end{aligned}$$

$$\begin{aligned} &2\left(\frac{\partial y_m(k+2)}{\partial u(k)}\right)^2 + 2(y_m(k+2) - \\ &y_r(k+2)) \frac{\partial^2 y_m(k+2)}{\partial u(k)^2} + 4\lambda, \end{aligned} \quad (13a)$$

$$\begin{aligned} \frac{\partial^2 y_m(k+1)}{\partial u(k)^2} &= LW((\mathbf{a}(k) - 3\mathbf{a}(k))^2 + 2\mathbf{a}(k))^3) \cdot * \\ &(IW(:, n_y+1))^2), \end{aligned} \quad (13b)$$

$$\begin{aligned} \frac{\partial^2 y_m(k+2)}{\partial u(k)^2} &= \frac{\partial^2 y_m(k+1)}{\partial u(k)^2} LW \times (((\mathbf{a}(k+1) - \\ &\mathbf{a}(k+1))^2) \cdot (IW(:, 1))) + LW \times \\ &(((\mathbf{a}(k+1) - 3\mathbf{a}(k+1))^2 + 2\mathbf{a}(k+1))^3) \cdot * \\ &(IW(:, 1) \frac{\partial y_m(k+1)}{\partial u(k)} + \\ &IW(:, n_y+2))^2), \end{aligned} \quad (13c)$$

$$\begin{aligned} \frac{\partial^2 J}{\partial u(k)\partial u(k+1)} &= \frac{\partial^2 J}{\partial u(k+1)\partial u(k)} = \\ &2\frac{\partial y_m(k+2)}{\partial u(k)} \frac{\partial y_m(k+2)}{\partial u(k+1)} + \\ &2(y_m(k+2) - y_r(k+2)) \times \\ &\frac{\partial^2 y_m(k+2)}{\partial u(k)\partial u(k+1)} - 2\lambda, \end{aligned} \quad (14a)$$

$$\begin{aligned} \frac{\partial^2 y_m(k+2)}{\partial u(k)\partial u(k+1)} &= LW((\mathbf{a}(k+1) - 3\mathbf{a}(k+1))^2 + \\ &2\mathbf{a}(k+1))^3) \cdot (IW(:, n_y+1)) \cdot * \\ &(IW(:, 1) \frac{\partial y_m(k+1)}{\partial u(k)} + \\ &IW(:, n_y+2)), \end{aligned} \quad (14b)$$

$$\begin{aligned} \frac{\partial^2 J}{\partial u(k+1)^2} &= 2\left(\frac{\partial y_m(k+2)}{\partial u(k+1)}\right)^2 + 2(y_m(k+2) - \\ &y_r(k+2)) \frac{\partial^2 y_m(k+2)}{\partial u(k+1)^2} + 2\lambda, \end{aligned} \quad (15a)$$

$$\begin{aligned} \frac{\partial^2 y_m(k+2)}{\partial u(k+1)^2} &= LW((\mathbf{a}(k+1) - 3\mathbf{a}(k+1))^2 + \\ &2\mathbf{a}(k+1))^3) \cdot ((IW(:, n_y+1))^2). \end{aligned} \quad (15b)$$

按式(2)–(3)(11)–(15)可完整计算出Hessian矩阵. 每次优化迭代都要计算Jacbian矩阵和Hessian矩阵, 然后按式(8)进行迭代, 则算法将收敛到目标性能函数 J 的一个极小值, 滚动优化得以实现.

2.3 反馈校正(Feedback correction)

在第 k 次采样周期内, 设检测的系统实际输出为 $y(k)$, 上一个周期时预测的本次系统输出为 $y_m(k)$, 则预测误差为 $\delta = y(k) - y_m(k)$, 在根据参考输入确定本次期望输出序列向量时做以下修正:

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

然后再进行本次预测优化控制, 就可以实现反馈校正.

3 初值问题(Initial value problems)

由式(8)不难看出: 算法运行前要给定一个初始的控制量值 $\mathbf{u}^0 = [u(k)^0 \ u(k+1)^0]^T$ 才能实现迭代过程, 在此将该值称为初值. 对局部优化算法(如上文提到的各种方法), 根据数值迭代的原理可知: 若参数选择适当, 算法最终将收敛到初值点处目标性能函数下降方向的一个最近极小值. 若 J 是 \mathbf{u} 的单峰函数, 只有一个极小点, 则将初值点选在何处只会影响算法的收敛速度, 并不会影响最优值的获取; 若 J 是 \mathbf{u} 的多峰函数, 具有多个极值点, 则初值问题不仅影响收敛速度, 还会影响优化结果, 有时甚至会得到一个远非期望的极小点, 导致控制系统的性能下降, 可靠性难以保证. 由于神经网络预测控制的应用对象大多具有经典方法难以建模的强非线性, 使得 J 一般都是多峰函数, 因此初值问题至关重要.

3.1 问题描述(Problems formulation)

为便于图示说明初值问题, 考虑一步预测 $d=1$, 式(6)变为

$$J = [y_m(k+1) - y_r(k+1)]^2 + \lambda[u(k) - u(k-1)]^2. \quad (17)$$

由式(1)知 f 对 $u(k)$ 具有任意复杂的非线性关系, 再由式(5)(17)知 J 是 $u(k)$ 的函数也具有复杂的非线性关系. 若按文献[3-5]提供的方法确定 $u(k-1)$ 的值为初值, 以 $u(k)$ 为横坐标, 由式(17)表达的 J 为纵坐标, 绘制可能的两条曲线如图2所示.

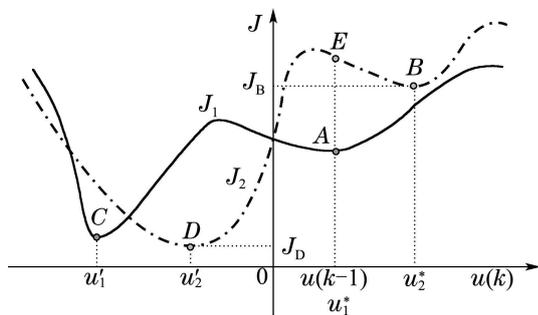


图2 初值问题示意图

Fig. 2 Schematic diagram of initial value problem

根据图2可以看出, 将 $u(k-1)$ 的值确定为初值可能会出现两个主要问题:

1) 优化后控制量不变, 不能实现有效控制.

对图中的曲线 J_1 , 初值确定在 A 点处, 此时参考输出改变: $y_r(k+1) \neq y_r(k)$, 系统期望的极值点在 C 点处, 然而由于 A 点本身是一个极小点, 所以导致局部优化算法收敛于 A 点而不是 C 点, 即优化后有: $u(k+1)$

$= u_1^* = u(k-1)$, 控制量优化前后没有发生改变. 从控制系统外面来看就是参考轨迹发生改变时不能进行有效控制.

2) 优化控制后输出误差较大.

对图中的曲线 J_2 , 初值确定在 E 点处, 系统期望的极值点在 D 点处, 局部优化迭代算法将收敛到 B 点. 对比 D, B 两点的目标性能函数值可知: $J_B \gg J_D$, 由图可以看出对式(17)所示的第2项有: $\lambda[u_2^* - u(k-1)]^2 < \lambda[u_2^* - u(k-1)]^2$, 所以 B 点在式(17)第1项的值远大于 D 点, 而第1项 $[y_m(k+1) - y_r(k+1)]^2$ 表征输出误差的大小, 因此在这种初值确定方法及局部优化控制作用下输出误差较大.

根据一步预测控制情况不难推知, 对多步预测控制也会出现相同的问题.

3.2 确定方法(Method to determine initial value)

目标性能函数式(6)的构造包含3部分, 若令

$$J_e = \sum_{j=1}^d [y_m(k+j) - y_r(k+j)]^2, \quad (18a)$$

$$J_u = \sum_{j=0}^{d-1} [u(k+j) - u(k+j-1)]^2, \quad (18b)$$

则式(6)可表示为

$$J = J_e + \lambda J_u, \quad (19)$$

其中: J_e, J_u 分别表征输出误差与控制增量的大小, λ 为权重因子, 且有: $\lambda \geq 0, J \geq 0, J_e \geq 0, J_u \geq 0$.

令

$$\begin{cases} J_e = 0, & \mathbf{u} = \mathbf{u}_r, \\ J_u = 0, & \mathbf{u} = \mathbf{u}_u. \end{cases} \quad (20)$$

由式(18b)(20)可直接求得 \mathbf{u}_u ,

$$u(k-1) = u_u(k) = \cdots = u_u(k+d-1). \quad (21)$$

即控制量 \mathbf{u}_u 中的全部元素等于上一时刻的控制量 $u(k-1)$ 不变. 当 $\mathbf{u} = \mathbf{u}_r$ 时, 由式(18a)(20)可得 $y_m(k+j) = y_r(k+j)$, 即预测输出等于给定参考值.

构造目标性能函数的目的是希望在最小的输出误差与最小的控制增量之间进行折衷, 换句话说, 笔者希望将控制量取在 \mathbf{u}_r 与 \mathbf{u}_u 之间的一个点, 该点处目标性能函数为极小值. 所以本文提出的初值确定思路是: 将初值定在 \mathbf{u}_r 处, 通过修正权重因子 λ 来确保 \mathbf{u}_r 与 \mathbf{u}_u 之间一定存在极小值, 这样局部迭代优化算法将收敛到该值, 从而保证系统的控制性能.

将 \mathbf{u}_r 定为初值, 首先要解决如何获取 \mathbf{u}_r 的问题. 如果根据式(18a)直接求 \mathbf{u}_r 要解复杂非线性方程组, 因此求解困难. 由上文分析知: $\mathbf{u} = \mathbf{u}_r$ 时, $y_m(k+j) = y_r(k+j)$, 此式表明 $y_m(k+j)$ 已知, 要求的是相应的控制量 \mathbf{u}_r , 显然这个问题恰好是建立预测模型的逆问题, 可以通过建立预测模型的逆神经网络来求解.

由图1所示的一步预测模型知, 只需将图中的 $y_m(k+1)$ 代替为 $y_r(k+1)$ 并与 $u(k)$ 位置互换即可构成逆神经网络模型. 逆神经网络的训练也无需额外的样本点, 只需将原样本点中的输入 $u(k)$ 改为输出, 输出 $y(k+1)$ 改为原输入 $u(k)$ 所在位置即可实现逆网络的训练. 设逆网络的映射关系数学表达式为

$$u_r(k) = LW' \times (1 + \exp(-IW' \times p'(k) - b'_1))^{-1} + b'_2. \quad (22)$$

将 $p'(k) = [y_r(k) \cdots y_r(k - n_y + 1) \ y_r(k + 1) \cdots u_r(k - n_u + 1)]^T$ 代入式(21)可求得 u_r 中的第1个元素 $u_r(k)$, 更新 $p'(k) = [y_r(k+1) \cdots y_r(k - n_y + 2) \ y_r(k + 2) \cdots u_r(k - n_u + 2)]^T$, 代入式(21)可求得 u_r 中的第2个元素 $u_r(k + 1)$, 如此不断更新 $p'(k)$ 代入式(21)即可求得 $u_r = [u_r(k) \ u_r(k+1) \ \cdots \ u_r(k+d-1)]^T$.

下面说明如何修正权重因子 λ 来确保 u_r 与 u_u 之间一定存在极小值. 为便于图示仍以一步预测为例, 根据式(17)–(20)绘制一步预测目标性能函数对控制量 $u(k)$ 的曲线如图3所示.

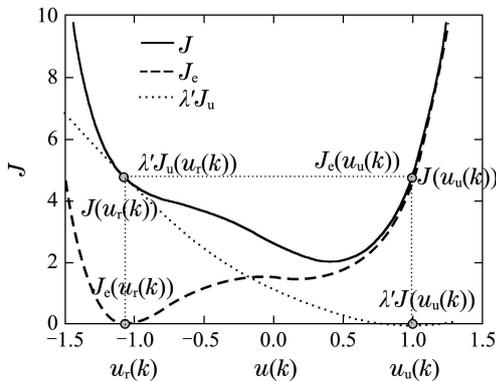


图 3 λ 修正方法图

Fig. 3 Figure of method to modify λ

图3中 $u_r(k) < u_u(k)$, 期望在区间 $[u_r(k) \ u_u(k)]$ 中存在极小值. 令 $J(u_r(k)) = J(u_u(k))$.

因 $J_e(u_r(k)) = \lambda' J_u(u_u(k)) = 0$, 故由式(18)–(19)及图3可得以下关系式成立

$$\lambda' = \frac{J_e(u_u(k))}{J_u(u_r(k))}. \quad (23)$$

注 1 需要说明的是当 $J_u(u_r(k)) = 0$ 或 $u_r(k) = u_u(k)$ 时, 根据式(18)–(20)此时只要保持控制量不变即有 $J = 0$, 不必再进行后续迭代优化. 因此如需继续优化必有: $J_u(u_r(k)) > 0, u_r(k) \neq u_u(k)$.

根据上文给出的方法求出 $u_u(k)$ 及 $u_r(k)$, 代入式(5)(17)–(19)(23)可计算出 λ' . 下面的定理1给出如何在一步预测中根据 λ' 修正 λ .

定理 1 对由神经网络建模表示的非线性系统, 若神经网络的数学模型使得由式(17)表示的一步预测目标性能函数满足以下条件:

- 1) J 对 $u(k)$ 连续且二阶可导;
- 2) 存在 $u_r(k)$ 使得 $J_e = 0$;
- 3) 式(17)中的 λ 满足 $\lambda \leq \lambda', \lambda'$ 由式(23)确定.

则当 $u(k) \in L(u_r(k), \tilde{u}_u(k))$ 时, J 存在极小值. 其中 $L(u_r(k), \tilde{u}_u(k))$ 表示从 $u_r(k)$ 到 $u_u(k)$ 的线段但不包含端点 $u_u(k)$.

证 当 $u_r < u_u$ 时, 首先证明 $\lambda = \lambda'$ 时的情况, 如图3所示: 显然根据1) J 在区间 $[u_r \ u_u]$ 上连续, 在区间 $(u_r \ u_u)$ 上一阶可导, 又有 $J(u_r(k)) = J(u_u(k))$, 因此根据罗尔中值定理知: 存在点 $\xi \in (u_r \ u_u)$, 使得 $J'(\xi) = 0$. 若满足条件的 ξ 存在极小值结论显然成立; 否则设 ξ_1 是距离 $u_r(k)$ 最近的一个极大值, 对任一 $\xi_i \in (u_r(k) \ \xi_1)$ 必有 $J(\xi_i) > J(u_r(k))$, 对于 $u_r(k)$ 左边的所有点 $u_i(k)$, 因为二次函数 J_u 单调递减, 而 u_r 是 J_e 的全局最小点, 因此有 $J(u_i) > J(u_r(k))$, 所以 u_r 是极小值, 定理结论成立.

其次证明 $\lambda < \lambda'$ 时的情况, 若在区间 $(u_r \ u_u)$ 上存在一阶导数为0的点, 则根据上面的证明知定理结论成立. 下面考虑不存在一阶导数为0的情况, 根据拉格朗日中值定理知: 在区间 $(u_r \ u_u)$ 上存在点 ξ 满足 $J'(\xi) = (J(u_u(k)) - J(u_r(k)))/(u_u(k) - u_r(k))$, 因 $\lambda < \lambda'$ 时根据式(17)–(20)及图3可得 $J(u_u(k)) > J(u_r(k))$, 故 $J'(\xi) > 0$. 由1)知 J 的一阶导函数连续, 进一步知区间 $(u_r \ u_u)$ 上所有点一阶导数大于0, 原因是若存在一阶导数小于0的点则根据连续性可以推知存在等于0的点, 这与前面的已知相矛盾. 换句话说, J 在区间 $(u_r \ u_u)$ 上单调递增, 所以此时 u_r 是极小值. 定理结论成立.

当 $u_r > u_u$ 时, 同理可证结论成立. 证毕.

根据定理1, λ 的修正方法为

$$\lambda = \begin{cases} \lambda', & \lambda > \lambda', \\ \lambda, & \lambda \leq \lambda'. \end{cases} \quad (24)$$

下面将一步预测中 λ 的修正方法推广到多步预测: 若令 $J(u_r) = J(u_u)$, 同理可得

$$\lambda' = \frac{J_e(u_u)}{J_u(u_r)}. \quad (25)$$

定理 2 对由神经网络建模表示的非线性系统, 若神经网络的数学模型使得由式(6)表示的 d 步预测目标性能函数满足以下条件:

- 1) J 对 u 连续且存在二阶偏导数;
- 2) 存在 u_r 使得 $J_e = 0$;
- 3) 式(6)中的 λ 满足 $\lambda \leq \lambda', \lambda'$ 由式(25)确定.

则 $u \in L(u_r, \tilde{u}_u)$ 时, J 存在极小值. $L(u_r, \tilde{u}_u)$ 表示 d 维空间中从点 u_r 到点 u_u 的线段但不包括端点 u_u .

证 对 d 维控制量 u 的后 $d - 1$ 个元素作以下线性变换:

$$u(k+i) = u_u(k+i) + \frac{u_r(k+i) - u_u(k+i)}{u_r(k) - u_u(k)} \cdot (u(k) - u_u(k)), \quad (26)$$

其中 $i = 1, 2, \dots, d-1$. 在此线性映射变换下, d 维空间中的点 \mathbf{u} 被一一映射为一维空间中的点 $u(k)$. 当空间中的点从 \mathbf{u}_r 沿直线逼近 \mathbf{u}_u , 相应的有一维空间中的点 $u(k)$ 从 $u_r(k)$ 沿直线逼近 $u_u(k)$. 将式(26)代入式(6), 则 J 是 $u(k)$ 的一元函数, 若能证明 $u(k) \in L(u_r(k), \tilde{u}_u(k))$ 时 J 有极小值, 则 $\mathbf{u} \in L(\mathbf{u}_r, \tilde{\mathbf{u}}_u)$ 时, J 存在极小值.

$$\text{令 } c_i = \frac{u_r(k+i) - u_u(k+i)}{u_r(k) - u_u(k)}. \quad (27)$$

因 \mathbf{u}_r 和 \mathbf{u}_u 是可求出的常值向量, 因此 c_i 为常数. 由式(26)–(27)(21)得

$$u(k+i+1) - u(k+i) = (c_{i+1} - c_i)(u(k) - u_u(k)). \quad (28)$$

将式(28)代入式(18b)可得

$$J_u = \left(1 + \sum_{i=1}^{d-2} (c_{i+1} - c_i)^2\right) (u(k) - u_u(k))^2. \quad (29)$$

式(29)左面是一个常数, 因此 J_u 也是一个二次函数且与一步预测的 J_u 具有相同的顶点. 分析定理1的证明过程知对 J_e 的形式没有其它要求, 所以完全可以按照定理1的方式, 根据式(25)确定 λ' , 并证明当 $u(k) \in L(u_r(k), \tilde{u}_u(k))$ 时 J 有极小值. 并进一步证明 $\mathbf{u} \in L(\mathbf{u}_r, \tilde{\mathbf{u}}_u)$ 时, J 存在极小值. 证毕.

综上本文提出的初值确定方法是: 在每一次滚动优化时, 进行迭代寻优前完成以下两步即可.

- 1) 通过递归调用由式(22)表示的逆神经网络获得 \mathbf{u}_r , 将 \mathbf{u}_r 确定为初值;
- 2) 对一步和多步预测分别根据式(23)(25)确定 λ' , 根据式(24)修正 λ .

4 系统控制原理(Principle of control system)

采用上文确定初值方法后, 神经网络预测控制系统结构如图4所示:

实施控制前先要通过实验方法获得非线性被控对象的输入输出采样数据, 利用采样数据对神经网络预测模型及逆神经网络计算模型离线进行学习, 使之能以一定精度逼近非线性被控对象和计算 \mathbf{u}_r .

控制开始后, 在每个采样周期内, 优化控制器先根据当前的系统输出 \mathbf{y} 和预测输出 \mathbf{y}_m 按式(16)对参考输出做反馈校正.

将校正后的参考输出 \mathbf{y}_r 作用于逆网络计算模型, 通过递归调用式(22)求得初值点 \mathbf{u}_r . 通过式(24)–(25)对参数 λ 进行修正.

将初始控制量 $\mathbf{u}^0 = \mathbf{u}_r$ 作用于神经网络预测模型得预测输出值 \mathbf{y}_m . 根据局部优化迭代方法式(8)进行

迭代至终止条件满足, 得优化后的次优控制量 \mathbf{u}^* , 用 \mathbf{u}^* 实现控制.

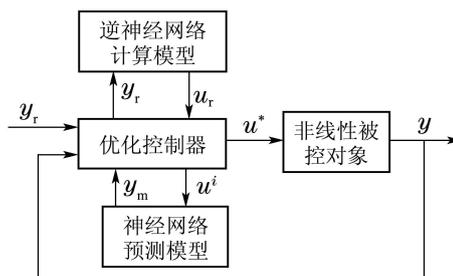


图4 控制系统结构图

Fig. 4 Structure diagram of control system

5 仿真实验(Simulation experiment)

仿真实验在MATLAB2010b环境下编程实现. 考虑如下离散非线性系统:

$$y_{k+1} = y_k^3 + \frac{y_k}{1 + y_k^2} - 0.2\sin(1.5y_k). \quad (30)$$

5.1 神经网络构建(Construct neural networks)

由式(30)知系统的预测模型输入 $\mathbf{p}(k)$ 为 $[y(k) \ u(k)]^T$, 维数 $R = n_y + n_u = 2$, 输出为1维的 $y(k+1)$; 逆网络模型输入 $\mathbf{p}'(k)$ 为 $[y(k) \ y(k+1)]^T$, 维数 $R = 2$, 输出为1维的 $u_r(k)$; 因此需要构造两个2输入1输出的BP神经网络, 其中一个网络用于预测模型, 另一个逆网络用于确定初值点 \mathbf{u}_r .

网络建立采用MATLAB神经网络工具箱中的函数newff. 预测模型神经元个数取30, 逆网络神经元个数取50. 隐层神经元取logsig函数, 输出神经元取线形purelin函数, 则神经网络的数学表达式分别如式(5)(22)所示.

通过以下方式产生样本: 在 $[-2 \ 2]$ 范围内随机生成1000个数据为输入控制量 u , 根据式(30)计算输出系统实际输出数据 y , 将这1000个输入输出数据对作为网络训练的样本.

BP神经网络的训练利用了MATLAB的神经网络工具箱中的train函数, 训练精度取 1×10^{-4} , 训练次数取500.

5.2 两种初值确定方法比较(Comparison of two methods to determine initial value)

为比较文献[3–5]的初值确定方法与本文提出的方法对控制系统的不同作用, 分别设计两种不同的控制方案:

方案1 神经网络预测控制制定为两步预测, 一步控制. 采用(N–R)局部优化算法进行滚动优化, 初值按文献[3–5]确定为上一时刻的控制量值, 即有: $\mathbf{u}^0 = [u(k-1) \ u(k-1)]^T$. 按式(8)进行迭代, 终止条件为: $\Delta u^i \leq 0.01$. 每次迭代按式(2)–(3)(11)–(12)计算Jacobian矩阵, 按式(2)–(3)(11)–(15)计算Hessian矩阵.

方案2 采用本文的方法在迭代优化前将初值设定在 $\mathbf{u}^0 = [u_r(k) \ u_r(k+1)]^T$, 按式(24)–(25)修正 λ , 其余全部同方案一。

分别按两种方案进行仿真实验, 在同一参考输出作用下将两方案的实际输出绘制如图5所示。

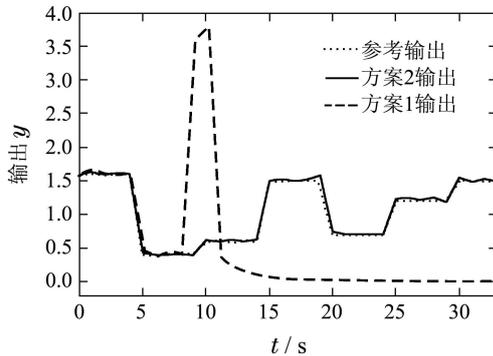


图5 控制效果对比

Fig. 5 Comparison of control accuracy

根据图5可知方案1中的初值确定方法出现了上文提到的两个问题, 先是出现较大误差, 然后控制量不变, 从而使得控制性能下降, 不能跟踪参考输出的变化。而方案2本文的方法不仅可以有效地跟踪参考输出, 而且误差较小, 有效的解决了初值问题。

6 结论(Conclusions)

本文指出了将初值确定为上一时刻的控制量会引起输出误差较大及控制量不变两个主要问题, 这会导致控制性能的下降。为解决此问题提出了一种新的动态确定初值的方法, 通过逆神经网络来求得输出误差最小点并将这一点确定为初值, 然后通过修正权重系数 λ 来保证在输出误差最小点与当前控制点之间存在极小值, 并在理论上进行了证明, 进一步保证了数值迭代局部优化算法能够收敛到该极小值。理论分析和仿真结果表明了这种方法可以提高控制系统的可靠性, 为神经网络预测控制的实际工程应用打下了良好的基础。

另外, 本文提出的初值确定方法对逆神经网络的要求较高, 实际应用时要根据样本数据适当选取神经元的个数, 不能产生较大计算误差, 否则控制系统的可靠性会降低。

参考文献(References):

[1] XI Y G, LI D W, LIN S. Model predictive control – status and challenges [J]. *Acta Automatica Sinica*, 2013, 39(3): 222 – 236.

- [2] 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 Weizhan, LOU Haichuan, YANG Aiping. An overview of Neural network predictive control for nonlinear systems [J]. *Control Theory & Applications*, 2009, 26(5): 521 – 530.)
- [3] TAN Y H, VAN CAUWENBERGHE A R. Optimization techniques for the design of a neural predictive controller [J]. *Neurocomputing*, 1996, 10(3): 83 – 96.
- [4] SØRENSEN P H, NØRGAARD M, RAVN O, et al. Implementation of neural network based non-linear predictive control [J]. *Neurocomputing*, 1999, 28(1): 37 – 51.
- [5] NAND K, 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.
- [6] SENDRESCU D, PETRE E, POPESCU D, et al. Neural network model predictive control of a wastewater treatment bioprocess. *Intelligent Decision Technologies* [M]. Berlin Heidelberg: Springer-Verlag, 2011: 191 – 200.
- [7] 李会军, 肖兵. 一种无约束多步递归神经网络预测控制器 [J]. *控制理论与应用*, 2012, 29(5): 642 – 648.
(LI Huijun, XIAO Bing. Multistep recurrent neural network model predictive controller without constraints [J]. *Control Theory & Applications*, 2012, 29(5): 642 – 648.)
- [8] ANTONIOU A, LU W S. *Practical Optimization: Algorithms and Engineering Applications* [M]. New York: Springer-Verlag, 2007.
- [9] NOCEDAL J, WRIGHT S J. *Numerical Optimization* [M]. New York: Springer-Verlag, 1999.
- [10] 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.)
- [11] ALIREZA R, SCOTT S. Identification of nonlinear systems using NARMAX model [J]. *Nonlinear Analysis*, 2009, 71(12): e1198 – e1202.
- [12] 谢新民, 丁锋. 自适应控制系统 [M]. 北京: 清华大学出版社, 2002.
(XIE Xinmin, DING Feng. *Adaptive Control Systems* [M]. Beijing: Tsinghua University Press, 2002.)
- [13] 丁锋. 系统辨识新论 [M]. 北京: 科学出版社, 2013.
(DING Feng. *System Identification – New Theory and Methods* [M]. Beijing: Science Press, 2013.)
- [14] 丁锋. 系统辨识—辨识方法性能分析 [M]. 北京: 科学出版社, 2014.
(DING Feng. *System Identification – Performance Analysis for Identification Methods* [M]. Beijing: Science Press, 2014.)
- [15] OSMAN GÜLER. *Foundations of Optimization* [M]. New York: Springer-Verlag, 2010: 377 – 383.

作者简介:

樊兆峰 (1972–), 男, 副教授, 博士研究生, 目前研究方向为神经网络预测控制、机器人控制, E-mail: fanzhaofeng72@163.com;

马小平 (1961–), 男, 教授, 博士生导师, 目前研究方向为计算机控制理论及应用, E-mail: xpma@cumt.edu.cn;

邵晓根 (1963–), 男, 教授, 目前研究方向为智能控制与信息处理, E-mail: shaoxg@xzit.edu.cn.