

文章编号: 1671-6612 (2019) 03-298-05

变风量空调系统温湿度解耦控制

闫秀联¹ 颜菲菲² 闫秀英³

- (1. 西安建筑科技大学华清学院 西安 710043;
2. 中联西北工程设计研究院有限公司 西安 710077;
3. 西安建筑科技大学信控学院 西安 710055)

【摘要】 变风量空调系统是一个非线性、大滞后、多变量的系统,简单的 PID 控制对非线性系统控制效果较差,控制精度不高。针对变风量空调系统控制过程中,送风量-房间温度、送风温度-房间湿度两控制回路间存在耦合的现象。采用神经网络算法进行解耦控制,以消除耦合。首先,应用最小二乘法建立被控对象控制回路的数学模型,其次,采用 PID 神经网络解耦控制算法和交叉解耦控制算法对存在耦合的回路进行解耦控制。最后,进行仿真分析,并加载到空调系统中进行验证。

【关键词】 变风量空调;温湿度控制;PID 神经网络;解耦控制

中图分类号 TP29 文献标识码 A

Variable Air Volume Air Conditioning System Temperature and Humidity Decoupling Control

Yan Xiulian¹ Yan Feifei² Yan Xiuying³

- (1. Xi'an University of Architecture and Technology Hua Qing College, Xi'an, 710043;
2. China United Northwest Institute for Engineering Design&Research Co.,Ltd, Xi'an, 710077;
3. School of Information and Control Engineering, Xi'an University of Architecture & Technology, Xi'an, 710055)

【Abstract】 Variable air volume air conditioning system is a non-linear, large delay, multi-variable system, simple PID control on the non-linear system control effect is poor, control precision is not high. In the control process of variable air volume air conditioning system, the two control loops of air supply - room temperature and air supply temperature-room humidity are coupled. In this paper, neural network algorithm is used for decoupling control to eliminate coupling. Firstly, the mathematical model between control loops of the controlled object is established by using the least square method. Secondly, PID neural network decoupling control algorithm and cross decoupling control algorithm are adopted to decouple control the coupled circuits. Finally, the simulation analysis is carried out and the validation is carried out in the air conditioning system.

【Keywords】 Variable air volume air conditioning system; Temperature and Humidity control; PID neural network; Decoupling control

0 引言

随着科技水平的不断提高和高精密仪器设备的不断投入使用,对科学实验环境的温湿度控制精度提出了更高的要求^[1]。实际的变风量空调系统 (Variable Air Volume, VAV) 中,多个控制回路之

间存在相互耦合,相互干扰的情况,为了使系统能够稳定运行,就必须对耦合回路进行解耦控制^[2-4],目前变风量空调系统解耦多采用前馈解耦,这种方法依赖于被控对象精确的数学模型,而实际系统中被控对象往往具有非线性时变特性,采用传统的解

基金项目:陕西省教育厅专项科研基金项目(17JK1002)

作者(通讯作者)简介:闫秀联(1985-),女,硕士,讲师, E-Mail: yanxiulian325@163.com

收稿日期:2019-03-11

耦控制方法很难达到预期的控制品质,甚至会引起系统的不稳定^[5-8]。基于此本文采用了 PID 神经网络智能解耦控制方法,该方法结构简单,易于实现。且解耦器与控制器于一体,不仅能够解决传统解耦控制方法精确度低的缺点,同时又能实现对系统的实时在线控制^[9,10]。目前大多数解耦控制都停留在仿真阶段,而本文设计的基于末端温湿度耦合现象的交叉解耦和 PID 神经网络解耦两种控制算法^[11-13],在软件仿真的基础上,进行了在线控制,并对仿真和在线控制效果进行了对比分析。

1 空调系统末端温湿度解耦控制原理

空调系统控制过程中,当送风温度降低时,房间温度会相应降低,而房间湿度则会相应升高。夏季工况,送风量增大时,房间温度会相应降低,房间湿度会相应升高。即房间温湿度与送风量、送风温度都有关系。因此本文研究了空调系统末端温湿度回路运行过程中存在的耦合作用,并对耦合回路采用 PID 神经网络算法进行解耦控制,如图 1 所示。房间温湿度实测值、房间温湿度设定值为 PID 神经网络的输入层,送风量设定值和送风温度设定值为 PID 神经网络的输出层,输出层参数与各自对应参数的实测值进行对比,控制房间温湿度,该控制方法是通过调节输入层与隐含层,隐含层与输出层之间的权值来达到房间温湿度解耦目的。

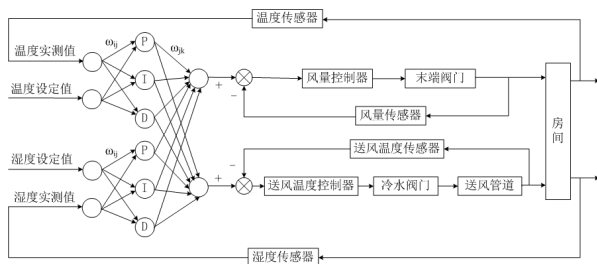


图 1 PID 神经网络解耦控制框图

Fig.1 PID neural network decoupling control block diagram

2 PID 神经网络解耦控制算法

PID 神经网络是一种动态前馈控制,在神经网络中融入了 PID 控制规律,不仅具有传统 PID 控制器的优点,同时又具有并行结构和学习记忆功能,可以实现任意精度逼近任意连续函数。通过在

线学习,根据对象参数变化时对系统输出性能的影响来调整连接权值。从而改变神经网络中比例、积分、微分的强弱,加快系统的收敛速度,使系统具备良好的动态性能和静态性能,来达到解耦目的。算法原理介绍如下:

2.1 前向算法

2.1.1 输入层

输入层包括 2 个神经元,分别为给定值和实测值,在任意采样 k 时刻,其输入:

$$net_1(k) = r(k) \tag{1}$$

$$net_2(k) = y(k) \tag{2}$$

输入层神经元的状态为:

$$u_i(k) = net_i(k) \tag{3}$$

输入层神经元的输出为:

$$x_i(k) = \begin{cases} 1 & u_i(k) > 1 \\ u_i(k) & -1 \leq u_i(k) \leq 1 \\ -1 & u_i(k) < -1 \end{cases} \tag{4}$$

式中: $i=1,2; j=1,2,3$ 。

2.1.2 隐含层

隐含层共 3 个神经元,分别为 1 个比例神经元,1 个积分神经元和 1 个微分神经元,它们各自的输入值均为:

$$net'_i(k) = \sum_{i=1}^2 \omega_{ij} x_i(k) \tag{5}$$

式中: $j=1,2,3$ 。 ω_{ij} 是输入层至隐含层的连接权值。

比例元状态为:

$$u'_1(k) = net'_2(k) \tag{6}$$

积分元状态为:

$$u'_2(k) = net'_2(k) + u'_2(k-1) \tag{7}$$

微分元状态为:

$$u'_3(k) = net'_3(k) - net'_3(k-1) \tag{8}$$

隐含层各神经元输出为:

$$x'(k) = \begin{cases} 1 & u'_j(k) > 1 \\ u'_j(k) & -1 \leq u'_j(k) \leq 1 \\ -1 & u'_j(k) < -1 \end{cases} \tag{9}$$

式中: $j=1,2,3$ 。

2.1.3 输出层

PID 神经网络输出层结构比较简单,包含一

个神经元，完成网络的综合输出功能，其总输入为：

$$net'_i = \sum_{j=1}^3 w'_j x'_j(k) \tag{10}$$

式中： $x'_j(k)$ 为隐含层各神经元输出值； w'_j 为隐含层至输出层的连接权值。输出层神经元的状态函数与比例元的状态函数相同，为：

$$u''(k) = net''(k) \tag{11}$$

输出层的输出 $x''(k)$ 为：

$$x''(k) = \begin{cases} 1 & u''(k) > 1 \\ u''(k) & -1 \leq u''(k) \leq 1 \\ -1 & u''(k) < -1 \end{cases} \tag{12}$$

而 SPIDNN 的输出 $v(k)$ 等于输出层神经元的输出，即：

$$v(k) = x''(k) \tag{13}$$

2.2 反传算法

PID 神经网络学习的目标是使式（14）为最小。

$$E = \frac{1}{l} \sum_{k=1}^l [r(k) - y(k)]^2 = \frac{1}{l} \sum_{k=1}^l e^2(k) \tag{14}$$

按照梯度法调节 PID 神经网络权值，使控制量不断接近控制目标。

2.2.1 隐含层至输出层

隐含层至输出层的权值迭代公式：

$$w'_j(n_0+1) = w'_j(k) - \eta_j \frac{\partial E}{\partial w'_j} \tag{15}$$

其中：

$$\begin{aligned} \frac{\partial E}{\partial w'_j} &= -\frac{2}{l} \sum_{k=1}^l [r(k) - y(k)] \text{sgn} \frac{y(k+1) - y(k)}{v(k) - v(k-1)} x'_j(k) \\ &= -\frac{1}{l} \sum_{k=1}^l \delta'_i(k) x'_j(k) \end{aligned}$$

2.2.2 输入层至隐含层

输入层至隐含层的权值迭代公式：

$$w_j(n_0+1) = w_j(n_0) - \eta_i \frac{\partial E}{\partial w_{ij}} \tag{16}$$

其中：

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}} &= \frac{\partial E}{\partial v} \frac{\partial v}{\partial x''} \frac{\partial x''}{\partial u''} \frac{\partial u''}{\partial net''} \frac{\partial net''}{\partial x'_j} \frac{\partial u'_j}{\partial u'_j} \frac{\partial net'_i}{\partial w_{ij}} \\ &= -\frac{1}{l} \sum_{k=1}^l \delta_i(k) x_i(k) \end{aligned}$$

3 仿真分析

把空调系统温湿度回路进行简化处理，对前向通道建立线性传递函数，在研究过程中仅对单个控制回路进行简化建模。实际应用中与其它环节还存在耦合。采用 PID 神经网络解耦控制方法，研究变风量空调系统中房间温湿度之间的耦合现象。在夏季工况下，采用 PID 神经网络解耦控制和交叉解耦控制，对辨识模型进行仿真控制和对比分析。

3.1 温湿度解耦控制建模

如图 1 所示，研究对象为双闭环控制系统，风量控制器与风阀执行器集成一体，无法直接采集风量对象的输入输出信号，外环温度反馈也无法断开，即不满足开环辨识条件，因此房间温度内外环控制回路需采用闭环辨识法。

风量设定值作为输入，风量反馈值作为输出，采用最小二乘法闭环辨识得到风量传递函数：

$$G_1(S) = \frac{S + 0.05983}{S + 0.06971} e^{-5s} \tag{17}$$

冷冻水阀门开度作为输入，送风温度作为输出，采用开环辨识得到送风温度传递函数：

$$G_2(S) = \frac{0.000962}{S + 0.1761} e^{-42s} \tag{18}$$

从图 1 可以看出，送风量、送风温度的变化，都会影响房间内温湿度，因此末端送风量——房间温度，送风温度——房间相对湿度两个控制回路间存在耦合关系，如图 2。图中 G11 表示送风量和房间温度之间的传递函数，G22 表示送风温度和房间湿度之间的传递函数，G12 表示送风温度和房间温度之间的传递

函数, G21 表示送风量和房间湿度之间的传递函数。

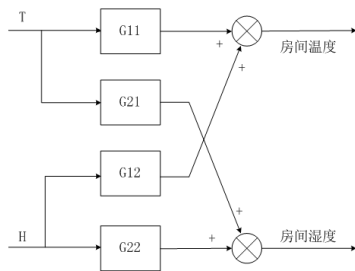


图 2 温湿度耦合传递函数框图

Fig.2 Diagram of temperature and humidity coupling transfer function

温湿度解耦过程中被控制对象传递函数如下:

$$G11 = \frac{0.003617s + 0.003466}{s + 0.9582} e^{-50s} \quad (19)$$

$$G22 = \frac{0.08532}{S + 0.1761} e^{-48s} \quad (20)$$

$$G12 = \frac{0.2206}{S + 0.4581} e^{-48\tau} \quad (21)$$

$$G21 = \frac{0.0164}{S + 5.668} e^{-50\tau} \quad (22)$$

3.2 仿真结果分析

将房间温湿度分别设定在 25℃ 和 50%, 通过系统建模, 采用交叉解耦、PID 神经网络解耦对变风量空调系统末端温湿度进行控制仿真, 得到房间温度、相对湿度解耦控制图, 其与交叉解耦控制进行对比, 如图 3, 图 4 所示。

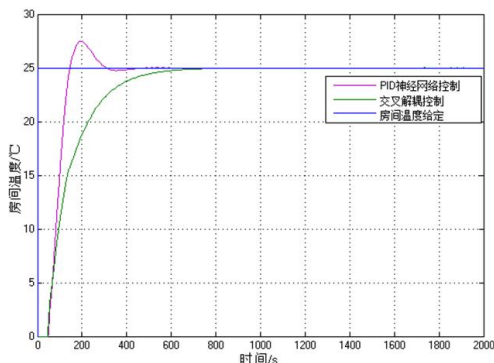


图 3 房间温度解耦控制

Fig.3 Room temperature decoupling control

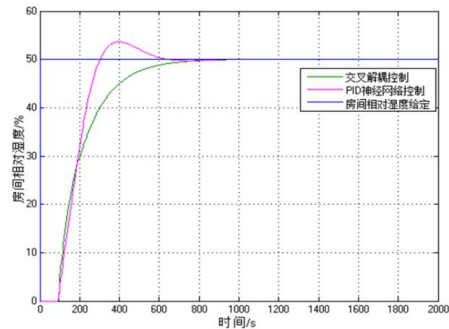


图 4 房间相对湿度解耦控制

Fig.4 Room relative humidity decoupling control

房间温度解耦控制仿真过程中, PID 神经网络解耦控制系统响应时间为 450s, 超调量为 10% 左右; 交叉解耦控制系统响应时间为 700s, 超调量为 0%。房间相对湿度解耦控制过程中, PID 神经网络解耦控制系统响应时间为 620s, 超调量为 5% 左右; 交叉解耦控制系统响应时间为 800s, 超调量为 0%。由图 3、图 4 综合对比可以看出, PID 神经网络解耦控制相比交叉解耦控制可以使系统尽快的达到稳定但在控制过程中会出现超调现象。

4 实验验证

4.1 解耦前控制

对实验控制对象采用 LabVIEW 软件编写在线控制程序, 加载到实际空调系统, 将房间温湿度分别设定为 25℃ 和 50% 进行在线控制。图 5 为解耦前房间温度控制, 图 6 为解耦前房间相对湿度控制。

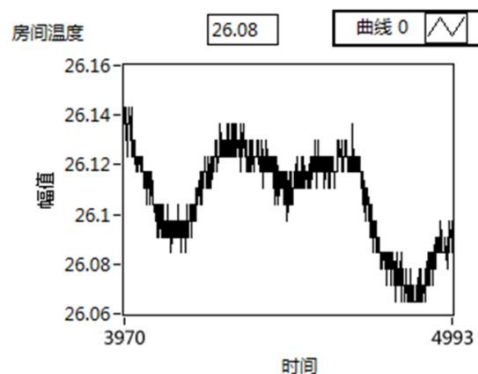


图 5 解耦前房间温度控制

Fig.5 Room temperature control without decoupling

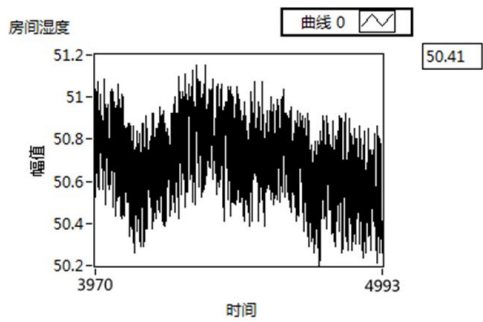


图 6 解耦前房间相对湿度控制

Fig.6 Room relative humidity control without decoupling

解耦前控制程序在线实验验证，系统运行到 3970s 时，房间温湿度基本达到稳定，房间温度在 $26.1\pm 0.04^{\circ}\text{C}$ 波动，房间相对湿度在 $50.6\pm 0.4\%$ 波动。

4.2 交叉解耦控制

实验控制对象采用 LabView 软件编写交叉解耦控制程序，并将程序加载到系统上，对系统进行在线控制，图 7 为交叉解耦后房间温度控制图，图 8 为交叉解耦后房间相对湿度控制图。

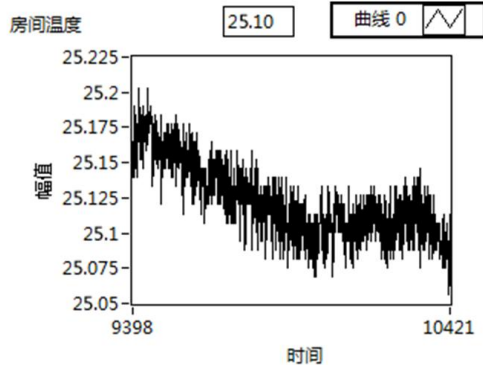


图 7 交叉解耦后房间温度控制

Fig.7 Room temperature control after cross decoupling

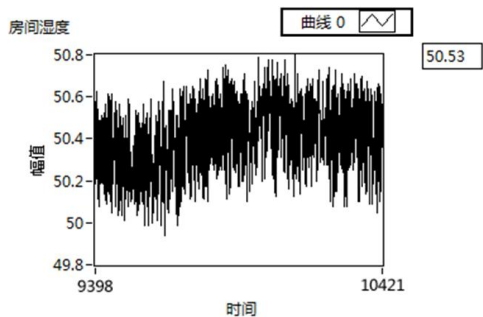


图 8 交叉解耦后房间相对湿度控制

Fig.8 Room relative humidity control after cross decoupling

交叉解耦控制在线进行实验验证过程中，系

统运行到 9398s 时，房间温湿度基本达到稳定，房间温度在 $25.1\pm 0.05^{\circ}\text{C}$ 波动，相对湿度在 $50.4\pm 0.2\%$ 波动。

4.3 PID 神经网络解耦控制

对实验控制对象采用 LabView 软件编写 PID 神经网络解耦控制程序。图 9 为 PID 神经网络解耦后房间温度控制，图 10 为 PID 神经网络解耦后房间相对湿度控制。

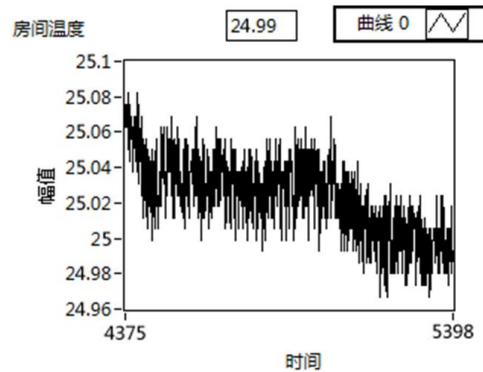


图 9 PID 神经网络解耦后房间温度控制

Fig.9 PID neural network decoupled room temperature control

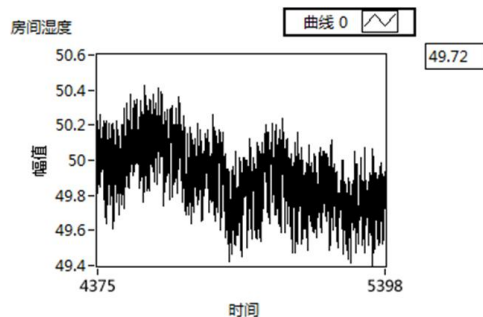


图 10 PID 神经网络解耦后房间相对湿度控制

Fig.10 Room relative humidity control after PID neural network decoupling

PID 神经网络解耦控制在线实验验证，系统运行到 4375s 时，房间温湿度基本达到稳定，房间温度在 $25.02\pm 0.04^{\circ}\text{C}$ 波动，相对湿度在 $50\pm 0.4\%$ 波动。

由以上实验结果表明，将房间温湿度分别设定在 25°C 和 50% 时，交叉解耦控制和 PID 神经网络解耦控制相比未解耦前，都有效消除了房间温湿度控制回路之间的耦合现象，其中 PID 神经网络解耦比交叉解耦控制使系统达到稳定时，所用时间更短。

(下转第 342 页)