摘 要

随着科学技术和生产力的不断发展，在实际工业生产过程中受控对象机理越来越复杂，表现出如下一些特征：多输入多输出、时变性、强耦合、时滞性、高指标要求等特点。此类复杂系统和对象，由于难以建立精确数学模型，使得传统控制理论很难应用，甚至不能控制。这就要求在控制中，控制不仅不依赖于对象数学模型，并且能够在线自调整，以满足实时控制的要求。

在众多具有以上特性的对象中，多温区温控系统无疑是一种较为典型的此类对象。该对象不仅具有参数不确定的特性，而且由于其本身的“多温区”特点，温区间往往存在较强的耦合。因此鉴于多温区温控系统滞后、强耦合、模型不确定的特性，针对传统 PSD 算法所存在的不足之处，本文重点研究了一种能够克服滞后，且不依赖于精确模型的自适应 PSD 算法。本文在单温区滞后温度对象中对该算法进行了研究，提出了一定的改进措施，并通过大量的仿真证明了所提出算法的有效性。

同时，本文将 PSD 算法在一多温区温控系统中作了推广。传统的 PSD 算法所具有的易于工程实现的特点和神经网络在非模型控制上所具有的优势，二者的结合对模型不确定性对象的控制是一个好的方法。本论文即结合二者之优点，提出了一种 PSD 网络算法的结构。并比较了现有的几种 PSD 网络控制器的结构特点，从理论上阐述了 PSD 网络算法的优势所在。

最后，在所建立的多温区测控系统模型基础上，针对模型的大滞后、强耦合这个特点，将 PSD 网络解耦控制算法应用到该对象中，给出了算法的实现步骤。仿真结果表明，所提 PSD 网络算法不仅具有 PSD 算法的能克服对象的滞后、参数时变且简单易行的特性，而且具有一定的解耦能力。

关键词：多温区系统，滞后，耦合，PSD 算法，仿真

ABSTRACT

With the continuous development of science and technology of productive forces, objects in the actual process of industrial production become more and more complex, and some obvious characters were shown, such as MIMO、time-variability、coupling、large-delay、high-indicators needed and so on. For such complex systems and controlled plants, it is very difficult to establish its accurate module and get a good control result with traditional control algorithms. So model free and self-tuning on line was required to realize real-time control.

Among many plants having characters as above, Multi-temperature zone measurement&control system is a common type. Not only the plant have parameter uncertainty, but also it have strong coupling factor because of its multi-temperature zones structure. So because of large-delay、strong coupling、model uncertainty, especially this paper study one kind of self-tuning PSD algorithm, which doesn't need the plant's model and can overcome the plant's time-delay. The paper validate the self-tuning PSD algorithm in the temperature plant with time-delay, meanwhile some improved methods was prosposed. Then, the simulation results have proved that the algorithm is effective in applications.

Meanwhile, the PSD algorithm was generalized in a multi–temperature zone measurement&control system. That the easy realization in applications of traditional PSD algorithm combine with the advantages of neuron network for model free control, is a good method for the plant having uncertainty model. By combining the advantages of those two methods, and a new structure of PSD network was proposed finally. Furthermore, various kinds of PSD network controller were compared in this paper, and the advantage of PSD network algorithm was analyzed in theory.

Lastly, according to the multi-temperature zone plant with time delay and strong coupling, a kind of PSD network algorithm was

applicated, and the approach of algorithm is given, too. The simulation results have proved that the algorithm is availability and adaptability.

KEY WORDS: multi-temperature zone plant, time delay, coupling, PSD algorithm, simulation

原创性声明

本人声明，所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了论文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得中南大学或其他单位的学位或证书而使用过的材料。与我共同工作的同志对本研究所作的贡献均已在论文中作了明确的说明。

作者签名：李长征　日期：2008年5月26日

学位论文版权使用授权书

本人了解中南大学有关保留、使用学位论文的规定，即：学校有权保留学位论文并根据国家或湖南省有关部门规定送交学位论文，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以采用复印、缩印或其它手段保存学位论文。同时授权中国科学技术信息研究所将本学位论文收录到《中国学位论文全文数据库》，并通过网络向社会公众提供信息服务。

作者签名：李长征　导师签名：[illegible]　日期：2008年5月26日

第一章 绪论

1.1 研究背景及意义

随着科学技术和生产力的不断发展，在实际工业过程中受控对象机理越来越复杂，表现出如下一些特征：多输入多输出、时变性、强耦合、时滞性、非线性、不确定性、高指标要求等特点。诸如此类的复杂系统和对象，由于无法建立系统的精确数学模型，从而使得传统控制理论很难取得很好的控制效果，甚至不能控制。同时，对象在噪声、负载扰动等因素的影响下，过程参数甚至模型结构均会发生较大变化，也限制了一些控制方法的使用。时滞的存在，使得被控量不能及时地反映控制信号的动作，控制信号的作用需要经过纯滞后时间以后才影响被控量；另外，当对象受到干扰而引起被控量改变时，控制器产生的控制作用不能立即对干扰产生抑制作用。因此，时滞对象的闭环控制系统必然存在较明显的超调量和较长的调节时间。这就要求在控制过程中，控制方法不仅能不依赖于数学模型，并且能够在线自调整，以满足实时控制的要求。

在众多具有以上特性的对象中，多温区温控系统无疑是一种较为典型的此类对象。该对象不仅过程参数变化难以确定，而且由于其本身的“多温区”特性，往往存在较强的耦合特性和时滞特性。同时，有着多个温区的测控系统是一种典型的多输入多输出(MIMO)对象，在无线电元器件的烧结、军工产品的热处理等方面有着非常广泛的应用，它是多种大型加工生产中的重要生产环节，对各个温区温度控制的精度可以直接影响到产品的质量和可靠性。对于多温区测控系统，各个温区之间由于存在着三种热传递方式：传导、对流、辐射，会使得各个回路之间存在较强的耦合性，同时使单个温区的对象参数难以确定；因而解决多回路的耦合，克服对象参数时变，对提高多温区测控系统的控制品质有重要意义。

因此鉴于多温区温控系统大滞后、强耦合、模型不确定的特性，寻求一种能够克服滞后，且不依赖于精确模型的算法具有一定的理论和工程价值。

1.2 国内外研究现状

现在解决多温区对象的主要方法有自适应控制、鲁棒控制、预测控制[1]。然而，这些方法的前提均是以获得较为精确的对象数学模型为基础的。然而，大多数工业过程对象的模型都是很难或者不可能完全获得其稳定的精确模型的。当模型失配时，若仍然采用以上控制方法系统性能难以得到保证，使得控制效果难以

达到实际的目标和要求。

与此同时，过程控制理论也在我国已进入到一个崭新的发展时期。指导过程控制的理论从“经典”到“现代”，从“现代”到“智能”也提出了很多不同类型的控制策略。当然，理论研究与工业实际应用有统一性，但是，理论研究与工业应用又有着各自追求的不同特点。工业应用特别强调选取运行可靠、调试方便、效果显著、技术成熟的控制策略。所以，理论上完整且系统的控制策略在实践上不一定能获得广泛的应用。

文献[2]依据有关对日本工业过程控制技术现状及其动向的调查资料，对各类控制策略的目前应用和发展前景作了分析和预测。由于日本的工业过程控制技术在国际上较先进且又注重实效，因此，其调查结果具有相当的代表性。据统计，以有效回答企业控制回路的总数为基数，各类控制策略的应用以及企业计划对各种控制策略占有回路数改进情况如表(1-1)示：

表 1-1 各种控制策略应用现状及前景

控制策略 比较项目	常规 PID	增强 PID	高级算法	智能算法	手动
目前应用情况	84%	7%	2%	1%	6%
计划改进情况	67%	17%	7%	7%	2%

表(1-1)中，增强 PID 控制包括二自由度 PID，纯滞后时间补偿 PID，增益自整定 PID，自整定 PID。高级控制算法包括最优控制、优化控制、观测器、卡尔曼滤波、H_∞最优控制、模型预测控制及自适应控制方法。智能算法则包括基于规则控制、模糊控制及神经网络控制方法。

由表(1-1)可见，常规 PID 现仍将大面积占领过程控制领域，但将会逐步缩小其占领面积；增强型 PID 倍受企业青睐，将被继续较多的开发应用；人工智能和高级算法控制也将得到逐渐的推广。

同样也可以看出理论上完整、先进且具有优秀控制品质的控制算法不一定能真正付诸实践，得到广泛的推广和应用。由此，我们可以洞悉理论研究与工业应用之间的接轨还需我们付出很大的努力。

大量的工程实践证明过程控制的主要目的：一是调节，即“应使最初的干扰以期望的动态特性消除”；二是跟踪，即“应使受控对象的输出渐近跟踪给定值”[3]。常规 PID 调节以消除误差和减少外扰为目的，对于受控对象结构和参教未知的场合，通过调整调节器参数，便可获得较好的控制效果，并且由于其参数物理概念明确，调试简单，因而在实际工业过程控制中得到广泛应用。但随着社会生

产力的提高，许多工业被控对象和过程变的越来越复杂，其明显的特点是含有大纯滞后，并且一般都具有一定的非线性和参数时变性，而 PID 参数的整定依赖于对象数学模型，不易在线调整，只对很小范围内的不确定性有效，同时一些增强型 PID 参数的整定也是困难所在，耗时费力，这就使得仅用常规 PID 调节难以做到任意工况下的最优控制，不能满足控制票求。

但由于 PID 控制的参数物理概念清晰、调试方向明确性以及在工业控制应用中的简单易实现性，PID 控制算法仍不失其对于进一步研究控制策略的借鉴性。近年来，重新认识常规 PID 调节的研究工作在国内外受到人们的广泛关注并且取得了很大的研究成果。由于调节质量的双重要求，开发了两自由度 PID 控制[7][8]；由于工业过程变量间的相关性，开发了解耦 PID 控制[9][10]；由于对象的纯滞后性，开发了 Smith 预估补偿控制[11][13]；由于系统的非线性和时变性，开发了增益整定和自整定 PID 控制[14]；这些增强功能的 PID 控制算法在实际的工业过程控制中也遇到了困难。由于许多被控过程机理较复杂，具有高度非线性、时变不确定性和纯滞后等特点。在噪声、负载扰动等因素的影响下，过程参数甚至模型结构均会发生变化。这就要求在控制中，控制不仅不依赖于数学模型，并且能够在线自调整，以满足实时控制的要求。

目前，人们已经注意将智能控制、模糊控制和神经网络技术引入常规 PID 控制之中，成为目前国际上最热门的研究课题之一。这里浅谈神经网络的一些特点。神经网络能够充分任意地逼近任何复杂的非线性系统，所有定量和定性分析都等势分布储存于神经网络内的各种神经元中，能够学习和适应严重不确定系统的动态特性，故有很强的鲁棒性和容错性。这些特点显示了神经网络在解决高度非线性和严重不确定性系统方面的潜力。因此，将神经网络技术引入 PID 控制中成为新的研究热点之一。其中，自适应 PSD 算法以其易于实现，结构明确的特点成为了其中的佼佼者。考虑到神经网络的结构复杂、学习速率、以及实现难度等方面的局限性以及工业现场控制设备的局限性，PSD 控制相对于基于神经网络的 PID 控制来说，则兼具了自学习能力和易于工程实现的两方优点。

PSD 算法是数字 PID 算法与神经网络的有效结合，利用单个神经元节点对 PID 的三个参数快速拟合，以达到实时控制的效果，其实质是一种非线性在线自适应 PID 控制。PSD 控制器由于其物理意义明确、参数自适应、不需要对控制对象建模等优点在许多工业控制领域得到了应用，包括：调速、炉温调节、船舶自在调速系统[15~18]。人们在应用过程中对 PSD 控制器作了不少改进，例如引入模糊控制以加快大误差时的动态响应，性能指标函数的变化，变步长以加快权值更新，修改对系统影响过小或过大的输入变量等。但是大部分文献在讨论 PSD 控制器时重点放在改进措施上，对影响神经元性能的各种问题如权值调整规律、

输入变量可能对算法造成的影响、非线性函数间的区别及其对单神经元的影响等的深入分析较少。为此有必要对 PSD 的基本原理，特别是反传误差调节规律进行分析，找到影响 PSD 控制器性能的根本原因，在此基础上提出改进措施，设计一种快速性和稳态性都较好的控制算法。同时，由于 PSD 单元结构上同 PID 单元的相似性，因此传统的对 PID 算法的一些改进措施可以为我们提供有益的借鉴。

1.3 研究内容及论文安排

针对工业生产中所用的多温区测控系统多变量、强耦合、大滞后的特点，为适应系统控温高精度，迅速升温，强抗干扰能力的需要，解决多回路的耦合，达到稳定运行是极为关键和重要的。根据研究目标，本文主要包括以下研究内容：

(1) 以神经网络和数字PID为理论基础，结合它们的优势，引出了自适应PSD算法，并作出了一定的算法改进，且将它们应用于电阻炉的炉温控制系统中，通过仿真进行了验证。

(2) 将PSD方法在多温区测控系统中的进行了推广，指出了存在的问题，进行了仿真研究，给出了结论。

(3) 对论文中所存在的尚未解决的问题做了总结，为后续的工作做好基础。全文的篇章结构安排如下：

第一章 绪论

主要介绍了课题的研究背景及研究的目标和意义，并分析了国内外对滞后不确定性对象的研究现状，给出了篇章构成。

第二章 多温区测控系统组成及特征分析

介绍了本文所研究的对象——多温区测控系统的结构特点及其建模过程和控制方法，并针对已知的数学模型分析多温区温控系统的耦合特性，影响系统耦合度的因素以及关联系统的稳定性，为后续章节做铺垫。

第三章 单温区中自适应 PSD 算法研究

针对大滞后系统的一般控制方法，引出了自适应 PSD 控制算法。并针对改种算法的不足，分析了相应的改进措施，并给出了改进的 PSD 算法仿真效果，做了比较和总结。

第四章多温区测控系统中 PSD 算法研究

在对比分析了几种 PSD 网络控制方法的优缺点的基础上，提出了一种 PSD 网络算法，并在多温区测控系统中作了仿真应用，给出了仿真结果并做了总结。

第五章 总结和展望

总结了本文的主要研究内容，并对今后的工作进行了展望。

第二章　多温区测控系统组成及特征分析

多温区测控系统在工业生产的各个方面都有着广泛的应用。它在在电子行业中如回流焊接，波峰焊接，电加热炉等众多对象中都广泛存在；本文以一个双输入双输出的多温区测控系统为例来说明温度滞后对象以及多温区解耦控制的方法。仿真实验结果表明，此种方法对具有滞后、耦合的一般对象也有一定的适用意义。

2.1 多温区测控系统介绍

本文所研究的多温区测控系统是由热电偶、单片机及其外围器件、计算机构成的一个典型计算机测量控制系统。多温区测控系统功能模块主要由温度采集和处理模块、温度控制模块和上位机构成，其系统结构框图如下：

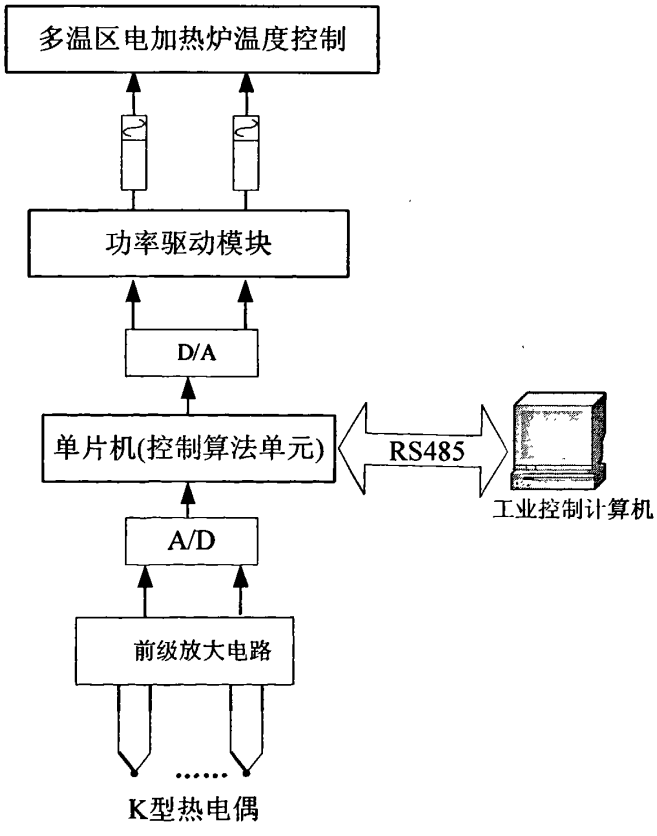


图 2-1 多温区测控系统总体结构框图

温度采集和处理模块包括热电偶和前级放大电路。普通 K 型热电偶作为传感器实现对温度变化信号的测量，后经信号调理电路（如放大、滤波等）处理后，送至由 MCU 构成的控制器。由于测量炉内温度和热电偶安放的位置有很大的关系，通常的作法是将温度采集器件热电偶安放在每个温区靠近加热体的位置上，这就使得测出来的是加热体温度的近似值，也可以更好地反应温度控制效果。前级放大电路包括放大电路和滤波电路，用来将热电偶的微小电位差信号经过滤波放大后形成标准的 0～10V 的电信号。

温度采集和处理模块转化后的信号通过 A/D 转化电路后送入单片机中与预先设置的理论值进行比较，再将前几次的采样与本次采样加以比较，以分析出实际温度的变化趋势，经过温度控制算法处理后得到控制信号，在现有的信号基础上作某些调整后经过 D/A 转换为 0～10mA 的电流信号便可输出到温度控制模块。

温度控制模块主要是将 MCU 输出的“0”“1”电平变成电加热器的功率信号，单片机输出的是 TTL 电平，不能直接控制红外加热管，经过功率驱动芯片之后接到 SSR(固态继电器)上，通过输出不同占空比的 PWM 数字脉冲来控制固态继电器的通断，再由红外发热管的通断完成自动调节发热功率[24]。

上位机工业控制计算机通过 RS-485 通信方式跟单片机相联，通过 MODBUS 协议实现温度数据及控制动作信号的传输，基于工控机上研发的监控软件实现了多温区电加热炉的监控、温度数据记录、绘图及打印等功能。

2.2 多温区测控系统模型及对象特征分析

2.2.1 多温区测控系统模型的确立

传统的工业过程动态建模方法主要分为两类：机理建模方法与系统辨识建模方法。机理建模从基本物理规律出发，即利用各个专门学科领域提出来的能量守恒性和连续性，以及系统(设备)的结构数据推导出对象的数学模型。机理建模有时可以获得精度较高的模型，而且如果在装置上不能或无法进行试验，或者被建模装置尚不存在时，机理模型是取得模型的唯一可行的途径。这类建模方法的最大好处就是变量和方程有明确的物理意义，往往可以准确表述系统的定性信息。直接从基本原理出发建立的原始微分方程往往相当复杂，一般在得到机理模型后需要简化后才可以作为控制用的数学模型[22]，文献[23]用机理建模的方法建立了热风回流焊机多温区的数学模型。系统辨识方法利用过程观测数据构造数学模型，即通过过程输入、输出信号的量测得出该过程输入、输出形态的模型。显然，

辨识方法可以对任意结构的过程进行建模[24]。即使是对其内部机理不甚了解，也可以得出模型。常用的辨识建模方法包括：阶跃(冲激)响应曲线辨识方法、相关分析辨识方法、最小二乘辨识方法等。辨识得到的模型，只反映过程输入、输出特性，对系统的内在信息反映不出来，此时机理模型可以弥补辨识所得模型在反映内在机理方面的不足。

求解传递函数的方法有几种，工程中常用的方法是利用飞升曲线的方法求传递函数，实测时在其中一路的输入端加阶跃信号，而在其它两端的加热电压为零，同时记录两个回路的温度变化曲线，然后在另一路上加电压再测，依次测得两组共 4 条响应曲线，以一阶惯性加纯滞后环节来拟合各路传递函数，采用文献[23]中的传递函数：

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} \dfrac{12.8e^{-s}}{16.7s+1} & \dfrac{-18.9e^{-3s}}{21s+1} \\ \dfrac{6.6e^{-7s}}{10.9s+1} & \dfrac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \cdot \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} \tag{2-1}$$

系统的输入输出关系可以表示为：

$$Y_1(s) = G_{11}(s)U_1(s) + G_{12}(s)U_2(s) \tag{2-2}$$

$$Y_2(s) = G_{21}(s)U_1(s) + G_{22}(s)U_2(s) \tag{2-3}$$

其中

$$G_{11} = \frac{12.8e^{-s}}{16.7s+1},\quad G_{12} = \frac{-18.9e^{-3s}}{21s+1},\quad G_{21} = \frac{6.6e^{-7s}}{10.9s+1},\quad G_{22} = \frac{-19.4e^{-3s}}{14.4s+1}$$

2.2.2 多温区测控系统机理模型特征分析

实际热风回流焊生产中，准确地测量炉内温度和反映电路板的实际温度跟热电偶安放的位置有很大的关系，通常的作法是将热电偶固定在每个炉膛布满通孔的加热板表面中心位置，在炉膛的内表面，这也是测量电路板温度的最佳位置，固定电路板的输送链爪会根据电路板大小来调整宽度，但其中一条链爪是固定输送通道的一方，另一条链爪可以自由调整，正是这种传输机构，热电偶就能比较正确地反映电路板的温度。该热风回流焊温区温度的控制方法是通过工业控制计算机内的监控软件来设定炉温，以单片机为核心的温度控制器实现数字式PID算法，输出不同占空比的数字脉冲来控制固态继电器的通断，再由红外发热管的通断完成自动调节发热功率；期间配以调整链条速度，优化热风马达转速等辅助手段来保证其控温效果，炉膛的数学模型是实施控制策略的基础。

全热风回流焊炉膛内部以及炉膛与外界的热交换过程是极其复杂的过程，炉内的热交换包含加热器与炉内空气的对流换热、电路板内部的热传导、输送链与炉内空气的对流换热、炉膛内空气与炉壁之间的对流换热、炉壁内部的热传导。

炉膛与外界之间的热交换包括炉体的热风循环系统造成炉内空气跟炉体外空气的对流换热、炉膛对外辐射。这其中加热器与炉内空气的对流换热、电路板跟炉内空气的对流传热、炉体热风循环系统造成的炉内空气跟外界空气的对流换热三种传热方式在热量传中占主要作用。

对于本文所要研究的热风回流焊机多温区对象主要有如下特点：

(1)系统多输入：热风回流焊机各温区均有一个发热源。

(2)强耦合性：由于各个温区之间并未完全隔离，而且热量通过炉膛内的热风循环装置在各温区之间相互传递。

(3)模型的误差：通过传热学机理建立数学模型，是在一定的范围获取物质特性参数，这些取值并非严格意义上的准确，这是模型的计算误差，而模型本身是建立在一定假设条件上的，这就会与实际情况总会有或多或少的差别。

(4)滞后：热风回流焊机多温区实际上是相当于多个加热炉，而加热炉本身的温度对象就是一个大滞后对象。

(5)非线性：热风回流焊机工作在不同时间段内，它的工艺条件不一样，因此依据此建立的模型参数就会有差别。

本章从热风回流焊炉的热传递和能量平衡方程出发，经过适当的简化，对建立多温区回流焊炉的炉温动态数学模型进行分析。

2.3 多温区测控系统耦合分析

对对象模型的耦合程度分析即是对控制回路相互关联分析的一个过程。通过从一些可供选择的输入输出搭配中，确定一个控制变量与被控变量的一组最佳搭配。如果所选择的最佳搭配使其相互关联很小，则可以采用 SISO 的方法来设计控制器；否则，则需要采用解耦技术或者采用补偿的方法进行多变量解耦设计[4]。

2.3.1 多温区测控系统耦合机理分析

对于该多温区测控系统，每个温区都是一套单独的温度控制系统，但由于各个温区不是绝对隔离的，当某一个温区因设定值改变或者受到干扰发生变化时，必然将立刻影响到其它温区尤其是相邻温区的温度变化，如果两个温度控制系统都要同时投入运行，其耦合是很严重的，这样的系统设计也无法满足生产过程对控制的要求的。

同时，该多温区测控系统工艺要求温度控制在$\pm 3^{\circ}C$的误差范围内。由于各个温区不是绝对隔离的，当某一个温区因设定值改变或者受到干扰发生变化时，将立刻影响到其它温区尤其是相邻温区的温度变化，如果两个温度控制系统都要

同时投入运行，其耦合是很严重的，这样的系统设计是无法满足生产过程对控制的要求的。

从传热学角度看，造成各温区输入输出关系耦合的直接原因，是热量在各温区之间的流动；这种流动主要发生在相邻的温区之间，根据传热学知识可知，在两个相邻温区之间的传热方式主要有以下几种：

(1)热量沿炉管的热传导；

(2)热量沿绝热材料的热传导；

(3)炉膛中通过空气对流的热传递；

(4)炉膛中空气的热辐射。

根据热传递原理[26]，在 1000℃以下，相临温区间的热传递方式主要为热对流，这意味着多温区电加热炉温区中主要是通过隔热板传热，所以相临温区间存在传热关系(即耦合)[27]。热对流的强度大体可以用下式表示：

$$Q/A = K \cdot \Delta t \tag{2-4}$$

式中，Q/A 为每单位面积的传热量；Δt 为相临温区间的温差；而 K 为传热系数，可当作常数，与相临温区间的距离、隔热板的隔热效果相关，它反映了相临温区间温度的耦合强度。

2.3.2 多温区测控系统耦合模型分析

对控制回路间相互关联的分析方法主要有相对增益法[28]、奇异值分析法[29]、相对动态增益阵法[30]、Jacobi 特征值准则[31]等方法。本文采用相对增益阵的方法对所建立的数学模型进行了简单的耦合强度分析。

对多温区温控系统的数学模型进行分析，对象的传递函数模型如下：

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} \dfrac{12.8e^{-s}}{16.7s+1} & \dfrac{-18.9e^{-3s}}{21s+1} \\ \dfrac{6.6e^{-7s}}{10.9s+1} & \dfrac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \cdot \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} \tag{2-5}$$

式中，$Y_1(s)$、$Y_2(s)$ 为系统两路输出量；$U_1(s)$、$U_2(s)$ 为系统控制输入量。利用相对增益阵的方法很容易判定这个系统的耦合强度。其中计算所得对象相对增益阵如下：

$$\begin{aligned} \Lambda = K \cdot (K^{-1})^T &= \begin{bmatrix} 12.8 & -18.9 \\ 6.6 & -19.4 \end{bmatrix} \cdot \left(\begin{bmatrix} 12.8 & -18.9 \\ 6.6 & -19.4 \end{bmatrix}^{-1} \right)^T \\ &= \begin{bmatrix} 0.6656 & 0.3344 \\ 0.3344 & 0.6656 \end{bmatrix} \end{aligned} \tag{2-6}$$

其中（•）为点乘运算表示两个矩阵中各自相对应元素的乘积。由相对增益的物理意义知，回路之间关联严重，且当其中一个通道闭环后，相对的另外一个通道增益将增大，系统的稳定性往往有所下降。因此，为实现对对象的良好控制，对其实现解耦是必要的。

2.4 本章小结

本章首先介绍了所研究的对象——多温区测控系统的组成结构及控制原理，并简单的分析了文献[22]所建立的一多温区数学模型。然后在系统机理分析的基础上，对系统耦合强度进行了定性的分析，总结了对象滞后、强耦合和参数时变的对象特性，为后续章节做准备。由系统中控制通道的相对增益的物理意义可知，系统之间存在强耦合，因此有必要进行解耦设计，才能实现系统的较高控制要求。

第三章 单温区中自适应PSD算法研究

经典控制理论和现代控制理论发展较为成熟，在空间技术，军事科学和工业控制等各领域均获得了较为显著的成效。但是，它们的共同特点就是依赖于系统的精确模型。不管是根据机理所建立的数学模型还是根据实验数据建立的数学模型，虽然它们都能够近似描述对象，但其最终的模型表示形式是确定的，得到的只是精确的定量解。然而，随着科学技术的快速发展，被控对象也越来越复杂，一个复杂系统的突出表现便是它的对输入－多输出变量间的强耦合性、系统参数的时变性、系统结构的非线性和不确定性. 这类系统大多没有确定的物理规律可以遵循，即使作出多种假设，要进行传统的定量分析也是十分困难的，甚至是无法实现的[16]。

因此，基于非辨识，无模型的控制方法便应运而生。无模型的控制方法至今应用广泛的有，黑龙江大学韩志刚教授创立的“无模型控制理论”[33][34]，基于神经网络的控制方法、基于模糊控制论的模糊控制方法[36]。如今虽然模糊控制技术可以获得良好的控制效果而且更容易获得工程上的实现。但是，模糊控制要求具有较为完善的控制规程，这些控制规则依赖于人们对被控过程（对象）经验信息的归纳和总结。由于被控过程的非线性，高阶次，时变性以及随机干扰等因素，造成模糊规则不够完善，都会不同程度的影响控制效果，难以达到控制的要求，限制了模糊控制技术在复杂对象中的应用。

常规的控制虽然算法简单，鲁棒性好，可靠性高，在工业控制中得到了广泛应用[2]。但是实际工业生产过程往往具有大滞后、非线性、参数不确定性，导致难以建立精确的数学模型，因此常规控制经常达不到理想的控制效果。此外，在实际生产现场中由于受到参数整定方法繁杂的困扰，常规控制器参数往往整定不良，性能欠佳——对运行工况的适应性很差，不能适应复杂的工况和高指标的性能要求。“对于一个动态过程，它的工作机理、规律、特性，一般用模型进行全局描述”，不可否认基于模型的控制理论的重要性，但是它们在应用方面存在两个困难：首先，对于大多数不确定系统而言，不可能为其建立一个精确的数学模型，这是最优控制在实际应用中效果不佳的主要原因。更为严重的是，有些模型失配严重导致无法进行控制器的设计，虽然智能控制模糊控制，人工神经网络控制等在一定程度上能够解决类似的“黑箱“的控制问题，但是其设计理论及技术实现的复杂性对工程应用提出了较高的要求，并且控制作用的实时性较差。其次，从传统统的模型描述看，不同的被控对象具有不同的模型结构和参数，这就导致

了所设计的控制器具有较强的专用性，从而在一定程度上限制了它的自适应范围。

因此，有必要对具有以下优良品质的控制算法进行研究：

1. 结构简单，易于工程实现；
2. 不依赖于对象的精确数学模型；
3. 能够很好的适应各种复杂的动态过程和参数变化；
4.对各种扰动具有良好的抑制能力。

由此可见，要求该类控制算法不仅应具有控制的特征，而且还应该吸取自适应的优点。本文即以此为原则，对一种能够适应对象参数时变的 PSD 算法在一滞后温度对象中进行了研究，针对所存在的问题提出了一定的改进措施，并通过大量的仿真证明了所提算法的有效性。

3.1 克服对象滞后特性的一般控制方法

传统的对大滞后对象的控制方法，一般有，史密斯预估控制[37]，达林算法，预测控制方法[38][39]等，这些算法在模型精确的条件下会取得良好的控制效果。但是有些算法却对模型的失配不具有鲁棒性（如 Smith 预估控制和达林算法），有的虽然对模型失配具有良好的鲁棒性，但是其算法本身和优化的复杂性（如预测控制方法），使得不仅对算法理论的研究十分困难，而且也给算法在工程实际中的推广应用带来了很多不利。

3.1.1 Smith 预估控制

在下图所示的单回路控制系统中，D(s)表示调节器的传递函数，用于校正 $G_p(s)$ 部分；$G_p(s)e^{-\tau s}$ 表示广义被控对象的传递函数，$G_p(s)$ 为广义被控对象中不包含纯滞后部分的传递函数，$e^{-\tau s}$ 为被控对象传滞后部分的传递函数。

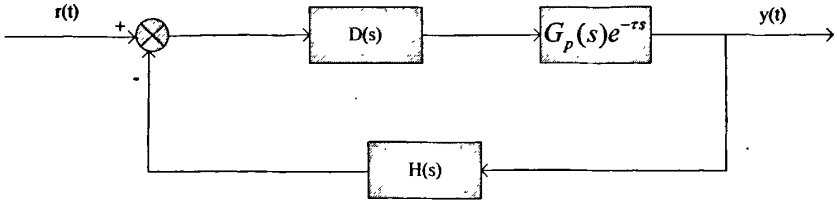


图 3-1 纯滞后环节的控制系统结构框图

Smith 预估控制基本原理：与 D(s)并接一个补偿环节，用来补偿被控对象中的纯滞后部分。这个补偿环节成为预估器，其传递函数为 $G_p(s)(1-e^{-\tau s})$，τ 为纯滞后时间，补偿后的系统框图如下：

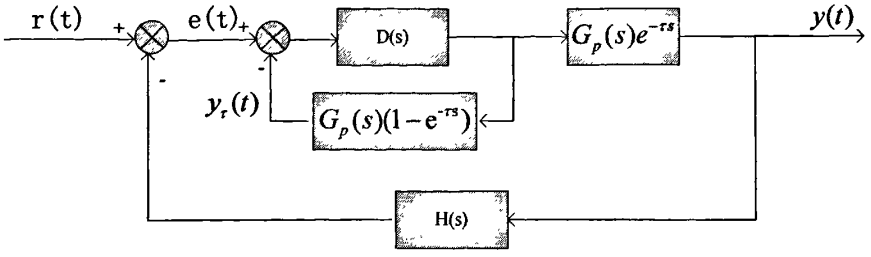


图 3-2 带 Smith 预估器的控制系统

由史密斯预估器和调节器 D(s)组成的补偿回路称为纯滞后补偿器，其传递函数为 $D'(s)$，即

$$D'(s)=\frac{D(s)}{1+D(s)G_p(s)(1-e^{-\tau s})}$$

经补偿后的系统闭环传递函数为

$$\Phi(s)=\frac{D'(s)G_p(s)e^{-\tau s}}{1+D'(s)G_p(s)e^{-\tau s}}=\frac{D(s)G_p(s)}{1+D(s)G_p(s)}e^{-\tau s}$$

上式说明，经过补偿后，已经消除了纯滞后部分对控制系统的影响，因式中的 $e^{-\tau s}$ 在闭环控制回路之外，不会影响系统的稳定性。根据拉氏变化的位移定理说明，$e^{-\tau s}$ 仅仅是将控制作用在时间坐标上推移了一个时间 τ，控制系统的过渡时间以及它的性能指标都与对象特性为 $G_p(s)$ 的对象完全相同。

3.1.2 达林（Dahlin）算法

设被控对象 $G_p(s)$ 是带有纯滞后的一阶或二阶惯性环节，即

$$G_p(s)=\frac{K}{1+T_1 s}e^{-\tau s}\quad 或，\quad G_p(s)=\frac{K}{(1+T_1 s)(1+T_2 s)}$$

式中：τ —纯滞后时间；T_1、T_2 —时间常数；K—放大系数。

达林算法的设计目标式使整个系统的闭环系统所期望的传递函数 $\Phi(s)$ 相当于一个延迟环节和一个惯性环节相串联，即

$$\Phi(s)=\frac{1}{T_\tau s+1}e^{-\tau s}$$

并期望整个闭环系统的纯滞后时间和被控对象 $G_p(s)$ 的纯滞后时间 τ 相同。式中

T_τ为闭环系统的时间常数，纯滞后时间τ与采样周期T有整数倍关系；

τ =N.T　(N=1，2……)

以一数字控制系统为例，原理框图如下（单位负反馈，即H(s)=1），

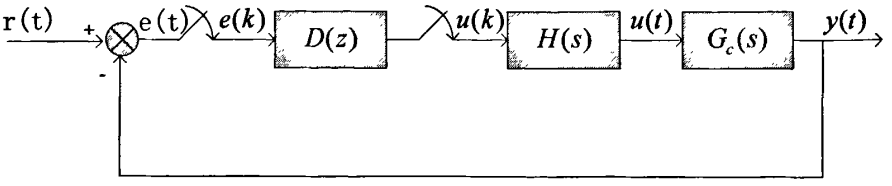


图 3-3 数字控制系统原理框图

由离散原理知，可以由脉冲传递函数近似法求得与$\Phi(s)$对应的闭环脉冲传递函数$\Phi(z)$：$\Phi(z)=\frac{Y(z)}{R(z)}=Z[\frac{1-e^{-Ts}}{s}.\frac{e^{-\tau s}}{T_\tau s+1}]$，代入$\tau$ =N.T，并进行z变换

$$\Phi(z)=\frac{(1-e^{-T/T_\tau}).z^{-N-1}}{1-e^{-T/T_\tau}.z^{-1}}$$

由上图知，控制器传递函数为

$$D(z)=\frac{1}{G(z)}.\frac{\Phi(z)}{1-\Phi(z)}=\frac{1}{G(z)}.\frac{(1-e^{-T/T_\tau}).z^{-N-1}}{1-e^{-T/T_\tau}.z^{-1}-(1-e^{-T/T_\tau}).z^{-N-1}}$$

因此，假若已知被控对象的脉冲传递函数$G(z)$，就可以由上式求出数字控制器的脉冲传递函数$D(z)$。

3.1.3 仿真分析

仿真实例1：采用Smith预估控制算法的阶跃响应仿真分析

被控对象：采用文献[2]中所提到的一个工业电加热炉模型，对象传递函数如下，

$$G_p(s)=\frac{e^{-80s}}{60s+1}$$

采用Smith预估控制方法，采样时间为10s。在PI控制中，取$k_p=4.0$，$k_i=0.022$，分别在模型精确，保持PID控制参数不变，模型T参数和τ参数变化的情况下的仿真结果如下（见附录1）：

（一）Smith预估控制算法对模型参数T的适应性

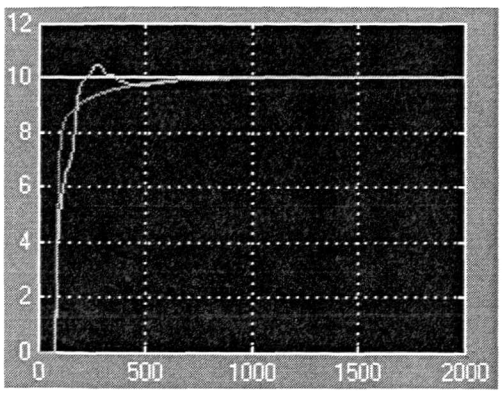


图 3-4 T 为模型参数 50％时阶跃响应

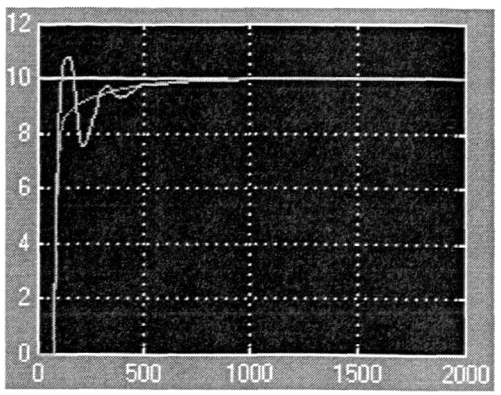


图 3-5 T 为模型参数 150％时阶跃响应

（二）Smith 预估控制算法对模型参数 τ 的适适应性

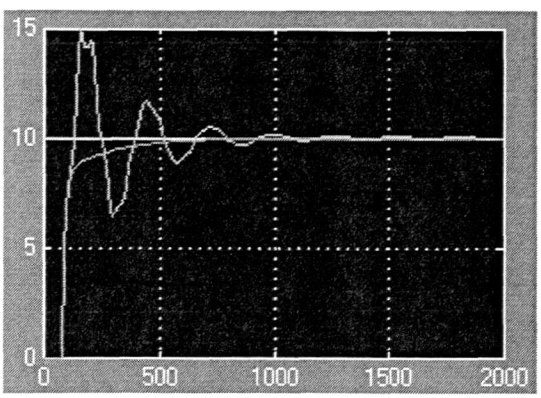


图 3-6 τ 为模型参数 50%时阶跃响应

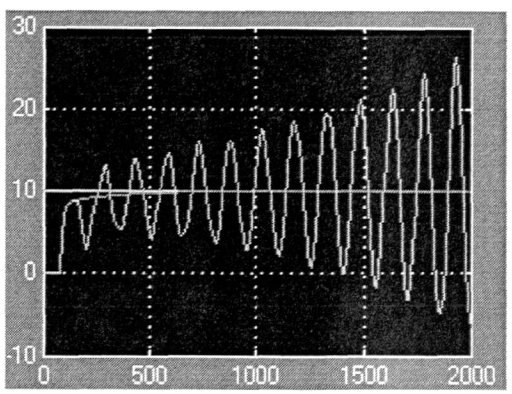


图 3-7 τ 为模型参数 150％时阶跃响应

小结：Smith 预估控制算法，在模型精确的条件下，可以较好的克服对象的滞后特性，取得满意的控制效果；但当对象参数如 T 或 τ 变化时，该算法控制品质明显变差，甚至失去收敛性。因此，Smith 预估算法对模型匹配要求较高，不适合于对模型参数变化较大的对象的控制。

仿真实例 2：采用达林算法的阶跃响应仿真分析

设被控对象为：

$$G_p(s)=\frac{e^{-0.76s}}{0.4s+1}$$

采样时间为 0.19s，期望的闭环响应设计为：

$$\Phi(s)=\frac{Y(s)}{R(s)}=\frac{e^{-0.76s}}{0.15s+1}$$

取相同的期望闭环响应设计，在模型各参数失配情况下，达林算法的适应性仿真结果如下（见附录 2）：

（一）达林算法对模型参数 T 的适应性

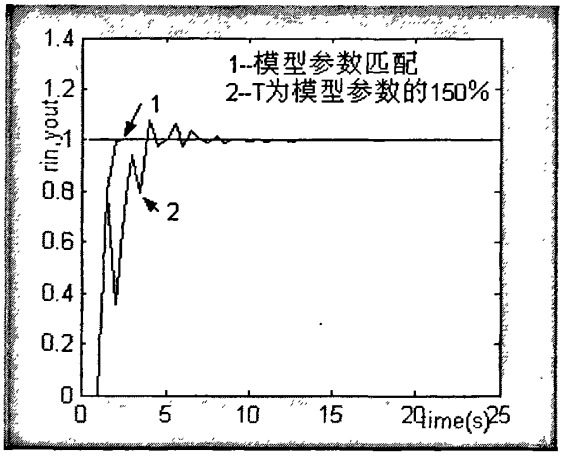


图 3-8 T 为模型参数 150％时的阶跃响应

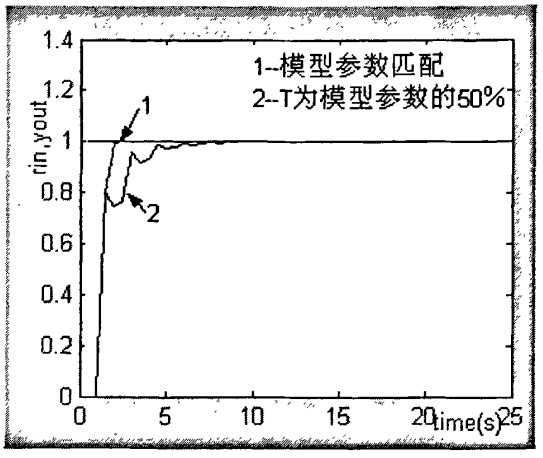


图 3-9 T 模型参数 50％时的阶跃响应

（二）达林算法对模型滞后时间参数 τ 的适应性

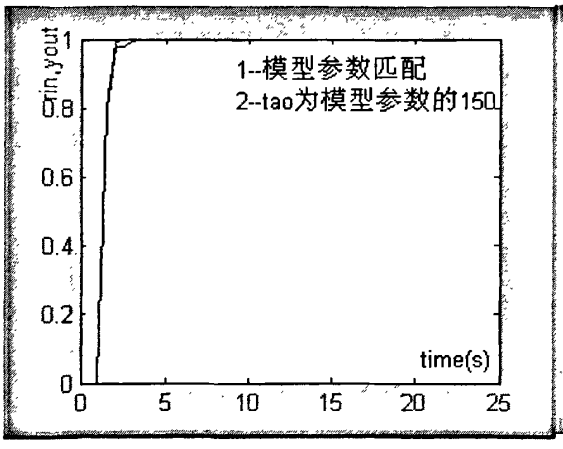


图 3-10 τ 为模型参数 150％时阶跃响应

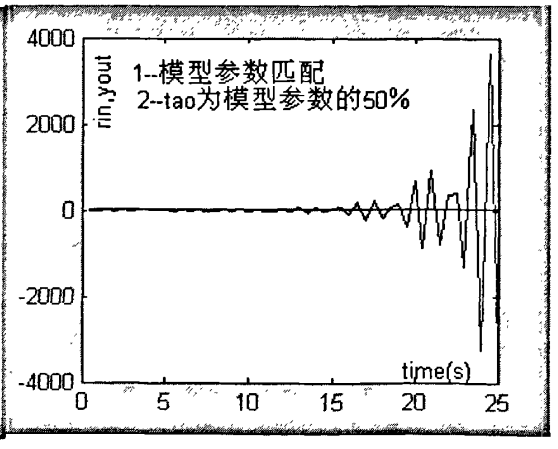


图 3-11 τ 为模型参数 50％时阶跃响应

小结：仿真结果表明，达林算法在模型匹配的情况下，设计简单易行，控制效果较好。但是，当参数 T 或者 τ 辨识不准确出现失配时，控制效果恶化，甚至

失去算法的收敛性。

对比总结以上的仿真结果，分析可以看到如下结论：

虽然 Smith 预估控制和达林算法在获得对象精确模型时都能克服对象的滞后特性，取得满意的控制效果。但当对象模型参数出现失配后，控制效果明显变差。然而，实际中的对象大多都是具有一定的参数不确定性和时变性，因此以上两种算法并不完全适用于现实的工程中，需要一定的改进。所以找到一种对对象模型匹配要求不高，具有较强鲁棒性的算法很有必要。

3.2 自适应 PSD 算法原理及其稳定性分析

传统的 PID 控制器由于其结构简单、调节方便。因而获得了广泛的应用，但对于一些复杂过程且参数时变的系统，由于 PID 的参数不易实时在线调整，因而在应用中遇到了一些困难。针对上述情况，由 Marsik 和 Strejc 提出的无辨识的自适应算法其机理是根据过程误差的几何特性建立性能指标，从而形成自适应 PSD（比例、求和、微分）控制规则[3]。该方法无须辨识过程参数，只要在线检测过程的期望输出和实际输出以形成自适应控制规律，因而这类自适应控制器具有明显的简单性和实现性。可以较好的克服算法对对象模型精确性的依赖。

单变量系统自适应 PSD 控制方法通常是用给定值与对象输出误差的平方构成性能指标函数，再用性能指标函数在线训练神经网络控制器的权值，使系统具有适应环境变化的能力[41][42][43]。

3.2.1 自适应 PSD 算法原理

本文所设计的控制器结构如图 3-12：

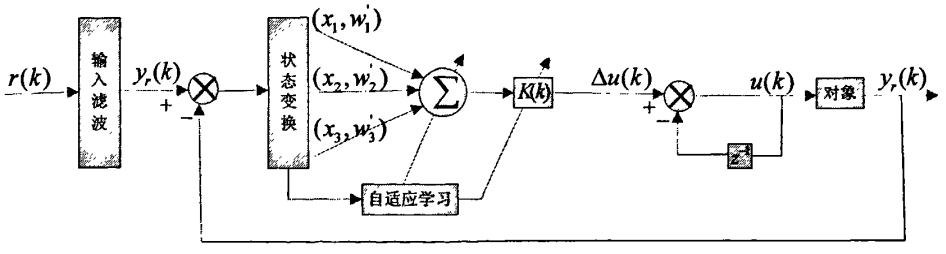


图 3-12 自适应 PSD 控制器结构图

该自适应 PSD 控制器主要包括：输入滤波、PSD 算法单元和权值自适应学习算法及增益自适应调整单元三个环节。

输入滤波：为提高系统的鲁棒性和稳定性，输入滤波采用一阶形式：

$$y_r(k+1)=(1-\alpha)r(k)+\alpha y_r(k)$$

其中$r(k)$为设定值，$y_r(k)$为滤波处理后的设定值，$0<\alpha<1$为滤波系数，可调节系统鲁棒性强弱。α越大，响应越慢，鲁棒性越强而快速性较差；相反，α越小，快速性提高而鲁棒性则会变差。因此，α的调整可以平衡系统快速性和鲁棒性之间的矛盾。

PSD算法单元和权值自适应学习算法：单神经元的输入取，

$$\begin{cases} x_1(k)=e(k) \\ x_2(k)=e(k)-e(k-1) \\ x_3(k)=e(k)-2e(k-1)+e(k-2) \end{cases} \tag{3-1}$$

PSD单元特性取为，$\Delta u(k)=K(k).w_i'(k).x_i(k)$ (3-2)

其中$\Delta u(k)$为k时刻单神经元的输出增量；$K(k)$为权值放大系数；$w_i'(k)=\sum_{i=1}^{3}w_i(k)/[\sum_{i=1}^{3}|w_i(k)|]$为权值的规一化，$w_i(k)$为$k$时刻状态$x_i(k)$所对应的权值；$x_i(k)$为$k$时刻单神经元的输入状态。权系数$w_i(k)$的学习策略采用Hebb学习（$p_i(k)=\Delta u(k).x_i(k)$）和监督学习（$p_i(k)=z(k).x_i(k)$）结合的方式，其中$p_i(k)$为学习策略，$z(k)=y_r(k)-y(k)$为教师信号，通过关联搜索对未知外界作出反应，并隐含着对单神经元作用信号的评价。即权系数调整公式，

$w_i(k+1)=w(k)+\eta_i.z(k).\Delta u(k).x_i(k)$（$\eta_i$为学习效率；$i$取1、2、3） (3-3)

增益的自适应调整方法：

采用由Marsik给出的递推算式，如下：

$$\begin{cases} K(k)=K(k-1)+C\times\dfrac{K(k-1)}{T_v(k-1)} & \text{(当sign(e(k))=sign(e(k-1))时)} \\ K(k)=0.75K(k-1) & \text{(当sign(e(k))}\neq\text{sign(e(k-1))时)} \end{cases} \tag{3-4}$$

$$T_v(k)=T_v(k-1)+L\times sign[|\Delta e(k)|-T_v(k-1)|\Delta^2 e(k)|] \tag{3-5}$$

其中，$0.025\le C\ge 0.05$，$0.05\le L\ge 0.1$，定义符号函数$\mathrm{sign}(x)=\begin{cases}+1 & x>0\\ 0 & x=0\\ -1 & x<0\end{cases}$

常规PID控制器同所设计的自适应PSD控制器比较可以得出如下结论：常规增量式PID控制增量如下，

$$\Delta u(k)=K_p\left[e(k)-e(k-1)\right]+K_i e(k)+K_d\left[e(k)-2e(k-1)+e(k-2)\right]$$

将此式同式（3-2）比较可知，自适应PSD控制器其实质是一种非线性在线自适应PID控制器。同时，自适应PSD控制器也克服了传统PID控制器在状态

变量 $x_i(k)$ 上选择的限制，PSD 控制器的状态量选取可以根据经验有几种不同的搭配方式。

3.2.2 自适应 PSD 算法实现

自适应 PSD 控制器的仿真实现具体步骤如下：

第一步：给一组初始权值赋值，即给 w_1_1，w_2_1，w_3_1 赋迭代初值。

第二步：确定增益 K_1 和 T_1 的赋迭代初值。

第三步：根据式(3-1～3-3)求取控制器的输出量 $\Delta u(k)$。

第四步：从系统求出输出量 $y(k)$。

第五步：计算误差信号 $e(k)$，并根据 $x_i(i$取1、2、3) 的算式得到控 x_i(k)。

第六步：根据算法式(3-5)修正权值 $w_i(k)(i$取1、2、3)。

第七步：根据(式 3-1)调整增益 K 和 T 的值。

第八步：返回第三步。

3.2.3 PSD 控制器的闭环特性和算法稳定性分析

PSD 控制器系统结构如图所示，

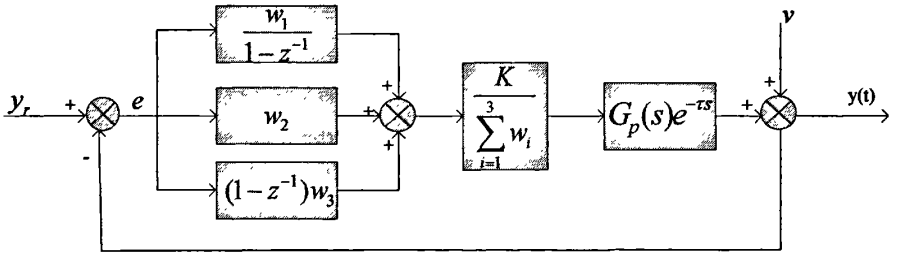


图 3-13 PSD 控制系统的结构图

设被控过程模型为

$$A(z^{-1})y(k)=z^{-d}B(z^{-1})u(k)+v(k) \tag{3-6}$$

其中 y，u ——系统输出和控制器输出信号；y_r，v ——系统给定输入信号和干扰输入信号。

$$A(z^{-1})=1+\sum_{i=1}^{p}a_i z^{-i}$$

$$B(z^{-1})=\sum_{i=0}^{q}b_i z^{-i}$$

PSD 控制器方程由式（3-3～3-5）有

$$R(z^{-1})u(k)=KQ(z^{-1})e(k) \tag{3-7}$$

其中 $Q(z^{-1})=\sum_{i=1}^{3}q_i z^{-1}$，$R(z^{-1})=1-z^{-1}$。

参数分别为，

$$q_1=\sum_{i=1}^{3}w_i/\sum_{i=1}^{3}|w_i|,\quad q_2=-(w_2+2w_3)/\sum_{i=1}^{3}|w_i|,\quad q_3=w_3/\sum_{i=1}^{3}|w_i| \tag{3-8}$$

闭环系统方程可通过将式子(式 3-7)代入（式 3-6）中求得：

$$y(k)=\frac{z^{-d}KBQ}{AR+z^{-d}KBQ}y_r(k)+\frac{Rv(k)}{AR+z^{-d}KBQ} \tag{3-9}$$

闭环特征方程为：

$$T(z^{-1})=A(z^{-1})R(z^{-1})+z^{-d}KB(z^{-1})Q(z^{-1}) \tag{3-10}$$

因此，PSD 控制算法实质上为一种变系数的比例、积分、微分复合控制算法，由于其学习算法是自适应的，本质上为非线性系统，要分析其稳定性有一定的难度，但是从式子（式 3-8）可以看出：q_1，q_2，q_3仍为一小于或等于 1 的有界变量，从理论上讲可以借助线性系统分析稳定性的方法，应用特征方程求出在系统稳定条件下学习参数与参数间的相互关系和约束条件。

系统的收敛性取决于学习步长的选取，以下利用 Lyapunov 稳定性原理证明保证系统收敛的学习步长范围。

定理 1[45] 当 PSD 网络的学习步长η满足

$$0<\eta<\frac{1}{\varepsilon^2},\ \text{其中}\ \varepsilon=-\frac{1}{2\sqrt{J}}\cdot\frac{\partial J}{\partial W} \tag{3-11}$$

权值调整算法，$W(k+1)=W(k)-\eta\frac{\partial J}{\partial W}$ (3-12)

可以保证控制系统在学习过程中收敛，其中 W 表示 PSD 神经元权值网络的连接权值w_{ij}的矢量，其中$J=\sum_{p=1}^{n}E_p=\frac{1}{m}\sum_{p=1}^{n}\sum_{k=1}^{m}e_p^2(k)$确定。

证明：

定义 Lyapunov 函数，

$$L(k)=\frac{1}{2}\bar{E}^2(k) \tag{3-13}$$

$$\text{其中}\ \bar{E}^2(k)=\sqrt{\frac{1}{m}\sum_{p=1}^{n}\sum_{k=1}^{m}[r_p(k)-y_p(k)]^2}=\sqrt{J} \tag{3-14}$$

式中$r_p(k)$为系统被控量给定值，$y_p(k)$为系统被控变量实际值。

$$\text{令}\ \Delta L(k)=L(k+1)-L(k)=\frac{1}{2}[\bar{E}^2(k+1)-\bar{E}^2(k)] \tag{3-15}$$

$$\text{其中有，}\ \bar{E}(k+1)=\bar{E}(k)+\Delta\bar{E}(k) \tag{3-16}$$

$\Delta\bar{E}(k)$为误差变化量，$\Delta\bar{E}(k)=\frac{\partial\bar{E}(k)}{\partial W}.\Delta W$　　（3-17）

网络权值变化量，$\Delta W=-\eta\frac{\partial J}{\partial W}=-\eta\frac{\partial}{\partial W}[\bar{E}^2(k)]=-2\eta\bar{E}(k)\frac{\partial\bar{E}(k)}{\partial W}$　　（3-18）

将式（3-16～3-18）代入（式3-15），则有

$$\Delta L(k)=\frac{1}{2}[\bar{E}^2(k+1)-\bar{E}^2(k)]=\frac{1}{2}\{[\bar{E}(k)+\Delta\bar{E}(k)]^2-\bar{E}^2(k)\}$$

$$=\frac{1}{2}\{2\bar{E}(k)\Delta\bar{E}(k)+[\Delta\bar{E}(k)]^2\}$$

$$=\frac{1}{2}\{2\bar{E}(k).\frac{\partial\bar{E}(k)}{\partial W}.[-2\eta\bar{E}(k)\frac{\partial\bar{E}(k)}{\partial W}]+[\frac{\partial\bar{E}(k)}{\partial W}.[-2\eta\bar{E}(k)\frac{\partial\bar{E}(k)}{\partial W}]]^2\}$$

$$=2[\bar{E}(k).\frac{\partial\bar{E}(k)}{\partial W}].[-\eta+[\eta\frac{\partial\bar{E}(k)}{\partial W}]^2]\qquad(3\text{-}19)$$

为了保证系统的收敛性，因此必须满足$\Delta L(k)<0$，即

$$-\eta+[\eta\frac{\partial\bar{E}(k)}{\partial W}]^2<0\qquad(3\text{-}20)$$

解以上不等式，可得：$0<\eta<\frac{1}{[\frac{\partial\bar{E}(k)}{\partial W}]^2}$　　（3-21）

而由（式3-14）可求出，$\frac{\partial\bar{E}(k)}{\partial W}=-\frac{1}{2\sqrt{J}}\frac{\partial J}{\partial W}$　　（3-22）

证毕。

同时文献[45]中还给出了一个有效的推论，为学习效率的选择提供了依据：

推论 如果一PSD单神经元网络多变量系统是收敛的，则其隐含层至输出层权值的调整算法

$$\mathrm{W(k+1)=W(k)}-\eta\frac{\partial \mathrm{J}}{\partial W}\qquad(3\text{-}23)$$

中的学习效率η必然要满足，$0<\eta<\frac{1}{\varepsilon^2}$　　(3-24)

其中$\varepsilon=-\frac{\sum_{p=1}^{n}\sum_{k=1}^{m}\delta_{ij}(k)x_{ij}(k)}{2\sqrt{m\sum_{p=1}^{n}\sum_{k=1}^{m}[r_p(k)-y_p(k)]^2}}$

证略。

3.3 自适应 PSD 算法设计中所涉及的问题及改进方法

3.3.1 PSD 算法设计中所存在的问题

通过对具有单神经元结构的 PSD 控制器的原理分析，总结了设计中存在的问题如下：

1. 关于权系数初值的选择问题

由于神经元的自学习能力，权系数的初值不影响权系数以后的学习效果，也不会影响控制器的控制效果。一般，我们选择 0.01～1 之间的三个相等的数即可。

2. 增益 k 值的选择问题

k 值的选择非常重要。k 大，则快速性更好，但超调量更大，甚至可能使系统不稳定。当被控过程时延较大时，k 的值必须减少，以保证系统稳定。k 值选择过小，虽然可以保证系统的稳定性，但却会使系统的快速性变差。

3. 可调参数 η_p、η_i、η_d 的选取规则

以上三个参数同自适应 PSD 学习算法的运行效果有关。由于 PSD 算法在结构上同 PID 相通，因此 PID 参数的调节方法在原则上也是同样适用的。通过以上仿真与试验结果分析总结出调节规则和顺序如下：

（1） 对阶跃输入，弱输出有大的超调，且多次出现正弦衰减现象。应较少 k 值，同时维持 η_p、η_i、η_d 不变。若上升时间过长，且无超调，则应增大 k，相应维持 η_p、η_i、η_d 不变。

（2） 对阶跃输入，若被控对象产生多次正弦衰减现象，可减少 η_p 的值，维持其他参数不变。

（3） 若被控对象响应特性出现上升时间短且有较大超调，应减少 η_i 的值，维持其他参数不变。

（4） 若被控对象上升时间长，增大 η_i 又导致超调过大，可以适当增加 η_p，维持其他参数不变。

（5） 在开始调整时，η_p 应该选择较小值，先调整 η_p、η_i 使对象具有良好特性后，再逐步增加 η_d，而同时保持其他参数不变，调整系统以获得更好的特性。

4. 关于状态变量 $x_i(t)(i=1,2,3)$ 的数值预处理问题

在构成实际控制系统时，由于被控物理量的不同，其数值可能差别很大。如果直接采用实际物理量作为状态量，必然造成 $x_1(t)$，$x_2(t)$，$x_3(t)$ 在数值上差别很大，使学习算法中有些项被其他项淹没。为此，可采用标度变换方法，如对设定点采用给定值为基准，进行标幺化处理：误差变化率乘上某一系数，以保证

$x_1(t)$，$x_2(t)$，$x_3(t)$具有基本相同的数量级，或每个状态变量采用不同的学习步长。

5. 算法的复杂程度和控制器的鲁棒性问题

虽然 PSD 控制器可以适应参数大范围的变化，但是却很难适用对于不同的控制对象。这一点主要是由于系统在k的取值上往往比较敏感而且相差较大，导致系统在参数尚未调整到所需要值附近时就已振荡或跑飞。为了保证系统的稳定性和较高的稳态精度，算法的复杂程度会相应增加，参数的调节也趋于复杂，增加了现场调试的难度和工作量，减弱了控制器的鲁棒性。

3.3.2 自适应 PSD 算法的改进方法

正如前所述，自适应 PSD 控制器在实际的应用中也存在许多问题，后来的许多学者在这方面做了大量的有益工作。由于自适应 PSD 是拟合 PID 控制器，结构上同 PID 具有相似性，因此对 PID 的一些改进方法可以作为对自适应 PSD 算法改进的有益参考，具体表现在自适应 PSD 中就是稳定性与快速性的折衷。在自适应 PSD 控制器中，为了保持权值的相对稳定性，常常以牺牲快速性为代价，即减小步长。这里就潜伏着另一个问题，虽然权值更新缓慢了，系统在一定时间内、扰动不频繁的情况下表现了较好的特性，但是无法检验长时间运行后系统的稳定性。

因此首先应当加快调节器权值的更新，并在仿真时加入较为恶劣的负载变化或扰动，观察权值的更新过程和变化趋势，根据权值的变化情况设计系统的控制策略。

（1）变结构控制

为抑制大范围误差变化时积分器过早饱而采用变结构控制，即误差较大时采用 P 控制，同时更新权值；误差较小时，恢复为 PSD 控制。应当注意的是，在参数变化前后必须保证 u 基本不变，以避免由于结构变化导致输出量剧变。

（2）积分饱和作用及其抑制

由于积分的作用，尽管计算差分方程式所得的积分项运算结果继续增大或减小，但执行结构已无相应的动作，控制信号则进入深度饱和区。饱和引起输出超调，甚至产生震荡，使系统不稳定。传统控制器中为了抑制超调一般都要对积分项限幅，这种方法也可以引入进来。

改进的方法一般有，遇限削弱积分法、积分分离法。他们的基本思路是，当被控量和给定值偏差大时，取消积分控制，以免超调量过大；当被控量和给定值接近时，积分控制投入，消除静差。这实质上是一种复合控制方法。

（3）针对权值更新趋势的控制

针对权值$w_i(i=1,2,3)$不收敛的情况有几种改进的思路：①改变单神经元的结构，变为其他形式的径向基网络；②修改误差指标函数；这种方案虽然可行但是对系统的改动较大，而且收敛性仍然难以保证；③对权值的硬限幅；④改变权值更新算法。后两者虽然简单、易于实现但是单独使用效果不好，故可以将其综合使用。

文献[3]和文献[42]分别提到了使用梯度最小法极化函数和采用二次型性能指标函数以及PQ性能指标函数修改误差指标函数的方法，并取得了满意的效果，值得借鉴。

（4）减少计算量的措施

通过仿真研究表明，由于P、I参数的自调整，微分的作用已经不明显了，对应于微分项的权值变化非常缓慢。为了减小计算量可将其除去。在系统误差较小时，积分项权值的变化没有多大意义，而且单神经元函数在输出为零附近时具有最大的导数，即最大的更新量，为了减小计算量和保证系统稳定，在判断系统进入空载后停止更新积分项权值。此种方法可以满足系统对实时性的要求。

（5）算法动态过程的加速方法

比例项系数与积分项系数的符号关系为：若被控量继续偏离给定值，则这两项符号相同；被控量向给定值方向变化，则这两项符号相反。

当被控量接近给定值时，反号的比例作用阻碍了积分作用，因而避免了积分超调及随之带来的振荡，单如果被控量远未接近给定值仅刚开始向给定值变化时，则由于比例项和积分项反向，将会减慢控制过程。

因此，为了加快开始的动态过程，可人为选择一偏差范围ε，当$|e_i|\leq\varepsilon$时按正常规律调节；而当$|e_i|>\varepsilon$时取其绝对值，去掉微分项的作用。

3.4 一种改进的自适应PSD算法及仿真分析

3.4.1 改进的PSD算法原理

本文借鉴吸收以上所作出的成果的基础上，提出并采用一种复合控制方法，提高其快速性以弥补PSD算法快速性方面的不足。即采用“积分分离“方法，在误差范围较大时，为提高快速性和避免积分饱和，神经元输入选为2维，分别为x_1、x_3；在误差范围较小时，为保证其控制精度，神经元输入选为3维，分别为x_1、x_2、x_3，即采用PSD控制。

同时，为了保证算法的收敛性，对误差量和控制器输出量进行了限幅处理；为提高系统的稳定性和鲁棒性，将设定值输入进行了滤波处理，可以消弱过量的控制作用，从而减少甚至消除超调，使系统能平滑地达到设定值。滤波器形式采

用一阶形式即可，即：

$$y_r(k+1)=(1-\alpha)r(k)+\alpha y_r(k) \tag{3-25}$$

其中：$r(k)$为设定值；y_r为滤波处理后的设定值；$0<\alpha<1$为滤波系数，可调节系统鲁棒性强弱。α越大，响应越慢，鲁棒性越强而快速性较差；相反，α越小，快速性提高而鲁棒性则会变差。因此，α的调整可以平衡系统快速性和鲁棒性之间的矛盾。

其复合控制算式如下：

$$\begin{cases}\Delta u(k)=K_v[w_1(k).x_1(k)+w_3(k).x_3(k)] & |e(k)\geq\varepsilon| \\ \Delta u(k)=K_v'[w_1(k).x_1(k)+w_2(k).x_2(k)+w_3(k).x_3(k)] & |e(k)<\varepsilon|\end{cases} \tag{3-26}$$

大量的工程实践证明，η_i参数的在线学习修正，主要与$e(k)$和$\Delta e(k)$有关[3]。基于此改进的PSD算法为：

$$\Delta u(k)=u(k)-u(k-1)=K(k).w_i'(k).x_i(k) \tag{3-27}$$

$$w_i'(k)=\sum_{i=1}^{3}w_i(k)/[\sum_{i=1}^{3}|w_i(k)|] \tag{3-28}$$

$$\begin{cases}w_1(k+1)=w_1(k)+\eta_p.z(k).\Delta u(k).(e(k)+\Delta e(k)) \\ w_2(k+1)=w_2(k)+\eta_i.z(k).\Delta u(k).(e(k)+\Delta e(k)) \\ w_3(k+1)=w_3(k)+\eta_d.z(k).\Delta u(k).(e(k)+\Delta e(k))\end{cases} \tag{3-29}$$

3.4.2 算法仿真分析

仿真实例：

仿真对象仍然采用文献[2]中所提到的一工业电加热炉对象，对象传递函数如下：

$$G_p(s)=\frac{e^{-80s}}{60s+1}$$

离散化对象的采样周期为1s。

采用改进后的自适应PSD算法，采样时间为10s。依照上节中所介绍的参数选取原则，分别取$K_0=0.03$、$T_0=10$、$C=0.04$、$L=0.08$，权系数调整初值取$w_1=0.10$、$w_2=0.10$、$w_3=0.10$。保持以上参数不变，分别在模型T参数和τ参数分别变化±50%的情况下的仿真结果如下（见附录3）：

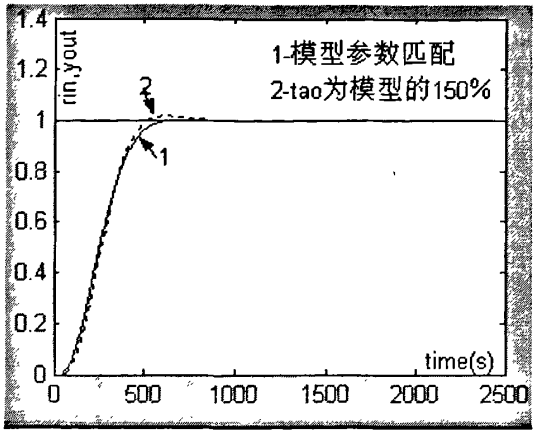


图 3-14 τ 为模型的 150%时阶跃响应

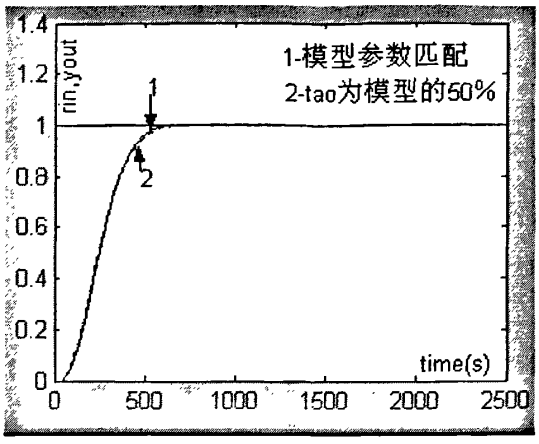


图 3-15 τ 为模型的 50%时阶跃响应

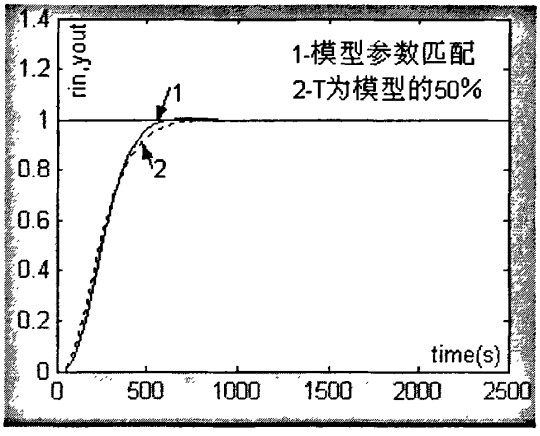


图 3-16 T 为模型的 50%的阶跃响应

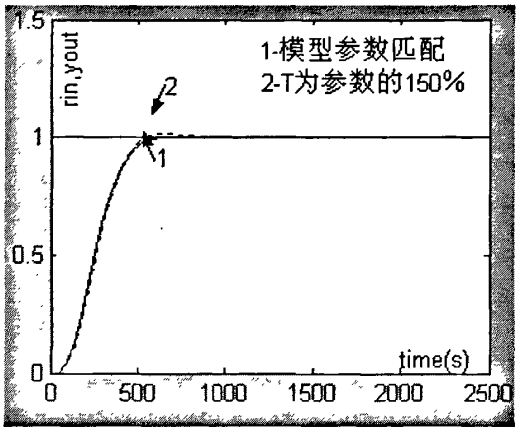


图 3-17 T 为模型的 150%的阶跃响应

通过以上仿真表明，所研究的 PSD 算法设计过程不仅所需设置参数少，调节方便；而且，在模型参数改变的条件下，仍然能够较好的克服对象的滞后特性，取得满意的控制效果，满足之前我们所提出的要求。

3.5 本章小结

本章主要针对单温区中温度对象所具有的滞后、对象参数时变的特性，对比分析了几种应用于滞后对象的控制算法的特点。在此基础上，重点分析了自适应 PSD 算法的原理并从理论上证明了其稳定性。

探讨了自适应 PSD 控制器设计中所存在的问题，并给出一些可行的改进方法。在此基础上，提出了一种能够改善 PSD 算法快速性不足之处的一种复合控

制方法，并在一工业电加热炉对象上验证了改进的自适应 PSD 算法的特性，总结了对于可调参数的选择的经验原则，分析了仿真结果。结果表明，改进后的自适应 PSD 算法在模型参数时变的条件下，仍然能保证系统快速性、收敛性等指标要求，能够取得满意的控制效果。

第四章 多温区测控系统中 PSD 算法研究

4.1 引言

经第二章分析，所建立的多温区测控系统模型不仅存在时滞特性，而且也存在一定程度的耦合。因此，为了提高系统的控制品质，有必要对系统的相关解耦方法进行研究。同时，第三章中所提自适应 PSD 算法对时滞、参数时变对象是行之有效的；而对 PSD 算法在多变量系统中的应用却少有文献论述。本章将以多温区测控系统模型为基础，将 PSD 算法推广应用到此多变量系统中，重点研究了 PSD 控制器组成网络过程中的结构变化，并讨论了 PSD 网络在设计中所涉及的一些问题。最后，给出了对模型的仿真结果及分析。

4.1.1 多变量系统控制方法的提出

多变量系统的控制目标就是要调整被控系统的多个输入作用使系统输出达到某些指定的目标。由于多变量系统的回路间广泛存在着耦合，即系统的某一个输入与系统的所有输出相互影响，或者系统的某一个输出受到所有系统输入的影响，因此为了获得满意的控制效果，有必要对系统实现解耦控制[63]。

目前许多多回路的控制系统按照单变量系统设计之所以也能正常工作，是因为系统间的耦合程度不高，或者说这些系统间只是松散联系。但是，现实中不少控制对象的变量关系比较密切，甚至相当密切，一个变量变化对其它变量有严重的影响。使得按照被控制变量与控制变量单一结合构成多个单回路控制系统时，系统控制品质明显下降，常常满足不了生产过程对控制系统的要求。因此，解耦控制一直以来都是众多学者争相研究的焦点[46][47]。

控制系统的耦合情况可以通过传递函数矩阵来分析。以一个 2×2 系统为例说明，其系统传递函数可写为，

$$\mathrm{Y}(s)=\begin{bmatrix}Y_1(s)\\Y_2(s)\end{bmatrix}=\begin{bmatrix}G_{11}(s) & G_{12}(s)\\G_{21}(s) & G_{22}(s)\end{bmatrix}\begin{bmatrix}U_1(s)\\U_2(s)\end{bmatrix} \tag{4-1}$$

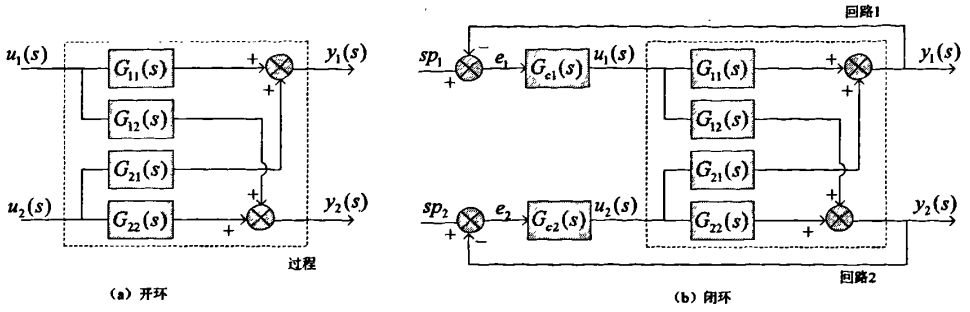


图 4-1 2×2 多变量控制系统开、闭环结构图.

如果传递函数$G_{12}(s)$和$G_{21}(s)$都等于零，两条调节通道各自独立，不存在关联，系统间无耦合。一个控制回路不管是处于开环还是闭环状态，对另一个控制回路无影响。此时，过程的输入输出关系为，

$$Y_1(s)=G_{11}(s).U_1(s) \tag{4-2}$$

$$Y_2(s)=G_{22}(s).U_2(s) \tag{4-3}$$

如果$G_{12}(s)$和$G_{21}(s)$有一个不等于零，则称系统为半耦合或者称单方向耦合系统；如果两个都不等于零，则称为耦合或者双向耦合系统。这种情况比较复杂。例如，当回路 2 开环时，$u_1 \to y_1$的传递函数是$G_{11}(s)$，只有一条通道；当回路 2 闭环时，$u_1 \to y_1$除了上述直接通道外，$u_1 \to y_2 \to u_2 \to y_1$有间接通道的影响。结果，使得这两个系统在控制过程中彼此形成干扰，使两个系统都控制不好。

所以，如果对象存在耦合，就会明显降低控制系统的调节品质，在耦合严重的情况时会使各个系统均无法投入运行。如果设计者回避了事实上的回路间的耦合而采用近似处理的办法，仍然按照单变量的设计方法设计控制器，这种人为的简化将会导致以下的一些问题：

(1)由于没有考虑被控对象中各回路间的关联，因而很难同时使各个单变量系统稳定地运行，也就无法有效地对对这种多输入－多输出、变量间紧密关联的系统实现统一的控制。

(2)对于存在耦合的系统，由于各回路不能独立考虑，例如采用 PID 控制，各回路 PID 参数的整定不得不多次进行，以便找到一个合适值。而在很多实际场合，是很难得到一个令人满意的整定组合的。即使可以得到一个一般的参数，但是当对象参数发生变化时，控制品质也会明显变差。

(3)从理论上讲，PID 控制器具有较好的鲁棒性，但是在多个单回路之间存在耦合的情况下，整个系统的鲁棒性却无法得到保证。

因此，研究如何实现解耦控制是多变量过程控制理论与实践中的一个突出问题。研究如何解耦控制，首先要研究各个回路间的耦合关系。控制回路相互关联

分析的方法即是为了确定多变量控制系统中回路间的关联程度。因此，它可以用于从一些可供选择的搭配中，确定控制变量和被控变量的一组最佳搭配。如果能找到这样的一组最佳搭配或是关联程度相对较小的一组，就可以用单输入单输出(SISO)的设计方法来设计控制器；如果所选择的最佳搭配仍呈现出明显的相互关联，则只能采用解耦技术或是采用能补偿相互关联的多变量设计方法。

随着控制理论的发展，解耦控制的研究对象已经从线性系统扩展到非线性系统，解耦方法应用的理论基础也从经典控制理论发展到了现代控制理论。近年来随着许多先进的控制理论的出现，与此对应的解耦方法也应运而生，呈现百花齐放的局面。如特征结构配置解耦，模糊解耦，神经网络解耦，自校正解耦，线性二次型解耦，奇异摄动解耦，变结构解耦，H_∞设计法解耦，等等。

总体来说，控制回路间耦合的分析方法主要有相对增益法[28]、奇异值分析法[29]、相对动态增益阵法[30]、Jacobi 特征值准则[31]等方法。Jensen，Yu 和 Luyben 等对其中的一些方法进行了评价和对比[32]。

4.1.2 多变量系统中解耦方法分类

通常将解耦方法主要分为传统解耦方法、基于现代控制理论的解耦方法、自适应解耦方法、神经网络解耦四类方法。

（1）传统解耦方法

传统解耦方法主要适用于线性定常 MIMO 系统，如下所述：

1. 基于古典控制理论的串联解耦

由 Bristol 提出的相对增益分析法和由 Bolsenom、Hood 和钱学森首先提出的对角矩阵解耦方法，是古典解耦的代表。其基本思想是：适当设计，使得联系 MMO 控制系统输入变量与输出变量之间的系统传函矩阵成为对角矩阵。在此基础上的进一步改进是改变目标矩阵(目标矩阵一般取原模型对角线矩阵的减阶形势)的解耦。它除了解耦外，能同时改变各个控制通道特性，使之更易于控制。

2. 基于多变量频域理论的逆 Nyquist 曲线法、序列回差法和特征曲线分析法。这几种方法本身引用的概念多，计算复杂，不易于工程实现。

（2）基于现代控制理论的解耦方法

由 Falb 等人发展起来的状态变量法，主要有线性状态反馈解耦和线性输出反馈解耦。其基本思想是通过从状态变量或输出变量处引出一个反馈阵，使得系统传递函数阵成为一个对角形有理多项式矩阵。但是，这种方法首先需要进行能解耦性判定。

（3）自适应解耦方法

自适应控制的思想与解耦控制技术相结合并用于多变量系统中，就形成了自

适应解耦方法。自适应解耦的目标是使系统的闭环传递函数成为对角阵，通常把耦合信号作为干扰处理。自适应解耦实质上采用了最优控制的方法，建立目标函数并对参数寻优是该方法的核心所在，这是与传统解耦方法的本质区别，是解耦理论的重大突破，也是智能解耦理论的基础。

（4）神经网络解耦方法

近几年来，随着智能控制技术的发展，“智能”的思想已运用于解耦控制中并取得了一定的成果。其中，以神经网络解耦方法为代表，成果丰富。

文献[52]中利用神经网络的可训练性与结构通用性，引入神经网络作为补偿环节而达到解耦的目的。当对象的输入输出之间存在耦合，又没有确定的映射关系，可以建立相应的模糊规则，进行模糊解耦。文献[54][55]中采用模糊概念表述相对耦合度，用模糊控制的方法设计了模糊解耦补偿器，使系统能按不同的被控过程特性达到一定的解耦要求。文献[57]中将预测控制的思想引入解耦控制中，进行预估补偿解耦控制。

在工程应用中，上述方法还存在着各种各样的问题。基于古典控制理论设计的解耦控制系统常常是物理上不可实现的系统。而基于现代控制理论设计的解耦控制系统算法过于复杂，而且还要求系统矩阵满足一定的条件。除此之外，以上方法具有一个共同特点，均建立在被控过程的数学模型已知的前提之上，数学模型已知成为使用这些方法的“瓶颈“。当系统建模建立困难或模型呈现非线性特性时，这些方法几乎不能应用；同时，当系统变量间的耦合强度或耦合关系随负载或时间而改变时，这些方法同样无能为力。

为此人们不断地探索新的方法。其中人工神经网络以其所具有的良好的学习能力、泛化能力、容错能力和非线性映射能力，将它用于解耦，为传统解耦方法所不能解决的问题提供了很多新的思路。人们先后提出了基于神经网络理论的解耦控制算法和容错解耦控制算法[59]。但是，这两种方法需要知道对象控制变量关于被控变量的雅可比矩阵，不便于在工程中应用。还有利用神经网络α阶积分逆系统来实现连续非线性 MMO 系统线性化解耦的方法[60]。它是由一个静态神经网络加若干积分器构成，将其串联在原系统之前，原系统则被解耦成若干个彼此间无关联的 SISO 伪线性积分系统。对于离散非线性 MMO 系统线性化解耦，则提出由一个静态神经网络和若干时延因子组成的神经网络α阶时延逆系统来进行解耦。这种方法不需要知道系统的精确模型，且结构简单，易于工程实现。但使用它的一个前提是系统必须可逆，而且神经网络是采取离线训练[61]。

此外，文献[45]构造了一种新的多层前向神经元网络，其隐含层单元分别为比例(P)、积分(I)、微分(D)项，其各层神经元个数、连接方式、连接权初值是由 PID 控制律的基本原则所确定的，它可用于多变量系统的解耦控制，其神经网络

训练时用的是单位阶跃信号。本文即在参考文献[45]中所提出的结构的基础上，将 PSD 算法推广到多变量系统中，用 PSD 单元取代其输出层，构成了 PSD 网络解耦结构。

4.1.3 PSD 网络解耦方法分类

上述利用神经网络进行解耦的方法，虽然结构各异，然而大部分都是离线训练神经网络，而后再将之应用于实际对象的解耦[59][60][61]。离线训练神经网络，所使用的训练样本是在某一固定工况下所采集的数据，当工况发生变化时，已训练好的神经网络却起不到好的解耦作用，需要重新离线进行训练，显然很不方便，也并不适用于对象时变较强的过程对象。而在实际生产过程中，被控对象间的耦合强度或耦合关系则经常是随负载或时间变化而改变的，这时离线解耦则显得很不经济。在线解耦相对离线解耦而言具有明显的优势[63]。它能够适应变化的工况，可不断地在线进行学习，当耦合关系发生变化时，神经网络可及时修改权值，使系统变量间的耦合关系得以解除。

根据连接方式和权值学习算法的不同，可将神经网络分为：感知器、多层前馈网络即 BP 网络、自适应线性神经元、径向基函数神经网络、小脑模型神经网络、PID 神经网络、全递归神经网络和连续型 Hopfield 网络等。其中，PID 神经网络在实际应用中因兼具了神经网络和广泛应用的 PID 控制的特点，得到了广泛应用和研究。

因此，研究和探索利用神经网络的可训练性同 PID 的易于实现性等特点实现在线解耦算法，具有重大的理论价值和应用前景。其中，PSD 网络解耦方法吸收了 PID 神经网络解耦控制器的结构特点和部分调节规律，兼具了 PSD 算法和神经网络的双重优点，可实现在线解耦和控制的统一。

PSD 网络解耦方法主要是将 PSD 算法和具有学习功能的神经元相结合，参考 PID 控制器的结构以及参数调节特点，以 PSD 单元取代原输出层，通过权系数的自调整达到解耦的目的。其中 PSD 网络借鉴了 PID 神经网络的结构特点同时也包含了 PSD 算法本身的不依赖模型的优点，是本文研究的重点。虽然 PSD 网络属于多层前向神经元网络的范畴，但它与一般多层前向神经元网络有所不同，一般多层前向神经元网络中的全部神经元的输入一输出特性都是相同的和静态的，而本问所研究的 PSD 网络的隐含层是由比例元、积分元和微分元组成的，是一种动态前向网络，更适合于系统控制；一般多层前向神经元网络的隐层神经元个数、连接权初值是随机确定的，在其控制下的系统稳定性得不到保证，控制系统初始不稳定，网络的收敛性失去了基础，这是很多神经元网络控制系统不能实用的原因之一。而 PSD 网络的各层神经元个数、连接方式、连接权初值是由

PSD 控制规律的基本原则和已有经验确定的，不仅能够保证系统稳定和参数的迅速收敛，而且又具备类似 PSD 方法的易于实现性的特点。

根据权值修正位置的不同及是否串联解耦，一般 PSD 网络有以下两种结构：

（一）设多输入多输出系统被控对象为 n 个输入，n 个输出的多变量系统。其控制器由单神经元 PSD 控制器构成，下面以 n=2 为例说明此种多输入多输出 PSD 解耦控制器的系统结构。系统结构如下图所示。

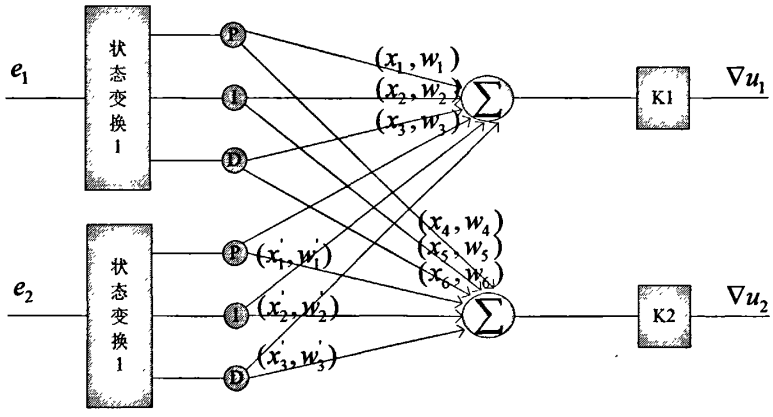


图 4-2 PSD 解耦控制器结构图

由上图所示，此种单神经元 PSD 解耦控制器由两个 PSD 神经元子网络组成，各个 PSD 自适应神经元到输出层之间不是独立的，互相之间有连接权值，这样两个 PSD 子网络构成了一个整体网络即 PSD 网络。它通过对加权系数的调整来实现自适应、自组织的功能，从而实现多变量系统的解耦控制。文献[45]和文献[62]中较为详细的分析了该种结构下算法的收敛性和稳定性，并给出了解耦仿真实例和结论。结论表明该种算法虽然继承了 PSD 算法的结构简单易于实现的特点，但是其解耦能力有限，对一些强耦合，滞后特性较强的对象仍然不能获得满意的解耦效果[45][62]。

（二）另外一种结构如下图，同样以一个双入双出系统为例说明其系统结构。系统结构如下图所示。

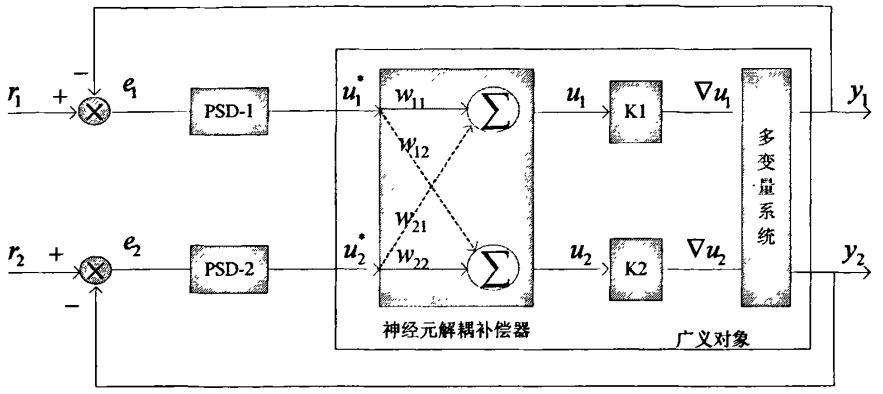


图 4-3 PSD 解耦补偿控制系统结构图

图中，r_1、r_2为系统设定值，y_1、y_2为系统测量输出值，PSD-1、PSD-2 是两个 PSD 控制器。采用图(4-3)中介绍的自适应 PSD 控制器结构，设计时只考虑单变量而不考虑其它变量之间的影响。在没有加解耦补偿装置时，与传统 PID 控制进行比较，这种控制器结构简单易于实时控制；并且参数能够自行调整。使用神经元解耦补偿器将来自其它通道的耦合影响视为可测干扰进行补偿，通过训练网络权值，使神经元解耦补偿器与被控对象组成的广义系统成为无耦合或耦合程度较小的系统[44][48]。采用解耦补偿装置位于控制器之后的串联开环解耦方案，在每对输入输出通道上设置一个单神经元，每个神经元有两个输入，均接受前端的两个自适应 PSD 控制器的控制信号u_1^*，u_2^*。后端的每个神经元的输出u_1，u_2作为补偿后的控制信号送至多变量系统，通过神经元权值的修正以达到解耦目的。

从整个解耦算法来看，神经元的各权值作为解耦环节，可以不需对象模型，仅根据一些过程信息就能达到自学习、自修正的目的。无需任何先验知识，仅需选择适当的学习速率 β，并且，神经元权值初始矩阵$\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$为相当于无解耦状态，在运行过程中，不断修正$w_{ij}$，实现在线解耦控制，使多变量系统的输出趋向于局部最优，取得满意的控制效果，具有很强的自适应性。故又将该补偿器称之为自适应神经元非模型多变量补偿器[48]。

仿真实验表明[48][67]，此种结构 PSD 解耦补偿控制算法虽然能够取得较好的控制效果，但是当输入个数增加时，神经元解耦补偿器的设计会较为复杂，而且不能保证系统的收敛性和稳定性。

4.2 PSD 网络算法原理

4.2.1 PSD 网络控制器结构及原理

综上所知，虽然图（4-2）、图（4-3）所设计的 PSD 解耦算法虽然结构上简单易行，但却各自存有缺点，不具有工程实现的普遍性。因此，有必要对 PSD 算法在多边量系统中的应用进行再研究。

本文借鉴文献[49]中提到的一种将 PID 控制器的优点与神经网络的优点相结合的方法，创建了一种 PSD 网络型控制器。结构如图（4-4），具体的说就是所创建的控制器是一个网络结构，因此它能够自适应的调整网络权值，而且该网络的输出层是由多变量输入的 PSD 单元组成；网络隐含层为混合的局部递归网络结构；网络权值具有较为明确的比例、积分和微分计算的意义。

设计者可以根据被控变量的数目就可以设计出网络的结构，并可以通过网络权值的在线修正来自动调整由系统参数和环境变量所引起的控制器参数变化。这样，它就克服了传统神经网络控制器设计时结构复杂，初始权值不容易设置获得等不利因素。

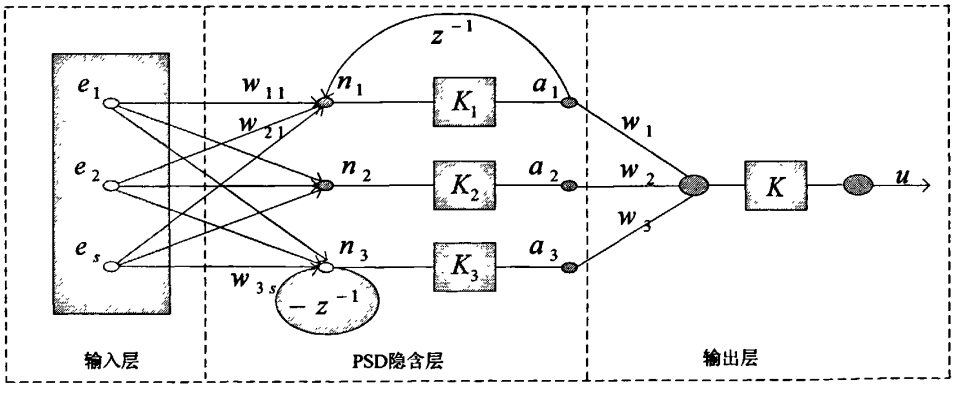


图 4-4 PSD 网络控制器结构图

该 PSD 网络控制器继承了 PSD 控制器与神经网络的双重优点。网络结构如图，网络控制器由 s 个输入节点 e_j $(j=1,2\cdots s)$、1 个输出节点 y 和 3 个隐含层节点组成；隐含层与输出层的激活函数取为线性函数，在隐含层的输入端和输出端分别存在部分自递归回路。其中，隐含层的第 1 个节点 a_1 具有输出反馈，它对该节点的输出进行单位延时，反馈到该节点的加权求和节点 n_1；隐含层的第 3 个节点 a_3 带有激活反馈，它对该节点的加权求和节点 n_3 的输出进行负单位延时，再

反馈到该加权求和节点 n_3 的输入；而隐含层的第 2 个节点 n_2 为不带有任何反馈的常规节点。

4.2.2 PSD 网络的输入、输出关系

由图(4-4) PSD 网络结构，可得隐含层各节点在 k 时刻的输出 $a_i(k)$ $(i=1,2,3)$ 分别为：

$$a_1(k)=\sum_{j=1}^{s}w_{1j}(k)e_j(k)+a_1(k-1) \tag{4-4}$$

$$a_2(k)=\sum_{j=1}^{s}w_{2j}(k)e_j(k) \tag{4-5}$$

$$a_3(k)=\sum_{j=1}^{s}w_{3j}(k)e_j(k)-\sum_{j=1}^{s}w_{3j}(k-1)e_j(k-1) \tag{4-6}$$

网络的最终输出为：

$$u(k)=\sum_{i=1}^{3}w_i(k)a_i(k) \tag{4-7}$$

由网络结构图及其所得到的网络输入/输出关系式可以看出：由于带有输出反馈，隐含层第 1 个神经元节点的输入/输出关系式（式 4-4）等价于：$a_1(k)=\dfrac{\sum_{j=1}^{s}w_{1j}(k)e_j(k)}{1-z^{-1}}$，表现出积分特性；而由于带有激活反馈，隐含层第 3 个神经元节点的输入/输出关系式（4-6）等价于：$a_3(k)=[\sum_{j=1}^{s}w_{3j}(k)e_j(k)](1-z^{-1})$ 呈现出微分特性；隐含层第 2 个神经元节点的输入/输出关系式（4-5）本身就呈现出比例特性，因而称之为比例节点。所以，整个网络在对输入/输出关系的调整上，表现出的是比例(第 2 个节点)+微分(第 3 节点)+积分(第 1 个节点)的特性，具有非常明确的物理意义。

由此可见，不同于以往采用神经网络调整 PID 参数的做法，所创建的控制器是由一种混合局部连接递归型神经网络组成。其结构简单，便于网络设计；加上隐含层的 3 组权值 w_{1j}、w_{2j} 和 w_{3j} $(j=1,2,\cdots)$ 分别起着比例因子 k_P、微分因子 k_D 和积分因子 k_I 的作用，网络参数物理意义明确；与一般只适用于单变量情况的 PID 控制不同，所创建的控制器可以很自然地扩展应用到多变量控制器的设计中。如输出变量增加一个时，只需要在隐含层增加三个节点，输出层增加一个输出节点即可。

4.2.3 PSD 网络参数的计算方法

本节将利用弹性 BP 算法中的符号函数来推导网络权值在线修正公式。作为

一般控制器的设计原则，我们的控制目标是使被控系统在 PSD 网络作用后，闭环系统参考输入与系统输出之间的误差平方最小。

设系统的输出变量为 $w_{ij}(k),(i=1,2,3;j=1,2,\cdots,s)$，j 为可测输出变量的数目；系统参考输入为 $r_j(k)(j=1,2,\cdots,s)$，见图 3-3，则 k 时刻对任意参考输入 $r_j(k)$ 和 $y_j(k)$ 之间的系统输出误差为

$$e_j(k)=r_j(k)-y_j(k) \tag{4-8}$$

那么，误差函数为：

$$J_j(k)=\frac{1}{2}[r_j(k)-y_j(k)]^2 \tag{4-9}$$

则 k 时刻系统的总误差函数为：

$$J(k)=\sum_{j=1}^{s}J_j(k) \tag{4-10}$$

采用所定义的误差函数 $J(k)$ 作为控制器设计的性能指标，这意味着所设计的 PSD 网络为闭环控制器，整个控制系统的结构图如图(4-5)所示。

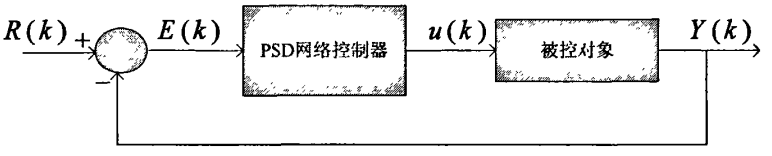


图 4-5 采用 PSD 网络控制器的系统结构图

其中 R(k)、Y(k) 均为向量。在所创建的网络控制器中，需要在第 k 个周期里先修正隐含层中的网络权值参数 $w_{ij}(k),(i=1,2,3;j=1,2,\cdots,s)$，然后计算出控制律 $u(k)$。在第 k 个周期获得系统输出值 $y(k)$、误差 $e(k)$ 以及性能指标 $J(k)$ 后，可得：

$$\begin{aligned}\frac{\partial J(k)}{\partial w_{ij}(k-1)}&=\sum_{j=1}^{s}[\frac{\partial J_j(k)}{\partial y_j(k)}\cdot\frac{\partial y_j(k)}{\partial u(k-1)}]\frac{\partial u(k-1)}{\partial w_{ij}(k-1)}\\&\triangleq-\sum_{j=1}^{s}\left\{e_j(k).\mathrm{sgn}[\frac{y_j(k)-y_j(k-1)}{u(k-1)-u(k-2)}]\right\}.e_j(k-1)\end{aligned} \tag{4-11}$$

在权值修正公式的计算中，考虑到系统输出 $y_j(k)$ 与输入 $u(k-1)$ 之间函数关系的复杂性和未知性，我们用差分概念来近似求解 $y_j(k)$ 对 $u(k-1)$ 的一阶偏导数。因此有，$\frac{\partial y_j(k)}{\partial u(k-1)}\approx\frac{\Delta y_j(k)}{\Delta u(k-1)}=\frac{y_j(k)-y_j(k-1)}{u(k-1)-u(k-2)}$ 并借用弹性 BP 算法的概念，对其使用符号函数，不考虑计算结果的大小，只根据其结果的正负取±1。这样不仅可以避免由于梯度太小所导致的停止修正权值的问题，还可以大大地简化计算，满足实时在线修正和控制的要求。由此可得隐含层权值修正公式为：

$$\begin{aligned}\Delta w_{ij}(k-1) &= -\eta_{ij}\cdot\frac{\partial J(k)}{\partial w_{ij}(k-1)} \\ &= \eta_{ij}\cdot\sum_{j=1}^{s}[e_j(k).\mathrm{sgn}\frac{\Delta y_j(k)}{\Delta u(k-1)}].e_j(k-1)\end{aligned} \tag{4-12}$$

其中，η_{ij}为隐含层权值的学习速率。由此可见，网络权值的变化与系统控制误差$e_j(k)$与符号函数$\mathrm{sgn}\frac{\Delta y_j(k)}{\Delta u(k-1)}$乘积的和以及网络输入$e_j(k-1)$成正比。隐含层在第$k$个周期里的权值计算公式为：

$$w_{ij}(k)=w_{ij}(k-1)+\eta_{ij}.\sum_{j=1}^{s}[e_j(k).\mathrm{sgn}\frac{\Delta y_j(k)}{\Delta u(k-1)}].e_j(k-1) \tag{4-13}$$

其中，$i=1,2,3$，$j=1,2,\cdots,s$

由于所创建网络的隐含层和输出层的激活函数都是取为线性的函数，且 PSD 网络的参数功能主要表现在隐含层。输出层只是起到一个多变量输出求和的目的，所以输出层的权值没有必要再进行调整计算。因为从网络角度来看，两层线性网络等于一层线性网络。w_i的变化可以由隐含层w_{ij}的来完成，为此我们取$w_i=1,i=1,2,3$。这也更加符合 PID 控制器的要求。

由此可见，不同于一般神经网络，PSD 网络的参数并不是通过不断的训练获得的，而是在每个采样周期里在线进行计算的。更重要的是，网络权值能够根据闭环系统误差性能指标进行在线调整，对被控对象提供非线性、自适应的实时在线控制，从而能够有效的克服对象参数时变对控制品质的影响。

4.2.4 PSD 网络初始权值的确定原则

PSD 网络的初始权值起着稳定系统的作用。根据这一要求，我们对初始权值的设计本着只要使系统稳定即可的原则，可以由多种方法来确定初始权值。

最直接的方法可以获取任意一组在某种策略稳定控制下的控制器的输入/输出数据，用此组数据对网络控制器进行适当的几千次的简单训练，以训练结束后的网络隐含层权值参数作为网络权值的初始值。不过在实际应用中，可以根据实际被控系统参数的特点以及与 PSD 网络参数之间的关系，利用一些已设计的常数控制器对 PSD 网络的参数进行初始化，如一组已知的 PSD 控制器参数或一组稳定运行的 LQR 参数等，都可以直接用来作为 PSD 网络的初始值。

PSD 网络在利用已有参数作为初始值的基础上，通过在线调整，能够自适应由系统非线性以及未建模因素所造成的影响，使系统的控制效果更佳，使其对各种干扰具有更强的鲁棒性。

由于网络权值初始值能够与常规线性控制器的参数相对应，具有明确的物理意义，因而所提出的网络控制器不仅在内部结构上具有很好的透明性，而且在网

络初始权值的设计与确定上也带来了一定方便之处。

4.2.5 算法实现

综上所述，网络控制器及其控制方法的设计及在线控制策略的具体实现步骤如下：

第一步，确定所要控制的被控对象的输出变量的数目，并以此作为确定网络输入节点的数目 s；

第二步，根据常规控制方法在被控系统的一般控制情况，确定隐含节点是采用 1 个、2 个还是 3 个；

第三步，使用常规控制方法获得参数 K_1、K_2、K_3 的值或者通过实际运行一种稳定的控制器获得一组控制器数据，对网络进行训练，来获得网络隐含层权值的初始值 $w_{ij}(0)$、网络输出层权值 $w_i=1$（$i=1,2,3$），并计算 $u(0)$；

第四步，根据调整规则确定合适的学习速率 η_{ij} 的值；

第五步，运行网络控制器同被控对象所组成的系统，在每一个 k 采样周期内读取系统输出量 $y_j(k)$，并根据式(4-8)计算系统输出误差值 $e_j(k)$；

第六步，根据式（4-13）计算权值调整系数 $w_{ij}(k)$ 的值；

第七步，根据式子（4-4~4-6），分别计算网络隐含层输出 $a_1(k)$、$a_2(k)$、$a_3(k)$ 的值；

第八步，根据式（4-7），计算控制器输出量。

第九步，迭代一次完成，$k=k+1$，返回第五步，循环运行。

4.3 PSD 网络算法仿真分析

为验证 PSD 网络控制方法对实际对象的解耦效果，选取第二章所建立的多温区温控系统动态数学模型进行仿真分析，对象的传递函数模型如下：

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} \dfrac{12.8e^{-s}}{16.7s+1} & \dfrac{-18.9e^{-3s}}{21s+1} \\ \dfrac{6.6e^{-7s}}{10.9s+1} & \dfrac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \cdot \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} \tag{4-14}$$

式中，$Y_1(s)$、$Y_2(s)$ 为系统两路输出量；$U_1(s)$、$U_2(s)$ 为系统控制输入量。

首先利用相对增益阵的方法很容易判定这个系统的耦合强度。其中计算所得对象的相对增益阵如下：

$$\Lambda = K\bullet(K^{-1})^T = \begin{bmatrix} 12.8 & -18.9 \\ 6.6 & -19.4 \end{bmatrix}\bullet\left(\begin{bmatrix} 12.8 & -18.9 \\ 6.6 & -19.4 \end{bmatrix}^{-1}\right)^T = \begin{bmatrix} 0.6656 & 0.3344 \\ 0.3344 & 0.6656 \end{bmatrix} \tag{4-15}$$

其中，定义（A•B）点乘运算表示 A、B 两个矩阵中各自相对应元素的乘积。由相对增益的物理意义知，回路之间关联严重，且当其中一个通道闭环后，相对的另外一个通道增益将增大，系统的稳定性往往有所下降。

为便于仿真研究，对对象进行离散化处理。离散化采样周期为 $T_s=2s$，

$$\begin{bmatrix} Y_1(z) \\ Y_1(z) \end{bmatrix} = \begin{bmatrix} \dfrac{0.744z+0.7007}{z^2-0.0071z} & \dfrac{-0.8789z-0.838}{z^2-0.909z} \\ \dfrac{0.5786z+0.5278}{z^2-0.8324z} & \dfrac{-1.302z-1.214}{z^2-0.8703z} \end{bmatrix}\cdot\begin{bmatrix} U_1(z) \\ U_2(z) \end{bmatrix}$$

转化为系统的差分方程为：

$$\begin{cases} (1-1.9163z^{-1}+0.0065z^{-2})y_1(k) = (0.744z^{-1}+0.02425z^{-2}-0.6370z^{-3})u_1 \\ \qquad\qquad +(-0.8789z^{-2}-0.8300z^{-3}+0.0076z^{-4})u_2 \\ (1-1.7027z^{-1}+0.7244z^{-2})y_2(k) = (0.5786z^{-4}+0.0242z^{-5}-0.4593z^{-6})u_1 \\ \qquad\qquad +(-1.302z^{-2}-0.1302z^{-3}+1.0105z^{-4})u_2 \end{cases} \tag{4-16}$$

采用 PSD 网络解耦方法仿真如下(见附录 4)：

分别给通道#1 和#2 输入以下信号：

$$rin1(k) = \begin{cases} 5 & k <= 200 \\ 10 & k > 200 \end{cases}，\quad rin2(k) = \begin{cases} 10 & k <= 600 \\ 5 & k > 600 \end{cases}$$

仿真过程及结果分析如下：

图(4-6)通道#1 在 200s 处，通过将设定从 5 突变至 10 给通道#2 施加干扰，对比图(4-6)和图(4-7)可知通道#2 输出几乎未受影响；同时，在通道#2 在其 600s 时，将设定从 10 突变至 5 以给通道#1 施加干扰，从图(4-6)的 600s 时的仿真结果可以看到通道#1 稍受其影响，但是很快便能恢复稳定。因此，以上对比仿真试验，可以证明文中所提的 PSD 网络控制方法在多温区测控对象中能实现较好的解耦控制效果。

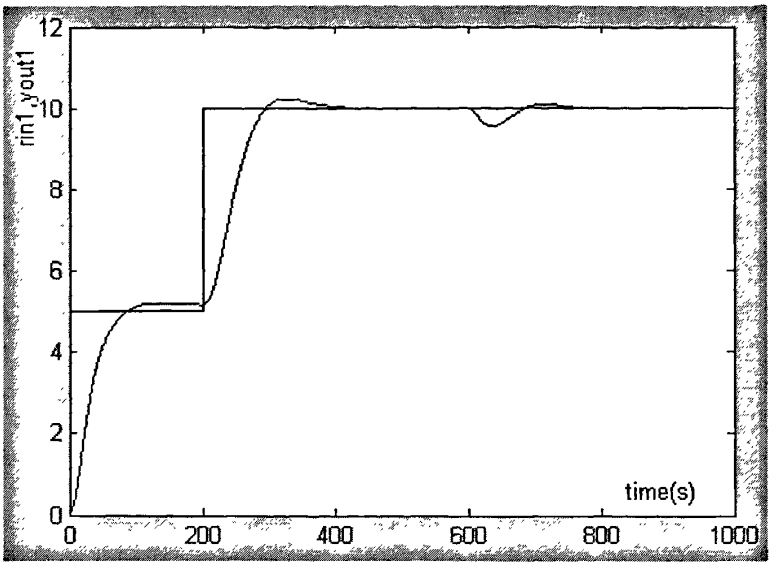


图 4-6 解耦后，通道#1 阶跃响应输出

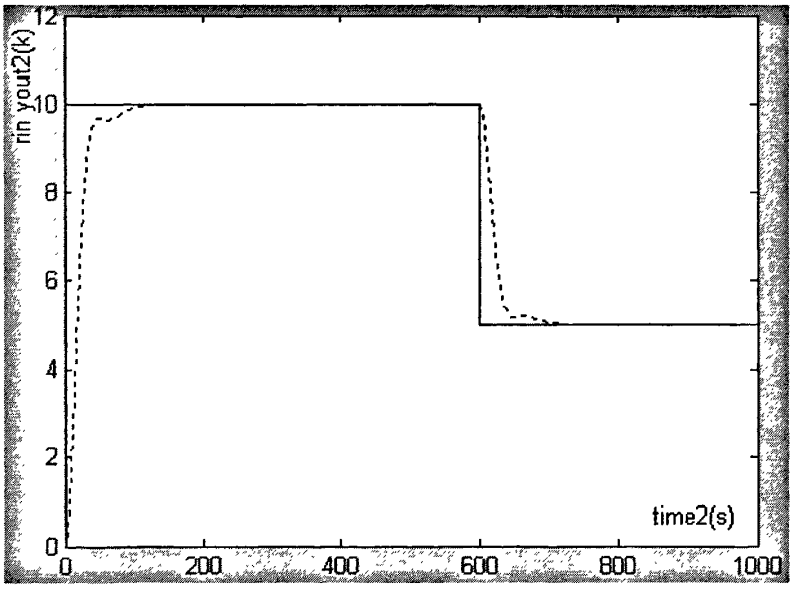


图 4-7 解耦后，通道#2 阶跃响应输出

4.4 本章小结

本章在对比了现有的解耦控制方法的基础上，总结了这些方法的优缺点，从理论上总结了神经网络方法对模型不确定对象进行解耦控制的优越性。结论认为，传统 PSD 算法所具有的易于工程实现的特性和神经网络在非模型控制上所

具有的优势，二者的结合对模型不确定的对象的控制是一个好的方法。基于此，本文将 PSD 算法在 MIMO 系统中作了推广，通过比较现有的几种 PSD 网络控制器的结构特点，提出了 PSD 网络的概念。并从理论上阐述了 PSD 网络算法的优势所在。

为验证所提算法的有效性，本文在第二章所建立的多温区测控系统模型基础上，针对对象的滞后、耦合的特性，将 PSD 网络控制算法应用到该系统中，给出了算法的仿真实现步骤，并对仿真结果做了总结与分析。结论表明，PSD 网络方法不仅具有 PSD 算法的能克服对象的滞后、参数时变且简单易行的特性，而且具有一定的解耦功能。

第五章 总结和展望

5.1 论文工作的总结

本论文在多温区测控系统模型基础上进行研究，主要完成了以下几个方面的工作：

1．根据多温区测控系统的运行机理，总结了对象模型所具有的大滞后、模型不确定、强耦合的特征。针对该特征，重点对比研究了适用于此类对象的传统控制方法，给出了仿真结果，总结了它们各自的优缺点。并在此基础上，引出了自适应 PSD 算法。并针对 PSD 算法所存在的一些问题，提出了一些相应的改进方法。针对增益设置的问题，引入了自适应律，形成了自适应 PSD 算法，实现了增益的自适应调整。通过仿真验证了该算法对大滞后、模型参数时变较大的对象具有良好的应用效果。

2．对 PSD 算法在 MIMO 耦合系统中的推广应用做了初步探讨。首先对传统的解耦方法做了总结，并重点对 PSD 同神经网络结合的方法做了总结分析，总结了各种 PSD 网络的结构特点。并将所研究的 PSD 网络算法应用于第二章所建立的多温区测控耦合系统。仿真结果表明，所提出的 PSD 网络算法不仅具有 PSD 算法的能克服对象的滞后、参数时变且简单易行的特性，而且具有一定的解耦能力。

5.2 后续工作的展望

在论文研究工作过程中，由于本人专业水平的局限性，在理论和技术上仍有以下几个方面需要进一步深入和完善：

1．模型算法的再研究

本文中将多温区测控系统模型看作为由存在耦合的多个单温区模型组成，各个温区之间存在耦合关系。其中，单温区数学模型用一阶惯性加纯滞后环节进行近似描述，虽然考虑到滞后的部分以及模型参数时变对系统的影响，但是对模型结构改变对算法的影响并没有进行研究。因此，有必要针对对象变结构的情况下，对算法的适应性进行再研究。

2．PSD 网络算法的实验方法

本章虽然将所提的 PSD 网络算法应用到所建立的多温区测控系统耦合对象中，从仿真的角度验证并讨论了该算法的非模型及解耦特性。但是并未对工程实

现过程中所要注意的问题做太深入的讨论。因此，后续工作中有必要针对 PSD 网络控制算法建立一个完善的实验平台，完成在合适硬件平台上的实验工作。并且，希望可以同时完善一套测控系统软件，为进一步的工作做好基础。

参考文献

[1] 韩江红，鲁照权、陆阳. 滞后不确定系统控制. 合肥工业大学学报，2000，23(1):16～20.

[2] 翁志成. 对各类过程控制策略应用前景的探讨. 基础自动化，1996，2:1～3.

[3] 王顺晃，舒迪前编著.《智能控制系统及其应用》(第二版). 北京：机械工业出版社，2005.

[4] 邵惠鹤 编著.《工业过程高级控制》(第二版). 上海：上海交通大学出版社，2003.

[5] 蔡自兴 编著.《智能控制》. 北京：电子工业出版社，1990.

[6] 邵裕森，戴先中 编著.《过程控制工程》(第二版). 北京：机械工业出版社，2004.

[7] 林瑞全，邱公伟，罗旺春. 一类二自由度 PID 控制的神经元实现研究. 系统仿真学报，2004-3，16(3):424～440.

[8] 邓刚毅，张卫东，顾诞英. 不稳定时滞过程的二自由度控制. 上海交通大学学报，2004-2，38(2):254～258.

[9] 肖军，任挺进等. 基于 PSD 算法的单神经元 PID 控制器在汽温控制中应用. 自动化仪表，2004，25(1):4～7.

[10] 赵锡龄，焦云婷. 单神经元自适应控制 PSD 在再热汽温控制中的应用. 中国机电工程学报，2001，21(2):94～96.

[11] 何立红等. 带 Smith 预估器的模糊 PID 组织温度控制系统. 中国纺织大学学报，1999，25(3):84～86.

[12] 孙德辉，穆志纯，赵仁涛. PSD-Smith 自适应网络控制机制与稳定性分析. 计算机工程， 2006，32(4):103～105.

[13] 李遵基，袁应钦，段志刚. 主汽温自适应预估控制方法的研究. 华北电力大学学报 1996-10，23(4):98～101.

[14] 姚荣斌，孙红兵. 一类非线性滞后系统的自整定 PID 控制. 淮阴师范大学学报（自然科学版）:289～292.

[15] 游文明，张友宏. 变结构神经元自适应 PSD 控制算法研究. 机械与电子，2004(4):36～39.

[16] 丁红，舒迪前. 无需辨识模型的 PSD 自适应控制器及其应用. 冶金自动

化. 1994，18(2):35～39.

[17] 孟令柏，姬晓飞，申东义，陈义娥. 滞后不确定系统的无辨识自适应智能控制方法. 控制理论与应用，2003，22(3):10～12.

[18] 陈东劲，蒋新华. 基于模糊控制的硬质合金炉温度控制器的设计. 福建工程学院学报，2005，3(6):608～611.

[19] Jin H P，Su W S，Lee IB. An enhanced PID control strategy for unstable processes. Automatica，1998，34(6):751～756.

[20] Huang H P，Chen C C. Control-system synthesis for open-loop unstable process with time delay. IEE Proc D:Control Theory Appl，1997，144(4):334～346.

[21] Zhang Zhiqiang ，Wang Shunhuang ， Shu Diqian. Intelligent Control of the Temperature in the Electronic Heating Furnace of Mechanics strength Machine. PLC. AMSE ，1992，Hefei，china.

[22] 向涛. 基于热平衡机理的多温区动态建模及控制研究:[硕士学位论文]. 长沙:中南大学，2006

[23] Powell，Reinhard Edison.Development of convective solder reflow and projection moire system and FEA model for PWBA warpage prediction. George W.Woodruff School of Mechanical Engineering，2006

[24] 常压塔二线凝点闭环控制的实现方法. 自动化仪表，1999，20(8):37～38.

[25] 陈金娥. 由实测响应曲线数值求解传递函数. 热能动力工程，1995，10(4):193～196.

[26] Zheng F J，Shi F.A Real-time Fuzzy Control for a Kind of Multivariable Object.Proceedings of The IEEE International Conference on Industrial Technology，1996.

[27] 王金廉，贾青，谢剑英. 用 SLC500 实现多温区电加热炉的 PID 解耦控制. 测控技术，2001，20(2):24～26.

[28] Bristol E H. On a New Measure of Interaction for Multivariable Process Control. IEEE Transactions on Automatic Control，1966.

[29] Bruns D D and C B Smith. Singular Value Analysis: A Geometrical Structure for Multivariable Processes. AICHE Winter Meeting，1982-34 Lau H.et al. Synthesis of Control structure by singlilar value Analysis: Dynamic Measures of Sensitivity and Interaction. Aich E J，1985，31(3):427:439.

[30] Tung L S and T F Edgar. Analysis of control output Interactions in Dynamic Systems. Aich E J，27，1981，4:690.

[31] Mijares G et al. A New Crierion for the Pairing og Control and Manipulated

Variable Control Systems. AichE J，1986，32(9):1439～1449.

[32] Jensen N，D G fisher and S L Shah. Interaction Analysis in Multivariable control systems. AichE J，1986，32(6):959～970.

[33] 韩志刚. 无模型控制的应用. 控制工程，2002，9(22):22～25.

[34] 蒋爱平，韩志刚，韩丹. 无模型控制系统在加热炉温度控制上的应用. 控制工程，2004，11(5):388～391.

[35] 曾贵娥，朱学峰，李海生， MFA 控制器控制大时滞过程的实验研究. 控制工程，2005，12(增刊):49～51.

[36] 诸静 等著. 模糊控制原理与应用（第二版）. 北京:机械工业出版社 2005.

[37] 王华强，李海波，胡平. 一种改进的自适应 Smith 预估器，合肥工业大学学报 2007-3，30(3):316～318.

[38] 李德伟，席裕庚. 预估网络控制系统的设计和分析[J]. 2007-9，22(9):1065～1069.

[39] 席裕庚. 预测控制. 北京:国防科技出版社，1993.

[40] Kaya I，Atherton D P. A new PI-PD Smith predictor for control of processes with long dead time. beijing: 14th IFAC World Congress，1999.

[41] 陈一秀 王永初 直线伺服系统的鲁棒保性能控制研究. 中国电机工程学报[J]. 2006，26(24):28～31.

[42] 徐英，徐用懋，杨尔辅 时变大纯滞后系统的单神经元预测控. 清华大学学报(自然科学版). 2002，42(3):383～385.

[43] 丁军，徐用懋 单神经元自适应 PID 控制器及其应用. 控制工程. 2004，11(1):27～20.

[44] 徐中 孙伟，刘晓冰 单神经元自适应 PID 控制器的研究. 大连理工大学学报 1999-9，39(5):667～672.

[45] 舒怀林. PID 神经元网络多变量控制系统分析. 自动化学报 1999-1，25(1):105～111.

[46] 庞国仲等，多变量控制系统实践，中国科学技术大学出版社，1990.

[47] 方崇智等，过程辨识，清华大学出版社，1988.

[48] 李明，林永君，马永光 自适应神经元非模型多变量系统解耦控制. 计算机仿真 2003-3 20(3):68-71.

[49] 从爽，梁艳阳，李国栋. 多变量自适应 PID 型神经网络控制器及其设计方法. 信息与控制 2006-10，35(5):568～573.

[50] 陈鹏，郑应文，户占良 时变大纯滞后系统的单神经元自适应控制. 2005-3，24(1):17～19.

[51] 郭启刚 热工过程多模型控制理论与方法的研究 [博士学位论文], 2007-4.

[52] 余文. 多变量自适应解耦[硕士学位论文] 沈阳:东北大学. 1994.

[53] 王军等. PID 神经网络解耦控制技术在 VAV 中的应用. 西北大学学报(自然科学版). 2002, 32(3):237～239.

[54] 李学锋, 杨先发, 周凤岐 多变量模糊解耦控制系统研究与应用. 西北工业大学学报 1995-2, 13(1):56～60.

[55] 刘国荣, 多变量系统模糊解耦自适应控制. 控制理论与应用 1997-4, 14(2):152～156.

[56] 舒怀林,郭秀才,舒杰磊 注塑机料筒多段温度 PID 神经网络解耦控制系统, 计算技术与自动化, 2004-12, 23(4):55～57.

[57] 贺国艳, 杨马英, 俞立. 神经网络解耦预测函数控制. 控制工程 2002-7:9(4):63～70.

[58] 靳其兵, 曾东宁, 王云华, 顾树生. 多变量系统的神经网络解耦新方法. 东北大学学报 (自然科学版) 1999-6, 20(3):250～252.

[59] 冯恩波, 愈金寿, 蒋慰孙. 基于神经网络技术的解耦及容错解耦控制策略. 信息与控制, 1992. 21(6):363～368.

[60] 戴先中, 何月等 非线性 MIMO 系统线性化解耦的一种新方法(I)——连续时间系统. 控制与决策 1999, 14(5):403-406.

[61] 何月, 戴先中等. 非线性 MIMO 系统线性化解耦的一种新方法(II)——离散时间系统. 控制与决策 1999, 14(6):631～635.

[62] 杨金芳, 马平, 张华磊 自适应神经元网络在耦合系统中的应用. 信息技术 2004-10, 28(10):42～45.

[63] 柴天佑 多变量自适应解耦控制及应用 北京: 科学出版社, 2001.

[64] 张杰,邹继刚,李文秀. 多输入多输出系统的神经网络 PID 解耦控制器. 哈尔滨工程大学学报. 2000-10, 21(5):6～9.

[65] A.N.Gündes, H.özbay, A.B.öüler PID controller synthesis for a class of unstable MIMO plants with I/O delays. automatica, 2007, 43:135～142.

[66] Min Han, Bing Han, Jianhui Xi, Kotaro Hirasawa, Universal learning network and its application for nonlinear system with long time delay.Computers& Chemical Engineering, 2006, 31:13～20.

[67] 姜萍, 李遵基, 梁伟平, 胡风岗 一种基于神经元的解耦控制算法. 华北电力大学学报. 2000, 27(2):47～51.

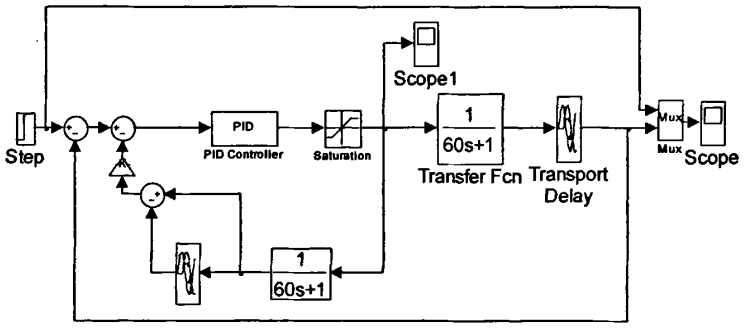
附 录

附录 1：达林算法对滞后对滞后对象的控制仿真

```
clear all;
close all;
ts=0.5;

%仿真对象，
sys1=tf([1],[0.4,1],'inputdelay',0.76);
dsys1=c2d(sys1,ts,'zoh');
[num1,den1]=tfdata(dsys1,'v');
%the non_jingque plant model
sys11=tf([1],[0.4,1],'iodelay',1.52);      %针对对象参数的变化，修改 T 和 tao，仿真对比
dsys11=c2d(sys11,ts,'zoh');
[num11,den11]=tfdata(dsys11,'v');

%期望闭环特性
sys2=tf([1],[0.15,1],'inputdelay',0.76);
dsys2=c2d(sys2,ts,'zoh');
sys22=tf([1],[0.15,1],'inputdelay',0.76);
dsys22=c2d(sys22,ts,'zoh');

%达林控制器输出表达式
dsys=1/dsys1*dsys2/(1-dsys2);
[num,den]=tfdata(dsys,'v');

u_1=0.0;u_2=0.0;u_3=0.0;u_4=0.0;u_5=0.0;              %%各量的迭代
y_1=0.0;

error_1=0.0;error_2=0.0;error_3=0.0;
ei=0;
%%%%%%%%%%%%%%以下对象用以做对比试验，%%%%%%%%%%%%%%%%%%%%
dsys22=1/dsys11*dsys22/(1-dsys22);
[num22,den22]=tfdata(dsys22,'v');

u2_1=0.0;u2_2=0.0;u2_3=0.0;u2_4=0.0;u2_5=0.0;
y2_1=0.0;

error2_1=0.0;error2_2=0.0;error2_3=0.0;
ei2=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k=1:1:50
time(k)=k*ts;

rin(k)=1.0;  %单位阶跃输入
rin2(k)=1.0;

%%%%%%%%%%对象输出，因为参数改变后会用所变化，要具体分析
yout(k)=-den1(2)*y_1+num1(2)*u_2+num1(3)*u_3;
error(k)=rin(k)-yout(k);
yout2(k)=-den11(2)*y2_1+num11(2)*u2_2;
```

```
%yout2(k)=1;
error2(k)=rin2(k)-yout2(k);

M=1;
if M==1          %求取达林控制器的输出
u(k)=(num(1)*error(k)+num(2)*error_1+num(3)*error_2+num(4)*error_3...
      -den(3)*u_1-den(4)*u_2-den(5)*u_3-den(6)*u_4-den(7)*u_5)/den(2);
u2(k)=(num22(1)*error2(k)+num22(2)*error2_1+num22(3)*error2_2+num(4)*error2_3...
      -den22(5)*u2_1-den22(6)*u2_2-den22(7)*u2_3-den22(8)*u2_4-den22(9)*u2_5)/den22(4);

elseif M==2      %Using PID Method
ei=ei+error(k)*ts;
u(k)=1.0*error(k)+0.10*(error(k)-error_1)/ts+0.50*ei;
end

u_5=u_4;u_4=u_3;u_3=u_2;u_2=u_1;u_1=u(k);
y_1=yout(k);
error_3=error_2;error_2=error_1;error_1=error(k);

u2_5=u2_4;u2_4=u2_3;u2_3=u2_2;u2_2=u2_1;u2_1=u2(k);
y2_1=yout2(k);
error2_3=error2_2;error2_2=error2_1;error2_1=error2(k);
end
figure(1);
plot(time,rin,'b',time,yout,'r',time,yout2,'b');
xlabel('time(s)');ylabel('rin,yout');
```

附录 2： Smith 预估算法仿真结构图



附录 3： 自适应 PSD 算法仿真程序

```
clear all;
close all;

x=[0,0,0]';
x2=[0,0,0]';

%xiteP=0.40;
xiteP=0.40;
xiteI=0.20;
xiteD=0.40;

%%%初始化 kp,ki and kd
```

```
wkp_1=0.10; wki_1=0.10; wkd_1=0.10;
Kk_1=0.03;   Tk_1=10;

wkp2_1=0.10; wki2_1=0.10; wkd2_1=0.10;
Kk2_1=0.03;   Tk2_1=10;
%%%%注意此处 K 的初值的选择，应该从较小的 0.001 选取，避免调节时间过长，而且 T
%%的处置幺选择一个较大值

error_1=0;
error_2=0;
y_1=0;y_2=0;y_3=0;
u_1=0;u_2=0;u_3=0;u_4=0;u_5=0;          u_6=0;u_7=0;u_8=0;u_9=0;

error2_1=0;
error2_2=0;
y2_1=0;y2_2=0;y2_3=0;
u2_1=0;u2_2=0;u2_3=0;u2_4=0;u2_5=0;                u2_6=0;u2_7=0;u2_8=0;u2_9=0;u2_10=0;
u2_11=0;u2_12=0;


%sys=tf([1.75],[3105 1],'iodelay',150);
%zsys=c2d(sys,1,'zoh');
%[num,den]=tfdata(zsys);

ts=5;
for k=1:1:500
    time(k)=k*ts;
    time2(k)=k*ts;

   rin(k)=1;
   rin2(k)=1;

    yout(k)=0.8465*y_1+0.1535*u_9;      error(k)=rin(k)-yout(k);

    %yout2(k)=0.8465*y2_1+0.1535*u2_12; error2(k)=rin2(k)-yout2(k);
    yout2(k)=0.8948*y2_1+0.1052*u2_8; error2(k)=rin2(k)-yout2(k);% 150Ts
   % yout2(k)=0.7165*y2_1+0.2835*u2_8; error2(k)=rin2(k)-yout2(k);
%Adjusting Weight Value by hebb learning algorithm

   wkp(k)=wkp_1+xiteP*error(k)*u_1*x(1);   %P
   wki(k)=wki_1+xiteI*error(k)*u_1*x(2);   %I
   wkd(k)=wkd_1+xiteD*error(k)*u_1*x(3);   %D
   if(sign(error(k))==sign(error_1))                %%效果和未修改的的符号函数 基本相同
%K=0.0028;K=0.0030;

%%%%%%          权值修正的自适应律          %%%%%%%%%%%%
if( ( (sign(error(k))==-1)&&(sign(error_1)==-1) )||( (sign(error(k))==1)&&(sign(error_1)==1) ) )
        K(k)=Kk_1+0.04*(Kk_1/Tk_1);
    else
        K(k)=0.75*Kk_1;
    end
    T(k)=Tk_1+0.08*sign(abs(error(k)-error_1)-Tk_1*abs(error(k)-2*error_1+error_2));
   %%%%%%%%%%%%    权系数迭代算式    %%%%%%%%%%%
   wkp2(k)=wkp2_1+xiteP*error2(k)*u2_1*x2(1);   %P
   wki2(k)=wki2_1+xiteI*error2(k)*u2_1*x2(2);   %I
   wkd2(k)=wkd2_1+xiteD*error2(k)*u2_1*x2(3);   %D
   if(sign(error2(k))==sign(error2_1))
```

```
        K2(k)=Kk2_1+0.04*(Kk2_1/Tk2_1);
    else
        K2(k)=0.75*Kk2_1;
    end
T2(k)=Tk2_1+0.08*sign(abs(error2(k)-error2_1)-Tk2_1*abs(error2(k)-2*error2_1+error2_2));

    x(1)=error(k)-error_1;                  %P
    x(2)=error(k);                          %I
    x(3)=error(k)-2*error_1+error_2;        %D
    wadd(k)=abs(wkp(k))+abs(wki(k))+abs(wkd(k));
    w11(k)=wkp(k)/wadd(k);
    w22(k)=wki(k)/wadd(k);
    w33(k)=wkd(k)/wadd(k);
    w=[w11(k),w22(k),w33(k)];
    %u(k)=u_1+0*w*x;
     u(k)=u_1+K(k)*w*x;
if u(k)>10
    u(k)=10;
end
if u(k)<-10
    u(k)=-10;
end
%%%%%%%%%%%%%          Control law      %%%%%%%%%%%%%%%
    x2(1)=error2(k)-error2_1;               %P
    x2(2)=error2(k);                        %I
    x2(3)=error2(k)-2*error2_1+error2_2;    %D

    wadd2(k)=abs(wkp2(k))+abs(wki2(k))+abs(wkd2(k));
    w112(k)=wkp2(k)/wadd2(k);
    w222(k)=wki2(k)/wadd2(k);
    w332(k)=wkd2(k)/wadd2(k);
    w2=[w112(k),w222(k),w332(k)];
    %u(k)=u_1+0*w*x;
     u2(k)=u2_1+K2(k)*w2*x2;
if u2(k)>10
    u2(k)=10;
end
if u2(k)<-10
    u2(k)=-10;
end
 %%%%%%%%%  Control law    %%%%%%%%
error_2=error_1; error_1=error(k);
u_9=u_8;u_8=u_7;u_7=u_6;u_6=u_5;u_5=u_4; u_4=u_3;u_3=u_2;u_2=u_1;u_1=u(k);

Kk_1=K(k);   Tk_1=T(k);
y_1=yout(k);

wkp_1=wkp(k); wkd_1=wkd(k); wki_1=wki(k);

error2_2=error2_1;
error2_1=error2(k);

u2_8=u2_7;u2_7=u2_6;u2_6=u2_5;u2_5=u2_4; u2_4=u2_3;u2_3=u2_2;u2_2=u2_1;u2_1=u2(k);

Kk2_1=K2(k);   Tk2_1=T2(k);
y2_1=yout2(k);
```

```
wkp2_1=wkp2(k); wkd2_1=wkd2(k); wki2_1=wki2(k);
end
figure(1);
plot(time,rin,'b',time,yout,'r');
xlabel('time(s)');ylabel('rin,yout');
hold on
plot(time2,yout2,'b:');
hold off
figure(2);
plot(time,rin,'b',time2,error2,'r');
xlabel('time2(s)');ylabel('error2');
```

附录 4： 自适应 PSD 算法解耦例程

```
%%%%%%%%%%%%%%%%%%%%%%
clear all;   close all;

x=[0,0,0]';   x2=[0,0,0]';

xiteP=0.40;   xiteI=0.20;   xiteD=0.40;

% kp,ki 和 kd 初始化
wkp_1=0.10; wki_1=0.10; wkd_1=0.10;
K_1=0.001;   T_1=10;
wkp2_1=0.10; wki2_1=0.10; wkd2_1=0.10;
K2_1=0.001; T2_1=10;

%%%%注意此处 K 的初值的选择，应该从较小的 0.001 选取，避免调节时间过长
error_1=0;
error_2=0;
y_1=0;y_2=0;y_3=0;
u1_1=0;u1_2=0;u1_3=0;u1_4=0;u1_5=0;          u1_6=0;u1_7=0;u1_8=0;u1_9=0;

error2_1=0;
error2_2=0;
y2_1=0;y2_2=0;y2_3=0;
u2_1=0;u2_2=0;u2_3=0;u2_4=0;u2_5=0;          u2_6=0;u2_7=0;u2_8=0;u2_9=0;u2_10=0;
u2_11=0;u2_12=0;

ts=0.1;
for k=1:1:10000
    time(k)=k*ts;
    time2(k)=k*ts;
   if k>2000
   rin(k)=10;
  else
   rin(k)=5;
  end

  if k<6000
      rin2(k)=10;
  else
      rin2(k)=5;
  end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
yout(k)=1.9771*y_1-0.9772*y_2+0.152*u1_1-0.00005*u1_2-0.015*u1_3+0.0037*u2_3-0.00365
*u2_4;
error(k)=rin(k)-yout(k);
```

```
yout2(k)=1.9844*y2_1-0.9845*y2_2+0.00433*u1_2+0.00001*u1_3-0.00426*u1_4+0.01412*u2_
2-0.014*u2_3;
error2(k)=rin2(k)-yout2(k);
%仿真控制算法选择

    wkp(k)=wkp_1+xiteP*error(k)*u1_1*(2*error(k)-error_1);   %P
    wki(k)=wki_1+xiteI*error(k)*u1_1*(2*error(k)-error_1);   %I
    wkd(k)=wkd_1+xiteD*error(k)*u1_1*(2*error(k)-error_1);   %D

       if(sign(error(k))==sign(error_1))
           K(k)=K_1+0.04*(K_1/T_1);
       else
           K(k)=0.75*K_1;
       end
     T(k)=T_1+0.08*sign(abs(error(k)-error_1)-T_1*abs(error(k)-2*error_1+error_2));
    %%%%%%%%%%%%%%%%----------control---------%%%%%%%%%%%%%%%%%%%%
    wkp2(k)=wkp2_1+xiteP*error2(k)*u2_1*(2*error2(k)-error2_1);   %P
    wki2(k)=wki2_1+xiteI*error2(k)*u2_1*(2*error2(k)-error2_1);   %I
    wkd2(k)=wkd2_1+xiteD*error2(k)*u2_1*(2*error2(k)-error2_1);   %D

     if(sign(error2(k))==sign(error2_1))
           K2(k)=K2_1+0.04*(K2_1/T2_1);
       else
           K2(k)=0.75*K2_1;
       end
     T2(k)=T2_1+0.08*sign(abs(error2(k)-error2_1)-T2_1*abs(error2(k)-2*error2_1+error2_2));

    x(1)=error(k)-error_1;                    %P
    x(2)=error(k);                             %I
    x(3)=error(k)-2*error_1+error_2;     %D

    wadd(k)=abs(wkp(k))+abs(wki(k))+abs(wkd(k));
    w11(k)=wkp(k)/wadd(k);
    w22(k)=wki(k)/wadd(k);
    w33(k)=wkd(k)/wadd(k);
    w=[w11(k),w22(k),w33(k)];
    u1(k)=u1_1+K(k)*w*x;
    %u1(k)=u1_1+K1*w*x;
if u1(k)>100
    u1(k)=100;
end
if u1(k)<-100
    u1(k)=-100;
end
%%%%%%%%%%%%%%          Control law     %%%%%%%%%%%%%%
    x2(1)=error2(k)-error2_1;                  %P
    x2(2)=error2(k);                            %I
    x2(3)=error2(k)-2*error2_1+error2_2;     %D

    wadd2(k)=abs(wkp2(k))+abs(wki2(k))+abs(wkd2(k));
    w112(k)=wkp2(k)/wadd2(k);
    w222(k)=wki2(k)/wadd2(k);
    w332(k)=wkd2(k)/wadd2(k);
    w2=[w112(k),w222(k),w332(k)];
    %u(k)=u_1+0*w*x;
```

```
    u2(k)=u2_1+K2(k)*w2*x2;
if u2(k)>10
   u2(k)=10;
end
if u2(k)<-10
   u2(k)=-10;
end
 %%%%%%%%          控制律          %%%%%%%%%
error_2=error_1; error_1=error(k);
u1_9=u1_8;u1_8=u1_7;u1_7=u1_6;u1_6=u1_5;u1_5=u1_4;
u1_4=u1_3;u1_3=u1_2;u1_2=u1_1;u1_1=u1(k);

%Kk_1=K(k);   Tk_1=T(k);
y_3=y_2;y_2=y_1;y_1=yout(k);

wkp_1=wkp(k); wkd_1=wkd(k); wki_1=wki(k);

error2_2=error2_1;   error2_1=error2(k);
u2_12=u2_11;u2_11=u2_10;u2_10=u2_9;u2_9=u2_8;u2_8=u2_7;u2_7=u2_6;u2_6=u2_5;u2_5=u
2_4; u2_4=u2_3;u2_3=u2_2;u2_2=u2_1;u2_1=u2(k);

K_1=K(k);   T_1=T(k);
y2_3=y2_2;y2_2=y2_1;y2_1=yout2(k);

wkp2_1=wkp2(k); wkd2_1=wkd2(k); wki2_1=wki2(k);
K2_1=K(k);   T2_1=T2(k);
end
figure(1);
plot(time,rin,'b',time,yout,'r');
xlabel('time(s)');ylabel('rin,yout');

figure(2);
plot(time,rin2,'r',time,yout2,'b:');
%plot(time,K,'r');
xlabel('time2(s)');ylabel('yout2(k)');
```

致 谢

在本论文的完成过程中，得到了很多很多人的帮助。我首先衷心感谢我的导师鲁五一教授，感谢他在我攻读硕士研究生学位期间对我学业上的悉心培养和生活上无微不至的关照。感谢鲁老师在读研期间所给予我的机会和信任；感谢他赠予我的“做事要用脑，更要用心”，这些谆谆教诲，将会让我受益终生。

同时感谢熊红云老师和王莉老师，熊老师严谨的治学之风、积极探索的奋斗精神、高尚的个人修为都将是我以后学习、工作的参照与指导。王老师不论是生活上还是学业上，都给了我深切的关怀和指导，并给予我无私的帮助。

感谢我的师兄张宏亮、汪学军以及韦小惠师姐、许寰师弟以及实验室所有的兄弟姐妹们对我的帮助、关心和支持。在工程实践中，有他们帮助让我能进入MCU、编程的天地。他们的经验让我受益匪浅，他们的耐心帮助让我终身难忘，让我在实验室度过了一段美好而充实的时光。衷心感谢他们。

最后，感谢我的家人和朋友对我的一贯支持、鼓励和理解。他们的坚忍、勤劳和无私付出，无时无刻不在鞭策着我，让我可以冲破一切困难，潜心做事；在最困难的时候，能够让我重获力量。

衷心的感谢他们，祝福他们身体健康、万事如意！

李长德

2008 年 4 月

攻读硕士学位期间科研及论文完成情况

一. 参与科研项目情况：

1. 石膏矿规范开采远程监控系统 (2006.9 月中--2006.12 月初)
2. FreeScale 智能小汽车大赛 (2006.12 月初--2006.寒假)
3. 讲授《微型计算机控制技术》课程 (2007.3.12--2007.4.24)
4. 汽车故障诊断平台 (2007.7.31--2007.9 月中)

二. 发表论文情况：

1. 李长德，鲁五一，王莉. 一种自适应 PSD 伺服控制器的研究. 计算机工程与科学，2008，3(30):139~141.