

## 摘要

在过去的十几年中, 化学加工工业 (CPI) 发生了巨大的变化, 间歇过程又重新受到了人们的普遍重视。间歇过程指的是将有限量的物料按规定的加工顺序在一个或多个设备中加工以获得有限量的产品的加工过程。

由于间歇过程通常具有非线性、时滞和不确定性的特点, 因此用常规的建模方法建立模型比较困难。智能系统理论是以众多新兴学科为基础发展起来的, 智能系统辨识方法理论是属于智能系统的一个研究范畴, 其中所包含的模糊系统理论、神经网络理论和进化计算理论都提供了面向这类复杂对象建立模型的方法。

本文在研究了包括传统系统辨识与智能辨识方法的基础上, 对氧乐果合成间歇生产过程的温度对象进行模型辨识。通过了解与分析氧乐果合成间歇过程的生产工艺, 找出了影响温度的主要因素以及温度的变化规律。然后将从现场采集到的各种历史数据进行必要的的数据预处理, 构成辨识所使用的输入输出数据。

本文的重点是利用神经网络方法来实现对象的模型辨识。通过研究 BP 神经网络的特点, 采用了 L-M 学习算法来完成辨识。利用该种算法, 学习训练的速度大大加快, 而且网络的泛化能力也较好。根据工艺了解到, 采用单一的神经网络模型具有一定的局限性。因此, 论文中提出了分阶段实现对象模型的方法, 即将温度按照工艺要求分成相应的几个阶段, 对各个阶段建立阶段的 BP 网络模型。BP 网络实现的是静态过程辨识, 如果要实现过程的动态辨识, 就要用到回归网络。本文是在 BP 网络的基础上加上 TDL 环节构成了回归网络。将温度数据通过一个 TDL 环节引入到神经网络输入端作为一个新的输入量来完成辨识。此外, 论文中还将利用各种方法辨识出来的模型引入到控制系统中, 进行现场情况的模拟仿真。

本文还研究了利用遗传算法优化模糊 T-S 模型参数的模糊系统辨识。在这一部分, 文章引入了广义 T-S 模型的概念, 从而更加方便了遗传算法的寻优过程, 并利用这种方法建立了对对象的广义 T-S 模糊系统模型。

另外, 本文还讨论了将仿真中使用的一甲胺流量控制信号转换为实际阀门开度信号。主要利用了曲线拟合的技术, 做出了流量曲线并给出了相应的经验方程。

**关键词:** 间歇过程; 系统辨识; 智能控制; 神经网络; 模糊 T-S 模型; 遗传算法

## Abstract

In the past decades, the chemical process industry has changed enormously. The batch process has been paid attention again. Batch process is a kind of process that gains finite production following job order by treating finite materials in one or more devices.

It is difficult to build a model for batch process using common methods because of the characteristics of non-linear, time-delay and indetermination. Intelligent system theory is based on many kinds of new knowledge, and intelligent identification theory belongs to it. The fuzzy system theory, neural network theory and evolve calculation theory in intelligent control theory provide model methods for batch process plants.

Based on researching traditional identification methods and those of intelligent control, the temperature plant of omethoate synthesis batch process will be identified. By comprehend and analysis the technology of it, the main factors that influences temperature and the laws of temperature variable are found. Input and output data for identification is made up of history data that has been pretreated from spot.

The important part of this paper is model identification with neural network method. After reaching the features of BP network, L-M learning algorithm is adopted. With this algorithm, the training is faster than ever and the ability of net generalization is better. According to technology, there is some limitation using simple neural network to build model. So the stage model is presented. That is, separate the temperature into some stages in accordance with technology and establish the BP model for every stage. BP net realizes the static process identification. For dynamic process, recurrent neural network is reasonable. In this paper, the recurrent neural network consists of BP net added TDL blocks. Put the temperature data through a TDL block and make it as a new input data of neural network. In addition, the models established using all kinds of methods are put into control system to simulate the practice spot situation.

Fuzzy T-S system identification whose parameters are optimized with GA is researched. In this part, generalized T-S fuzzy model is introduced, so it is feasible for finding the best parameters. With this method, the generalized T-S fuzzy model of omethoate synthesis batch process plant is built.

Otherwise, this paper talks about the transformation from monomethyl amine flux

in simulation to open status of valve in practice. In order to realize it, curve-fitting technology is adopted. In a result, the flux curve and the experience equation are obtained.

**Keywords:** batch process; system identification; intelligent control; neural network; fuzzy T-S model; genetic algorithm

## 1 绪论

### 1.1 引言

系统辨识是控制理论研究的一个重要的分支，它是设计控制系统的基础。随着控制过程复杂性的提高，控制理论应用日益广泛，控制理论的实际应用不能脱离被控对象的数学模型。然而在多数情况下，被控对象的数学模型是不知道的，或者在正常运行期间，模型的参数可能发生变化。因此利用控制理论去解决实际问题时，首先需要建立被控对象的数学模型，这是控制理论能否成功地用于实际的关键之一<sup>[1]</sup>。

控制对象的高度复杂性是过程控制的特点，它表现为非线性，时滞性普遍存在；内部关联错综，机理分析困难；先验知识少而且表达非常规。方便而且有效的建立过程对象的数学模型是控制系统设计的一个重要基础，有了适当的模型，才便于开展有关的分析、设计、预报和计算仿真等研究工作。

目前，尽管对线性系统已经有了较为成熟的辨识方法，如最小二乘法，但是对非线性系统的辨识尚处于探索阶段，没有类似的手段对其进行逼近。已有的一些非线性系统的辨识方法，往往需要有关被辨识系统结构形式等各种先验知识和假设，因此它们基本上是针对某些特殊非线性系统而言的。过去几十年中，对线性、非时变和具有不确定参数的系统进行辨识的研究已取得了很大的进展。但是辨识模型的结构选取和保证整个系统全局稳定的自适应调节参数规律的构成等，都是建立在线性系统理论基础上的，对于非线性系统的辨识问题却一直难以找到相应的数学方法。智能系统理论中包含了许多系统辨识的新方法，这些方法在非线性领域有很大的发展和应用潜力，是现阶段比较有效的方法<sup>[2]</sup>。

### 1.2 课题背景

在过去的十几年中，化学加工工业发生了巨大的变化。其主要趋势表现为从商品化学品生产转向专用的功能化学品生产，从大规模过程转向小规模具有弹性的过程；从连续加工转向间歇加工等等。为了适应这种变化，间歇过程又重新受到了人们的普遍重视。这主要是因为间歇过程具有灵活多变的特性，即它可以用同一套多用途、多功能的设备生产多种类型的产品。这种转变的内在动力反映



在化学过程工业方面，则是精细化工、生物化工等高度技术密集和知识密集的新兴产业的蓬勃发展。

间歇生产过程，又称作批量生产过程。由于生产设备的物理结构，或者其他经济上的因素，致使生产过程由一个或多个按一定顺序执行的操作步骤（或操作阶段）组成，那么从性质上说，这个生产过程就认为是一个间歇生产过程。完成这个操作步骤顺序就生产出一定量的最终产品。如果需要生产更多的产品，则必须重复执行这个操作步骤顺序。简而言之，间歇过程指的是将有限量的物料按规定的加工顺序在一个或多个设备中加工以获得有限量的产品的加工过程<sup>[8,45-46]</sup>。

间歇生产过程最大的优点是柔性好。它的设备要求能生产多种产品，进行多种生产操作。设备具有操作柔性，使工厂生产新产品不用添置新设备，生产不同规格的产品不需要修改设备，这样工厂只需要加少量投资就能适应市场需求的变化。间歇生产过程不仅具有明显的经济规模效益，而且其重要性在于这类生产过程在工业生产中占有很大的比重。

目前，间歇操作广泛应用于食品、聚合物、药品、染料和涂料等产品的生产。随着间歇过程普遍的应用与发展，随之也出现了一些问题，主要有：（1）由于间歇过程通常具有非线性、时滞和易燃易爆的特点，因此用常规的建模方法建立模型比较困难；（2）间歇过程控制系统很难进行结构调整，不同厂家的控制系统互不兼容<sup>[3]</sup>。

化工间歇过程的特点决定了它必然属于难控制的系统，针对这类系统，有很多学者和专家提出了各种各样新的智能控制方法<sup>[47-55]</sup>，而且其中的一些方法也已经得到了广泛的应用，解决了一些复杂间歇过程的控制问题。

还有一些研究人员致力于化工间歇过程的模型辨识研究，其一是侧重于宏观上间歇过程调度与配比的建模，这一点不属于本课题所研究的范畴；其二是侧重于具体的化工间歇过程对象。如发酵生产过程<sup>[2,57,58,63-67]</sup>。工业发酵过程是与微生物的生长和代谢活动紧密相联系的生物反应过程，反应机理复杂，影响因素较多。目前，基于酶动力学、发酵动力学和生化反应工程的研究，建立机理模型的理论建模方法尚不成熟，难以实际应用。文献[2]中针对间歇生产过程动态特性中的非线性、时变性和不确定性，提出了梯形神经网络群模型。它是将间歇过程中的状态在时间上进行分段，每一段用一个基本的神经网络表出，这样只要在不同的时间段调用不同的基本神经网络就可逐步得到整个生产过程的行为。有了这样的模型，使得不必再去针对不同的反应类型，作复杂的分析和论证，只要具有每段的操作数据和直观的系统阶次估计，就可以通过它的自学习能力，适应不同的工况或反应机制。所以，这是一种通用模型结构，实际生产过程表明，大部分间歇过程以2阶近似已经足够。

文献[57]中运用模糊集和模式识别理论，提出了一种具有自学习功能的在线识

别间歇过程特性的新方法，将用该方法所建立的螺旋霉素间歇生产过程菌体生长期的在线模糊识别模型，在杭州华东制药厂加以实际应用，取得了良好效果。

文献[58]中利用微分神经网络来辨识发酵过程。这个微分神经网络在时域中体现出的是动态网络。这种方法不需要动态方程和参数的先验信息。利用实际数据，经过训练该网络，仿真表明该网络对于发酵过程这类对象具有很好的辨识能力，而且网络结构简单，只用了六个神经元。

在文献[63]中，提出从发酵机理出发选择影响过程的关键因素，利用发酵批报数据所提供的信息，以改进后的二次多项式模型集为基础，采用逐步回归和统计推断方法进行模型选择和参数估计，建立描述产物合成率随关键变量变化的回归方程的通用建模方法。通过在线递推建立部分数据的回归模型，实现对发酵过程的预测和优化，对于典型的氨基酸发酵过程批报数据的仿真研究证明此方法确实可以达到较高的预测精度。

文献[64]中提到，发酵过程的控制通常需要获取状态变量，如菌体、基质和产物的浓度，但由于生物传感器的缺乏，这些变量很难在线测量。基于卡尔曼滤波器的状态估计技术曾经应用于发酵过程，而与之相比的研究成果已经证明，神经网络技术在发酵过程的应用比卡尔曼滤波器估计技术估计精度高、预测准确<sup>[66]</sup>。神经网络刚开始应用于发酵过程的建模时，最常用的建模方式是“黑箱”式标准神经网络模式<sup>[67]</sup>，后来人们提出了将神经网络与过程的部分机理模型如物料、能量平衡方程相结合，组成混合神经网络模型，这种混合神经网络模型明显比标准神经网络模型建模精度高、泛化能力强<sup>[65,66]</sup>。文献[64]当中将过程的内部结构融入神经网络建模中，按这种思路建立的模型有一个共同的特点是模型中各个部分完成不同的任务。一种方式是将过程内部已知的线性模型与非线性神经网络模型相结合，发挥神经网络的非线性特性。非线性网络对过程非线性特性建模，这样使得整个模型能够描述过程的复杂动力特性，这种建模方法统称为混合神经网络建模方法。目前在发酵过程中应用的混合神经网络模型有如下几种结构：串联结构、并联结构及串并联结构。通过仿真试验对比，混合神经网络模型在泛化能力上明显优于标准的“黑箱”式神经网络模型，而在混合神经网络模型中，串联结构和串并联结构及文献[64]提出的新型结构的混合神经网络模型由于有物料平衡方程式的约束条件，其泛化能力比并联结构模型强。但串联结构和串并联结构混合模型训练方法复杂。混合神经网络模型不仅保持了串联和串并联结构模型的优点，而且简化了训练方式。

间歇聚丙烯过程具有非线性、时变和易燃易爆的特点，由机理分析或实验方法来得到过程的模型十分困难。影响丙烯聚合反应的因素较多，在一段时间内，可以认为每批所用的物料情况基本一致，而主要考虑操作条件的影响，如反应温度、时间均对聚合反应有较大的影响。对于间歇聚丙烯过程，聚合釜内压力十分接近丙



烯的饱和蒸气压, 压力的高低完全与温度对应, 因此只把釜压作为输出参数。文献[56]利用前馈神经网络建立间歇聚丙烯过程反应阶段的压力数学模型。测试结果表明, 经过学习的神经网络有足够的精度, 用它充当对象的数学模型是可以的。值得注意的是, 并非训练的次数越多, 越能得到正确的输入输出映射关系。在文献[56]的实验中发现, 训练的次数如果再增加, 测试的误差就会变大。这是由于所搜集的数据都是包含噪声的, 训练的次数过多, 网络将包含噪声的数据都准确记忆下来, 在极端情况下, 训练后的网络可以实现相当于查表的功能, 但是对于未经训练的输入数据却不能给出合适的输出, 即并不具备很好的泛化功能。泛化性能好的网络才能恰当地表示实际过程的输入输出关系。

蒸馏过程也是一类典型的间歇生产过程。常压蒸馏作为石油加工的关键环节, 其侧线馏出产品的质量直接影响全厂的经济效益, 在实际生产过程中各段油品质量的检测是通过抽取成品油样本, 进行离线试验分析, 这样很难适应在线质量控制和在线指导生产过程操作的需要, 因此实现油品质量的在线估计具有重要的意义。在文献[59]中, 利用三层神经网络, 结合遗传 BP 算法训练出一个神经网络模型, 该模型可在工程所要求的误差范围内对常一线干点及常三线柴油倾点进行有效估计, 由于神经网络可以根据实际运行对象对新的数据样本进行自适应学习, 从而为常压塔的监控和操作指导提供了可以实时实现的产品质量估计模型, 进而为原油常压蒸馏的最优化控制奠定了基础。而在文献[60]中, 主要比较了世界几个公司已经辨识几种间歇蒸馏过程模型的优缺点。

在对这些间歇生产过程对象模型的辨识过程中, 采用了基于智能系统理论的辨识方法, 辨识的效果也比较理想。但是也存在一些不足, 比如有些神经网络的泛化能力较差; 混合神经网络需要与过程的部分机理模型结合, 如果在过程的机理模型知之甚少的情況下, 这种模型也是不容易实现的; 利用回归模型需要较多的数学知识, 计算起来较为繁琐等等。

在众多的文献中所列举的较多的间歇生产过程大都集中在发酵、蒸馏和聚丙烯等。尽管有些文献也提出一种通用的模型, 但是在其实现模型的过程中使用的还是发酵这类间歇过程。此外建模的方法多使用的是神经网络辨识方法, 利用模糊方法来完成过程的建模则比较少, 尤其是利用 T-S 模型来建模的方法更少。

对于本文所研究的对象——氧乐果合成间歇生产过程, 在参阅了大量的国内外文献后, 发现还未给出一种合适的模型。该对象具有大时滞、时变和不确定性的特点, 无法用合适的数学表达式来描述。为了能够有效的对该对象进行控制, 预测对象输出, 提高生产效率, 有必要对这类间歇生产过程对象进行模型的建立。

### 1.3 本文的主要工作

本文在吸收了文献[2, 56-67]所提供方法的基础上, 重点研究了基于神经网络的系统辨识方法, 并简要讨论了基于遗传算法的模糊系统辨识。

本论文的主要内容可分为以下几个方面:

1. 氧化乐果间歇过程工艺的研究;
2. 系统辨识方法的研究;
3. 现场采集数据的预处理;
4. 基于神经网络的系统辨识方法的研究以及仿真实现;
5. 基于遗传算法的 T-S 模糊模型的模糊辨识方法的研究以及仿真实现;
6. 结论与展望。



## 2 氧乐果合成间歇过程生产工艺分析

常见的间歇生产过程，各个设备单元，如反应器、混合器、储罐等都是用管道固定连接的，这类间歇生产过程称作有管间歇生产过程。此外，根据间歇生产过程生产的产品，可以分为单产品、多产品和多用途三类。一个单产品间歇生产过程，其每批产品的生产都是按照配方规定的同一操作顺序、相同的原料配比和工艺参数进行。氧乐果合成间歇过程就是属于这种有管单产品间歇生产过程。

### 2.1 工艺介绍

氧乐果生产中的合成（胺解）工序是整个氧乐果生产的关键，直接影响着氧乐果粗原油的收率、含量等指标。氧乐果的合成反应是一个剧烈的放热反应。反应物是一甲胺和精酯，反应产物是氧乐果粗原油。合成反应工艺流程如图 2-1 所示。

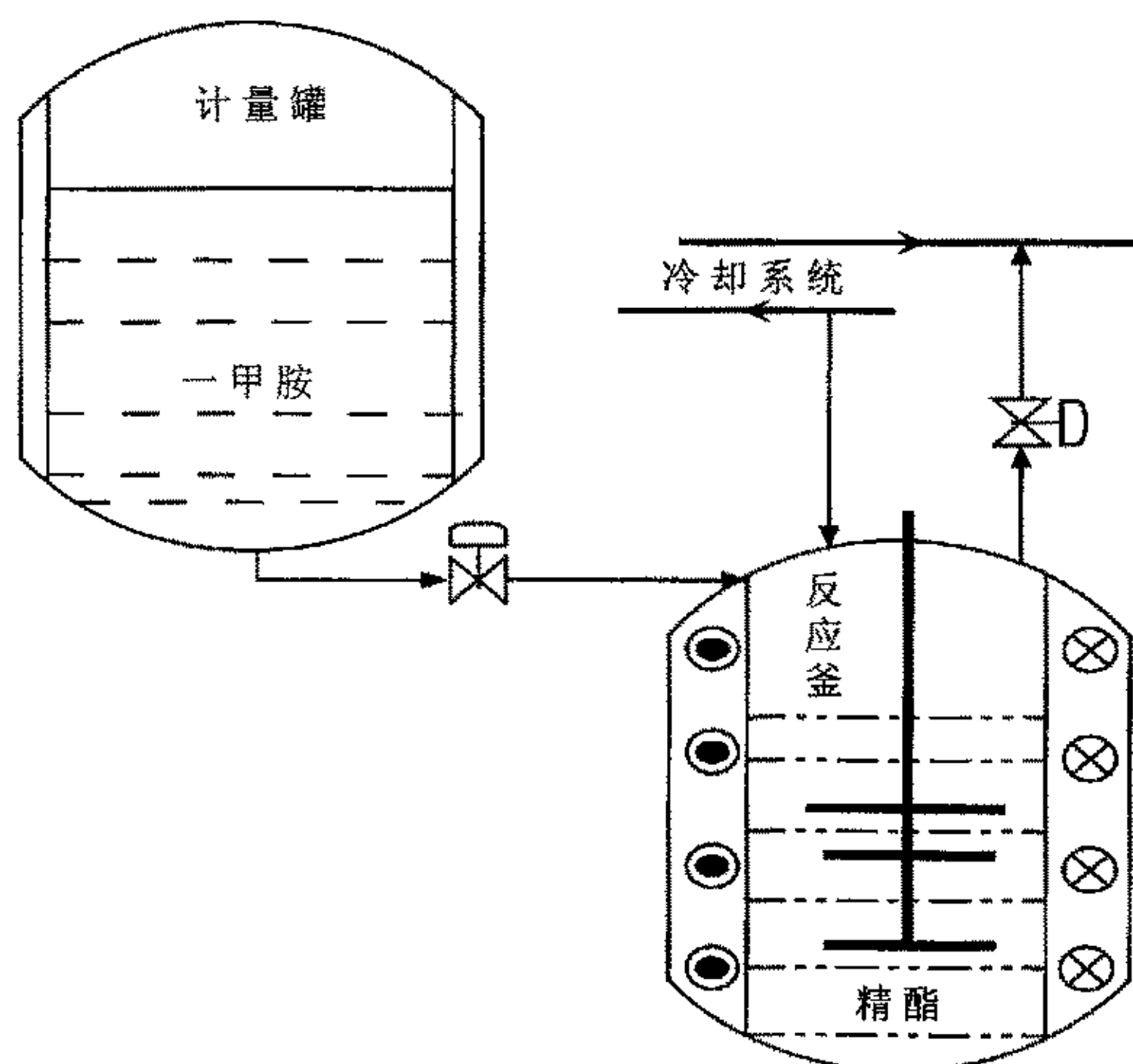


图 2.1 氧乐果合成反应工艺流程图

Fig 2.1 Flow chart of omethoate synthesis

计量罐中的一甲胺滴注到反应釜内，经搅拌后均匀地喷洒到反应釜中，与其中预先备好的精酯混合，发生剧烈的放热反应，生成氧乐果粗原油。反应釜外壁缠有冷却盐水管，反应放出的热量经过热交换后被冷却盐水带走。有的计量罐内也装有冷却盘管。开始投一甲胺以前，要把精酯冷却到 $-20^{\circ}\text{C}$ 以下。

氧乐果的合成反应过程对温度和一甲胺投料速度均有较高的要求。将精酯冷却到 $-20^{\circ}\text{C}$ 以下后，开始投一甲胺进行合成反应。当反应釜温度上升到 $-14\sim-12^{\circ}\text{C}$ 后，在温度不超过 $-12^{\circ}\text{C}$ 前提下，要求尽可能快地完成一甲胺投料。

反应初始阶段，即使反应温度很低，温升较慢，甚至是负增长，一甲胺的投料速度也不宜过快。此时，由于反应釜内精酯含量较多，合成反应比较剧烈，容易造成局部温度过高，降低合成质量。

一甲胺累积已投料达到约 $60\sim70\text{kg}$ 时，合成反应的总体放热效果开始超过系统的冷却效果，系统的热平衡被打破，反应温度开始快速增长，此时要采用较低的投料速度，抑制合成反应的放热量和放热速度，从而使反应温度的上升趋势放缓。

反应釜温度上升到接近 $-12^{\circ}\text{C}$ 后，保持适当的投料速度，系统处于热平衡状态，反应温度在小范围内变化。

一甲胺的剩余量少于 $50\text{kg}$ 后，由于反应釜内的精酯大部分已经被消耗掉，此时可以以较快的速度进行投料，而反应釜温度不会出现快速攀升。

## 2.2 氧乐果合成间歇过程建模影响因素分析

根据上面所介绍的工艺流程，可以看出整个过程当中温度控制的重要性。然而该过程存在着比较大的时变性和非线性，同时也存在着比较大的滞后，除此之外，该过程还受到很多干扰因素的影响。这些都对温度的控制不利。如果能够利用一个与氧化乐果间歇过程接近的预估模型，那么将有利于改善控制效果，在一定程度上提高生产效益。

### 2.2.1 系统可测参数的影响

影响温度的因素有很多，经过认真分析，主要有以下几个方面：

#### 1. 一甲胺流量

一甲胺流量是影响反应釜温度的最主要因素，流量大，则合成反应放热量多，温升快；流量小，则反应放热量少，温升慢。其中一甲胺流量主要受调节阀的影响，调节阀开度增大，则流量增加；反之，则流量减少。一甲胺流量还受计量罐中一甲胺液位的影响，阀位不变时，液位越高，则流量越大；反之，则流量越小。

## 2. 冷却盐水流量

冷却盐水流量越大，则冷却效果越好，温升越慢；反之，则温升越快。由于农药厂共有两个车间，每个车间有三套反应釜，但是却公用一套冷却系统，由一条冷却盐水母管连接在一起，各个反应釜的冷却效果随距离冷却盐水母管入口的远近依次降低。因此，在一个车间里冷却盐水流量还受另外两个反应釜冷却盐水阀门开关状态的影响，其余两个反应釜冷却盐水阀门全关时，第一个反应釜冷却盐水流量最大；其余两个反应釜冷却盐水阀门只有一个关闭时，第一个反应釜冷却盐水流量次大；其余两个反应釜冷却盐水阀门全开时，第一个反应釜冷却盐水流量最小。此外两个车间反应釜冷却盐水阀门的开关状态还相互影响和制约着反应釜冷却盐水流量。

## 3. 冷却盐水温度

冷却盐水温度越低，则冷却效果越好；反之越差。

## 4. 精酯量

精酯量越多，则所需的一甲胺量也越多，反应放热量就越多；反之就越少。

## 5. 一甲胺累积投料量

随着一甲胺累积投料量的增加，合成反应的放热量也在增加。投料量大约在 70kg 时，反应最为剧烈，温升较快；投料量剩余 50kg 时，由于精酯已大部分被消耗掉，即使以较快的速度投一甲胺，反应温度也不会有较大上升。

## 6. 反应进行时间

反应刚开始时，由于精酯较多，反应较为剧烈，局部温升较大。随着反应时间的延长，精酯逐步被消耗掉，与反应刚开始时相比，在一甲胺流量相等的情况下，反应放热量减少，温升变慢。

## 7. 一甲胺温度

反应温度同时受一甲胺温度的影响，一甲胺温度越低，则反应温度上升越慢；反之，则越快。但一甲胺温度比较稳定，波动不大。

## 8. 反应起始温度

反应刚开始时，精酯被冷却到的温度称为反应起始温度。起始温度越低，则温升越慢；反之，则越快。

## 9. 环境温度

环境温度影响冷却效果，环境温度越低，则冷却效果越好，温升越慢；反之，则温升越快。

## 10. 其他因素

包括精酯含量、一甲胺含量、冷却系统热交换的好坏等等。

以上这些因素同时作用，共同影响着反应温度，如图 2.2 所示。

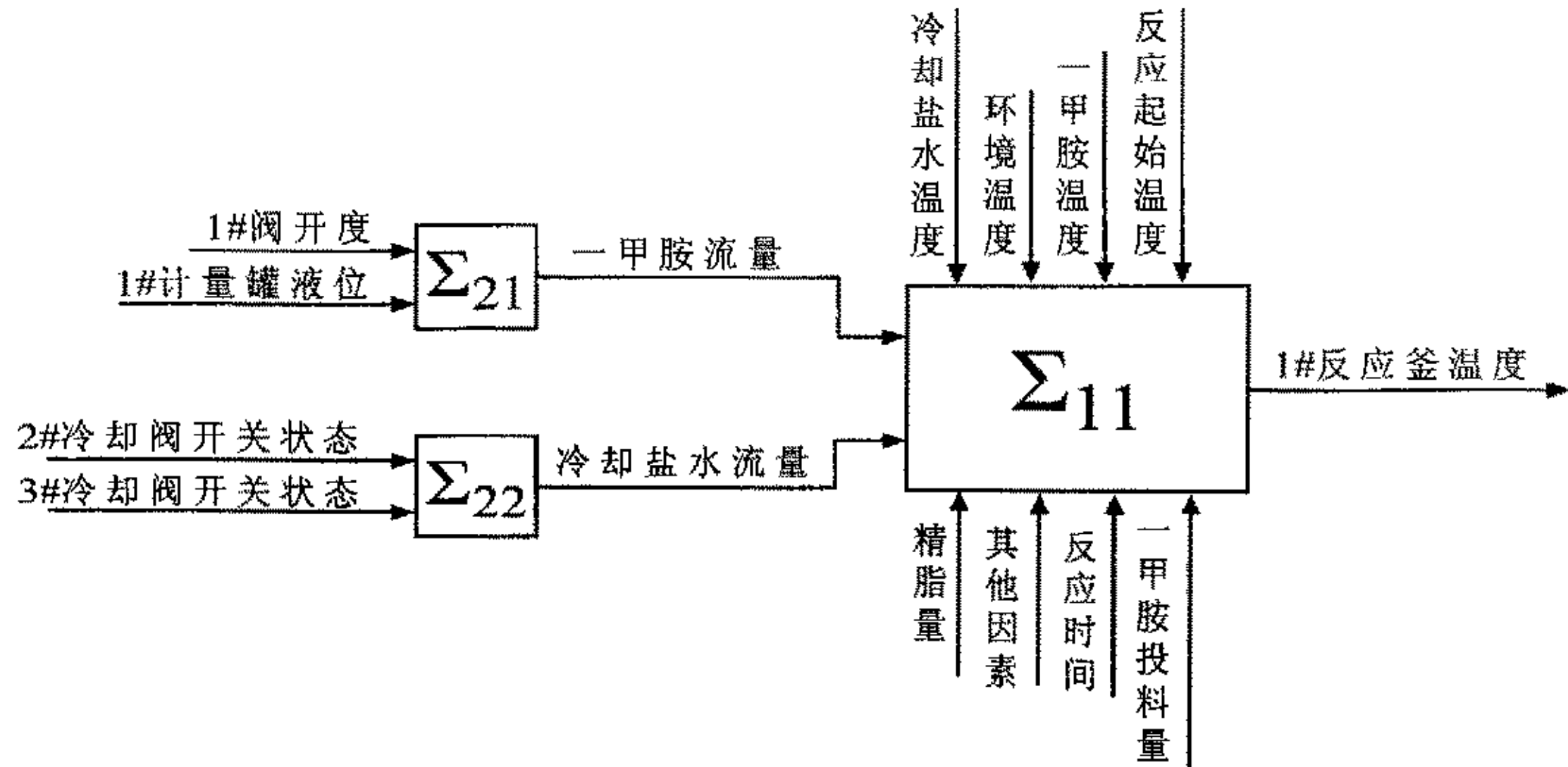


图 2.2 温度影响因素示意图

Fig 2.2 Schematic diagram of factors that effect the temperature

### 2.2.2 系统不可测参数的影响

氧化乐果合成反应的温度对象除了受到各种因素的影响之外，对象的参数也是时变的，变化最大的就是纯滞后时间  $\tau$ 。从反应开始到最后结束， $\tau$  从几秒钟变化到几分钟。 $\tau$  与反应进行时间  $t$ ，精酯含量  $M_{JZ}$ ，一甲胺含量  $M_{LG}$ ，反应釜初始温度  $T_0$  和其它因素  $O$  有关，可以用下式表示：

$$\tau = f(t, M_{JZ}, M_{LG}, T_0, O) \quad (2-1)$$

由此可以看出， $\tau$  是一个多因素决定的、在反应过程中不断变化的、而且变化范围比较大的参数。也就是说，对同一釜的合成反应， $\tau$  在不断的变化；对同一个反应釜的不同批次合成反应， $\tau$  的变化曲线不完全相同；不同反应釜的合成反应， $\tau$  的变化曲线也存在差异。其变化曲线可以参看文献[6]。

另外一个变化比较大的参数是增益  $K$ 。在反应初期，增益较大，一甲胺累积投料量约为 70kg 左右时， $K$  达到最大，然后慢慢减小，到投料结束减至最小。 $K$  的大小与一甲胺含量  $M_{LG}$ ，总投料量  $M_{TOTAL}$  有关，可以用下式表示：

$$K = f(M_{LG}, M_{TOTAL}) \quad (2-2)$$

$K$  的变化曲线参看文献[6]。增益  $K$  的特性使得对象具有高度非线性。

综合上述分析，可以看出该温度过程具有多变量、非线性、时变、分布复杂等特点，利用传统的经典系统辨识方法很难预测出较为理想的温度值，因此需要适合于这类对象的新辨识方法，也就是前面的提到的基于智能系统理论的系统辨识方法。



### 3 间歇过程对象辨识的理论基础

L. A. Zadeh 曾给辨识下过这样的定义：“辨识就是在输入和输出数据的基础上，从一组给定的模型中，确定一个与所测系统等价的模型。”这个定义明确的表示了辨识的三大要素：①输入输出数据；②模型类；③等价准则。其中数据是辨识的基础；准则是辨识的优化目标；模型类是寻找模型的范围。当然按照这个定义寻找一个与实际过程完全等价的模型无疑是非常困难的。从实用的观点出发，对模型的要求并非如此苛刻，为此对辨识又有一些比较实用的定义。比如 L. Ljung 给辨识下的定义：“辨识有三个要素——数据，模型类和准则。辨识就是按照一个准则在一组模型类中选择一个与数据拟合得最好的模型。”另外，由于观测到的数据一般都含有噪声，因此辨识建模实际上是一种试验统计的方法，它所获得的模型只不过是与实际过程外特性等价的一种近似描述。

#### 3.1 辨识算法的基本原理

辨识的实质就是从一组模型类中选择一个模型，按照某种准则，使之能最好地拟合所关心的实际过程的动态特性。下图给出了常见的辨识结构<sup>[1]</sup>。

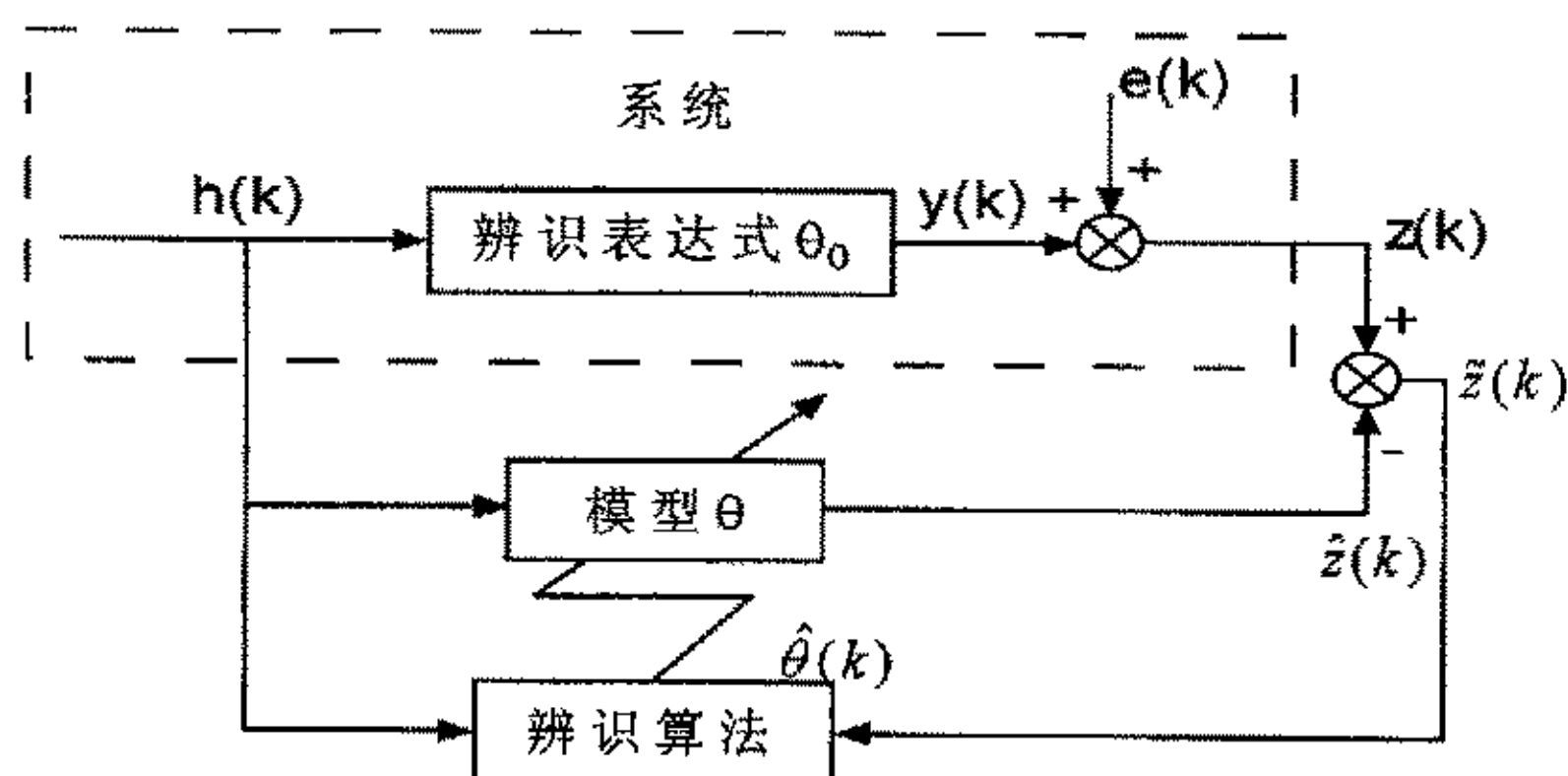


图 3.1 辨识原理

Fig 3.1 Identification theory

为了得到模型参数  $\theta$  和估计值  $\hat{\theta}$ ，通常采用逐步逼近的办法。在  $k$  时刻，根据前一时刻的估计参数计算出该时刻的输出，即过程输出的预报值

$$\hat{z}(k) = h^T(k)\hat{\theta}(k-1) \quad (3-1)$$

同时计算出预报误差或称新息

$$\tilde{z}(k) = z(k) - \hat{z}(k) \quad (3-2)$$

其中过程输出量

$$z(k) = h^T(k)\theta_0 + e(k) \quad (3-3)$$

及辨识表达式的输入量  $h(k)$  都是可以测量的。然后将新息  $\tilde{z}(k)$  反馈到辨识算法中去，在某种准则条件下，计算出  $k$  时刻的模型参数估计值  $\hat{\theta}(k)$ ，并据以更新模型参数。这样不断迭代下去，直至对应的准则函数取最小值。这时模型的输出  $\hat{z}(k)$  也已在该准则意义下最好地逼近过程的输出值  $z(k)$ ，于是便获得了所需要的模型。

## 3.2 线性模型的辨识方法

迄今为止，已经有许多不同的辨识方法。这些辨识方法就其所涉及的模型形式来说，可以分成两类。一类是非参数模型辨识方法，另一类是参数模型辨识方法。

非参数模型辨识方法亦称为经典的辨识方法。它在假定过程是线性的前提下，不必事先确定模型的具体结构，因而这类方法可适用于任意复杂的过程。参数模型辨识方法亦称为现代的辨识方法。它必须假定一种模型结构，通过极小化模型与过程之间的误差准则函数来确定模型的参数。如果模型的结构无法事先确定，则必须利用结构辨识的方法先确定模型的结构参数，再进一步确定模型参数<sup>[1]</sup>。

### 3.2.1 经典的辨识方法

线性过程的动态特性通常用传递函数，频率响应，脉冲响应和阶跃响应来表示。后面三种为非参数模型。在经典的辨识方法中，主要有：阶跃响应法、脉冲响应法、频率响应法、相关分析法和谱分析法。利用这些方法可以将非参数模型转化成传递函数。

其中阶跃响应法、脉冲响应法和频率响应法都是在过程中施加试验信号，记录下相应的响应曲线，通过这些响应曲线可以求解过程的传递函数。但是这三种方法原则上只有在高信噪比的情形下才是有效的，因此这便成了上述辨识方法的致命缺点。然而在工程实际中，所获得的数据总是含有噪声的，相关分析法和谱分析法是这类辨识问题的有效方法。

相关分析法就是根据输出数据和输入数据之间的互相关函数来估计过程的频率响应，此时的输出数据中是含有噪声的。换句话说，就是利用带有噪声的输出数据来估计过程的频率响应。

谱分析法的特点是不需要对过程施加试验信号，只需要利用正常操作下的输

入输出数据就可以辨识过程的动态特性，因此使用比较方便，而且有较强的抑制噪声的能力。利用谱分析法辨识频率响应的基本公式为

$$S_{xy}(j\omega) = G(j\omega)S_x(\omega) \quad (3-4)$$

关键在于估计输入数据的自谱密度和输入输出数据的互谱密度。

### 3.2.2 现代的辨识方法

这类辨识方法就其不同的基本原理来说可分成三种不同的类型。

第一种称作最小二乘法。它利用最小二乘原理，通过极小化广义误差的平方和函数来确定模型的参数。在这类方法中，最小二乘法是最基本、最典型，也是应用最广的一种方法。线性系统的模型可通过一定的数学变换成最小二乘格式

$$z(k) = h^T(k)\theta + n(k) \quad (3-5)$$

其中， $z(k)$  是过程的输出量， $h(k)$  是可观测的数据向量， $n(k)$  是均值为零的随机噪声。利用数据序列  $\{z(k)\}$  和  $\{h(k)\}$ ，极小化下列准则函数

$$J(\theta) = \sum_{k=1}^L [z(k) - h^T(k)\theta]^2 \quad (3-6)$$

使  $J(\theta) = \min$  的  $\theta$  估计值记作  $\hat{\theta}$ ，称作参数  $\theta$  的最小二乘估计值。上述概念表明，未知模型参数  $\theta$  最可能的值是在实际观测值与计算值之累次误差的平方和达到最小值处，所得到的这种模型输出能最好地接近实际过程的输出。

最小二乘法具有两方面的缺陷：

- ① 当模型噪声是有色噪声时，最小二乘参数估计不是无偏、一致估计；
- ② 随着数据的增长，最小二乘法将出现所谓的“数据饱和”现象。这是由于增益矩阵  $K(k)$  随着  $k$  的增加将逐渐趋近于零，以致于递推算法慢慢失去修正能力之故。

针对这两个问题，在文献[1]提出了几种相应的改进最小二乘辨识方法。

第二种称作梯度校正法。它利用最速下降法原理，沿着误差准则函数关于模型参数的负梯度方向，逐步修改模型的参数估计值，直至误差准则函数达到最小值。其主要特点是计算简单，可用于在线实时辨识；缺点是在极小点附近，收敛速度很慢，其原因是它只利用了一阶导数的信息。

第三种为极大似然法和预报误差方法。极大似然法需要构造一个以数据和未知参数为自变量的似然函数，根据极大似然原理，通过极大化似然函数来定模型参数的估计值。这意味着模型输出的概率分布将最大可能地逼近实际过程输出的概率分布。为此，极大似然法通常要求具有能够写出输出量的条件概率密度函数的先验知识。

对另外一种称作预报误差方法来说，它需要事先确定一个预报误差准则函数，

并利用预报误差的信息来确定模型的参数。从某种意义上说，预报误差方法和极大似然法是等价的，或者说是极大似然法的一种推广。原因在于极大似然法要求数据的概率分布是已知的，通常假设为高斯分布。然而实际问题不一定都能满足这种假设，但是预报误差方法能够适应这种问题，它不要求关于数据概率分布的先验知识。因此预报误差方法是解决更加一般问题的一种辨识方法，是极大似然法的一种推广。当数据的概率分布服从高斯分布时，预报误差方法等价于极大似然法。这两种方法的参数估计量具有良好的渐近性质，但是计算量比较大。

### 3.3 智能系统理论中的辨识方法

上面所介绍的一些辨识方法，其研究对象的模型主要是集中参数、离散、定常、线性动态随机模型，而本课题所研究的间歇过程对象具有高度的非线性，时变性和时滞性，因此想利用某一类数学表达式模型来描述这类对象，显然是比较困难的。这就需要采用合适的辨识方法——基于智能系统理论的辨识方法来解决问题。

#### 3.3.1 基于神经网络的系统辨识与建模

神经网络在系统建模、辨识与控制中的应用，大致以 1985 年 Rumelhart 的突破性研究为界。在极短的时间内，神经网络就以其独特的非传统表达方式和固有的学习能力，引起了控制界的普遍重视，并取得了一系列重要结果。神经网络应用于控制领域的主要吸引力在于：

- ① 多层前馈神经网络能够以任意精度逼近任意非线性映射，给复杂系统的建模与辨识带来了一种新的、非传统的表达工具；
- ② 固有的学习能力降低了不确定性，增加了适应环境变化的泛化能力；
- ③ 并行计算特点，使其有潜力快速实现大量复杂的控制算法；
- ④ 分布式信息存储与处理结构，从而具有独特的容错性。

对于非线性系统，基于非线性自回归滑动平均模型（NARMA），难于找到一个恰当的参数估计方法，传统的非线性控制系统辨识方法，在理论研究和实际应用中都存在极大的困难。

相比之下，神经网络在这方面显示出明显的优越性。与传统的基于算法的辨识方法相比较，基于神经网络的辨识具有以下几个特点：

- ① 不要求建立实际系统的辨识格式，即可省去系统结构建模这一步。因为神经网络本质已作为一种辨识模型，其可调参数反映在网络的内部权值上。
- ② 可以对本质非线性系统进行辨识。辨识是非算法式的，由神经网络本身体现。辨识的结果为网络外特性拟合系统的输入输出数据中的系统特性。



③ 辨识的收敛速度不依赖于待辨识系统的维数，只与神经网络本身及其所采用的学习算法有关，而传统的辨识算法随模型参数维数的增大变得很复杂。

④ 神经网络具有大量连接，其连接权值在辨识中对应于模型参数，通过调节这些参数可使网络输出逼近系统输出。

⑤ 神经网络作为实际系统的辨识模型，实际上也是系统的一个物理实现，可以用于在线控制。

非线性系统的模型可以分为两类：输入输出模型和状态空间模型；神经网络模型也分为两类：静态神经网络和动态神经网络。在神经网络的建模方法中，静态神经网络被用作对象的输入输出模型；而动态神经网络既可用于对象的输入输出模型，又可用于对象的状态空间模型。

在系统辨识中最常用的静态神经网络有多层前馈神经网络（MPLs）。多层前馈神经网络具有逼近任意非线性映射的能力；在文献[5]中，已经证明了当满足三个条件时，这种网络必可以任意精度逼近任意非线性连续或分段连续函数。利用静态多层前馈神经网络建立系统的输入输出模型，本质上是基于网络的逼近能力，通过学习获知系统的未知非线性函数。这种网络具有三要素，即多层次结构、S型神经元和反向传播（BP）算法。

在系统辨识中最常用的动态网络是回归网络。回归网络近年来受到了人们的普遍关注，它的一个最显著特征就是它的输出端信号通过延时环节或者一阶惯性环节的反馈机构连接到输入端。回归网络本质上是动态系统，其代表是 Hopfield 网络和 Elman 网络。Hopfield 网络是单层的，它代表了一类非线性系统；Elman 网络是在多层前馈网络的基础上，加入了连接权值不能修正的“结构”单元，这种网络可以映射任意的非线性动态系统。

动态网络在建立动态系统模型方面有其长处，同时也应指出，它的学习算法采用的是动态反传算法，其运算比静态 BP 算法大得多。因此，在实际使用采用神经网络辨识时，应首先考虑静态网络，其动态关系可由输入输出经过多分头延时（TDL）环节获得。

### 3.3.2 基于模糊理论的系统辨识与建模

系统越复杂，对该系统进行精确的描述就越困难。但不管如何复杂，其经验的语义描述还是可以获得的，而且表现出一定的稳定性。因此，用模糊集合理论，从系统的输入输出值来建立和辨识系统的模糊模型，是处理复杂系统的一种有效途径。尽管这种模糊模型与通常的精确模型相比显得粗糙，但是它也能对复杂系统的基本特性给出严格的定量描述。从某种意义上说，这种模糊模型也是明确的，它并不是像人想象的那样是随意的。

模糊逻辑系统的主要应用在于它能够作为非线性系统的模型，文献[11]中给出了由高斯型隶属度函数构成的模糊逻辑系统（简称高斯型模糊逻辑系统），可以在任意精度上一致逼近任何定义在一个致密集上的非线性函数。

模糊模型辨识主要有基于模糊关系模型的系统辨识和基于 T-S 模型的模糊系统辨识等。

一个模糊关系模型可以表示为： $M(A, Y, U, F)$ 。其中  $A$  表示模糊算法； $Y$  表示过程的有限离散输出空间； $U$  表示过程的有限离散输入空间； $F$  表示过程的有限离散输入输出空间中所定义的所有基本模糊子集的集合。

对于 SISO 系统，设系统的输入空间  $U$  和输出空间  $Y$  分别由  $M$  个点  $u_1, u_2, \dots, u_M$  和  $N$  个点  $y_1, y_2, \dots, y_N$  构成； $B_1, B_2, \dots, B_m$  和  $C_1, C_2, \dots, C_n$  分别是  $U$  和  $Y$  中的模糊集合。所谓的模糊模型是指描述系统特性的一组模糊条件语句，即

$$\begin{aligned} \text{if } u(t-k) = A \text{ or } B \text{ and } y(t-l) = C \text{ or } D \text{ then } y(t) = E \\ \dots \dots \dots \end{aligned} \quad (3-7)$$

其中  $A$  和  $B$  为  $U$  中模糊集合， $C$ 、 $D$  和  $E$  为  $Y$  中的模糊集合，称(3-7)中每一条模糊条件语句为一条规则，而将(3-7)中一组描述系统特性的模糊条件语句称为模糊算法。每一条规则可以根据模糊集合运算规则写成如下形式

$$E = u(t-k) \circ [(A + B) \times E] \bullet y(t-l) \circ [(C + D) \times E] \quad (3-8)$$

根据每一条规则以及已知的  $u(t-k)$  和  $y(t-l)$ ，可由(3-8)计算出一个  $E$ 。若系统的特性由  $p_1$  条规则描述，则模糊变量  $y(t)$  的值可以写为

$$y(t) = E_1 + E_2 + \dots + E_{p_1} \quad (3-9)$$

若(3-8)中的系统输入和系统的输出分别用它们的隶属度函数表示，那么其形式为

$$E = \min\{\max[\mu_A(i), \mu_B(i)]; \max[\mu_C(i), \mu_D(i)]; \mu_E\} \quad (3-10)$$

由(3-9)计算得到的  $y(t)$  是一个模糊集合，从  $y(t)$  中选择确切的值一般有两种方法：面积中心法和最大值平分法。

上面介绍的是模糊关系模型，在 1985 年日本的高木(Takagi)和杉野(Sugeno)在论文中针对非线性动态系统的实际，提出了一种模糊动态线性模型，将一般的 MISO 动态系统由  $n$  个模糊规则组成的集合来表示，即 T-S 模型。

这种模糊模型的具体形式为

$$\begin{aligned} R^l : \text{if } x_1 \text{ is } A_1^l, x_2 \text{ is } A_2^l, \dots, x_n \text{ is } A_n^l, \text{ then } y^l = P_0^l + P_1^l x_1 + P_2^l x_2 + \dots + P_n^l x_n \\ l = 1, 2, \dots, M \end{aligned} \quad (3-11)$$

其中  $R^l$  表示第  $l$  条规则； $A_i^l$  是一个模糊子集，其隶属度函数的参数称作前提参数； $x_i$  是第  $i$  个输入变量； $n$  是输入变量的数目； $y^l$  是第  $l$  条模糊规则的输出； $P_i^l$  是第  $l$  个模糊规则结论中的第  $i$  个参数，称作结论参数。

如果给定输入模糊向量  $(x_1^0, x_2^0, \dots, x_n^0)$ ，那么输出  $y^l (l = 1, 2, \dots, M)$  由各条规则输出的加权平均求得为

$$\hat{y} = \sum_{l=1}^M G^l y^l / \sum_{l=1}^M G^l \quad (3-12)$$

其中  $M$  为模糊规则的数量,  $y^l$  由第  $l$  条规则的结论方程式计算,  $G^l$  为第  $l$  条规则的真值, 由下式计算

$$G^l = \prod_{j=1}^n A_j^l(x_j^0) \quad (3-13)$$

其中  $\prod$  为模糊化算子, 通常采用取小运算。

在应用 T-S 模型对被控对象动态模型进行系统辨识过程分为结构辨识和参数辨识, 而结构辨识又可分为前提结构和结论结构辨识; 参数辨识也分为前提参数和结论参数辨识。在文献[4, 5, 9]中给出了辨识的流程和辨识的具体实现步骤。

### 3.3.3 基于遗传算法的系统辨识与建模

进化计算是当今科学研究的热点之一, 遗传算法(Genetic Algorithm, GA)是其中一种, 其基本思想由美国密歇根大学的 Holland 教授在 60 年代提出的。遗传算法是人工智能的重要分支, 是基于达尔文进化论, 在计算机上模拟生命进化机制而发展起来的一门学科。它根据适者生存, 优胜劣汰等自然进化来进行搜索计算和问题求解, 对许多用传统数学难以解决或明显失效的复杂问题, 特别是优化问题, 提供了一个行之有效的新途径, 也为人工智能的研究带来了新的生机。

GA 是一类可以解决复杂优化计算的鲁棒搜索算法, 与其他优化算法相比, 它具有以下特点: GA 是对参数编码进行操作, 而不是对参数本身, 通过目标函数来计算适应度值, 因此 GA 不受诸如连续性、可导性等函数约束条件的限制, 也不需要其他的推导与辅助信息; GA 的搜索过程是从问题解的一个集合开始的, 具有隐含并行搜索特性, 从而大大减小了陷入局部极小值的可能; GA 对于待寻优函数基本无限制, 既可以是数学解析式表达的显函数, 也可以是映射矩阵或神经网络等隐函数, 也就是说 GA 能较好的解决复杂问题与非线性问题。

GA 是自然遗传学与计算机技术相结合而成的一种新的优化方法, 因此在研究和应用 GA 时经常要用到一些自然进化中的基本术语, 在这里简要介绍如下。

染色体: 遗传物质的主要载体, 指多个基因的集合。

基因: 控制生物性状遗传物质的功能和结构的基本单位, 又称遗传因子。

基因座: 染色体上遗传因子的位置称作基因座, 各个位置决定了遗传什么样的信息。

基因型: 由基因组合的模型叫做基因型, 它是性状染色体的内部表现。

表现型: 由染色体决定性状的外部表现, 或者说, 根据基因型形成的个体。

个体：染色体带有特征的实体。

群体：染色体带有特征的个体集合。该集合内的个体数称作群体的规模。

适应度：各个个体对各自适应环境的程度。

选择：决定以一定的概率从群体中选取若干对个体的操作。

交叉：把两个染色体换组的操作。又称重组。

突变：突然让基因以一定的概率变化的操作。

编码：从表现型到基因型的映射。

解码：又称作译码，是从基因型到表现型的映射。

在 GA 的结构中，主要包含选择、交叉和突变三个基本操作。选择操作体现了优胜劣汰的竞争和进化思想，而优秀的个体是靠交叉和突变的操作而获得，交叉和突变其目的都是为了产生优秀的个体，实际上，突变是为了更好地交叉，在这个意义上说，交叉和突变实质上都是交叉。因此，可以认为交叉是 GA 的本质操作，交叉的思想是 GA 的核心。有了不断的交叉才能不断的产生新的个体，才能不断推陈出新。

GA 的应用研究比理论研究更为丰富，目前已被广泛应用于许多实际问题，作为一种优化算法，本身不能直接用来实现系统辨识，而是作为一种寻优的手段来辅助实现对象模型的辨识。比如在线性系统辨识中，最常用的是最小二乘法。由于基于最小二乘法的辨识方法是以函数梯度信息为基础，在系统辨识的迭代过程，因而就不可避免地存在搜索区域的局部化、迭代过程收敛于局部解。利用 GA 可以在比较大的设计变量空间内迅速寻优，并更有可能获得全局最优解，所以可以克服最小二乘法的一些缺陷。在利用神经网络或者模糊模型来对非线性系统进行辨识时，网络的权值和阈值，利用神经网络的学习算法，有可能会使得它们陷入局部最小；模糊关系模型的隶属度函数参数、模糊规则，T-S 模糊模型的前提参数与结论参数，以及神经网络的神经元个数，这些参数的确定大都带有主观因素，如果选取的不得当，就可能无法实现对象的辨识。对于利用神经网络学习算法会引起神经网络参数陷入局部最小值这种情况，仿照 GA 解决克服最小二乘法缺陷的方法，可以得到改善；对于参数不易确定这样的问题，GA 将这些参数编码成初始种群，然后从初始种群开始寻优，根据适应度函数最终可以找到合适的参数。由于 GA 的寻优是全局的，因此即使选择的初始种群不甚合适，最后也能找到较为理想的参数，从而可以实现对象模型的辨识。

上面简要介绍了智能系统理论当中所涉及到的系统辨识的方法，此外还有一些将这些方法结合起来使用的综合性方法，也能较好的实现非线性系统的辨识工作。总之，基于智能系统理论的辨识方法为系统辨识的发展开辟了一个新的方向。在下面的章节中将具体介绍这些系统辨识方法在氧乐果合成间歇生产过程模型建立中的应用。



## 4 基于神经网络的氧乐果合成间歇过程系统辨识

随着人工神经网络应用研究的不断深入,新的神经网络模型不断推出,现有的神经网络模型已达近百种。在智能辨识方法中,应用最广的是以下几种基本模型和它们的改进型,即 BP 网络、Hopfield 网络、自组织神经网络、动态递归网络和动态记忆网络(BAM)等。在系统辨识当中,最常用的就是 BP 网络与动态递归网络,下面一一介绍他们在系统辨识中的应用。为了能更好的完成辨识工作,首先要对采集到的现场数据进行必要的预处理。

### 4.1 数据预处理

根据第二章介绍的氧乐果间歇生产工艺过程,可以知道该过程的温度受到了多种因素的影响。从现场采集到的数据来看,主要归结为一甲胺流量  $F_{yja}$ , 冷却盐水温度  $T_{lqs}$ , 计量罐液位  $L_{jlg}$ , 投料阀开度  $V_{uf}$  和反应当前时刻。其中一甲胺流量、计量罐液位和冷却盐水温度为直接测量到的影响因素;根据计量罐液位可以计算出一甲胺累积投料量  $sum_{yja}$ , 根据反应当前时刻  $t$  可以计算出反应进行时间  $t_{fy}$ , 将它们作为间接测量到的影响因素。因此反应釜温度  $T_{fy}$  与这些影响因素之间的关系可以表示为,

$$T_{fy} = f(F_{yja}, T_{lqs}, sum_{yja}, t_{fy}) \quad (4-1)$$

温度与各种影响因素之间的关系可以从现场采集到的温度数据以曲线的形式体现出来。温度曲线如图 4.1 所示。

图 4.1 给出的是一个月里不同日期在不同的运行批次所绘制的温度曲线,从这些曲线当中我们可以明显的看到每批次的温度数据也参差不齐,从而影响了各批次温度曲线也是相差甚远。如果用这样的数据来作为辨识模型的数据是不合理的,由于输入输出数据存在测量噪声,以及系统本身存在动态干扰,故在系统辨识之前有必要对这些原始数据进行数据的预处理。

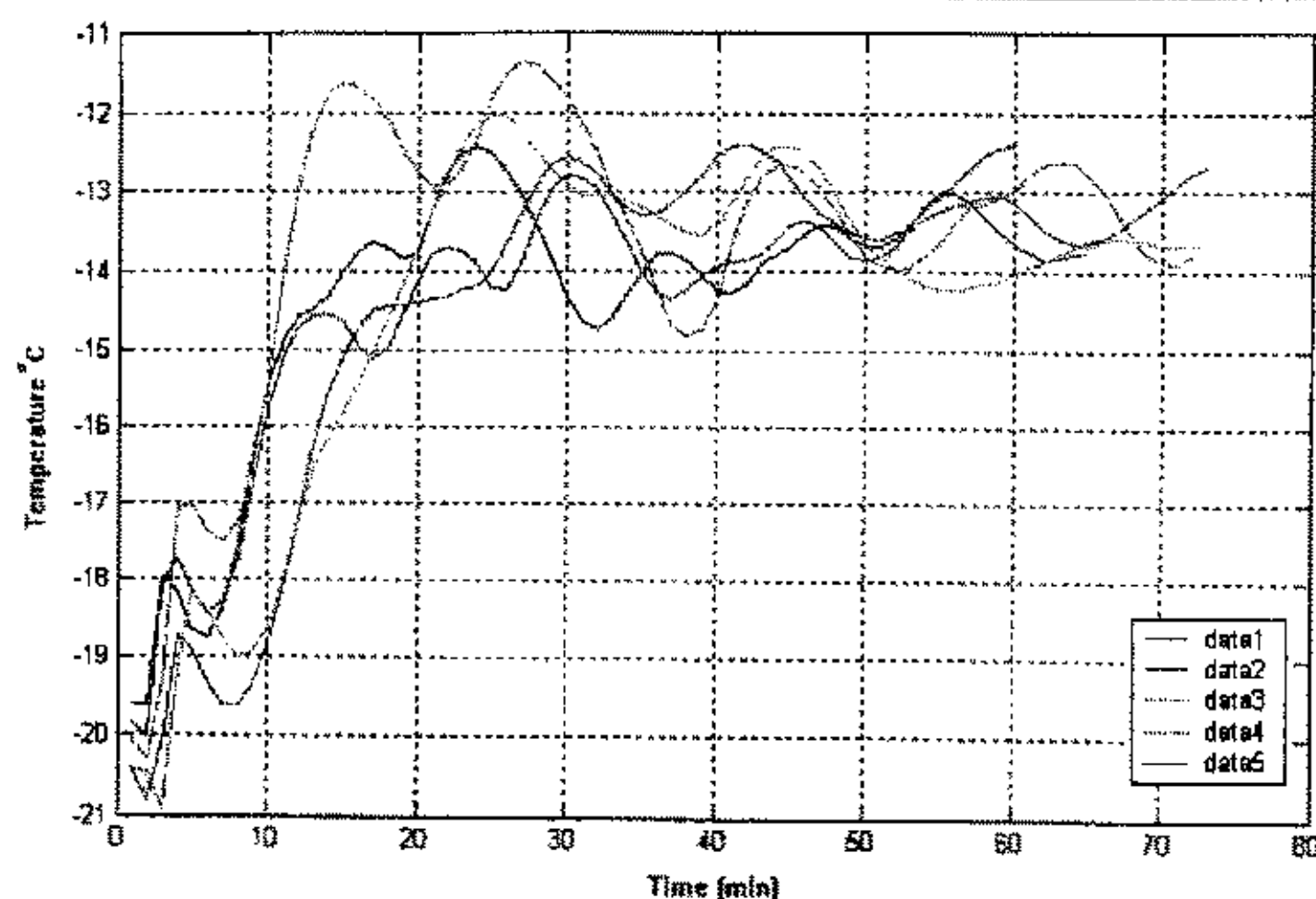


图 4.1 氧乐果合成反应温度曲线

Fig 4.1 Temperature curve of omethoate synthesis

所谓过程数据预处理，就是不要求知道过程的物理和化学机理或系统的结构，仅根据过程中所测量的各参数值，应用某些理论算法，例如误差理论、概率统计理论等等，在数据处理装置上分析、估计、校正部分误差，抑制噪声，得到过程参数的真实数值。工程上往往根据需要，将静态处理和动态处理分成两个独立的部分进行。

反应釜温度、计量罐液位和冷却盐水温度为直接测量到数据，它们都或多或少的同时存在着随机误差、系统误差和粗差。测量值中存在粗差时，无法进行统计分析。因此首先必须分析粗差，而系统误差的分析又是建立在判断偏离正态分布的基础上，所以第二步应分析随机误差，第三步则是分析系统误差。如果已知真值则应先分析系统误差，后分析随机误差。分析处理三类误差时，应有正确的步骤。这里以反应釜温度为例来说明怎样处理这些误差。

粗差是由于仪器仪表故障，或者是操作人员的大意而造成的，因此首先应该将采集数据中的这些数据剔除，这个过程称作坏值剔除。剔除过程的理论依据是小概率事件不可能发生<sup>[17]</sup>。对于反应釜温度数据，作这样的处理：首先将所有采集到的数据按照月份分成三组（四月、五月和六月），然后从每月当中均选择温度波动较小的 10 条批次曲线，如图 4.2 所示。

从图 4.2 中可以明显的看到黑色（3 号曲线）的曲线是含有粗差的曲线，这条曲线的最大值已经超过了  $-11^{\circ}\text{C}$ ，这在实际生产过程中就已经造成了事故，生产出来的就是次品甚至是废品，这样的数据是不能够作为辨识数据来使用的，根据拉依达准则<sup>[17]</sup>，这个数据应当被剔除。类似的，对三组数据分别进行坏值剔除，消除粗差。

对过程数据的某一测点进行  $n$  次测量，若误差的大小和方向均不相同，其变化

规律不能用一确定的数学形式描述，但服从统计规律，此种误差称为随机误差。

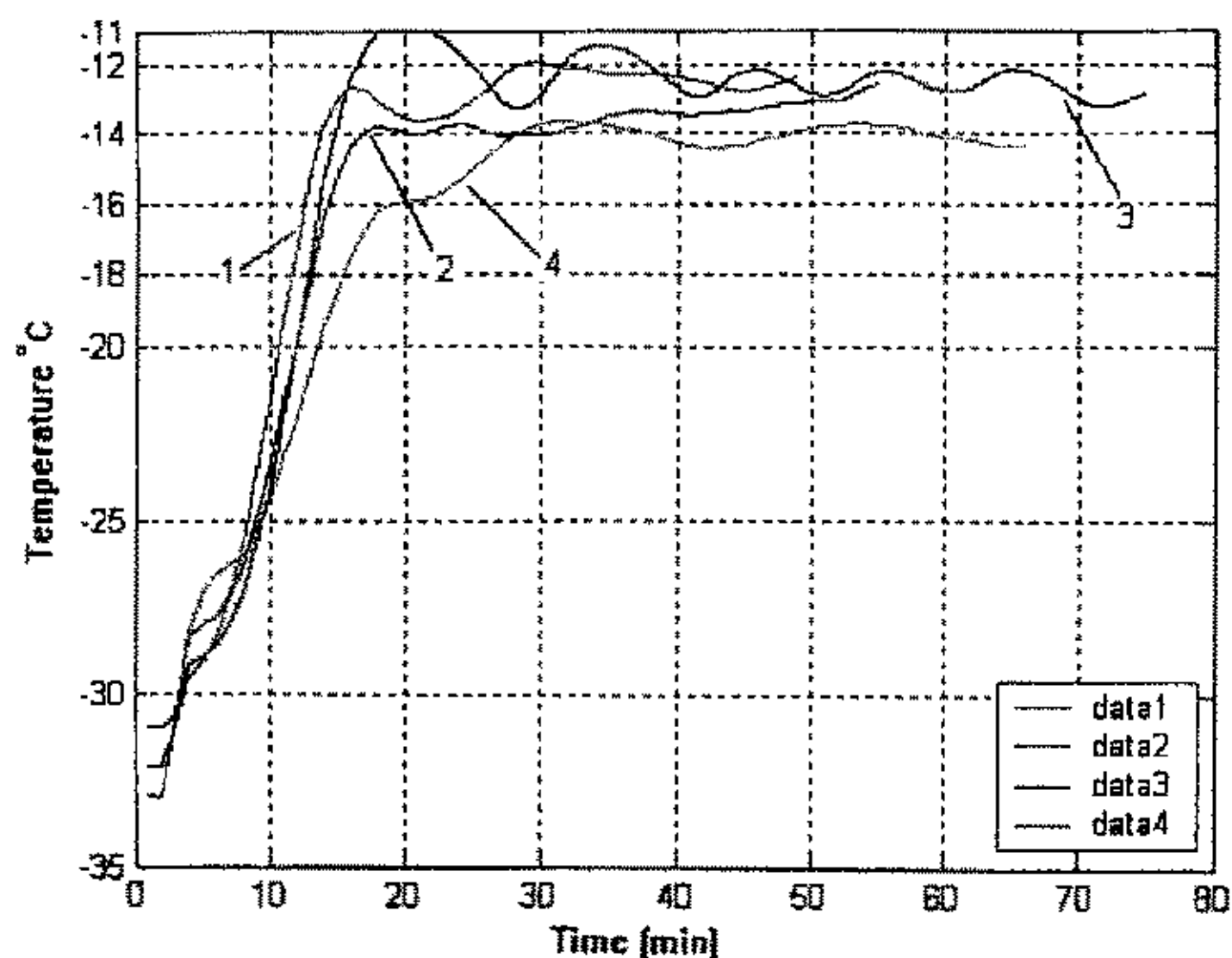


图 4.2 经过分组的氧乐果合成反应温度曲线（部分）

Fig 4.2 Temperature curve grouped of omethoate synthesis

由于随机误差只服从统计规律，不可能逐点校正处理。因此只能根据给定的概率来估计误差的区间，我们定义 $\Delta$ 为随机不确定度，并用下式表示：

$$\Delta = K \hat{\sigma}_x \quad (4-2)$$

其中 $K$ 为某一置信度(置信概率)下的置信系数，误差的区间则为 $-\Delta \rightarrow +\Delta$ ，故某点测量值范围应为：

$$x = x_i \pm \Delta = x_i \pm K \hat{\sigma}_x \quad (4-3)$$

显然对应于每个置信度，均有一个置信系数。

为方便起见，下面给出工程实验中常用的，正态分布中某些置信度与置信系数的一一对应的非线性关系。

置信度	置信系数
68%	1
95%	2
99%	2.58
99.73%	3

为了估计误差范围和分析极限误差，首先需要根据测量数据估计子样均值和方差，由统计理论知：

算术平均值：

$$\bar{x} = \hat{\mu}_x = \frac{1}{n} \sum_{i=1}^n x_i \quad (4-4)$$



$$\text{母体均方差有偏估计: } S_b^2 = \hat{\sigma}_x^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (4-5)$$

$$\text{母体均方差无偏估计: } S_b^2 = \hat{\sigma}_x^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (4-6)$$

因此, 只要知道测量数据值和测量次数, 就可估计母体均值和方差。

由于测量的次数  $n < 30$ , 因此由文献[17]可知, 随机误差区间估计的步骤为:

- (1) 将测量结果按测量顺序列表;
- (2) 按照(4-4)式计算测量值的算术平均值;
- (3) 求偏差  $\delta_i$  并列表, 找  $\delta_{\max}$  和  $\delta_{\min}$ , 计算公式为:  $\delta_i = x_i - \bar{x}$ ;
- (4) 求偏差  $\delta_i^2$  并将  $\delta_i^2$  列表, 然后求  $\delta_i^2$  之和;
- (5) 根据测量过程初步估计是否是正态分布的误差, 并根据计算  $\hat{\sigma}_x$  的精度要求  
和测量次数  $n$ , 选定计算  $\hat{\sigma}_x$  的方法并计算  $\hat{\sigma}_x$ ;
- (6) 由子样求随机不确定度(置信区间)。

在等精度测量条件下, 对某点数据进行  $n$  次测量, 如果测量误差的大小和方向不变, 或按某一确定的规律变化(如按单调增或单调减、周期性、指数、对数等规律变化), 这种误差称为系统误差。

按系统误差与测量顺序的关系, 可分为固定的系统误差, 即误差与测量顺序无关, 恒为常数。周期性的系统误差, 即误差的大小和符号随测量顺序周期性的变化。复杂变化规律的系统误差, 即误差的大小和符号随测量顺序按对数曲线、指数曲线、多项式规律, 甚至更复杂的形式变化。这种系统误差往往和随机误差混杂在一起, 很难区分开, 因而一般采用随机误差的方法进行处理。

系统误差存在的形式很多, 发现也很困难, 所以系统误差的分析处理是十分麻烦的, 但是可通过数据处理的方法来消除。系统误差很难通过统计分析方法(即用算术平均值作真值求偏差), 来发现系统误差的存在。此外, 当系统误差和随机误差同时存在, 但其数值远远小于随机误差时, 以及系统误差的性质不明显时, 都是不好处理的。

常用的分析系统误差的方法有作图法, 通过误差分析检验法和判据分析法<sup>[17]</sup>。在判据分析法中主要有平均误差判据、阿贝判据和马克夫判据。系统误差分析的基础是检验误差概率分布偏离正态分布的程度, 由于工程实验数据的测量次数有限, 不易得到标准正态分布, 因此这些判据也是很粗糙的。所以在处理氧乐果合成间歇过程数据的时候, 均把系统误差当成是复杂变化规律的系统误差, 按照处理随机误差的方法进行了处理。

某五批次的温度原始数据的一部分如表 4.1 所示, 选择置信度为 95%, 查置信度与置信系数的对应关系得到置信系数为 2, 故将数据进行随机误差与系统误差预处理后的结果如表 4.2 所示, 其对应的温度曲线如图 4.3 所示。



表 4.1 温度原始数据表

Tab.4.1 Original data of temperature

时间	温度 1	温度 2	温度 3	温度 4	温度 5	温度平均值	标准差(无偏)	$\Delta$
1	-31.6519	-30.971	-30.9586	-33.5118	-31.8967	-31.798	1.0436	2.0873
2	-31.6619	-30.9908	-30.9604	-33.5179	-31.6304	-31.7523	1.0425	2.085
3	-29.9461	-30.5158	-30.6528	-32.8019	-29.8148	-30.7463	1.2037	2.4074
4	-28.9082	-28.3326	-28.183	-30.3536	-29.6293	-29.0813	0.9106	1.8211
5	-28.7977	-27.9264	-26.9421	-29.9828	-29.6493	-28.6597	1.2487	2.4974
6	-28.6893	-27.7099	-26.5347	-30.0026	-29.503	-28.4879	1.3951	2.7903
7	-28.2279	-27.2056	-26.2971	-29.8261	-29.0079	-28.1129	1.4026	2.8053
8	-27.3641	-26.2593	-25.981	-29.327	-28.0665	-27.3996	1.3669	2.7338
9	-26.1716	-24.8999	-25.3111	-28.4433	-26.8828	-26.3417	1.4035	2.807
10	-24.7084	-23.3202	-23.1369	-27.2103	-25.4241	-24.76	1.67	3.3399
11	-22.9796	-21.5531	-21.9863	-25.7476	-23.7939	-23.2121	1.664	3.3281
...	...	...	...	...	...	...	...	...

表 4.2 预处理后的温度数据表

Tab.4.2 Pretreatment data of temperature

时间	温度 1	温度 2	温度 3	温度 4	温度 5
1	-29.5646	-28.8837	-28.8713	-31.4245	-29.8094
2	-29.5769	-28.9058	-28.8754	-31.4329	-29.5454
3	-27.5387	-28.1084	-28.2454	-30.3945	-27.4074
4	-27.0871	-26.5115	-26.3619	-28.5325	-27.8082
5	-26.3003	-25.429	-24.4447	-27.4854	-27.1519
6	-25.899	-24.9196	-23.7444	-27.2123	-26.7127
7	-25.4226	-24.4003	-23.4918	-27.0208	-26.2026
8	-24.6303	-23.5255	-23.2472	-26.5932	-25.3327
9	-23.3646	-22.0929	-22.5041	-25.6363	-24.0758
10	-21.3685	-19.9803	-19.797	-23.8704	-22.0842
11	-19.6515	-18.225	-18.6582	-22.4195	-20.4658
...	...	...	...	...	...

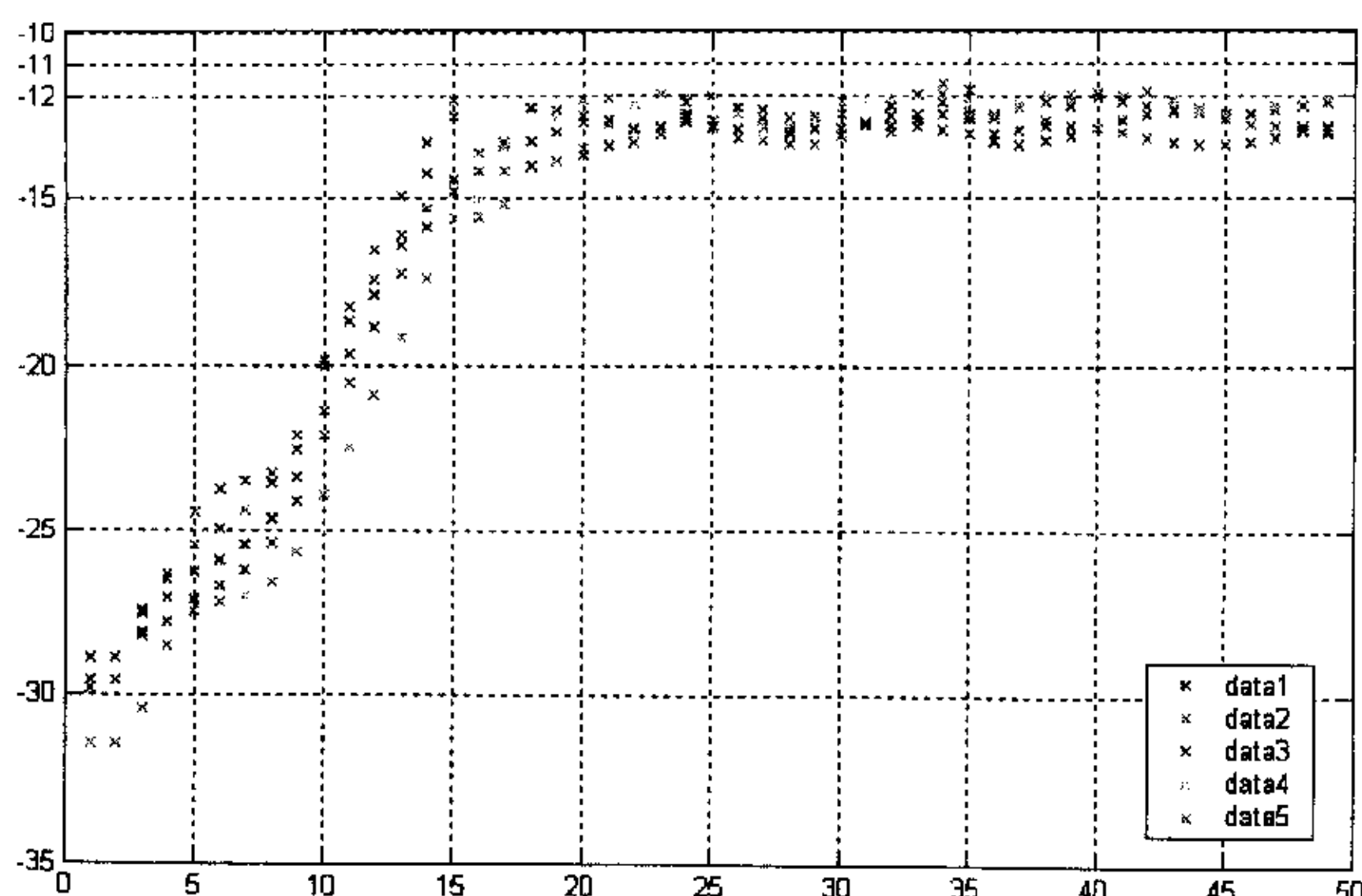


图 4.3 经过预处理的氧乐果合成反应温度曲线

Fig. 4.3 Pretreatment temperature curve of omethoate synthesis

## 4.2 BP 神经网络

在神经网络辨识方法当中，最常用的就是 BP 神经网络，因此在本论文中也采用 BP 神经网络来对氧乐果合成间歇生产过程中的温度对象进行模型辨识。

BP 模型是一种采用误差反向传播算法 (Error Back Propagation) 进行有监督训练的多层神经网络 (Multi-layer Feed Forward Network)。网络有输入层节点，输出层节点以及隐含层节点。隐含层可以是一层，也可以是多层。对于输入信号，要先向前传播到隐含节点，经过激活函数  $f(\cdot)$  后，再把隐层节点的输出信息传播到输出节点，最后给出输出结果。节点激活函数通常选取 S 型函数。网络结构图如图 4.4 所示。这是一个典型的多层前向网络模型。输入层和输出层的输入输出关系为线性。隐含层的输入输出关系为  $f(\cdot)$ 。

BP 网络可看作是一个从输入到输出的高度非线性映射，即  $F: R^n \rightarrow R^m, F(X)=Y$ 。对于样本集合：输入  $x_i \in R^n$  和输出  $y_i \in R^m$ ，可认为存在某一映射  $g$  使得，

$$g(x_i) = y_i \quad i = 1, 2, \dots, n$$

要求出一个映射  $f$  使得在某种意义下  $f$  是  $g$  的最佳逼近。神经网络通过对简单的非线性函数进行数次复合，可近似复杂的函数。定理 1 就给出了映射网络的存在性。

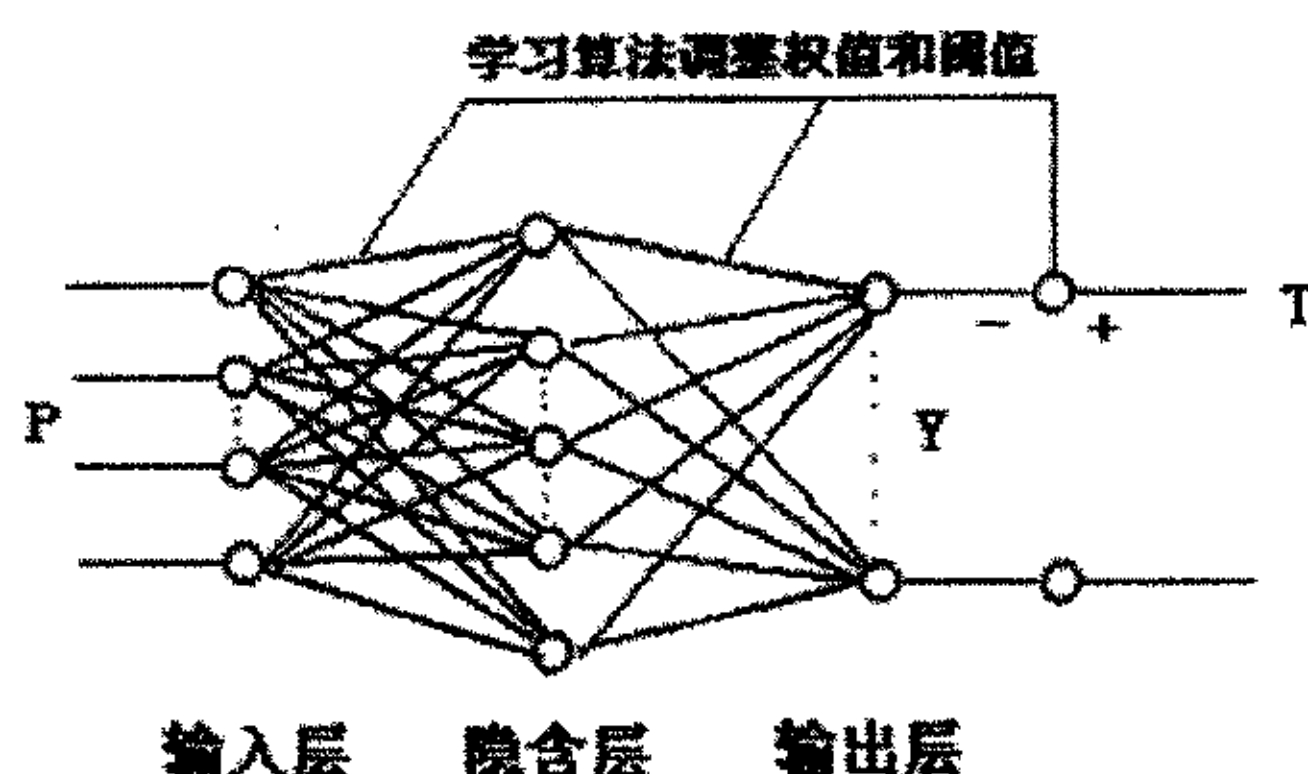


图 4.4 BP 网络拓扑结构图

Fig.4.4 Topology of BP net

定理 1 (Kolmogorov 定理) 给定一个连续函数  $f:U^n \rightarrow R^m, f(X)=Y$ , 这里  $U$  是闭区间  $[0,1]$ ,  $f$  可以精确的用一个三层前向网络实现, 网络的第一层 (输入层) 有  $n$  个处理单元, 中间层 (隐层) 有  $2n+1$  个处理单元, 第三层 (输出层) 有  $m$  个处理单元。

尽管定理 1 保证任意一个连续函数可以于一个三层前向神经网络来实现, 但是没有提供任何构造这样一个网络可行的方法。定理 2 给出了在 BP 网络中它可以在任意的精度上实现任意的连续函数。

定理 2 (BP 定理) 给定任意  $\varepsilon > 0$  和任意  $L_2$  函数  $f:[0,1]^n \rightarrow R^m$ , 存在一个三层 BP 网络, 它可在任意  $\varepsilon$  平方误差精度内逼近  $f$ 。

网络的学习算法由正向传播和反向传播两部分组成。在正向传播的过程中, 输入信息从输入层经过隐层逐层处理, 并传向输出层, 每一层神经元的状态只影响下一层神经元的状态, 同层神经元之间没有作用。若输出结果与期望的结果不符, 则反向传播误差信号, 将误差信号从输出层开始, 沿原来的连接通路返回, 通过修改权值, 反复上述过程使得误差信号最小。系统辨识工作分为两个工作阶段: 一是有监督的训练阶段, 调整网络权值, 使训练样本的实际输出与期望输出之间的误差达到一定的可接受范围; 二是工作阶段, 将实际输入 (可能含有未知模式) 赋予网络并得到输出的过程。

以三层 BP 网络为例, 输入层有  $r$  个神经元, 输入向量  $P \in R^r, P=(p_1, p_2, \dots, p_r)^T$ 。隐层有  $S1$  个神经元, 隐层输出向量  $A1 \in R^{S1}, A1=(a_{11}, a_{12}, \dots, a_{1S1})^T$ , 激活函数为  $F1$ , 阈值为  $B1, B1=(b_{11}, b_{12}, \dots, b_{1S1})^T$ 。输出层有  $S2$  个神经元, 输出向量  $A2 \in R^{S2}, A2=(a_{21}, a_{22}, \dots, a_{2S2})^T$ , 激活函数为  $F2$ , 阈值为  $B2, B2=(b_{21}, b_{22}, \dots, b_{2S2})^T$ 。

输入层与隐层之间的权连接为  $W1=[w1_{ij}]_{i=1,2,\dots,S1;j=1,2,\dots,r}$ , 隐层与输出层之间的权连接

为  $W2 = [w2_{ij}]_{i=1,2,\dots,S1; j=1,2,\dots,S2}$ 。

BP 网络是一个非线性映射系统，给定输入向量  $P$ ，通过网络映射得到输出向量  $A2$  为：

$$\begin{aligned} a1_i &= f_1\left(\sum_{j=1}^r w1_{ij} p_j + b1_i\right), i=1,2,\dots,S1 \\ y = a2_k &= f_2\left(\sum_{i=1}^{s1} w2_{ki} a1_i + b2_k\right), k=1,2,\dots,S2 \end{aligned} \quad (4-7)$$

通过  $Q$  对样本  $(P^1, T^1), (P^2, T^2), \dots, (P^Q, T^Q)$  训练网络，调整各神经与之间的连接强度，即  $W1$ 、 $W2$ 、 $B1$  和  $B2$  这些权值及阈值，达到在一定误差范围内正确映射的目的，使网络对非训练样本输入也能给出近似正确的输出。定义网络中所有可变参数的集合为  $\theta$ ，BP 网络所代表的映射即为  $A2 = F(P, \theta)$ ，其中  $F$  是  $F_1$  和  $F_2$  的复合。 $\theta$  包括  $W1$ 、 $W2$ 、 $B1$  和  $B2$ 。由此可见，BP 网络完成的是一个多重复合非线性映射，随着权值的改变，它可以实现各种复杂的非线性映射。

#### 4.2.1 标准 BP 算法

以图 4.4 的三层网络为例，(4-7) 给出了隐层和输出层的表达式。设取拟合误差的代价函数为

$$E = \frac{1}{2} \sum_{k=1}^{s2} (t_k - a2_k)^2 \quad (4-8)$$

BP 算法的优化计算方法采用的一阶梯度法，即最速下降法，根据梯度法，将误差  $E$  反向传播，修改权值和阈值。对输出层权系数，从第  $i$  个输入到第  $k$  个输出的权值有：

$$\Delta w2_{ki} = -\eta \frac{\partial E}{\partial w2_{ki}} = -\eta \frac{\partial E}{\partial a2_k} \frac{\partial a2_k}{\partial w2_{ki}} = \eta (t_k - a2_k) f2' a1_i = \eta \delta_{ki} a1_i \quad (4-9)$$

其中， $\eta$  为学习速率， $\delta_{ki} = (t_k - a2_k) f2' = e_k f2'$  (4-10)

同理可得， $\Delta b2_k = -\eta \frac{\partial E}{\partial b2_k} = -\eta \frac{\partial E}{\partial a2_k} \frac{\partial a2_k}{\partial b2_k} = \eta (t_k - a2_k) f2' = \eta \delta_{ki}$  (4-11)

对隐层权系数，从第  $j$  个输入到第  $i$  个输出的权值有：

$$\Delta w1_{ij} = -\eta \frac{\partial E}{\partial w1_{ij}} = -\eta \frac{\partial E}{\partial a2_k} \frac{\partial a2_k}{\partial a1_i} \frac{\partial a1_i}{\partial w1_{ij}} = \eta \sum_{k=1}^{s2} (t_k - a2_k) \cdot f2' \cdot w2_{ki} f1' \cdot p_j = \eta \delta_{ij} p_j \quad (4-12)$$

其中， $\delta_{ij} = e_i f1'$ ， $e_i = \sum_{k=1}^{s2} \delta_{ki} w2_{ki}$  (4-13)



$$\text{同理, } \Delta b_{1j} = \eta \delta_{ij} \quad (4-14)$$

$$\begin{aligned} w_{1j}(k+1) &= w_{1j}(k) + \Delta w_{1j}(k+1), b_{1j}(k+1) = b_{1j}(k) + \Delta b_{1j}(k+1) \\ w_{2_{ki}}(k+1) &= w_{2_{ki}}(k) + \Delta w_{2_{ki}}(k+1), b_{2_{ki}}(k+1) = b_{2_{ki}}(k) + \Delta b_{2_{ki}}(k+1) \end{aligned} \quad (4-15)$$

式 (4-7)、(4-9)、(4-11)、(4-12)、(4-14) 和 (4-15) 构成了 BP 网络的学习算法。

只要有足够多的隐层和隐层结点, BP 网络就可以逼近任意的非线性映射关系。然而标准 BP 网络学习算法的收敛速度慢, 而且容易陷入局部最小值, 因此又出现了一些改进的 BP 算法。

#### 4.2.2 改进的 BP 算法

一种改进的方法就是附加动量法。附加动量法使网络在修正其权值时, 不仅考虑误差在梯度上的作用, 而且考虑在误差曲面上变化趋势的影响, 其作用如同一个低通滤波器, 它允许网络忽略网络上的微小变化特性。在没有附加动量的作用下, 网络可能陷入浅的局部极小值, 利用附加动量的作用则有可能滑过这些极小值。

该方法是在反向传播法的基础上在每一个权值的变化上加上一项正比于前次权值变化量的值, 并根据反向传播往来产生新的权值变化。带有附加动量因子的权值调节公式为

$$\begin{aligned} \Delta w_{ij}(k+1) &= (1-mc)\eta\delta_i p_j + mc\Delta w_{ij}(k) \\ \Delta b_i(k+1) &= (1-mc)\eta\delta_i + mc\Delta b_i(k) \end{aligned} \quad (4-16)$$

其中  $k$  为训练次数,  $mc$  为动量因子, 一般取 0.95 左右。

附加动量法的实质是将最后一次权值变化的影响, 通过一个动量因子来传递。当动量因子取值为零时, 权值的变化仅是根据梯度下降法产生; 当动量因子取值为 1 时, 新的权值变化则是设置为最后一次权值的变化, 而依梯度法产生的变化部分则被忽略掉了。以此方式, 当增加了动量项后, 促使权值的调节向着误差曲面底部的平均方向变化, 当网络权值进入误差曲面底部的平坦区时,  $\delta_i$  将变得很小, 于是  $\Delta w_{ij}(k+1) \approx \Delta w_{ij}(k)$ , 从而防止了  $\Delta w_{ij}(k) = 0$  的出现, 有助于使网络从误差曲面的局部极小值中跳出。

根据附加动量法的设计原则, 当修正的权值在误差中导致太大的增长结果时, 新的权值应被取消而不被采用, 并使动量作用停止下来, 以使网络不进入较大误差曲面; 当新的误差变化率对其旧值超过一个事先设定的最大误差变化率时, 也得取消所计算的权值变化。其最大误差变化率可以是任何大于或等于 1 的值。典

型的值取 1.04。

另外一种常用的改进方法是自适应学习速率法。对于一个特定的问题,要选择适当的学习速率不是一件容易的事情。通常是凭经验或实验获取,但即使这样,对训练开始初期功效较好的学习速率,不见得对后来的训练合适。为了解决这一问题,人们自然会想到在训练过程中,自动调整学习速率。通常调节学习速率的准则是:检查权值的修正值是否真正降低了误差函数,如果确实如此,则说明所选取的学习速率值小了,可以对其增加一个量;若不是这样,而产生了过调,那么就应该减小学习速率的值。下式给出了一种自适应学习速率的调整公式,

$$\eta(k+1) = \begin{cases} 1.05\eta(k), & SSE(k+1) < SSE(k) \\ 0.7\eta(k), & SSE(k+1) > SSE(k) \\ \eta(k), & \text{其他} \end{cases} \quad (4-17)$$

$SSE(k)$  为误差平方和函数,初始学习速率  $\eta(0)$  的选取范围可以有很大的随意性。

与采用附加动量法时的判断条件相仿,当新误差超过旧误差一定的倍数时,学习速率将减少;否则其学习速率保持不变;当新误差小于旧误差时,学习速率将被增加。此方法可以保证网络总是以最大的可接受的学习速率进行训练。当一个较大的学习速率仍能够使网络稳定学习,使其误差继续下降,则增加学习速率,使其以更大的学习速率进行学习。一旦学习速率调得过大,而不能保证误差继续减少,则减少学习速率直到使其学习过程稳定为止。

除了以上两种方法之外,还有误差函数改进法和双极性 S 型压缩函数法等等 BP 算法的改进算法,具体的可以参看文献[4]、[21]。

### 4.3 氧乐果合成过程辨识实例

#### 4.3.1 BP 网络辨识温度对象

将整个过程的温度对象用一个 BP 网络来表示。对象的输入为一甲胺流量  $F_{ya}$ , 冷却盐水温度  $T_{lqs}$ , 一甲胺累积投料量  $sum_{yja}$  和反应进行时间  $t_{fy}$  共四个输入量, 输出为反应釜温度  $T_{ffy}$ 。其中, 一甲胺流量  $F_{ya}$ , 冷却盐水温度  $T_{lqs}$  和反应釜温度  $T_{ffy}$  是可以直接测量的数据, 这些数据经由前面介绍的方法预处理后得到; 一甲胺累积投料量  $sum_{yja}$  由计量罐液位  $L_{jlg}$  计算得到, 反应进行时间  $t_{fy}$  由反应时刻  $t$  计算得到。

计算方法为,以开始反应的时刻为初始时刻记液位和时间分别为  $L_{jlg}(0)$  和  $t(0)$ , 并

设  $sum_{yja}(0)=0, t_{fy}(0)=0$ , 那么  $sum_{yja}(i)=L_{jlg}(i)-L_{jlg}(0), t_{fy}(i)=t(i)-t(0), i=1, 2, \dots, n$ ,

$n$  为反应终止时刻。其中  $L_{jlg}(i), t(i)$  为第  $i$  时刻的液位和第  $i$  反应时刻。

经过预处理和计算后的数据并不能直接拿来训练 BP 神经网络,而是要经归一化处理后才能够形成适合于训练的数据。归一化所采用的公式为:  $y' = \frac{y - y_{min}}{y_{max} - y_{min}}$ ,

其中  $y, y'$  分别为归一化前后的数据,  $y_{min}, y_{max}$  分别是所有采样数据中的最小值和最大值。由于冷却盐水温度变化波动不大, 并且最低温度也不会低于  $-40^{\circ}\text{C}$  (从采集

数据中得到), 故冷却盐水温度  $T_{lqs}$  的归一化采用  $T_{lqs}' = \frac{T_{lqs}}{-40}$ ; 根据工艺知道反应釜

温度超过了  $-12^{\circ}\text{C}$  就会出现不合格的产品, 所以反应釜温度  $T_{ff}$  的归一化采用

$T_{ff}' = \frac{-12}{T_{ff}}$ 。归一化后的数据就构成了训练用的输入矩阵  $P_{tr}$  和目标矩阵  $T_{tr}$ , 如表

4.3 所示 (部分数据)。

表 4.3 输入矩阵  $P_{tr}$  和目标矩阵  $T_{tr}$

Tab.4.3 Input matrix  $P_{tr}$  and target matrix  $T_{tr}$

	$P_{tr}$				$T_{tr}$
	$F_{yja}$	$T_{lqs}$	$sum_{yja}$	$t_{fy}$	
1	0	0.9060	0.0016	0	0.3790
2	0.6451	0.9065	0.0138	0.0133	0.4007
3	0.7564	0.9075	0.0272	0.0267	0.4151
4	0.7650	0.9075	0.0408	0.0400	0.4167
5	0.7583	0.9075	0.0544	0.0533	0.4183
6	0.7491	0.9075	0.0675	0.0667	0.4251
7	0.7502	0.9075	0.0810	0.0800	0.4385
8	0.7421	0.9075	0.0941	0.0933	0.4585
9	0.7364	0.9075	0.1070	0.1067	0.4857
10	0.7327	0.9080	0.1201	0.1200	0.5222
...	...	...	...	...	...

确定了输入矩阵和目标矩阵也就确定了 BP 网络的输入层和输出层结点数,即输入层结点数为 4,输出层结点数为 1。接下来就是要确定 BP 网络的隐含层结构。由文献[22, 29]可知,用一个三层神经网络可以逼近任意一个非线性函数,只要具有足够多的隐层结点,因此辨识所使用的 BP 网络也采用三层网络。

一般来说,一个多层网络需要多少隐层,每层需要多少结点数,这要由网络的用途来决定。可以说隐层结点数与问题的要求,输入输出单元的多少都有直接关系。根据对隐含结点的几何解释知道,第一隐含层的每个结点确定了一个判决面,它把  $N$  为输入空间 ( $N$  为输入向量的分量数) 分为两部分。第二隐含层的每个结点又将第一隐含层结点形成的多个判决面组合成凸域空间或者判决域。最后输出结点又把多个凸域组成任意形状的判决空间或判决边界。显然,隐层结点有些用来提取输入图形的特征,有些则用来完成某些特殊功能。力图根据任务来确定隐层结点数,确定比较困难,因为网络映射的复杂性和由于许多成功地完成训练过程的不确定性的性质,目前还没有确切的方法和理论,通常是还是以经验为依据来选取。

有一些确定隐层结点数的经验公式,如,1987 年,Hechi-Nielsen 在讨论了具有单隐层的人工神经网络的功能之后,指出它可以实现输入的任意函数,并提出隐层结点数为  $2N+1$ ,  $N$  为输入结点数;同年 R.P.Lippmann 利用他对多层网络功能的几何解释,提出了隐层结点的估算。对于图形识别问题,第二个隐层结点数为  $M \times 2$ ,  $M$  为输出结点数,对于模式分类问题,隐层结点数  $H = \log_2 T$ ,  $T$  为输入训练模式数。1988 年, Kuarycki 根据实验发现,在高维输入时,第一隐层对第二隐层的最佳结点数的比例为 3:1;对医疗诊断问题,不少人作了研究,认为具有单隐层的网络就可以得到满意的结果。Lippmann 认为最大隐层结点数目为  $M_1(N+1)$ ; Kuarycki 认为最大隐层结点数目为  $M_1 \times 3$ ; A.J.Maren 等人认为对小型网络而言,输入结点数大于输出结点数时,最佳隐层结点数等于输入何输出结点数的几何平均数即  $(M_1 N)^{1/2}$ ,  $M_1$  为输出结点数。另外还有其他一些经验公式,  $J = \sqrt{n+m} + a$ ;  $J = \log 2n$  等等,  $J$  为隐层结点数,  $m$  为输出神经元数,  $n$  为输入神经元数,  $a$  为 1~10 之间的常数。其中  $J = \log 2n$  常用于数据压缩。

根据以上经验公式,辨识所采用的 BP 网络隐含层结点数取 10,其结构如图 4.5 所示。图 4.6 给出的 BP 网络在控制系统中所辨识的对象,其中  $D$  表示扰动。

代价函数采用(4-8)式,学习算法可以采用 4.2.2 节中提到的自适应学习速率法与附加动量法相结合的方法来,但是由于代价函数选取的是误差平方和的形式,因此可以使用专门用于误差平方和最小化的方法——Levenber-Marquart 算法(L-M 算法)。



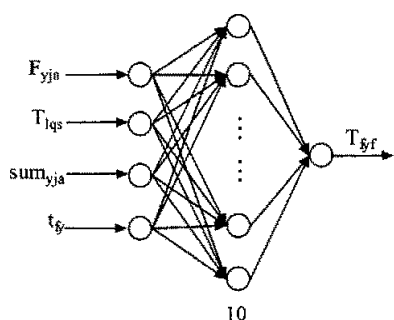


图 4.5 辨识所采用的 BP 网络

Fig.4.5 BP net for identification

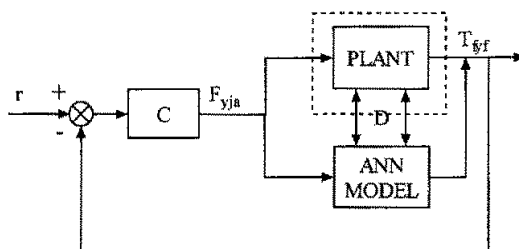


图 4.6 辨识结构图

Fig.4.6 Identification structure

(4-8)式可以改写为

$$E = \frac{1}{2} \sum_p (\varepsilon^p)^2 = \frac{1}{2} \|\varepsilon\|^2 \quad (4-18)$$

其中  $p$  是第  $p$  个样本,  $\varepsilon$  是以  $\varepsilon^p$  为元素的向量。假设当前老的值  $w$  向新的位置  $w'$  移动, 如果移动量为  $w' - w$  很小, 则可将  $\varepsilon$  展成一阶泰勒级数

$$\varepsilon(w') = \varepsilon(w) + Z(w' - w), \text{ 其中 } (Z)_{pi} = \frac{\partial \varepsilon^p}{\partial w_i}$$

于是(4-18)可以改写为  $E = \frac{1}{2} \|\varepsilon(w) + Z(w' - w)\|^2$ , 对  $w'$  求导以使  $E$  最小, 可得

$$w' = w - (Z^T Z)^{-1} Z^T \varepsilon(w) \quad (4-19)$$

由(4-18)式, 其 Hessian 阵<sup>[25]</sup>的元素是  $H_{ik} = \frac{\partial^2 E}{\partial w_i \partial w_k} = \sum_p \left[ \frac{\partial \varepsilon^p}{\partial w_i} \frac{\partial \varepsilon^p}{\partial w_k} + \varepsilon^p \frac{\partial^2 \varepsilon^p}{\partial w_i \partial w_k} \right]$ 。

若忽略其中第二项, 则 Hessian 阵可表示为  $H = Z^T Z$ , 但(4-19)给出的步长可能过大, 为此把误差表达式改写为  $E = \frac{1}{2} \|\varepsilon(w) + Z(w' - w)\|^2 + \lambda \|w' - w\|^2$ 。求  $E$  对  $w'$  的极小点, 可得

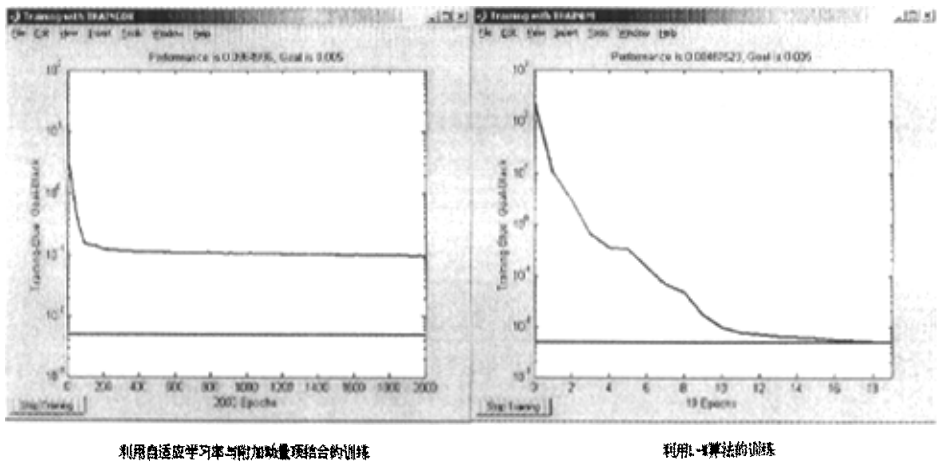
$$w' = w - (Z^T Z + \lambda I)^{-1} Z^T \varepsilon(w) \quad (4-20)$$

在计算过程中调节  $\lambda$  的大小, 首先任意选取  $\lambda$ , 再每步观察  $E$  的变化, 如果用了(4-20)后误差减小, 则保留  $w'$ ,  $\lambda$  减小 10 倍, 重复此步骤。如果误差增加, 则维持  $w$  且  $\lambda$  增大 10 倍, 然后再按(4-20)计算  $w'$ , 如此重复直到  $E$  合乎要求。以上就是 L-M 算法, 利用该算法可以比较快的完成训练过程。

目标误差即  $\varepsilon$  取 0.005。图 4.7 为训练过程中的误差变化曲线图。

图 4.7 中将利用自适应学习率与附加动量项结合方法的误差变化曲线与利用 L-M 算法训练的误差变化曲线进行比较, 可以明显的看到利用自适应学习率加动

量项的方法在 2000 步的训练完毕后并没有达到训练目标, 在这里学习速率选取  $0.01^{[21]}$ , 动量因子取  $0.95^{[21]}$ , 自适应速率的递增乘因子取  $1.05^{[21]}$ , 递减乘因子取  $0.7^{[21]}$ 。之所以会出现这种现象, 主要在于首先是目标误差的选取, 目标误差的选取一般是相对于所需要的隐层结点数来确定, 因为较小的目标误差值是要靠增加隐层结点, 以及训练时间来获得的。从这个角度考虑没有必要把目标误差设定的那么小, 从实际的角度考虑, 将目标误差设定为 0.01 或者 0.02 就基本可以满足实际需要了。其次是训练总步数, 如果不改变目标误差, 只要有足够的训练步长, 就可以训练到目标误差<sup>[21-23, 25]</sup>。再次就是这种算法本身存在一些问题, 虽然结合了自适应学习率与附加动量项两种方法的优点, 但是并不能说明一定可以解决标准 BP 算法会出现的问题。因为这种方法本质上还是梯度下降法, 所以就不可避免的会出现局部最小值。当权值收敛于误差曲面上的某一个局部极小值时, 就难以跳出局部极小点, 而要跳出局部极小点, 就需要选择较之以前更大的一个学习速率, 从而获得跳出极小点的能量, 但是自适应学习率的调整已经不能给出满足条件的学习速率, 因此也会使得训练曲线出现图中的情况。



利用自适应学习率与附加动量项结合的训练

利用L-M算法的训练

图 4.7 训练过程中误差变化曲线

Fig.4.7 Curve of error changed during the course of training

而对比 L-M 算法训练的误差曲线, 可以看到误差曲线只用了 19 步很快就达到了训练目标。常用的 BP 算法是梯度下降法, 参数沿着误差梯度相反的方向移动, 使误差函数减小, 直到取得极小值, 其计算的复杂度主要是由计算偏导数引起的。但是, 这种基于梯度下降方法的线性收敛速度很慢。带有动量项的改进 BP 算法是共轭梯度法<sup>[26]</sup>的近似, 而 L-M 算法是一种利用标准的数值优化技术的快速算法, 它是梯度下降法与高斯-牛顿法<sup>[26]</sup>的结合, 也可以说成是高斯-牛顿法的改进形式,

它既有高斯-牛顿法的局部收敛性，又具有梯度下降法的全局特性。从(4-20)式中可以看出如果 $\lambda = 0$ ，则为高斯-牛顿法；如果 $\lambda$ 取值很大，则 L-M 算法接近梯度下降法，每迭代成功一步，则 $\lambda$ 减小一些，这样在接近误差目标的时候，逐渐与高斯-牛顿法相似。高斯-牛顿法在接近误差的最小值的时候，计算速度更快，精度也更高。由于 L-M 算法利用了近似的二阶导数信息，它比梯度下降法快得多。另外由于 $[Z^T Z + \lambda I]$ 是正定得，所以(4-20)式的解总是存在的，从这个意义上说，L-M 算法也是优于高斯-牛顿法的，因为对于高斯-牛顿法来说， $Z^T Z$ 是否满秩还是个潜在的问题。此外，在精度要求高的情况下 L-M 算法的计算量和存储量都非常大，每次迭代的效率显著提高，大大改善了训练过程的整体性能。图 4.7 的对比充分说明了 L-M 算法在速度与精度上的优越性。

表 4.4 是训练好的 BP 神经网络的权值和阈值。

利用训练好的权值和阈值，采用 1 或者 2 组不同于训练时的数据对 BP 神经网络模型进行验证。网络的输入矩阵与输出矩阵同样要进行归一化处理，方法同前面的类似。

图 4.8 给出了网络输出与实际输出的拟合情况，从图中可以看出网络输出较好的逼近了实际输出，表 4.5 给出了两者之间的误差，计算公式为

$$error = abs\left(\frac{T_{act} - T_{net}}{T_{act}}\right)。$$

表 4.4 网络的权值和阈值

Tab.4.4 weight values and threshold values of network

w1 (10×4)				b1 (10×1)			b2 (1×1)		
-0.0264	-2.365	-3.9666	-6.6521	-2.365			3.3589		
0.1387	-0.0124	1.0256	0.6713	-0.0124					
0.0012	1.507	2.2267	-0.7117	1.507					
0.0636	0.0438	-2.1624	3.8627	0.0438					
-2.7998	-9.6909	-0.5192	-2.7555	-9.6909					
-1.6885	-8.2641	6.483	4.673	-8.2641					
0.0454	1.9498	1.3464	-0.1761	1.9498					
-0.0348	-1.4448	6.419	-3.1428	-1.4448					
-4.0578	-14.5214	-4.8192	0.3679	-14.5214					
-2.2774	-8.1569	-1.4377	-6.3567	-8.1569					
w2 (1×10)									
-2.4537	0.5722	-5.9446	-1.1961	2.2359	-0.0107	0.3779	0.1687	-1.0237	-1.228

表 4.5 网络输出与实际输出误差表

Tab.4.5 Error between net output and actual output

1	2	3	4	5	6	7	8	9	10
0.0837	0.1177	0.0672	0.0826	0.0939	0.0996	0.1007	0.1061	0.1141	0.1312
11	12	13	14	15	16	17	18	19	20
0.1448	0.1486	0.1398	0.1140	0.0129	0.0079	0.0018	0.0150	0.0255	0.0236
21	22	23	24	25	26	27	28	29	30
0.0152	0.0007	0.0244	0.0637	0.0319	0.0522	0.0544	0.0709	0.0144	0.0686
31	32	33	34	35	36	37	38	39	40
0.0497	0.0435	0.0089	0.0404	0.0024	0.0156	0.0097	0.0086	0.0199	0.0187
41	42	43	44	45	46	47	48	49	50
0.0026	0.0621	0.0319	0.0257	0.0253	0.0261	0.0190	0.0071	0.0270	0.0649
51	52	53	54	55	56	57	58	59	60
0.0386	0.0503	0.0668	0.0222	0.0311	0.0279	0.0425	0.0169		

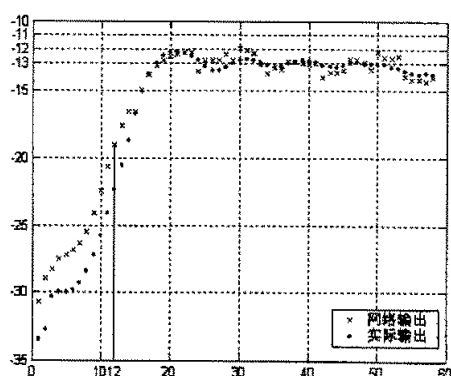


图 4.8 验证曲线

Fig.4.8 Verifying curve

从图 4.8 可以看到，网络输出与实际测量的温度是存在误差的，而且从误差表 4.5 可以看到最大误差高达 15%。这些误差比较高的点是出现在反应的开始阶段，也就是温度的上升阶段，这个阶段恰好是一甲胺累积投料量达到约 60~70 公斤的时候，这个阶段合成反应的总体放热效果开始超过系统的冷却效果，系统的热平衡被打破，反应温度开始快速增长。实际输出是由于在初始阶段温度的变化率较大，而工艺要求在初始阶段即使反应温度很低甚至是负增长，一甲胺的流量也不宜太大，因为此时反应釜内精脂含量较多，合成反应比较剧烈，容易造成局部温度过高，降低合成质量。从图 4.8 中可以看到由于温度变化率较大，反应在



控制量上就是一甲胺流得太快，就施加了减小一甲胺流量的控制信号，所以在后面时刻温度就上升的慢了，而网络输出曲线表现出的是从初始阶段到投料 60~70 公斤后，温度变化率并没有变得很大，因此温度就沿着这种趋势变化，所以在投料 60~70 公斤后比实际输出就高了一些。

以一甲胺流量作为控制量将该 BP 网络模型引入到控制系统当中进行控制，以手动的方式施加控制信号，其控制结构图如图 4.9 所示。

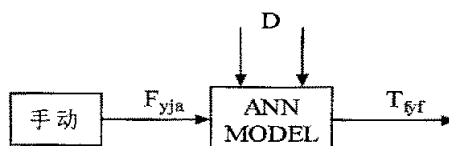


图 4.9 手动控制结构图

Fig.4.9 The structure of control with manual

在过程控制中，利用误差  $e$  这一信息来计算控制量是十分自然的，然而，如果只根据  $e$  的大小来进行控制，对于稍复杂一些的系统，很难得到令人满意的控制效果。当被控系统处于误差较大而又正向减小误差的方向快速变化时，如果只考虑误差大小而不考虑误差迅速变化的因素，必然要加大控制量，使系统尽快消除大的误差，这样的控制作用势必导致调节过头而又出现反向误差的不良后果。

表 4.6 符号变化表

Tab.4.6 Table of sign changed

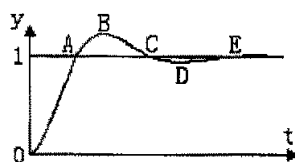


图 4.10 动态过程曲线

Fig. 4.10 Curve of dynamic

特征变量	AB 段	BC 段	CD 段	DE 段
$e$	$<0$	$<0$	$>0$	$>0$
$\Delta e$	$<0$	$>0$	$>0$	$<0$
$e \cdot \Delta e$	$>0$	$<0$	$>0$	$<0$
	超调段		回调段	

当采用两个输入变量  $e$  和  $\Delta e$  进行控制时就可避免上述的盲目性，根据模糊控制的基本思想，这是不难理解的。因此可以得出这样的结论：一个人工控制的复杂系统，在控制过程中，人对被控系统的状态、动态特征及行为了解的越多，控制的效果就会越好。

对于如图 4.10 所示的动态过程曲线的不同阶段，特征变量  $e$ 、 $\Delta e$  及  $e \cdot \Delta e$  的取值符号由表 4.6 给出。

当  $e \cdot \Delta e < 0$  时，如曲线 BC 段和 DE 段，表明系统的动态过程正向着误差减小的方向变化，即误差的绝对值逐渐减小，这时应减小控制作用或不加控制作用，

被控量将借助于惯性继续趋近于设定值。

当  $e \cdot \Delta e > 0$  时, 如曲线 AB 段和 CD 段, 表明系统的动态过程正向着误差增大的方向变化, 即误差的绝对值逐渐变大, 这时应加大控制作用, 改变误差变化的方向, 使被控量逐渐趋近于设定值。

按照上面的原则对 BP 网络模型手动施加控制信号。在实现手动控制的程序中, 神经网络输入矩阵中冷却盐水温度取  $-36^{\circ}\text{C}$ , 一甲胺流量由手动输入给定, 反应进行时间由初始 0 时刻开始, 每给一次一甲胺流量就累加 1。在程序中, 人为设定一甲胺总投料量 (250kg) 和计量罐初始液位 (200kg), 利用这两个量与一甲胺流量, 由公式

$$\begin{aligned} L_{jlg}(t+1) &= F_{yja}(t)/60 + L_{jlg}(t) \\ \text{sum}_{yja}(t) &= L_{jlg}(t) - L_{jlg}(0) \end{aligned} \quad (4-21)$$

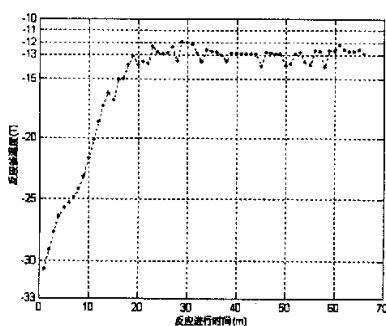


图 4.11 手动控制曲线图

Fig.4.11 Curve of control with manual

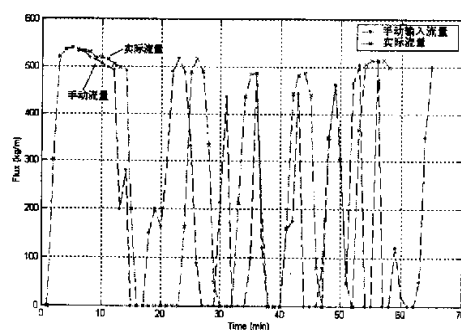


图 4.12 一甲胺流量 (BP 神经网络模型)

Fig.4.12 Monomethyl amine flux of BP net model

就可以计算出输入矩阵中的一甲胺累积投料量, 其中除以 60 表示将单位由  $\text{kg/h}$  转换成  $\text{kg/m}$ 。控制曲线如图 4.11 所示。

图 4.12 中的实际流量曲线对应于图 4.8 中的温度。从图中可以看到手动输入的控制量——一甲胺流量曲线与实际流量的变化趋势是一致的。比如在反应初始阶段, 尽管手动给定的流量比较小, 但是温度变化却比较大; 在反应结束阶段, 尽管手动给定的流量比较大, 但是温度变化却比较小。温度变化规律同实际的情况也是一致的。因此对于采用 BP 网络辨识出来的模型, 能够仿真实际的过程, 手动基本可以实现较好的控制。

氧乐果间歇过程的 BP 网络模型在控制系统中的仿真实验, 从另外一个方面也证明了该模型是较好的实现了实际过程。

根据氧乐果间歇生产的工艺可以了解到, 在整个反应过程中, 在不同的阶段要求的一甲胺流量速度是不同的, 这反映了实际对象参数的一种时变性。如果将反应过程按照工艺要求分成几个不同的阶段, 并利用神经网络建立各个不同阶段

的模型，那么将使得模型对生产过程来说更有针对性。另外在上面的仿真实验中，在反应的开始阶段，有一些数据误差较大，如果采用分阶段模型的话，有可能减小这个误差。

按照工艺要求以及氧乐果合成反应的静特性，将整个反应过程大致分为三个阶段：

1.  $M_i < M_1 = 60/70\text{kg}$
2.  $M_1 \leq M_i < M_2 = 120\text{kg}$
3.  $M_i \geq M_2$

其中  $M_i$  为一甲胺累积投料量。

训练数据仍旧采用前面训练时的数据，并根据分阶段的要求将训练数据和目标数据分成三部分，训练的参数也采用相同的训练参数。表 4.7~表 4.9 是训练完后各阶段 BP 网络模型的权值和阈值。

另外取一组数据（依旧采用前面验证时的），将该组验证数据的输入矩阵和输出矩阵也分成相应的三个阶段，然后计算各阶段的网络输出。

图 4.13 给出的是各个阶段的验证曲线。表 4.10~表 4.12 给出的是各阶段网络输出与实际输出之间的误差。结合曲线图与误差表，可以看到，在初始阶段尤其是在第二个阶段，即一甲胺累积投料量达到约 60~70 公斤的时候，这个阶段的误差已经明显的比采用单一神经网络时要小了很多。虽然这样的分阶段模型弥补了单一模型中的不足之处，但是也存在着一些问题。在第三个阶段出现了两个误差

表 4.7 第一阶段网络的权值和阈值

Tab.4.7 weight values and threshold values of network (1<sup>st</sup> stage)

w1 (10×4)				b1 (10×1)			b2 (1×1)		
-0.6739	1.4815	-1.6713	2.8886	0.7710			-0.3178		
-0.5316	4.2982	1.4188	2.0107	-0.9935					
0.0754	-4.0356	-5.9650	4.1715	-1.7886					
-0.0050	-3.9864	3.6239	-4.5264	5.2636					
-0.2964	3.1518	-4.3059	3.8910	-3.2299					
-0.1711	3.1623	2.0318	-3.5859	-0.0605					
0.6033	0.7851	3.6216	3.9204	-1.3608					
-0.2382	3.9625	0.7738	3.5376	-5.4255					
-0.3065	4.7656	4.0352	-0.3844	-4.5366					
-0.5014	0.4363	2.5587	-4.1172	-0.9645					
w2 (1×10)									
-0.4759	0.5237	-0.7513	0.7666	-0.8360	0.1718	0.4051	0.0894	-0.1325	0.7872

表 4.8 第二阶段网络的权值和阈值

Tab.4.8 weight values and threshold values of network (2<sup>nd</sup> stage)

w1 (10×4)				b1 (10×1)		b2 (1×1)			
-0.3880	2.1868	-3.3277	-0.5018	2.2259		0.2066			
0.0648	-4.6911	-3.0136	1.8324	-1.3351					
0.1024	3.4365	-4.9080	-4.0685	0.8456					
0.3551	-5.8302	-3.4157	-0.3534	2.6336					
-0.0501	2.9363	2.3546	-5.6501	0.0141					
0.3530	3.9804	-3.6573	2.9304	-2.9476					
-0.2636	5.1023	4.1424	0.9509	-4.0729					
-0.8796	-3.6800	-3.5498	-2.3024	1.4160					
0.0421	2.9474	-2.9130	-5.3374	3.2376					
0.2653	-3.1914	2.4882	-4.0111	6.6145					
w2 (1×10)									
-0.6390	0.4950	1.5615	-0.5624	-2.2602	-0.3363	0.9391	0.2273	0.2636	0.4643

表 4.9 第三阶段网络的权值和阈值

Tab.4.9 weight values and threshold values of network (3<sup>rd</sup> stage)

w1 (10×4)				b1 (10×1)			b2 (1×1)		
1.0534	5.7614	-2.7660	-0.1459	-0.7895			0.2863		
0.8104	-0.3544	2.1303	-2.2556	-0.7767					
0.3611	0.7204	3.8721	-6.9976	3.0352					
1.3590	4.6851	3.3770	-3.1672	-1.1584					
-0.1993	3.5321	3.8261	-3.9439	0.3275					
-0.0679	3.3398	-4.0727	3.2901	-4.8241					
-0.2465	-4.1006	-3.7016	7.0065	-2.0026					
1.5808	-4.9234	0.8292	-2.6659	5.0929					
0.4445	5.7302	-3.9174	-3.3498	1.3201					
0.0414	-1.8760	-3.6560	-3.8589	-0.2873					
w2 (1×10)									
-0.9388	1.3683	-1.5198	-0.8798	-3.6072	-0.2904	-2.8759	-1.3102	1.8671	0.3486

比较大的点，对应的反应釜温度分别为-11.3403℃和-12.1557℃。第一个点的温度已经超过了一12℃，是网络在计算输出时的一个错误点，那么这个点究竟会不会影响到对整个系统的控制，在后面将加以仿真验证。此外，在反应即将结束的



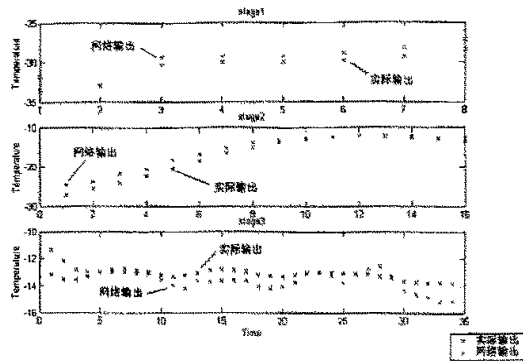


图 4.13 各阶段网络验证曲线图

Fig.4.13 Verifying curve of all stages

表 4.10 第一阶段误差表

Tab.4.10 Error of stage 1

	1	2	3	4	5	6	7	8
误差	0.0537	0.0052	0.0320	0.0228	0.0283	0.0342	0.0418	0.0479

表 4.11 第二阶段误差表

Tab.4.11 Error of stage 2

	1	2	3	4	5	6	7	8
误差	0.0976	0.0759	0.1082	0.0848	0.1172	0.1092	0.1043	0.0829
	9	10	11	12	13	14	15	16
误差	0.0511	0.0255	0.0054	0.0046	0.0127	0.0193	0.0248	0.0447

表 4.12 第三阶段误差表

Tab.4.12 Error of stage 3

	1	2	3	4	5	6	7
误差	0.1662	0.1118	0.0536	0.0255	0.0011	0.0153	0.0220
	8	9	10	11	12	13	14
误差	0.0207	0.0128	0.0233	0.0450	0.0662	0.0441	0.0596
	15	16	17	18	19	20	21
误差	0.0614	0.0523	0.0446	0.0609	0.0642	0.0543	0.0374
	22	23	24	25	26	27	28
误差	0.0047	0.0050	0.0138	0.0516	0.0023	0.0288	0.0618
	29	30	31	32	33	34	35
误差	0.0118	0.0455	0.0585	0.0698	0.0935	0.0874	

时候也出现了一些比较大的误差（与采用单一网络计算的输出比较），这是因为在反应结束阶段反应釜内的精脂大部分已经被消耗掉了，此时放出的热量较之以前已经有很大幅度的降低，所以网络输出温度可能要低一些。

将反应过程的各个阶段模型引入到控制系统当中，采用手动方式来进行控制。图 4.14 是手动控制时的温度曲线图。从控制曲线当中可以看到，采用分阶段的网络模型和采用单一网络模型时同样能够控制的很好，而且在第三阶段中出现较大误差的点并未对控制效果造成很大的影响，因此从这点可以说明虽然第三阶段的模型存在一定的误差，但是通过控制器施加合适的控制信号是可以弥补的。由控制曲线可以进一步说明，分阶段模型也在一定程度上实现了实际生产过程。

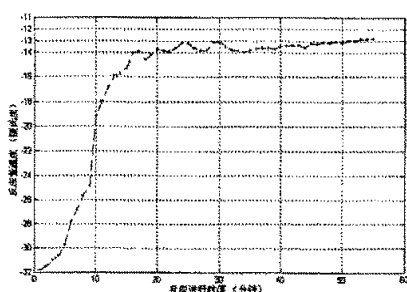


图 4.14 手动控制曲线图(分阶段模型)

Fig.4.14 Curve of control with manual

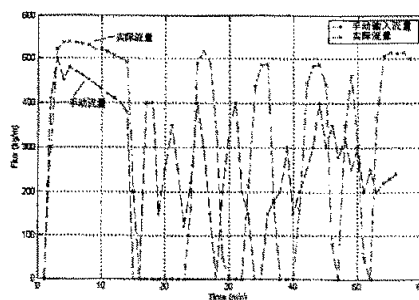


图 4.15 一甲胺流量（分阶段模型）

Fig.4.15 Monomethyl amine flux of stage model

图 4.15 给出了与图 4.13 类似的流量曲线。手动一甲胺流量曲线与实际流量的变化趋势也是一致的。综上，分阶段模型，不仅能够仿真实际的过程，而且也能够体现出实际过程的时变性。

#### 4.3.2 回归网络辨识温度对象

在 4.3 节中介绍了利用静态 BP 网络辨识的过程，但是实际的系统是一个动态的系统，前  $n-1$  个时刻的温度对第  $n$  时刻有影响作用，因此单纯的使用静态网络来完成辨识与实际的动态过程还是有一定差距的，在文献[7]中已经证明了利用 TDL 环节加上静态 BP 网络所构成的回归网络可以实现动态过程的辨识。

氧乐果生产过程中的实际温度数据是每隔一分钟采样一次，前一分钟的温度对当前时刻的温度影响最大，所以在 4.3.1 节单一 BP 神经网络模型的基础上，输入层多引入一个结点，作为延迟一拍的温度输入，网络结构如图 4.16 所示。

训练所使用的数据（部分）如表 4.13 所示，在输入矩阵中加入延迟一拍的反应釜温度。网络的隐层结点变为 12 个，其他训练参数与 4.3.1 节的一致。训练完

成后的网络权值和阈值如表 4.14 所示。

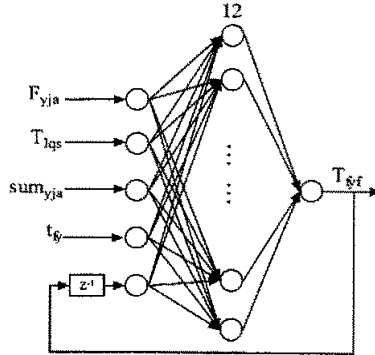


图 4.16 带有一个延迟环节的 BP 网络

Fig.4.16 BP net with a delay block

表 4.13 回归网络训练数据

Tab.4.13 Train data of regressive net

	$P_{tr}$					$T_{tr}$
	$F_{yja}$	$T_{lqs}$	$sum_{yja}$	$t_{fy}$	$T_{tr} \times 10^{-1}$	
1	0	0.9060	0.0099	0	0	0.4942
2	0.5509	0.9065	0.0196	0.0333	0.4942	0.5072
3	0.7023	0.9075	0.0322	0.0467	0.5072	0.5223
4	0.7429	0.9075	0.0454	0.0600	0.5223	0.5338
5	0.7551	0.9075	0.0588	0.0733	0.5338	0.5434
6	0.7353	0.9075	0.0718	0.0867	0.5434	0.5592
7	0.7187	0.9075	0.0839	0.1000	0.5592	0.5842
8	0.6691	0.9075	0.0947	0.1133	0.5842	0.6189
9	0.6027	0.9075	0.1038	0.1267	0.6189	0.6624
10	0.5511	0.9080	0.1112	0.1400	0.6624	0.7122
...	...	...	...	...	...	...

为了对比起见，图 4.19 给出了两种不同网络的验证情况。(A) 是利用不含有延迟环节的静态 BP 网络计算出的温度与实际温度的匹配情况，该 BP 网络模型的训练方式与 4.3.1 节中的单一 BP 神经网络模型的训练方式完全一样；(B) 是利用含有一个延迟环节与静态 BP 网络组成的回归神经网络计算出的温度与实际温度的匹配情况。虽然网络的结构是动态回归网络的一种形式，但是其训练方法还是

与 BP 网络的训练方法相同的。表 4.15 给出的是利用两种不同的网络计算出来的温度与实际温度之间的误差。结合图 4.19 和表 4.15 可以清楚的看到，在静态 BP 神经网络的基础上加入延迟环节后，网络的泛化能力较之静态 BP 网络有了很大的改善，尤其是在反应的初始阶段和一甲胺累积投料量达到约 60~70 公斤的时候，(B) 的误差比 (A) 的误差要小很多。

表 4.14 网络的权值和阈值

Tab.4.14 weight values and threshold values of network

w1(12×5)					b1(12×1)	b2(1×1)					
-0.1931	8.3848	-3.5955	-6.2902	-2.2481	3.4432	0.9257					
4.1266	1.2266	3.7976	-4.0619	-0.7835	-6.5842						
3.6841	-2.7986	-2.1287	6.811	-3.3902	0.5371						
0.8204	4.0364	-3.3166	8.9445	4.3449	-3.3675						
4.775	0.8009	3.1954	-4.0533	0.627	-4.6038						
4.5143	-2.4447	4.2329	3.3379	-1.5279	-2.1224						
-2.7524	5.7194	-4.5789	5.0294	0.1123	1.3573						
-0.1543	2.3599	-5.887	4.5162	-1.7562	3.4691						
-2.3001	7.0857	3.7977	-0.0309	-1.0673	-4.0765						
2.4527	8.8383	-3.9571	-0.4304	-2.9778	2.6888						
4.9258	0.8755	-3.7534	1.3653	1.386	0.7685						
-2.8715	9.1087	1.6946	-2.6043	1.5738	-6.5428						
w2 (1×12)											
1.0594	0.8748	-0.2022	1.0159	-0.4061	0.7403	1.1345	-1.45	-0.23	-1.4127	1.2735	-1.074

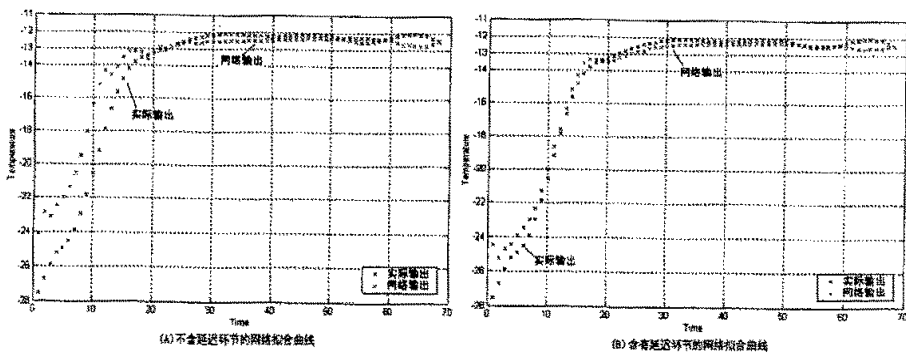


图 4.17 验证曲线比较

Fig.4.17 Comparison of testing curve



表 4.15 拟合曲线误差对比

Tab.4.15 Error comparison of matching curve

	1	2	3	4	5	6	7	8	9
含延迟	0.1265	0.0574	0.0496	0.0336	0.0432	0.0448	0.0390	0.0283	0.0251
不含延迟	0.1423	0.1689	0.1208	0.1247	0.1334	0.1474	0.1618	0.1775	0.2079
	10	11	12	13	14	15	16	17	18
含延迟	0.0269	0.0273	0.0151	0.0188	0.0317	0.0338	0.0449	0.0336	0.0116
不含延迟	0.2498	0.2598	0.2454	0.1423	0.1091	0.0959	0.0787	0.0520	0.0259
	19	20	21	22	23	24	25	26	27
含延迟	0.0048	0.0162	0.0219	0.0173	0.0244	0.0266	0.0277	0.0322	0.0250
不含延迟	0.0192	0.0010	0.0153	0.0092	0.0020	0.0080	0.0145	0.0132	0.0265
	28	29	30	31	32	33	34	35	36
含延迟	0.0256	0.0292	0.0293	0.0270	0.0263	0.0243	0.0250	0.0233	0.0201
不含延迟	0.0279	0.0271	0.0296	0.0320	0.0330	0.0325	0.0303	0.0277	0.0253
	37	38	39	40	41	42	43	44	45
含延迟	0.0169	0.0146	0.0143	0.0137	0.0139	0.0139	0.0154	0.0163	0.0174
不含延迟	0.0241	0.0227	0.0212	0.0204	0.0192	0.0195	0.0190	0.0196	0.0186
	46	47	48	49	50	51	52	53	54
含延迟	0.0179	0.0177	0.0178	0.0166	0.0142	0.0106	0.0057	0.0009	0.0065
不含延迟	0.0176	0.0170	0.0151	0.0151	0.0130	0.0073	0.0009	0.0112	0.0163
	55	56	57	58	59	60	61	62	63
含延迟	0.0102	0.0111	0.0083	0.0033	0.0052	0.0186	0.0290	0.0390	0.0478
不含延迟	0.0184	0.0189	0.0152	0.0093	0.0047	0.0089	0.0293	0.0441	0.0505
	64	65	66	67	68				
含延迟	0.0438	0.0400	0.0372	0.0164	0.0012				
不含延迟	0.0516	0.0521	0.0503	0.0232	0.0046				

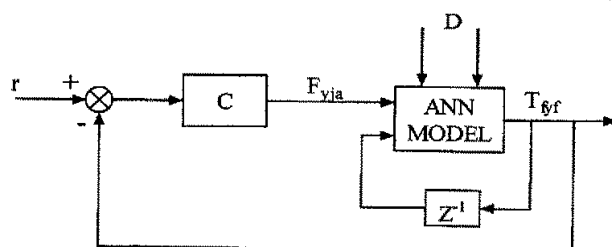


图 4.18 控制结构图

Fig.4.18 Structure of control

把该回归网络模型也引入到控制系统中来验证其合理性。控制系统结构图如图 4.18 所示。

图 4.19 是手动控制的曲线，控制程序中的变量设置与 4.3.1 节中手动控制程序一致，从图中可以看到在引入了一个延迟环节之后，温度能够比较好的在  $-12^{\circ}\text{C}$  处稳定，并且没有超调出现，在反应初始阶段以及稳态时没有振荡或者振荡很小。引入了延迟环节的回归网络模型较之静态 BP 网络模型在对实际过程的动态模拟上，效果更佳。

图 4.20 是对应的一甲胺流量曲线。手动与实际流量变化趋势接近。综合验证曲线的比较以及控制流量曲线，也从另外一个角度验证了引入延迟环节的回归网络模型较之静态 BP 网络模型在对实际过程的动态仿真上，效果更好。

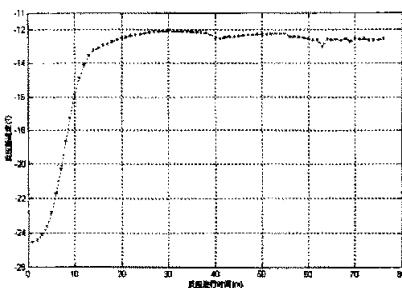


图 4.19 手动控制曲线图

Fig.4.19 Curve of control with manual

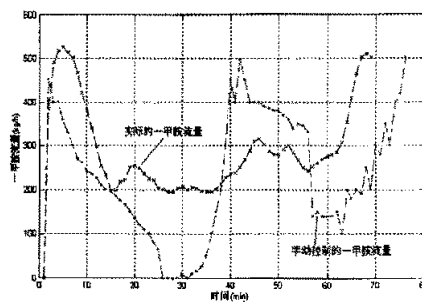


图 4.20 一甲胺流量（回归神经网络模型）

Fig.4.20 Monomethyl amine flux of recurrent net model

综上，各种网络模型有各自的特点：单个 BP 网络模型简单，在应用中方便，可以避免模型切换时候出现的扰动；分阶段 BP 网络模型体现出了实际过程的参数时变性；回归网络模型体现了实际的动态过程。

#### 4.4 一甲胺流量与投料阀开度关系曲线

在 4.3 节的仿真控制当中，控制器输出使用的是一甲胺流量，在工程实际中是不容易实现的，因此必须将其转换为在工程实际中便于实现的控制信号——投料阀开度信号，这样才可能真正实现对一甲胺流量的控制。一甲胺流量受投料阀开度和计量罐内一甲胺液位的影响。计量罐为敞口容器，不用考虑负压的影响。

因此，对于确定的计量罐液位  $L_{jlg}$  和投料阀开度  $V_{lf}$ ，就可以确定一个一甲胺流量

$F_{yja}$ ，即存在映射关系： $f: L_{jlg}, V_{lf} \rightarrow F_{yja}$ 。由于液位不同，即使投料阀开度相同，

一甲胺流量也不相同；而对投料阀开度不同，在不同的液位情况下，则一甲胺流量有可能会相同，因此  $f$  比较难以确定，在文献[6]中利用神经网络的方法来逼近  $f$ ，但是该神经网络的训练数据都是采用的一些典型的样本，所以训练出来的神经网络具有一定的局限性。采用[6]中的神经网络，从生产过程的历史数据中任意取一组异于训练数据的数据来验证，效果不是很理想，也就是说网络的泛化能力比较差。

如果将所有的历史数据都拿来作为神经网络的训练数据是不切合实际的，本文从统计的角度出发，利用大量的历史数据，结合数理统计与曲线拟合的相关知识，初步做出一甲胺流量的流量与投料阀开度之间的关系。

在实验数据处理中，经常会遇到这样的问题，即已经知道两个或三个变量之间存在着某种函数关系，但是不能从理论上推出方程的形式，而是要建立一个经验方程来表达它们之间的关系。把这类待定经验公式的诸实验点画出散点图，大多数不是直线，而是各种各样的曲线，而曲线拟合的任务就是恰当的选择一个经验方程式来较好的拟合实验数据。

经验公式的确定方法有很多种，比如平均值法，差分差商法，最小二乘法，以及更一般的多项式拟合，这些方法在文献[27-28]中均有详细的论述。

将生产过程历史数据中的 30 组（即在 4.1 节中选出进行预处理的数据）投料阀开度、计量罐液位和一甲胺流量组成一个查询表格，其格式如表 4.16 所示。

表 4.16 与液位、开度对应的流量表

Tab.4.16 Flux data correspond to level and valve

开度	液位

把由  $(L_{jig}(k), V_{df}(k))$  所确定的第  $k$  时刻的流量  $F_{yfa}(k)$  填入表格相应的位置，并

在坐标图中标出该点，如果在第  $k$  时刻同时有  $n$  个流量  $F_{yfa}(k)_i, i=1, \dots, n$ ，那么将这  $n$  个流量取算术平均作为第  $k$  时刻的流量，并将该值也标在坐标图中。依次将这些数据均在坐标图中标出，利用上面介绍的曲线拟合的方法，来作出近似的映射关系  $f$ 。

在进行曲线拟合时，如果全部都用人来完成是非常复杂的，本文采用了曲线拟合软件 TableCurve 来完成这些工作。该软件基本包括了所有的曲线拟合方法。

图 4.21 是把按照上面方法描出的流量点，用 TableCurve 软件绘制的流量曲线。在图中淡蓝色的点是曲线的主要分布点，黄色和红色的点被认为是错误点，所以

在绘制曲线时黄色点和红色点并未影响到曲线的绘制。同时利用 TableCurve 软件还给出了一些曲线拟合的经验方程式。如切比雪夫两变量 10 阶多项式拟合经验方程是精度比较高的一个经验方程，表 4.17 是该经验方程的拟合数据，从数据表中可以看到预测出的阀门开度比较接近实际开度，误差较小。

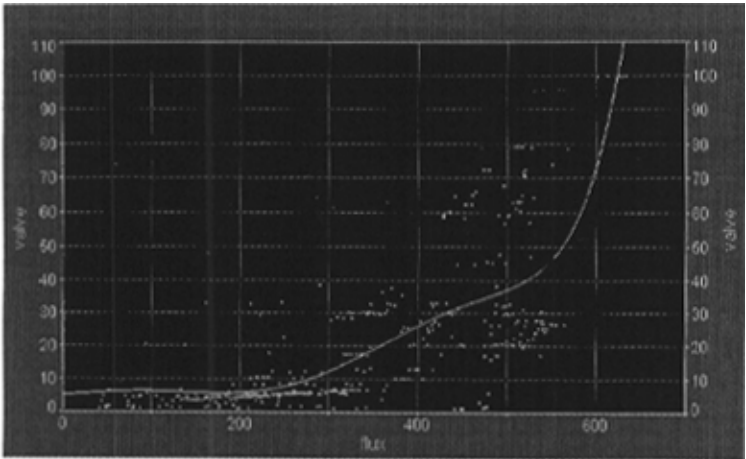


图 4.21 流量曲线图

Fig.4.21 Flux curve

表 4.17 切比雪夫两变量 10 阶多项式拟合数据

Tab.4.17 Chebyshev X,Y Bivariate Polynomial Order 10 matching data

XYZ	X(Level)	Y(Flux)	Z(Valve)	Z Predict
1	537.0869	495.3578	68.0097	68.717771
2	535.1193	472.4921	72.458	75.486936
3	527.2056	276.2939	70.339	66.736121
4	524.4053	478.8278	72.458	68.269631
5	522.7424	0	1.964	2.2350864
6	522.7424	0	1.964	2.2350864
7	522.5954	50.8939	1.964	1.028964
8	521.9404	192.7878	1.964	3.0762096
9	519.0485	205.4676	5.5227	6.4734706
10	517.6884	434.707	60.213	54.623159
...	...	...	...	...

## 5 基于遗传算法的 T-S 模糊模型氧乐果间歇过程 系统辨识

T-S 模糊模型可以任意精度逼近定义在紧集上的非线性函数，因而引起了人们的广泛关注。T-S 模糊模型可看成近似分段线性模型。该模型相当于将输入空间分为若干个模糊子空间，首先在每个模糊子空间建立一个局部线性模型，然后使用隶属度函数平滑地将各个局部模型连接起来，从而形成非线性函数的全局模糊模型。

### 5.1 广义 T-S 模糊模型

T-S 模糊模型是一种本质非线性模型，宜于表达复杂系统的动态特性。该模糊模型的辨识算法通常包括前提结构的辨识、前提参数的辨识、结论结构的辨识和结论参数的辨识。在第三章已经提到了 T-S 模糊模型的特点，虽然 T-S 系统模型算法简单，易于工程化，但该模型中具有以下一些需要进一步研究的问题：

- ① 后件参数  $P_i^l (l=1,2,\dots,M; i=1,2,\dots,n)$  的确定以及是否是最优；
- ② 隶属度函数  $\mu_{F_i^l}(x_i) (l=1,2,\dots,M; i=1,2,\dots,n)$  的确定以及是否具有自适应功能；
- ③ T-S 模型中模糊规则数  $M$  的最佳确定，它关系到模型的复杂程度。

这些参数及隶属函数的确定将直接影响 T-S 模型是否具有较好的自适应性去最佳匹配给定的实际模型，是否具有任意精度的有效逼近性和较低的模型复杂度。因此，引入了一类广义 Takagi-Sugeno 模糊逻辑系统(以下简称广义 T-S 系统)。

在模糊系统的工程应用中，模糊集合的隶属函数通常取三角型、梯型、高斯型或其它指数型，这些隶属函数的特点是一旦给定为某种类型就不能改变其大致形状，若存在一种参数化隶属函数，通过改变参数便能自适应地逼近以上所提到的各种隶属函数，这对于模糊系统将具有重要意义。下面将引入这样的隶属函数和相应的模糊系统。

定义 1<sup>[11]</sup>：函数  $\mu_F(x)$  称为一类广义高斯隶属函数，若  $\mu_F(x)$  具有如下表达式：



$$\mu_F(x) = \exp\left[-\left|\frac{x-\beta}{\alpha}\right|^\gamma\right], \alpha > 0, \beta \in R, \gamma \geq 0 \quad (5-1)$$

若取  $\alpha=1, \beta=0$ , 并选择合适的  $\gamma$  值, 广义高斯隶属度函数族  $\mu_F(x) = \exp(-|x|^\gamma)$  就可分别近似于三角形、梯形、高斯型等隶属度函数; 若进一步改变参数  $\alpha, \beta$  的值, 还可以对隶属度函数进行平移、压缩或扩展, 更能逼近三角形、梯形、高斯型等其他隶属度函数, 因而广义高斯隶属度函数具有自适应性。

定义 2<sup>[11]</sup>: 若 T-S 系统中隶属度函数  $\mu_{F_i}(x_i)$  均采用广义高斯隶属度函数族形式, 即

$$\mu_{F_i}(x) = \exp\left[-\left|\frac{x-\beta_i'}{\alpha_i'}\right|^{\gamma_i'}\right], \alpha_i' > 0, \beta_i' \in R, \gamma_i' \geq 0 \quad (5-2)$$

$(l=1, 2, \dots, M; i=1, 2, \dots, n)$

这样的模糊逻辑系统称为广义 T-S 模糊逻辑系统。由(3-12)式和(5-2)式可导出具有  $n$  输入单输出, 模糊规则数为  $M$  的广义 T-S 系统的输出为,

$$y = \frac{\sum_{l=1}^M y^l \prod_{i=1}^n \mu_{F_i}(x_i)}{\sum_{l=1}^M \prod_{i=1}^n \mu_{F_i}(x_i)} = \frac{\sum_{l=1}^M \left[ P_0^l + \sum_{k=1}^n P_k^l x_k \right] \prod_{i=1}^n \exp\left[-\left|\frac{x-\beta_i'}{\alpha_i'}\right|^{\gamma_i'}\right]}{\sum_{l=1}^M \prod_{i=1}^n \exp\left[-\left|\frac{x-\beta_i'}{\alpha_i'}\right|^{\gamma_i'}\right]} \quad (5-3)$$

广义 T-S 系统能进行实际建模的充分必要条件是: 该系统能以任意精度逼近实际模型, 也即是该系统能否具有全局逼近性。根据 Stone-Weirstrass 定理<sup>[19]</sup>, 可对广义 T-S 模糊系统的全局逼近性理论给出分析证明。限于篇幅, 仅给出如下定理, 证明请参看文献[11]。

定理 1: 广义 T-S 模糊逻辑系统具有全局逼近性。

由定理 1 可以知道, 利用广义 T-S 模糊模型可以实现氧乐果合成间歇生产过程温度对象的模型辨识。

## 5.2 遗传算法

引入广义 T-S 模糊模型, 目的就是为了使隶属度函数具有自适应性。具有自适应性的参数, 也能够减小在确定模糊隶属度函数时人为主观性因素的影响。遗传算法是一种全局寻优的进化算法, 如果把广义 T-S 模糊模型的参数进行简单有效的

编码, 那么利用遗传算法就可以寻找到一组全局最优或者全局次优的参数, 从而可以找到一个最优或者次优的被控对象模型。

GA 是一个迭代过程, 首先随机形成初始种群, 种群规模固定为  $N$ , 接着进行迭代, 每次迭代中都保留一组候选解, 形成一个新种群, 按照解的优劣进行排序, 并按照某种指标从中选出一些解, 利用遗传算子进行操作, 产生新一代的候选解, 重复此过程, 直到满足某种收敛指标, 求出最优解或者次优解为止。

简单遗传算法 (SGA) 包括三个基本算子, 即选择、交叉和变异。GA 的工作对象是编码字符串, 因此在进行遗传操作之前应该确定合适的编码方式, 以及合适的适应度函数。

### 5.2.1 编码方式

在遗传算法的运行过程中, 它不对所求解问题的实际决策变量直接进行操作, 而是对表示可行解的个体编码施加选择、交叉、变异等遗传运算, 通过这种遗传操作来达到优化的目的, 这是遗传算法的特点之一。遗传算法通过这种对个体编码的操作, 不断搜索出适应度较高的个体, 并在群体中逐渐增加其数量, 最终寻找出问题的最优解或近似最优解。在遗传算法中如何描述问题的可行解, 即把一个问题的可行解从其解空间转换到遗传算法所能处理的搜索空间的转换方法就称为编码。

编码是应用遗传算法时要解决的首要问题, 也是设计遗传算法时的一个关键步骤。编码方法除了决定了个体的染色体排列形式之外, 它还决定了个体从搜索空间的基因型变换到解空间的表现型时的解码方法, 编码方法也影响到交叉算子、变异算子等遗传算子的运算方法。由此可见, 编码方法在很大程度上决定了如何进行群体的遗传进化运算以及遗传进化运算的效率。一个好的编码方法有可能会使得交叉运算、变异运算等遗传操作可以简单地实现和执行。而一个差的编码方法, 却有可能会使得交叉运算、变异运算等遗传操作难以实现, 也有可能产生很多在可行解集合内无对应可行解的个体, 这些个体经解码处理后所表示的解称为无效解。虽然有时产生一些无效解并不全部是有害的, 但大部分情况下它却是影响遗传算法运行效率的主要因素之一。

由于遗传算法应用的广泛性, 迄今为止人们已经提出了许多种不同的编码方法。总的来说, 这些编码方法可以分为三大类: 二进制编码方法、浮点数编码方法、符号编码方法。下面从具体实现的角度出发介绍其中的几种主要编码方法。

#### ① 二进制编码

二进制编码方法是遗传算法中最常用的一种编码方法, 它使用的编码符号集是由二进制符号 0 和 1 所组成的二值符号集  $\{0,1\}$ , 它所构成的个体基因型是一个

$$\begin{array}{lll} 000000 \cdots 000000 = 0 & \rightarrow & U_{min} \\ 000000 \cdots 000001 = 1 & \rightarrow & U_{min} + \delta \\ \vdots & & \vdots \\ 111111 \cdots 111111 = 2^l - 1 & \rightarrow & U_{max} \end{array}$$

假设某一个体的编码是:  $X:b_1b_{i-1}b_{i-2}\cdots b_2b_1$ , 则对应的解码公式为:

$$x = U_{min} + \left( \sum_{i=1}^l b_i \cdot 2^{i-1} \right) \cdot \frac{U_{max} - U_{min}}{2^l - 1}$$

## ② 浮点数编码（实数编码）

首先是二进制编码存在着连续函数离散化时的映射误差。个体编码串的长度较短时,可能达不到精度要求;而个体编码串的长度较长时,虽然能提高编码精度,但却会使遗传算法的搜索空间急剧扩大。其次是二进制编码不便于反映所求问题的特定知识,这样也就不便于开发针对问题专门知识的遗传运算算子,人们在一些经典优化算法的研究中所总结出的一些宝贵经验也就无法在这里加以利用,也不便于处理非平凡约束条件。

在浮点数编码方法中，必须保证基因值在给定的区间限制范围内，遗传算法中所使用的交叉、变异等遗传算子也必须保证其运算结果所产生的新个体的基因值也在这个区间限制范围内。再者，当用多个字节来表示一个基因值时，交叉运算必须在两个基因的分界字节处进行，而不能在某个基因的中间字节分隔处进行。

- 49 -

计算复杂性，提高了运算效率；便于遗传算法与经典优化方法的混合使用；便于设计针对问题的专门知识的知识型遗传算子；便于处理复杂的决策变量约束条件。

其他还有一些编码方法，如格雷码法，符号编码法等等，具体的可以参阅文献[13-15], [23-24]。

## 5.2.2 适应度函数

遗传算法的目标函数不受连续可微的约束且定义域可以为任意集合。对目标函数的唯一要求是，针对输入可计算出能加以比较的非负结果。GA 利用种群中每个个体的适应度函数值，进行最优搜索，因此适应度函数的选取至关重要。一般而言，适应度函数是由目标函数变换而来的。

常见的适应度函数基本上有以下三种：

① 直接以待求的目标函数转化为适应度函数，即

若目标函数为最大化问题  $Fit(f(x)) = f(x)$

若目标函数为最小化问题  $Fit(f(x)) = -f(x)$

这种适应度函数简单直观，但存在两个问题，其一是可能不能满足常用的轮盘赌选择中概率非负的要求；其二是某些待求解的函数在函数值分布上相差很大，由此得到的平均适应度可能不利于体现种群的平均性能，影响算法的性能。

② 若目标函数为最小问题，则

$$Fit(f(x)) = \begin{cases} c_{max} - f(x), & f(x) < c_{max} \\ 0, & \text{其他} \end{cases}, \text{ 式中 } c_{max} \text{ 为 } f(x) \text{ 的最大值估计。}$$

若目标函数为最大问题，则

$$Fit(f(x)) = \begin{cases} f(x) - c_{min}, & f(x) > c_{min} \\ 0, & \text{其他} \end{cases}, \text{ 式中 } c_{min} \text{ 为 } f(x) \text{ 的最小值估计。}$$

这种方法是对第一种方法的改进，可以称为“界限构造法”，但有时存在界限值预先估计困难，不可能精确的问题。

③ 若目标函数为最小问题，则

$$Fit(f(x)) = \frac{1}{1 + c + f(x)} \quad c \geq 0, c + f(x) \geq 0$$

若目标函数为最大问题，则

$$Fit(f(x)) = \frac{1}{1 + c - f(x)} \quad c \geq 0, c - f(x) \geq 0$$

这种方法与第二种方法类似， $c$  为目标函数界限的保守估计值。

适应度函数是个体能否进入下一代的唯一标准，直接决定群体的进化行为，地位非常重要，并且为了能够将适应度函数与群体中个体优劣度相联系，在遗传



算法中适应值规定为非负，并且在任何情况下都是越大越好。适应度函数的标定是指在遗传算法不同的运行阶段，对个体适应度进行适当的扩大或缩小。它有双重意义，一是使染色体之间的适应度值保持合理的差距，二是阻止某些超级染色体太快的主导遗传过程，以满足早期限制竞争、晚期鼓励竞争的需要。

常用的适应度函数标定方法有：

### ① 线性标定法

假设原适应度函数为  $f$ ，标定后的适应度函数为  $f'$ ，则线性标定可用下式表示： $f' = \alpha \times f + \beta$ 。式中的系数确定的方法很多，但要满足两个条件，一是原适应度的平均值要等于标定后的适应度平均值，以保证适应度为平均值的个体在下一代的期望复制数为 1，即  $f'_{avg} = f_{avg}$ ；二是标定后的适应度最大值应等于原适应度

平均值的指定倍数，以控制适应度最大的个体在下一代中的复制数，即  $f'_{max} = c_{mult} f_{avg}$ ，

指定倍数  $c_{mult}$  一般取 1.0~2.0。根据上述条件可以计算出线性标定的系数：

$$\alpha = \frac{(c_{mult} - 1)f_{avg}}{f_{max} - f_{avg}}, \quad \beta = \frac{(f_{max} - c_{mult}f_{avg})f_{avg}}{f_{max} - f_{avg}}$$

线性标定法变换了适应度之间的差距，保持了种群内的多样性，并且计算简便，易于实现。如果种群内某些个体适应度远远低于平均值时，有可能出现标定后适应度值为负的情况，为此，考虑到保证最小适应度值非负的条件，进行如下的变换，

$$\alpha = \frac{f_{avg}}{f_{max} - f_{avg}}, \quad \beta = \frac{-f_{min}f_{avg}}{f_{max} - f_{avg}}$$

### ② 幂函数标定法

标定公式为： $f' = f^k$ 。幂指数  $k$  与所求的最优化问题有关，并且在算法的执行过程需要不断对其进行修正才能扩大或缩小到需要的范围。

### ③ 指数函数标定法

指数函数标定是一种用于正比选择的标定方法，公式如下： $f' = e^{-\alpha f}$ 。这种标定方法的基本思想来源于模拟退火过程，其中的系数  $\alpha$  决定了复制的强制性，其值越小，复制的强制就越趋向于那些具有最大适应度的个体。

以上是一些常用的标定方法，此外其它标定方法有对数标定，窗口技术，正规化，排序，动态线性标定等，可参阅文献[23]。

## 5.2.3 选择算子

从群体中选择优胜的个体，淘汰劣质个体的操作叫选择，选择算子又称作复



制算子。选择的目的是把优化的个体（或解）直接遗传到下一代或通过配对交叉产生新的个体再遗传到下一代。选择操作建立在对个体的适应度进行评价的基础之上。选择操作的主要目的是为了减少基因缺失、提高全局收敛性和计算效率。

最常用的选择算子是基本遗传算法中的比例选择算子。但对于各种不同的问题，比例选择算子并不是最合适的一种选择算子，所以又出现了其他一些选择算子。

### ① 适应度比例方法

适应度比例方法也叫做轮盘赌或蒙特卡罗选择。在该方法中，各个个体被选中的概率与其适应度大小成正比。由于是随机操作的原因，这种选择方法的选择误差比较大，有时甚至连适应度较高的个体也选择不上。

设群体的大小为  $M$ ，个体  $i$  的适应度为  $f_i$ ，则个体  $i$  被选中的概率  $p_{is}$  为

$$p_{is} = f_i / \sum_{i=1}^M f_i$$

由上式可见，适应度越高的个体被选中的概率也越大；反之，适应度越低的个体被选中的概率也越小。

### ② 排序选择方法

所谓排序选择方法是指在计算每个个体的适应度后，根据适应度大小顺序对群体中个体排序，然后把事先设计好的概率表按序分配给个体，作为各自的选择概率。在基于排序的适应度分配中，种群按目标值进行排序。适应度仅仅取决于个体在种群中的序位，而不是实际的目标值。排序方法克服了比例适应度计算的尺度问题，当选择压力<sup>[13-15][23-24]</sup>太小的情况下，以及选择导致搜索带迅速变窄而产生的过早收敛。因此，再生范围被局限。排序方法引入种群均匀尺度，提供了控制选择压力的简单有效的方法。

排序方法比比例方法表现出更好的鲁棒性，因此，不失为一种好的选择方法。

设定  $N$  为种群大小， $Pos$  为个体在种群中的序位， $SP$  为选择压力，个体的适应度可以计算如下：

$$\text{线性排序: } Fit(Pos) = 2 - SP + \frac{2(SP-1)(Pos-1)}{N-1}, SP \in [1, 2]$$

$$\text{非线性排序: } Fit(Pos) = \frac{NX^{Pos-1}}{\sum_{i=1}^N X^{i-1}}$$

其中  $X$  是下列多项式方程的根：

$$(SP-1) \cdot X^{N-1} + SP \cdot X^{N-2} + \dots + SP \cdot X + SP = 0, SP \in [1, N-2]$$

$$\text{选择强度: } SelInt_{Rank} = \frac{SP-1}{\sqrt{\pi}}, \text{ 多样化损失: } LossDiv_{Rank} = (SP-1)/4$$

选择方差:  $SelVar_{Rank}(SP) = 1 - ((SP - 1)^2 / \pi) = 1 - SelInt_{Rank} SP^2$

除了以上几种选择算子之外, 其他的一些选择算子可以参阅文献 [13-15], [23-24]。

#### 5.2.4 交叉算子/重组算子

遗传算法中的交叉运算, 是指对两个相互配对的染色体按某种方式相互交换其部分基因, 从而形成两个新的个体。交叉运算是遗传算法区别于其他进化算法的重要特征, 它在遗传算法中起着关键作用, 是产生新个体的主要方法。在对实数编码的个体进行交叉的时候, 交叉算子又称作重组算子。

最常用的交叉算子是单点交叉算子。但单点交叉操作有一定的适用范围, 故又出现了其他一些交叉算子。下面介绍几种适用于二进制编码个体或浮点数编码个体的交叉算子。

##### ① 单点交叉

单点交叉 (又称为简单交叉), 它是指在个体编码串中只随机设置一个交叉点, 然后在该点相互交换两个配对个体的部分染色体。

$A: 1001:111 \rightarrow A': 1001:000$

$B: 0011:000 \rightarrow B': 0011:111$

交叉点

##### ② 多点交叉

多点交叉又称作广义交叉, 在个体编码串中随机设计多个交叉点, 然后进行基因交换, 其交叉点数和位置有多种选择方法。多点交叉有可能破坏一些好的模式。二点交叉就是多点交叉的一个特殊情况。

$A: 10: \underline{110}: 11 \rightarrow A': 1001011$   
交叉点1,2

$B: 00: \underline{010}: 00 \rightarrow B': 0011000$   
交叉点1,2

##### ③ 均匀交叉

均匀交叉是指通过设定屏蔽字来决定新个体的基因继承两个旧个体中哪个个体的对应基因。均匀交叉的操作过程表示如下: 当屏蔽字中的位为 0 时, 新个体  $A'$  继承旧个体  $A$  中对应的基因, 当屏蔽字位为 1 时, 新个体  $A'$  继承旧个体  $B$  中对应的基因, 由此生成一个完整的新个体  $A'$ , 同样可生成新个体  $B'$ 。均匀交叉也是包括在多点交叉范围内的。

$A: xxxxxxxxxx \xrightarrow[\text{mask}=01010101]{\text{均匀交叉}} A': xyxyxyxyxy$

$B: yyyyyyyyyyyy \xrightarrow[\text{mask}=01010101]{\text{均匀交叉}} B': yxyxyxyxyx$

##### ④ 中间重组

中间重组只能适用于实变量而非二进制变量。子个体的产生按下列公式：

$$\text{子个体} = \text{父个体 1} + \alpha \times (\text{父个体 2} - \text{父个体 1})$$

这里  $\alpha$  是一个比例因子，可由  $[-d, 1+d]$  上均匀分布随机数产生。对于中间重组  $d=0$ ；一般选择  $d=0.25$ 。子代的每个变量的值按上面的表达式计算，对每个变量要选择一个新的  $\alpha$  值。考虑含有三个变量的两个个体：

父个体 1	12	25	5
父个体 2	123	4	34

$\alpha$  的样本为：

样本 1	0.5	1.1	-0.1
样本 2	0.1	0.8	0.5

计算出的新个体为：

子个体 1	67.5	1.9	2.1
子个体 2	23.1	8.2	19.5

#### ⑤ 线性重组

线性重组也只能适用于实变量而非二进制变量。其子代计算公式与中间重组类似，所不同的是  $\alpha$  是一个固定值。还以④中的父个体为例， $\alpha$  值的样本为：

样本 1	0.5
样本 2	0.1

计算出的新个体为：

子个体 1	67.5	14.5	19.5
子个体 2	23.1	22.9	7.9

其他形式的交叉算子/重组算子可以参阅文献[13-15]，[23-24]。

### 5.2.5 变异算子

变异操作模拟自然界生物体进化中染色体上某位基因发生突变的现象，从而改变染色体的结构和物理性状。遗传算法中的所谓变异运算，是指将个体染色体编码串中的某些基因座上的基因值用该基因座的其他等位基因来替换，从而形成一个新的个体。

从遗传运算过程中产生新个体的能力方面来说，交叉运算是产生新个体的主要方法，它决定了遗传算法的全局搜索能力；而变异运算只是产生新个体的辅助方法，但它也是必不可少的一个运算步骤，因为它决定了遗传算法的局部搜索能力。交叉算子与变异算子的相互配合，共同完成对搜索空间的全局搜索和局部搜索，从而使得遗传算法能够以良好的搜索性能完成最优化问题的寻优过程。变异算子不仅可以改善遗传算法的局部搜索能力，而且也能够维持群体的多样性，防

止出现早熟现象。

常用的变异算子有：

### ① 基本变异算子

基本变异算子是指对群体中的个体码串随机挑选一个或多个基因座并对这些基因座的基因值以变异概率  $P_m$  做变动。 $\{0, 1\}$  二进制编码串中的基本变异操作为，

$$A:1011011 \rightarrow A':1110011$$

### ② 逆转算子

逆转算子是变异算子的一种特殊形式。它的基本操作内容是，在个体码串中随机挑选两个逆转点，然后将两个逆转点之间的基因值以逆转概率  $P_i$  逆向排序。 $\{0, 1\}$  二进制编码串中的逆转变异操作为，

$$A:10 \underbrace{11010}_{\text{逆转点}} 00 \rightarrow A':10 01011 00$$

逆转操作的真正目的并不在变异而在实现一种重新排序操作，这种重新排序是对个体中基因排列所进行重新组合，但并不影响该个体的特征。在自然界中的基因重组就有这种重新排列的机制。对 GA 而言，采用这种重新排列的目的是为了提高积木块的繁殖率。

### ③ 自适应变异算子

该算子与基本变异算子的操作内容类似，唯一不同的是变异概率  $P_m$  不是固定不变而是随群体中个体的多样性程度而自适应调整的。一般是根据交叉所得到的两个新个体的海明距离进行变化，海明距离越小， $P_m$  越大，反之  $P_m$  越小。具体的，

$$P_m = \begin{cases} \frac{k_3(f_{\max} - f)}{f_{\max} - f_{\text{avg}}}, & f \geq f_{\text{avg}} \\ k_4, & f < f_{\text{avg}} \end{cases} \quad k_3, k_4 \in (0, 1)$$

在文献[15]，[23-24]中还提到其他一些变异算子和具体的表达方式。

## 5.3 氧乐果合成过程辨识实例

在 3.3.2 节中提到 T-S 模糊模型是利用多个线性表达式来逼近一个非线性的对象；在 5.1 节中也分析了 T-S 模糊模型的一些不足之处，并且也给出了利用广义 T-S 模糊模型的解决方法。本节就利用广义 T-S 模糊模型，结合遗传算法来实现氧乐果合成间歇生产过程中的反应釜温度模型进行辨识。辨识的数据仍采用 4.3 节中的数据，输入矩阵中的四个变量和输出温度均分成{小，中，大}三个等级，这样  $n=4, M=81$ 。由于冷却盐水温度基本上是没有变化的，为了编码方便，将冷却盐水温度作为一个常量忽略，因此  $n=3, M=27$ 。



从式 (5-3) 中可以看到要确定的参数有  $\alpha, \beta, \gamma$  和  $C$  共有  $(4n+1)M$  个变量有待确定, 因此在编码时就要采用合理的方法。

1. 个体编码: 由于  $n, M$  较大, 所以独立变量个数非常大, 采用常规二进制编码方式, 则一个完整的染色体编码就会过长, 将导致遗传算法搜索过程时间较长, 所以本文综合文献[26]中提到的编码方式, 将参数进行实数编码, 如式 (5-4) 所示。其中第一部分为规则 1 的编码, 依次类推。

$$\begin{array}{ccccccc} \alpha_1^1 & \cdots & \alpha_n^1 & \vdots & \alpha_1^2 & \cdots & \alpha_n^2 \cdots \alpha_1^M \cdots \alpha_n^M \\ \beta_1^1 & \cdots & \beta_n^1 & \vdots & \beta_1^2 & \cdots & \beta_n^2 \cdots \beta_1^M \cdots \beta_n^M \\ \gamma_1^1 & \cdots & \gamma_n^1 & \vdots & \gamma_1^2 & \cdots & \gamma_n^2 \cdots \gamma_1^M \cdots \gamma_n^M \\ C_0^1 & C_1^1 & \cdots & C_n^1 & \vdots & C_0^2 & C_1^2 \cdots C_n^2 \cdots C_0^M & C_1^M \cdots C_n^M \end{array} \quad (5-4)$$

2. 初始种群: 群体规模是一个重要的参数, 直接影响遗传优化的最终结果以及遗传算法的执行效率<sup>[14]</sup>。在此群体规模取为 30, 采用循环递归算法随机产生初始群个体。

3. 适应度函数: 适应度函数取误差平方和的倒数, 即 (4-8) 式的倒数

$$Fit(i) = \frac{1}{E(i)} \quad (5-5)$$

4. 选择: 采用在 5.2 节中提到的排序选择方法, 方式为非线性排序。

5. 交叉: 采用中间重组的方法, 即从当前代中随机选取两个广义 T-S 模型个体作为父代, 从两个父代上随机选取两个相同位置相互交换, 这样就形成了新一代。

6. 变异: 对于实数编码  $x_k$ , 在满足变异概率  $p_k$  的前提下, 变异按照下面的方式进行:

$$\tilde{x}_k = x_k \pm p_k \times range \times \eta \times delta, \quad delta = \sum_{i=0}^{m-1} \alpha_i 2^{-i}, m=20; \alpha_i = \begin{cases} 1, & p_k / m \\ 0, & else \end{cases} \quad (5-6)$$

其中  $x_k$  表示的为  $\alpha_i^l, \beta_i^l, \gamma_i^l, C_i^l$ ,  $\tilde{x}_k$  是  $x_k$  相应的变异,  $range$  为  $\alpha_i^l, \beta_i^l, \gamma_i^l, C_i^l$  变化范围的一半。

7. 结束条件: 当迭代次数超过给定代数或者最优个体误差低于给定值时, 结束算法。

整个遗传算法的过程如图 5.1 所示。

在初始化阶段, 给定了所有参数的变化范围, 其中  $\alpha_i^l, \beta_i^l, \gamma_i^l$  的变化范围为  $[0.5, 3]$ ,  $C_i^l$  的变化范围为  $[-1.0, 1.0]$ ; 变异概率为 0.01。循环代数设定为 2000 代。另外考虑到当新一代群体产生后, 如果说不是所有的新个体在每一代中都使用, 或者说如果个体数多于父代个体数 (两代间群体规模之差称作“代沟”), 那么必须采用恢复算子来确定在下一代中哪个个体可以存在。所以在程序中使用了恢复



算子来避免这种情况的出现。

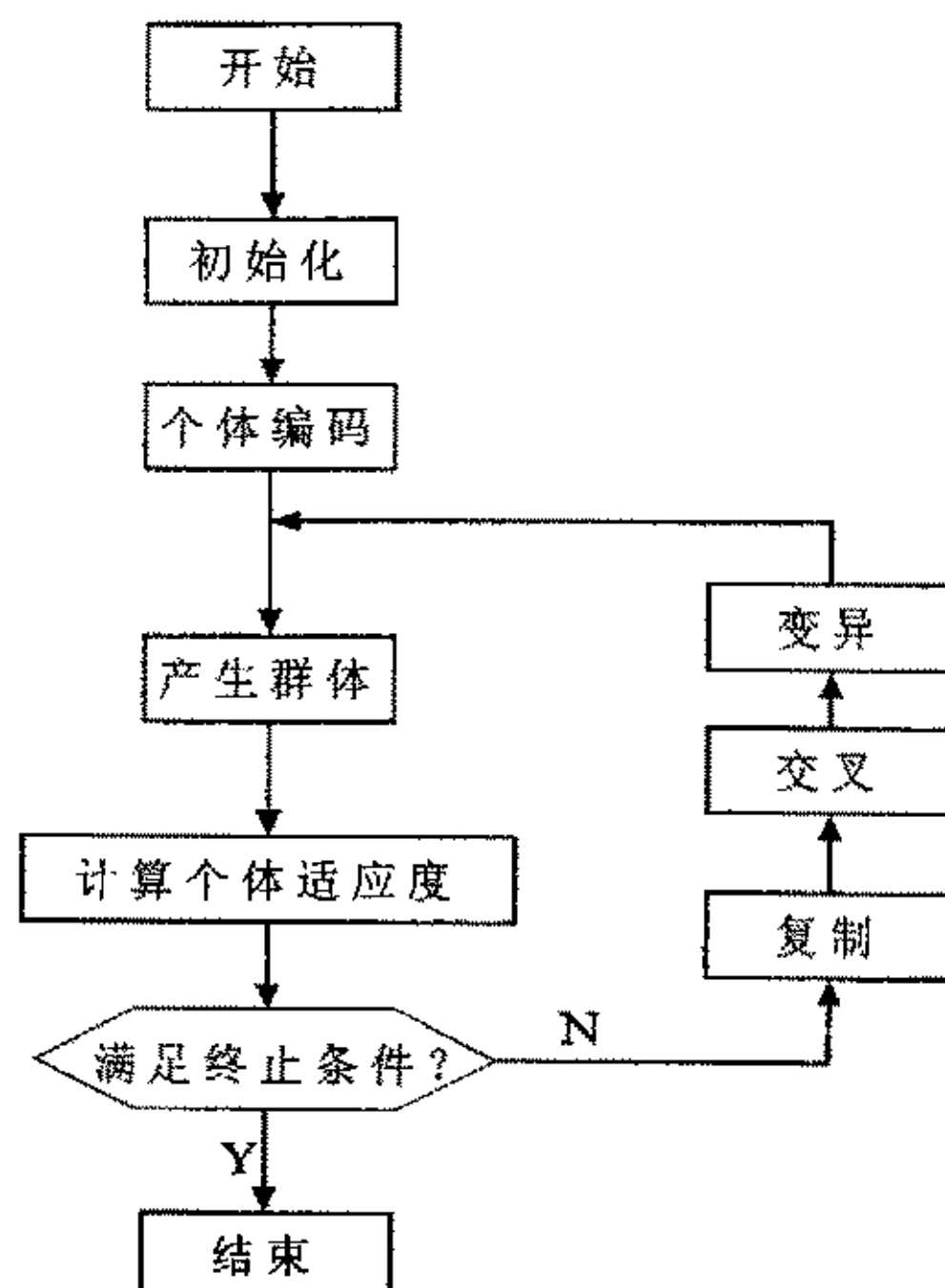


图 5.1 遗传算法的过程

Fig.5.1 Procedure of GA

图 5.2 给出的是遗传算法的进化曲线和数据拟合曲线。拟合所采用的数据是参数寻优进化中所使用的数据，所以误差比较小。表 5.1 与表 5.2 是循环结束后，最后一代群体中的最佳个体所代表的模糊逻辑系统模型的参数。为了验证模型的泛化能力，采用另外一组数据来验证，这组数据仍采用第四章中所使用的验证数据。模糊输出采用式 (5-3) 来计算。

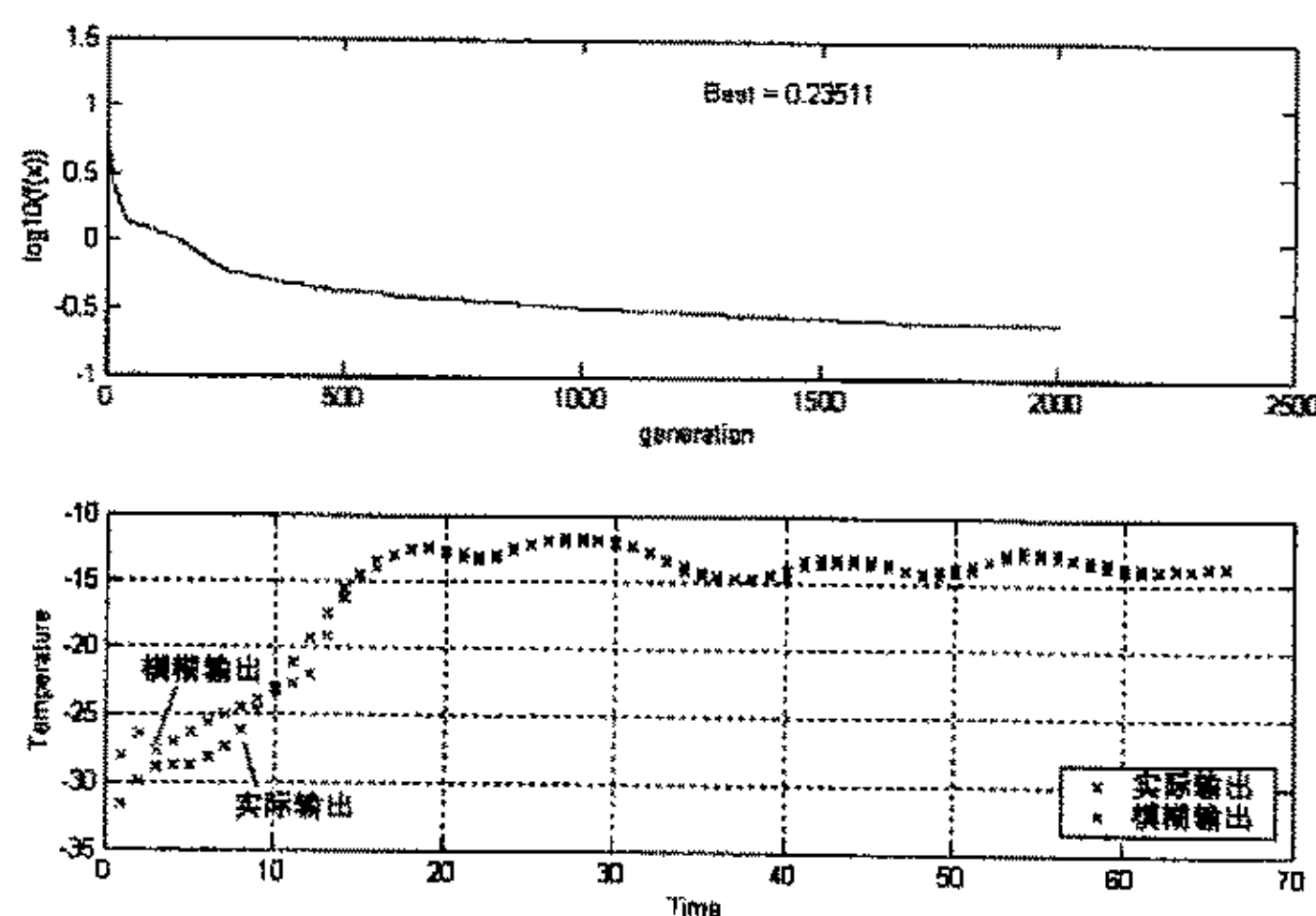


图 5.2 遗传算法进化曲线与拟合曲线

Fig.5.2 Curve of GA evolution and matching

表 5.1  $\alpha, \beta, \gamma$  参数表

Tab.5.1 Parameter:  $\alpha, \beta, \gamma$

$\alpha_1^i$	$\beta_1^i$	$\gamma_1^i$	$\alpha_2^i$	$\beta_2^i$	$\gamma_2^i$	$\alpha_3^i$	$\beta_3^i$	$\gamma_3^i$
0.8069	0.5003	1.6392	1.4158	1.1377	1.2934	1.9036	0.8985	2.4811
2.1595	1.3677	1.3944	2.9274	0.5625	1.6358	0.5859	2.9965	2.9996
2.255	1.6169	0.5249	0.7371	2.6069	1.3686	2.5229	2.9989	1.5199
1.7589	1.4964	2.6059	0.5015	2.9803	1.8943	1.239	2.0682	1.3127
2.4974	1.3099	2.159	0.5126	2.9255	2.9503	2.8646	0.9783	2.6998
1.5583	0.945	2.9865	1.3167	0.6908	2.9841	1.6995	1.401	1.8237
1.6598	1.7901	2.2977	2.9959	1.5669	1.2689	2.5303	1.9545	0.5851
2.855	1.4217	1.9824	2.0278	1.6528	0.5654	1.1673	1.7693	1.2902
2.3291	0.9096	2.2347	2.9425	1.4972	2.3308	2.9919	0.8352	2.1478
1.8925	1.7608	1.6672	1.3472	1.4436	1.7016	1.747	0.6341	1.0151
0.962	1.5596	0.6892	0.564	0.5002	3	2.7999	1.0757	2.1768
2.4837	0.6508	1.6031	1.6003	1.7491	1.7521	2.4941	0.5748	1.8293
1.9916	1.7663	1.9305	2.1796	0.7916	2.1289	2.7309	1.8789	2.2674
1.7259	1.7191	2.1506	2.2136	1.9128	2.0262	2.0474	2.2917	1.764
2.4768	1.2601	2.9803	2.6888	0.6178	1.0823	2.0013	1.5308	2.3535
1.6702	2.8541	1.2576	2.8664	2.0278	2.6146	2.0473	1.8561	2.2587
1.6535	1.3931	0.5024	2.9178	0.8364	2.2344	2.4544	0.7518	2.2313
1.7932	1.1784	0.5067	1.6373	3	1.7884	2.8275	1.4697	2.4283
2.7236	1.48	2.8843	1.263	2.092	1.6061	2.2956	1.3044	1.7895
2.1857	0.6233	1.7254	1.973	1.6866	2.9927	1.6544	1.0269	1.8913
2.6295	0.9662	1.8852	0.9499	0.8909	1.865	2.1654	1.0819	2.6184
2.1047	1.9435	1.3933	2.5612	2.709	1.5189	2.3257	0.9934	1.4922
1.2475	0.7797	2.6827	2.922	0.929	2.0414	2.2217	1.8544	2.0982
2.5736	1.7537	2.9531	1.6068	0.698	1.6627	1.9413	0.9258	2.6604
2.4977	1.2507	3	2.004	2.1075	2.9687	0.8591	0.5156	2.9224
1.2938	1.8089	1.1309	1.9035	1.7091	2.9945	2.2294	0.9791	1.4587
2.4412	0.805	1.6398	2.0213	0.9081	2.3745	1.6903	0.8417	2.8608

表 5.2  $C$  参数表

Tab.5.2 Parameter:  $C$

	$C_0^i$	$C_1^i$	$C_2^i$	$C_3^i$
1	-0.9639	-0.1834	0.0258	0.0901
2	-0.2711	0.9704	-0.5591	0.3075
3	0.9846	-0.5616	0.5319	0.7528
4	0.8655	0.3622	0.7848	0.3056
5	-0.9878	0.4031	0.2736	0.0172
6	-0.5289	0.8202	0.3529	0.2035
7	0.2431	-0.7167	0.8534	0.2873
8	0.3846	-0.2804	-0.2216	0.3683
9	-0.6408	0.1009	0.9991	-0.2403
10	-0.4631	-0.5046	0.9988	-0.5896
11	-0.6671	-0.3436	0.5937	-0.9633
12	-0.3608	0.0765	-0.207	-0.5369
13	1	-0.9136	-0.3668	0.6222
14	-0.7577	-0.0945	-0.0221	0.0013
15	-0.0317	-0.0105	-0.0768	0.566
16	-0.8837	0.4339	-0.9503	-0.8885
17	0.1862	0.3279	-0.2847	-0.0518
18	-0.9976	0.3667	-0.8028	-0.9953
19	-0.2279	0.1015	-0.0655	0.3018
20	-0.4442	0.9955	-0.2117	-0.3652
21	0.0561	0.9982	0.9957	-0.6335
22	0.7804	1	0.9965	-0.6366
23	0.9994	-0.3796	-0.0909	-0.4159
24	0.9989	0.7362	0.1272	0.3852
25	0.3632	0.9138	0.9574	0.0942
26	0.0119	0.8668	-0.1371	-0.8043
27	-0.3061	0.2759	0.575	-0.3373

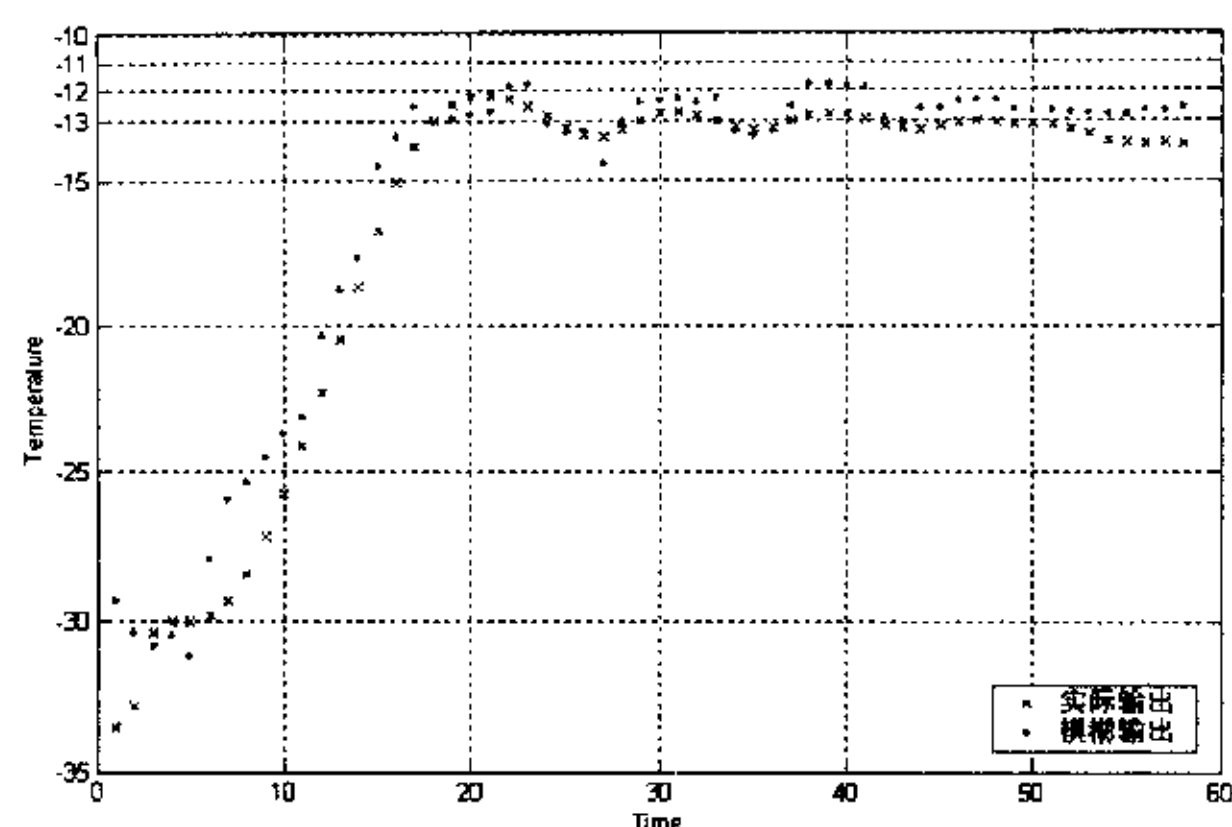


图 5.3 验证曲线

Fig.5.3 Testing curve

表 5.3 模糊输出与实际输出误差表

Tab.5.3 Error between fuzzy output and actual output

	1	2	3	4	5	6	7	8
误差	0.0837	0.1177	0.0672	0.0826	0.0939	0.0996	0.1007	0.1061
	9	10	11	12	13	14	15	16
误差	0.1141	0.1312	0.1448	0.1486	0.1398	0.1140	0.0129	0.0079
	17	18	19	20	21	22	23	24
误差	0.0018	0.0150	0.0255	0.0236	0.0152	0.0007	0.0244	0.0637
	25	26	27	28	29	30	31	32
误差	0.0319	0.0522	0.0544	0.0709	0.0144	0.0686	0.0497	0.0435
	33	34	35	36	37	38	39	40
误差	0.0089	0.0404	0.0024	0.0156	0.0097	0.0086	0.0199	0.0187
	41	42	43	44	45	46	47	48
误差	0.0026	0.0621	0.0319	0.0257	0.0253	0.0261	0.0190	0.0071
	49	50	51	52	53	54	55	56
误差	0.0270	0.0649	0.0386	0.0503	0.0668	0.0222	0.0311	0.0279
	57	58						
误差	0.0425	0.0169						

图 5.3 是验证数据的曲线图。表 5.3 是验证误差表。从验证曲线图和误差表中可以看到，误差比较大，尤其是在初始阶段，最大误差达到了 14%，此外在反应

中期即稳定运行阶段也有一些误差较大的点，最大误差也达到了 14%，但是并没有出现会导致生产出废品的温度，在 $-12^{\circ}\text{C}$ 附近只有很小的波动，不会对生产造成很大的影响。因此该模糊模型的辨识效果还是比较满意的。

在曲线验证中之所以会出现误差较大的情况，主要有两方面的原因。其一，从 T-S 模糊模型的角度来看，模糊等级划分的较少是一个因素。在训练 T-S 模糊模型参数时对输入数据均进行了三个等级的划分，这是为了编码方便，但是针对不同的输入变量，根据实际情况应该采用不同的等级划分。比如说反应进行时间可以分为三个等级{开始阶段，反应中期，结束阶段}，而一甲胺流量则分成五个阶段较好{流量小，流量较小，流量适中，流量大，流量较大}。使用了合理的等级划分，就会有合适的模糊推理规则，这样就会有精度较高的输出。其二，从遗传算法角度来考虑，首先是编码，本文所使用是将所有的参数编码排成一行，算法的实现就是一种串行算法，因此运行速度比较慢。本文是在 2000 步中就找到了合适值，假如编码更长就有可能找不到合适的代，因此也就搜索不到最优参数，这样势必就会造成模糊输出的精度不高。如果改用并行算法或者其他一些较好的编码方式，效果就会比较好<sup>[26]</sup>。其次是变异算子，在本文中采用的是一般的变异算子即变异概率是不会变化的，如果在进化到了一定代数，出现了不进化的情况，但是此时并没有找到最优的代，那么即便是设定了足够的训练代数，依旧找不到最优参数，从而也影响了辨识结果。



## 6 结论与展望

氧乐果反应釜温度对象是一个比较典型的间歇过程对象,该对象动态过程比较缓慢,具有很强的不确定性和时变性,用常规的建模方法建立起模型比较困难。智能系统理论的产生与发展为系统辨识开辟了新的方向,基于智能系统理论,可以对系统本身的性质不用做太多的假设,对系统内部机理知识也不必了解的太多,就能够建立起比较理想的模型。本文在研究了智能系统辨识方法的一些相关理论的基础上,对氧乐果合成间歇反应釜温度对象的辨识做了一些研究,并取得了一定的成果,主要包括以下几个方面:

1. 建立了 BP 神经网络间歇过程模型。本文针对 SSE 性能函数,采用 L-M 训练算法,大大减少了训练时间,网络的泛化能力也较好,验证数据的误差比较小。将 BP 神经网络模型引入到控制系统中,通过手动控制手段,能够达到比较好的控制效果,较好的仿真了实际过程。

2. 提出了分阶段建立模型的方法。氧乐果合成间歇反应釜温度对象,在不同的反应阶段对一甲胺流量的要求也不同,基于此本文提出了分阶段建立模型的方法。根据工艺要求,将整个过程的温度分成三个阶段,相应的输入输出数据也分成三个阶段,由此建立阶段模型。利用验证数据来检验网络,网络泛化能力较之单独的 BP 网络有很大的提高,尤其是在反应初始阶段的效果明显好于采用单独 BP 神经网络时的情况,而且将其引入到控制系统中,也能够很好的模拟现场情况,体现出了实际过程的参数时变性。

3. 建立了回归神经网络间歇过程模型。实际的过程是一个动态过程,前  $n-1$  时刻的温度对第  $n$  时刻的温度是有影响的。BP 神经网络是一种静态的辨识方法,在加上 TDL 环节后就构成了回归神经网络,这样就可以将前面时刻的温度引入到神经网络中,从而可是实现动态过程的辨识。文中在 BP 神经网络的基础上加上了一个 TDL 环节构成回归网络,即认为前一个时刻的温度对当前温度影响最大。在加上了 TDL 环节后,辨识精度明显的提高了,尤其是用验证数据验证时,拟合程度显著高于没有加上 TDL 环节的静态 BP 网络,数据验证误差较小,而且即使是利用手动控制也能够实现较好的控制效果,这也从另一个角度证明了回归神经网络模型更逼近于实际的间歇生产动态过程。

4. 建立了基于遗传算法的 T-S 模糊间歇过程模型。T-S 模糊模型可以任意精度逼近定义在紧集上的非线性函数,但是还存在一些缺点,因此文中引入了广义

T-S 模型的概念, 并将广义 T-S 模型中的参数进行编码, 利用遗传算法寻找出最优的参数, 构成广义 T-S 模型的间歇过程模型。从验证数据的结果来看, 效果还是比较令人满意的。

此外, 本文还将曲线拟合的技术引入到仿真控制信号与实际控制信号的转化当中, 绘制出的曲线和得到的经验方程在一定范围内能够加以应用。

氧乐果合成间歇反应釜温度对象是一个复杂的过程, 其模型的辨识并不是一次能够详尽的描述, 本文仅仅是在某些方面做了一些研究工作, 还有很多方面的工作值得去研究。

1. 进一步研究和探索神经网络的建模方法。本文仅研究了 BP 神经网络以及加上了 TDL 环节的回归网络在辨识中的应用, 还有其他一些网络结构没有进行研究, 以此更进一步完善氧乐果合成间歇过程的模型, 建立能体现较多特性的一个统一模型。此外在 BP 网络的结构上是根据一些经验公式来确定, 如果能够采用自适应变结构的网络来进行辨识, 那么会减小在确定网络结构时的一些主观性。在其他算法与神经网络的结合上可以拓宽一些研究, 如混沌神经网络, 基于遗传算法的神经网络以及灰色理论等等。

2. 进一步研究和探索遗传算法。本文仅仅是利用标准遗传算法来对广义 T-S 模型的参数进行了寻优, 而且还是一种串行算法, 时间较长, 并且还存在着有可能寻找不到最优参数的可能性。因此, 今后可以从编码方式上, 算法改进上做一些研究工作。

3. 寻求流量与阀门开度转换的更为理想的方法。虽然文中利用曲线拟合的方法做出了曲线, 并给出了经验方程, 但毕竟有一定的局限性。

4. 仿真试验与实际生产过程的有机结合。文中的所有试验均为仿真试验, 在实际生产过程中并未得到验证, 因此这一部分的工作还有待完善。

## 致谢

本文的研究工作是在导师冯冬青教授的悉心指导和热情鼓励下完成的。值此论文完成之际，衷心感谢导师冯冬青教授三年来对我学业上的精心指导和帮助。冯老师渊博的知识、敏捷的思维和丰富的创造力时刻帮助我在科研工作中把握正确的方向，引导我取得了一个个进步。非常感谢冯老师努力为我们创造一个良好的工作环境与和谐的工作氛围。

在我的学习过程中，董玲娇、温盛军、刘刚、王瑗琇、刘新宇、梁义涛等同学以及同实验室的周永庆、王丁磊、吴杰、邵丽红、李波和李玮给了我很大的启发和帮助，在此向他们表示深深的谢意。

衷心感谢所引文献作者，衷心感谢前来指导的专家和评委。

最后谨将此文献给我的父亲母亲，献给所有我爱的与爱我的人。

## 参考文献

- [1] 方崇智, 萧德云. 过程辨识. 清华大学出版社. 1988
- [2] 冯恩波, 俞金寿, 蒋慰孙. 基于神经网络技术的间歇过程建模与优化控制方法. 华东化工学院学报. 1992. 10. p113-118
- [3] 王保国, 王春燕, 许锡恩. 间歇过程标准化. 化工进展. 1997. 4. p23-27
- [4] 李士勇. 模糊控制、神经控制和智能控制论. 哈尔滨工业大学出版社. 1998
- [5] 孙增圻等. 智能控制理论技术. 清华大学出版社. 1997
- [6] 杨志晓论文
- [7] 乔俊伟, 詹永麒, 施光林. 基于结构的神经网络在系统建模中的应用. 系统仿真学报. 2002. 5. p557-561
- [8] 宋建成. 间歇过程计算机集成控制系统. 化学工业出版社. 1999
- [9] 汤兵勇. 模糊模型的辨识及应用. 中国环境科学出版社. 1994
- [10] 王杰, 金耀初. 智能控制系统理论及应用. 河南科学技术出版社. 1997.3
- [11] 王士同. 模糊系统、模糊神经网络及应用程序设计. 上海科学技术文献出版社. 1998
- [12] T.Takagi and M.Sugeno, Fuzzy Identification and Its Application to Modelling and Control, IEEE Trans, Syst, Man & Cybern, 15(1985), p116-132
- [13] 陆金桂, 李谦, 王浩, 曹一家. 遗传算法原理及其工程应用. 中国矿业大学出版社. 1997
- [14] 陈国良, 王煦法, 庄镇泉, 王东生. 遗传算法及其应用. 人民邮电出版社. 1996. 6
- [15] 王小平, 曹立明. 遗传算法——理论、应用及软件实现. 西安交通大学出版社. 2002
- [16] 王士同. 神经模糊系统及其应用. 北京航空航天大学出版社. 1998
- [17] 李兰馨. 过程数据的计算机处理. 清华大学出版社. 1989
- [18] 王守唐, 高东杰. 基于 T-S 模糊模型的辨识算法. 控制与决策. 2001. 9. p630-636
- [19] 王立新. 自适应模糊系统与控制: 设计与稳定性分析. 国防工业出版社. 1995
- [20] 张建华, 张荫芬, 侯国莲, 史运涛. 基于 T-S 模型的自组织模糊辨识算法研究. 现代电力. 2002. 10. p64-68



- [21] 丛爽. 面向 MATLAB 工具箱的神经网络理论与应用. 中国科学技术大学出版社. 1998
- [22] 飞思科技产品研发中心. MATLAB6.5 辅助神经网络分析与设计. 电子工业出版社. 2003
- [23] (日)玄光男, 程润伟著; 汪定伟等译. 遗传算法与工程设计. 科学出版社. 2000
- [24] 周明, 孙树栋. 遗传算法原理及应用. 国防工业出版社. 1999
- [25] 阎平凡, 张长水. 人工神经网络与模拟进化计算. 清华大学出版社. 2000
- [26] 李合生, 毛剑琴, 代冀阳. 基于遗传算法的广义 Takagi-Sugeno 模糊逻辑系统最优参数辨识. 自动化学报. 2002. 7, p581-586
- [27] 孙培勤, 刘大壮. 实验设计数据处理与计算机模拟. 河南科学技术出版社. 2001. p46-57
- [28] 石振东, 刘国庆. 实验数据处理与曲线拟合技术. 哈尔滨船舶工程学院出版社. 1991. p206-239
- [29] 袁曾任. 人工神经网络及其应用. 清华大学出版社. 1999
- [30] 王耀南. 智能控制系统——模糊逻辑·专家系统·神经网络控制. 湖南大学出版社. 1996
- [31] 冯冬青, 谢宋和. 模糊智能控制. 化学工业出版社. 1998. 9
- [32] M.O.Efe, O.Kaynak, A comparative study of neural network structures in identification of nonlinear systems, Mechatronics 9(1999),p287-300
- [33] Yacoub M.Najjar, Imad A.Basheer & Wissam A.Naouss, On the Identification of Compaction Characteristics by Neuronets, Computers and Geotechnics, Vol.18,NO.3, pp167-187,1996
- [34] Glenn K.Heitman, Complexity of identification of input-output systems, Mathematical analysis and Applications 192, 1007-1033 (1995)
- [35] Biao Huang and Sirish L.Shah, Closed-loop identification: a two step approach, J.Proc.Cont.Vol.7,NO.6,pp425-438,1997
- [36] P.M.Makila, J.R.Partington, On robustness in system identification, Automatica 35(1999) 907-916
- [37] Minkee Park, Seunghwan Ji, Euntai Kim, Mignon Park, A new approach to the identification of a fuzzy model, Fuzzy Sets and Systems 104(1999),169-181
- [38] X.Cao, Y.Sugiyama, Y.Mitsui, Application of artificial neural networks to load identification, Computers and Structures 69(1998) 63-78
- [39] A.Blanco, M.Delgado, I.Requena, Identification of fuzzy relational equations by fuzzy neural, Fuzzy Sets and Systems 71(1995) 215-226

- [40] A.T.C.Goh, Back-propagation neural networks for modeling complex systems, *Artificial Intelligence in Engineering* 9(1995) 143-151
- [41] Erik Weyer, Geoff Bell, Peter L.Lee, System identification for generic model control, *Journal of Process Control* 9(1999) 357-364
- [42] Byoung-Tak Zhang, Dong-Yeon Cho, System identification using evolutionary Markov chain Monte Carlo, *Journal of Systems Architecture* 47(2001) 587-599
- [43] 黄龙峰. 我国农药工业发展需要重视的问题. *安徽化工*, 1997, 87(2):1-5
- [44] 王律先. 我国农药工业概况及发展趋势. *现代化工*, 2000, 20(2):1-6
- [45] Sand G, Engell S, Markert A, et al. Approximation of an ideal online scheduler for a multiproduct batch plant [J]. *Compute Chem-Eng-*, 2000, 24(2):361-371
- [46] Kefeng Wang, Thomas Lohl, Mario Stobbe, et al. A genetic algorithm for online scheduling of a multiproduct polymer batch plant [J]. *Compute Chem-Eng-*, 2000, 24(2):393-401
- [47] 陈治纲, 许超, 邵惠鹤. 间歇过程优化与先进控制综述. *化工自动化及仪表*. 2003. 030(003). -1-6
- [48] 李永彪, 孙正贵, 赵清杰. 间歇聚丙烯生产的智能控制方案[J]. *工业仪表与自动化装置*, 2002(5):15-18
- [49] 李鸣, 李赣平, 龙伟. 氯乙酸间歇生产过程中控制方案的若干改进措施[J]. *南昌大学学报(工科版)*, 2002(1):59-61
- [50] 刘瞰东, 漆随平, 郑国祥, 方廷健等. 一种智能预估控制算法在间歇造气炉控制中的应用[J]. *化工自动化及仪表*, 2003 (3):16~20
- [51] Iwasa, Takeshi, et al. Temperature control of neural network in batch process control[J]. *Journal of Food Engineering*, 2001(4):403-421
- [52] Canton, J, Ruiz, D, et al. Integrated information system for monitoring, scheduling and control applied to batch chemical processes[C]. *IEEE Symposium on Emerging Technologies and Factory Automation, ETFA*, 1999(1): 211-217
- [53] Cohen A, Heqq D, De Michele M, et al. An intelligent controller for automated operation of sequencing batch reactors[J]. *Water Science and Technology*, 2003(12):57-63
- [54] Qingjie Zhao, Bo Yang, Fuchun Sun. Intelligent control scheme of a batch process[J]. *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, 2000(1): 439-443
- [55] Zhang, Jie. Multi-Objective Optimal Control of Batch Processes Using Recurrent Neuro-fuzzy Networks. *Proceedings of the International Joint Conference on Neural Networks.v1.2003.p304-309*



- [56] 孙正贵, 李永彪, 赵清杰. 间歇过程的神经网络模型. 工业仪表与自动化装置. 2000. 4. p12-14
- [57] 傅春生, 王骥程. 在线识别间歇过程特性的新方法及其在发酵过程中的应用. 自动化学报. 1994. 020(002). -162-168
- [58] A.Cabrera, A.Poznyak, T.Poznyak, etal. Identification of a fed-batch fermentation process: comparison of computational and laboratory experiments. Bioprocess and Biosystems Engineering. Vol2.NO.5.2002.319-327
- [59] 杨凌, 张子范. 原油常压蒸馏过程的人工神经网络建模方法. 兰州铁道学院学报 (自然科学版), Vol. 21, NO. 6, 2002, 10~13
- [60] Jimenez, L.; Basualdo, M.S.; Gomez, J.C.; Toselli, L.; Rosa, M., Nonlinear dynamic modeling of multicomponent batch distillation: A case study. Brazilian Journal of Chemical Engineering, v19, n3, July/September. 2002.p307-317
- [61] Faouzi M'Sahli, Ridha Ben Abdenmour, Mekki Ksouri. Identification and Predictive Control of a Nonlinear Process Using a Parametric Volterra Model. International Journal of Computational Engineering Science. Vol 2.NO.4.2001.12.633-651
- [62] Zhang; J.. A Reliable Neural Network Model Based Optimal Control Strategy for a Batch Polymerization Reactor. Industrial & Engineering Chemistry Research; 2/18/2004, Vol. 43 Issue 4, p1030, 9p
- [63] 张凤珊, 孙振环. 发酵过程控制的回归模型和过程预测. 天津轻工业学院学报, NO. 2 sum 41, 2002, 55~57
- [64] 隋青美, 王正欧. 发酵过程混合神经网络模型及其仿真. 系统仿真学报. Vol 14, NO. 4, 2002, 415~417
- [65] Psichogois D C. Ungar L H. A hybrid neural network-first principles approach to process modeling AIChE Journal, 1992, 38 (10): 1499~1511
- [66] Thompson M L, Kramer M A . Modeling chemical processes using prior knowledge and neural networks. AIChE Journal, 1994. 40(8): 1328~1340
- [67] Mavrovouniotis M L, Chang S. Hierarchical neural networks for process monitoring. Comp. Chem. Eng., 1992,6(4): 347~356

## 附录 1

### 攻读硕士学位期间发表的学术论文

1. Feng Dongqing, Zhang Xinzheng, Fei Minrui, Chen Tiejun. Intelligent Fuzzy Control Method for Soaking Pit Economical Combustion. IFAC Workshop on New Technologies for Automation of Metallurgical Industry. Shanghai, P.R.China, October 11-13, 2003, 123~127 (被 EI 收录)
2. 张新征, 冯冬青, 雷天友. 不确定时滞系统的模糊神经网络自适应 PID 控制. 自动化仪表 (已录用)
3. 张新征, 董玲娇. 基于小脑模型神经网络与 PID 的复合控制研究. 河南科学, Vol. 22, NO.3 (Tot.88) 2004, 310~312