

摘 要

在数控机床中，位置伺服系统作为数控装置与机床传动部件之间的中间连接环节，它的性能对于数控机床的整体性能具有较大影响。如工作台的最高运动速度、跟踪及定位精度、加工表面质量、生产率及工作可靠性等指标均取决于位置伺服系统的动态和静态性能。目前，位置伺服控制器的设计仍主要是采用常规的 PID 控制。PID 控制具有算法简单、鲁棒性强、可靠性高的特点，尤其适于能建立精确数学模型的系统。然而，随着数控机床加工精度要求的提高，数控加工工艺变得越来越复杂，加工过程往往具有非线性、时变性，难以建立被控对象的精确数学模型。常规的 PID 控制因其自身存在的一些局限性，如需要控制对象的精确数学模型、自适应能力差等，已越来越不能满足高精度数控机床的需要。因此，有必要对其进行适当的改进，以提高数控机床位置伺服系统的性能。

模糊逻辑和神经网络均不依赖被控对象的精确数学模型，且具有较强的自适应和自学习能力；遗传算法是一种基于生物进化过程的随机搜索的全局优化方法，它通过对个体进行编码，并对编码串进行复制、交叉和变异等操作，使算法最终优化到全局极值点。本文将模糊逻辑控制、神经网络和遗传算法运用于 PID 参数的整定过程中，设计了一种新型的智能 PID 控制器。这种新型的控制器一方面保留了常规 PID 控制鲁棒性强、可靠性高的特点，同时又兼备了模糊逻辑控制和神经网络的一些优点。例如，不再依赖控制对象的精确数学模型，具有较强的自适应、自学习能力和抗干扰能力等。通过对数控机床位置伺服系统的仿真说明，这种新型的控制器不但大大缩短了系统的过渡过程时间，实现了无超调控制，而且具有较强的自适应能力和抗干扰能力。因此，对于提高数控机床的定位精度和跟踪精度，从而提高机床的加工精度具有较强的现实意义。

关键词：位置伺服系统；PID 控制；模糊神经网络；BP 算法；遗传算法

Abstract

In numerical control machine, the position servo system links the computer numerical controller and the mechanical transmission equipment. So its performance is very important to the performance of the numerical control machine. Such as it can affect the fast ratio of the machine, position precision, the quality of surface of the manufacture, productivity, reliability and etc. At the present, we usually used conventional PID controller because its algorithm is every simple and it has strong robustness and fine reliability. And especially, it is suitable to the system that we can make its accurate mathematical model. However, with the development of industry, the manufacturing process is becoming more and more complicated. First, it is difficult to make the precise mathematical model of object. And the conventional PID controller can't adjust itself to the variation of the surroundings and the object. So we must find a new method to design the parameters of the PID controller.

Fuzzy logic control and neural network have many advantages which conventional PID control doesn't have. They don't need the accurate mathematical model of object, and have fine ability of self-studying and adapting. Genetic algorithm is a kind of random searching optimal algorithm based on biological evolution. It can find the best result by reproduction, crossover and mutation. So we come up with a new method to design the parameters of PID controller. This kind of controller has the advantages of conventional PID controller, fuzzy logic controller and neural controller. The simulation curve on position servo system of numerical control machine shows that it improves the ability of self-studying, adapting and anti-jamming of the system, and betters the robustness.

Keywords: Position servo system ; PID control; Fuzzy neural network; BP algorithm; Genetic algorithm

兰州理工大学

学位论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其它个人或集体已经发表或撰写的成果作品。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律后果由本人承担。

作者签名：刘鹏宇

日期：2006年5月10日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权兰州理工大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于

1、保密口，在_____年解密后适用本授权书。

2、不保密口。

(请在以上相应方框内打“√”)

作者签名：刘鹏宇

日期：2006年5月10日

导师签名：[Signature]

日期：2006年5月10日

第 1 章 绪论

1.1 课题的来源、目的和意义

本课题为兰州理工大学自选课题。

位置伺服系统是数控机床的重要组成部分，它的性能对于数控机床的性能有较大影响，如数控机床的最高移动速度、运动精度和定位精度等重要指标均取决于位置伺服系统的动态性能和静态性能。因此，研究和开发高性能的伺服系统一直是现代数控机床的关键技术之一。目前，数控机床的位置伺服系统大多仍采用常规的 PID 控制^[1]，它具有算法简单、鲁棒性好、可靠性高的特点，尤其适用于可建立精确数学模型的系统^[2]。然而，随着工业生产的进一步发展，数控加工的精度要求越来越高，数控加工工艺越来越复杂，加工过程往往具有非线性、时变不确定性，难以建立精确的数学模型，再加上常规的 PID 控制器缺乏自适应能力，不能根据被控对象特性的变化自动调整控制器的参数，这些都使得常规的 PID 控制难以适应高精度数控机床的需要，必须对其进行改进。

长期以来，虽然人们为了提高伺服系统的性能而在不断的努力，也提出了一些新的控制算法，如前馈控制（Feedforward Control）、预测控制（Predictive Control）和学习控制（Learning Control）等^[3]，取得了一定的效果。但这些算法本身并没有脱离传统控制理论的范畴，且算法本身仍存在的一定缺陷，因此，对于高精度数控机床的控制仍不甚理想。

随着微处理机技术、现代控制理论尤其是智能控制理论（如模糊逻辑、神经网络等）和新型优化算法（如遗传算法）的发展与应用，为控制复杂无规则系统开辟了新途径^[1]。人们将这些新型的控制方法与传统的 PID 控制算法相结合，也就是借助新型的控制算法优化传统 PID 控制的各个参数，进而提高 PID 控制器的控制效果和控制精度，目前已出现了许多新型的 PID 控制器，例如，模糊 PID 控制器、基于神经网络的 PID 控制器等等。它们吸收了智能控制与常规 PID 控制两者的优点，既具备一定的自学习、自适应能力，又保留了常规 PID 控制器鲁棒性强、可靠性高的特点。实践证明，对于复杂控制对象，其控制效果远远超过常规 PID 控制^[1]。

本课题正是从已经取得的成果的基础上得到启发，将模糊逻辑、神经网络和遗传算法应用于常规的 PID 控制中，即用模糊逻辑、神经网络

和遗传算法设计并优化 PID 控制器的各个参数，从而构造了一种基于模糊神经网络和遗传算法的智能 PID 控制器，为发展新型的 PID 控制技术，提高数控机床位置伺服系统的性能，作一些简单的理论探讨。

1.2 国内外的研究现状

目前，将模糊逻辑、神经网络与遗传算法用于 PID 参数的设计是控制专家们研究的热点，已经提出了一些具体的策略，有些已在工业中得到了广泛的应用，取得了良好的控制效果。综合起来，大致可分为以下几种类型：

1. 借助单纯的模糊逻辑、神经网络或遗传算法设计 PID 参数。如模糊 PID 控制器、模糊自适应 PID 控制器、神经网络 PID 控制器和基于遗传算法的 PID 控制器等。关于这方面的研究进行的较多，也取得了许多效果明显的研究成果，有些成果已在实际工业控制中得到广泛地应用，取得了良好的控制效果。如将模糊 PID 控制用于无刷直流电动机交流伺服系统、液压系统以及智能人工腿位置伺服系统的控制等等，将神经网络 PID 控制器用于高温力学试验机加热炉计算机控制系统等。具体的可以参见参考文献 [2]。

2. 将模糊逻辑、神经网络和遗传算法的任意两种进行结合，用于 PID 参数的设计。典型的如基于遗传算法的模糊 PID 控制器，其基本思想是在模糊 PID 控制器的基础上借助遗传算法优化模糊控制器的隶属度函数和控制规则；基于遗传算法的神经网络 PID 控制器，其基本思想是借助遗传算法优化神经网络 PID 控制器的连接权系数；基于神经网络的模糊 PID 控制器，其基本思想是借助神经网络的结构来实现模糊 PID 控制等等。具体可以参见文献 [2]。

3. 将模糊逻辑、神经网络与遗传算法三者进行结合用于 PID 参数的设计。目前关于此方向的研究也正在如火如荼的进行，主要的形式就是将遗传算法用于模糊神经网络的优化设计。这其中，重点是如何用遗传算法对隶属度函数和模糊控制规则，总结起来，大致可分为以下三种类型：

(1) 已知模糊控制规则，利用遗传算法优化隶属度函数。即首先根据控制专家的经验知识确定出控制器的模糊控制规则，然后保持控制规则不变，再利用遗传算法优化隶属度函数。Karr 利用这种方法为一个四输入单输出的倒立摆平衡系统成功设计了模糊控制器，并运用到了 PH 值的系统控制^[4]；王耀南提出了一种模糊神经网络模型，可以实现模糊控制规则的在线修改，同时又提出了一种自适应遗传算法，用于优化模糊

控制器的隶属度函数，使模糊控制具有了自学习和自适应能力，并将其运用到了全自动热电偶校验仪检测室炉温的控制中，取得了满意的效果^[5]；李琳将这种方法用于故障诊断，仿真结果显示其效果优于单纯采用神经网络或模糊神经网络进行故障诊断的效果^[6]；Buckley提出了一种无需事先确定隶属度函数形状的模糊遗传算法，并将其应用于函数优化及模糊控制器的设计^[7]。

(2) 已知隶属度函数，利用遗传算法优化模糊控制规则。即首先根据控制专家的经验知识确定出各个语言变量的隶属度函数，然后在此隶属度函数的基础上，再利用遗传算法对控制规则进行优化，得出最佳的控制规则。Thrift将这种方法应用于一个两输入单输出的小车定位模糊控制器的设计过程，取得了良好的控制效果^[8]；金耀初、蒋静坪提出了一种遗传算法的改进形式，引入了杂交算子和主动优生的选择机制，并将其应用到三角形和铃形隶属度函数下模糊控制规则的优化^[9]。

(3) 用遗传算法同时优化隶属度函数和模糊控制规则。这也是本论文要采用的方案。由于隶属度函数与模糊控制规则之间不是相互独立的，而是存在一定联系的，是互动的关系。因此，无论是固定隶属度函数优化控制规则，还是固定模糊控制规则优化隶属度函数都会人为地割裂两者之间的联系，即使优化出了一组参数，也只是相应于某一种具体规则或某一种具体隶属度函数的参数，并不一定是适合于所求问题的最优参数。Lee M. A.和Takagi H.以一个简化的倒立摆系统为例，用遗传算法同时优化一个二输入单输出的T-S型模糊控制器的隶属度函数和模糊控制规则，个体编码由输入变量的隶属度函数参数和控制规则的后件参数两部分组成。利用这种方法得到的模糊控制器能在没有专家经验知识的情况下达到满意的控制效果，但缺点是编码长度过长，导致搜索空间急剧膨胀，收敛过程缓慢^[10]；Homaifar A.和Me Connick E.通过固定输入输出变量的隶属度函数的个数、固定三角形隶属度函数顶点的位置、限制隶属度函数底部宽度的变化范围等措施，缩小了隶属度函数参数编码的长度，但这些措施有可能会损害模糊控制器的性能^[11]；王晶等提出了一种混合变长度编码策略，规则前件为二进制编码，规则后件为浮点数编码，对倒立摆系统进行了仿真研究^[12]。周志坚、毛宗源利用遗传算法优化神经网络模糊控制器的结构和参数，结果显示，系统的静态性能和动态性能均优于常规的模糊控制器^[13]。

通过对上述方案分析发现，这些方案存在以下几个特点：

第一，控制器的输出均为控制量或控制量的增量，而不是PID参数，因此无法反映出偏差、偏差的变化以及偏差的累积在控制量中所占的比

重；

第二，在遗传算法的运行过程中，对遗传算法性能有较大影响的交叉概率和变异概率固定不变，无法根据种群结构的变化进行调整，虽然自适应遗传算法使得交叉算子和变异算子具有了一定的适应能力，但同时又引入了新的参数，这些参数的选取具有较大的主观性；

第三，对模糊控制器性能有较大影响的量化因子和比例因子的确定主要是依据控制专家的经验知识，带有较大的主观性；

第四，模糊神经网络中初始连接权系数的确定主要是采用随机方法产生，导致算法的收敛速度较慢，甚至陷入局部极值点。

1.3 论文的主要内容

本文首先结合高精度数控机床对位置伺服系统的要求，指出了常规 PID 控制所存在的不足之处，提出了对其进行改进的必要性。然后介绍了有关模糊逻辑、神经网络和遗传算法的基本原理，提出了将模糊逻辑、神经网络和遗传算法引入 PID 控制器参数整定过程的思想，并在总结前人工作的基础上提出了自己的设计方案。最后，针对一个具体的数控机床位置伺服系统设计了相应的位置伺服控制器，基于 MATLAB 的仿真表明，采用这种方案设计的控制器不但大大缩短了系统的过渡过程时间，实现了无超调控制，而且具有较强的自适应能力和抗干扰能力。对于提高数控机床的定位精度、跟踪精度，从而提高机床的加工精度具有很强的现实意义。

全文共分为五章，具体内容分别如下：

第一章，绪论。主要介绍了课题的来源、目的和意义，并在总结前人工作的基础上提出了论文所要采用的控制策略，分析了目前国内外的研究现状。

第二章，数控机床位置伺服系统。主要介绍了数控机床位置伺服系统的概念和原理，并建立了直流位置伺服系统的数学模型，分析了目前常用的位置调节器的设计方法的不足之处，并在此基础上提出了自己的设想。

第三章，模糊神经网络和遗传算法。主要介绍了有关模糊逻辑、神经网络和遗传算法的一些基础知识。

第四章，位置调节器的设计。详细阐述了自己的设计思想，并针对一个具体的数控机床位置伺服系统，设计了相应的位置调节器，进行了 MATLAB 仿真。

第五章，结论。对全文的工作进行了总结。

第 2 章 数控机床位置伺服系统

2.1 数控机床伺服系统概述

2.1.1 伺服系统的概念

伺服系统是指以机械位置或角度作为控制对象的自动控制系统。在数控机床中，伺服系统主要指各坐标轴进给驱动的位置控制系统。伺服系统接受 CNC 控制器经过插补运算生成的进给脉冲或进给位移量，经变换和功率放大后，驱动各坐标轴按指令脉冲运动^[1]。这些轴有的带动工作台，有的带动刀架，通过几个坐标轴的联动，使刀具相对于工件产生各种复杂的机械运动，加工出所要求的复杂形状工件^[14]。伺服系统既是数控机床控制器与刀具、主轴间的信息传递环节，又是能量放大与传递的环节，它的性能在很大程度上决定了数控机床的性能。数控机床的最高移动速度、运动精度和定位精度等重要指标均取决于伺服系统的动态性能和静态性能。因此，研究和开发高性能的伺服系统一直是现代数控机床的关键技术之一。

早期的数控机床，尤其是大中型数控机床常采用电液伺服系统驱动。它由电液脉冲马达构成开环控制系统，用电液伺服阀、低速大转矩液压马达或液压缸及位置检测等反馈控制构成闭环驱动系统。从八十年代起，全电气伺服系统成为数控机床的主要驱动器。

电气伺服系统是指以各种伺服电动机作为驱动元件的伺服系统^[3]。

2.1.2 伺服系统的分类

1. 按控制方式分类

位置伺服系统可分为开环、半闭环和闭环三种。开环控制系统就是不具有任何反馈装置的控制系統。这种系统通常使用步进电动机作为执行机构。数控装置根据所要求的进给速度和进给位移，输出一定频率和数量的进给指令脉冲，经过驱动电路放大后，每一个进给脉冲驱动步进电动机旋转一个步距角，再经过传动系统转换成工作台的一个当量位移。开环控制系统相对半闭环和闭环控制系统精度较低。但开环控制系统结构简单、运行平稳、成本低、使用维护方便，故广泛应用于经济型数控机床上。

闭环和半闭环数控系统具有位置和速度检测元件。数控装置将位移指令与位置检测元件检测到的实际位置进行实时的比较，其差值与数控

装置给定的指令速度按一定的关系转换得到伺服驱动系统的速度指令。同时，速度检测元件将测得的实际速度反馈信号与该速度指令相比较，以对电动机的转速进行实时校正。利用这种位置控制和速度控制的双回路控制，可以获得比开环控制系统精度更高、响应速度更快、驱动功率更大的特性指标。

闭环和半闭环控制系统的区别在于检测元件的类型和检测的对象不同。闭环控制系统采用直线位置检测元件，直接测量移动部件的位移。半闭环控制系统采用角位移检测元件，测量电动机或丝杠的角位移来间接地检测移动部件的直线位移。由于角位移和直线位移之间多了一个机械传动环节，因而半闭环控制系统比闭环控制系统精度低，但角位移检测元件比直线位置检测元件结构简单、安装方便、稳定性好、价格便宜，所以应用非常广泛。

2. 按所驱动的伺服电动机分类

伺服系统可以分为步进电动机伺服控制系统、直流伺服系统和交流伺服系统。它们都是由伺服驱动器和伺服电动机两部分组成。由于交流伺服系统几乎保留了直流伺服系统的所有优点，具有调速范围宽、稳速精度高和动态响应特性好等优良的技术特性，而且继承了交流电动机本身固有的许多优良性能，因而成为迄今为止最为理想的伺服系统，对于传动功率要求较大的数控机床，交流伺服系统以成为首选方案。

3. 按进给驱动和主轴驱动分类

(1) 进给伺服系统

进给伺服系统就是我们通常所说的伺服系统，它包括速度控制环和位置控制环，主要完成各坐标轴的进给运动，具有定位和轮廓跟踪功能，是数控机床中要求最高的伺服系统。

(2) 主轴伺服系统

严格的来说，它只是一个速度控制系统，主要实现主轴的无级调速和满足功率与扭矩的输出，对它的精度和快速响应要求也没有进给伺服系统高。但现在许多要求较高的数控机床的主轴也具有位置控制环节，也能作为旋转坐标轴而高精度地完成定位和轮廓跟踪功能。

2.2 位置伺服系统

进给伺服系统是 CNC 系统中的一个重要组成部分，它的性能直接决定与影响 CNC 系统的快速性、稳定性和精确性。其一般结构如图 2.1 所示。

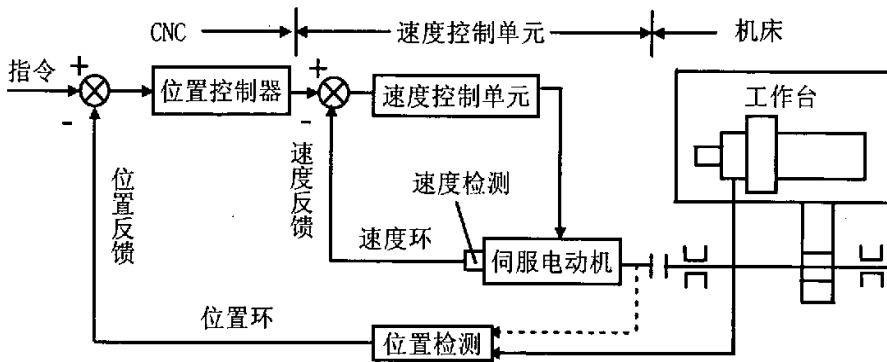


图 2.1 位置伺服系统的基本结构

这是一个双闭环系统，内环是速度环，外环是位置环。其中，速度控制单元是一个独立的单元部件，它由速度调节器、电流调节器及功率驱动放大器等各部分组成，其作用是抵抗负载扰动，抑制速度波动。而位置环则是由 CNC 装置中的位置控制模块、速度控制单元、位置检测及反馈控制等各部分组成。位置控制主要是对机床运动坐标轴进行控制，轴控制是要求最高的位置控制，不仅对单个轴的运动速度和位置精度的控制有严格要求，而且在多轴联动时，还要求各运动轴有很好的动态配合，才能保证加工效率、加工精度和表面粗糙度。

进给伺服系统是以位置为控制对象的自动控制系统，对位置的控制是以对速度的控制为前提，而伺服电动机及其速度控制单元，只是伺服控制系统中的一个组成部分。对于位置闭环控制的进给系统，速度控制单元是位置环的内环，它接收位置控制器的输出，并将这个输出作为速度环路的输入命令，去实现对速度的控制。对于性能较好的速度控制单元，它将包含速度控制及加速度控制，加速度控制环路是速度环路的内环^[14]。

2.3 位置控制的基本原理

为说明位置控制的工作原理，下面以 FANUC 系统为例加以说明。

7M 系统是由日本富士通 FANUC 公司和德国西门子公司联合设计的，于 1976 年研制成功，主要用于数控铣床和加工中心，目前这个系统已经被更为先进的系统所代替，但它的位置控制原理仍具有普遍意义。

7M 系统的位置控制是一个闭环或半闭环系统，其基本原理如图 2.2 所示。

位置控制装置的作用是将插补计算得出的瞬时位置指令值 D_{oi} 和检测到的实际位置 D_{Ai} 相比较，产生位置偏移量 ΔD_i ，再把 ΔD_i 变换为瞬时速

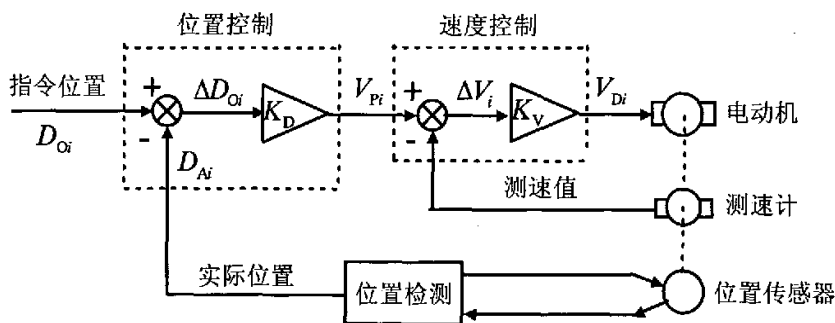


图 2.2 位置控制系统示意图

度指令电压 V_{P_i} 。速度控制装置的作用是将瞬时速度指令电压 V_{P_i} 与检测到的实际速度电压比较后放大为驱动直流伺服电动机的电枢电压 V_{D_i} ；位置检测装置的作用是把位置检测元件检测到的信号转换为与指令位置量级相同的数字量 D_{A_i} ，供它们比较。位置指令是由计算机插补运算得到的。7M 系统的插补周期为 8ms，根据加工程序给出的速度指令和插补计算公式，计算机算出在每个插补周期内各坐标的位置增量 ΔD_{O_i} ，并计算出各坐标方向上的指令位置 D_{O_i} 。当电动机起动时，先输入第一个指令位置值 D_{O_1} ，若按相对坐标计算， $D_{O_1} = 0 + \Delta D_{O_1}$ ，由于此时电动机还未转，因而检测值 $D_{A_1} = 0$ ，位置控制部分先计算 $\Delta D_1 = D_{O_1} - D_{A_1}$ ，再将得到的 ΔD_1 进行模数转换和放大，经速度控制部分控制电动机转动。由于 $D_{O_1} = \Delta D_{O_1}$ ， $D_{A_1} = 0$ ，因而 $\Delta D_1 = D_{O_1}$ 。经过一个插补时间后，计算机送来第二个位置指令值 D_{O_2} ， $D_{O_2} = D_{O_1} + \Delta D_{O_2}$ ，检测装置也把检测到的实际位置 D_{A_2} 送来，位置控制部分再作 $\Delta D_2 = D_{O_2} - D_{A_2}$ 计算。若坐标移动没有误差，则实际位置检测值 D_{A_2} 应等于 D_{O_1} ， $\Delta D_2 = \Delta D_{O_2}$ ，但误差一定存在，因而 $\Delta D_2 \neq \Delta D_{O_2}$ 。若 $D_{A_2} < D_{O_1}$ ，则 $\Delta D_2 > \Delta D_{O_2}$ ，因而使坐标移动量增加，来弥补前一次的不足；若 $D_{A_2} > D_{O_1}$ ，则 $\Delta D_2 < \Delta D_{O_2}$ ，使本次坐标的移动量减少，缩小上一次移动误差。每经过一个插补时，计算机都要算出一个瞬时位置指令值 D_{O_i} ，检测装置也必须送来一个位置反馈值 D_{A_i} ，位置控制装置完成一次 $\Delta D_i = D_{O_i} - D_{A_i}$ 计算。实际上 ΔD_i 是实际检测位置滞后指令位置的滞后量，电动机的转动就是由它控制的。

坐标的移动速度根据程序中指令速度值 $F(\text{mm}/\text{min})$ 而定， F 值大，使每次插补运算得出的坐标增量值 ΔD_{O_i} 大，因而使 ΔD_i 增大，送给电机的电压高，电机的转速增高；若 F 值小，则情况正好相反。

2.4 对位置伺服系统的要求

由于各种数控机床所完成的加工任务不同，它们对位置伺服系统的要求也就不尽相同，但通常可概括为以下几方面：

第一、精度高。伺服系统的精度是指输出量能复现输入量的精确程度。例如数控加工对定位精度和轮廓加工精度要求都比较高，定位精度一般为 $0.01-0.001\text{mm}$ ，甚至 $0.01\ \mu\text{m}$ 。轮廓加工精度与速度控制和联动坐标的协调一致控制有关。在速度控制中，要求高的调速精度，比较强的抗干扰能力，即对伺服系统的动态和静态精度要求都比较高。

第二、稳定性好。稳定性是指系统在给定输入变化或外界干扰作用下，能在短暂的调节过程后，达到新的或者恢复到原来的平衡状态。数控机床要求伺服系统具有较强的抗干扰能力，保证进给速度均匀、平稳。稳定性直接影响加工的精度和产品表面质量。

第三、快速响应。快速响应是描述伺服系统动态品质的重要指标，它反映了系统的跟踪精度。数控机床为了保证轮廓切削形状和高的加工表面质量，要求伺服系统跟踪指令信号的响应要快。这一方面要求过渡过程时间要尽可能短，一般在几毫秒以内，甚至达到微秒级；另一方面要求超调要小，甚至无超调。这两方面的要求往往是矛盾的，实际应用按工艺要求作出选择。

第四、调速范围宽。调速范围是在额定负载时电动机能够提供的最高转速和最低转速之比。对于少数负载很轻的机械，也可以是实际负载下的转速。在数控机床中，由于加工用刀具，被加工材质及零件要求的不同，为保证在任何情况下都能得到最佳切削条件，就要求伺服系统具有足够宽的调速范围。

第五、低速大转矩。机床加工的特点是在低速时进行重切削，因此要求伺服系统在低速时要有大的转矩输出。

2.5 建立位置伺服系统的数学模型

位置伺服系统非常复杂，含有分布参数、非线性、时变性等因素，如果全部因素都予以考虑，建立的数学模型往往极其复杂，不但分析起来不方便，而且在工程上也很难有实用价值。因此，在建模时必须忽略一些次要因素，对系统进行合理的简化。工程上常用的简化方法包括：用集中参数代替分布参数，用定常参数代替时变参数，用等效的线性特性代替非线性特性，用单自由度力学系统代替多自由度力学系统等^[14]。

这里以采用直流伺服电动机驱动的闭环位置伺服系统为例，建立其数学模型。系统的结构如图 2.3 所示。

该系统由双环组成，内环是速度环，外环是位置环。为获得系统的数学模型，下面首先分析各组成环节的传递函数。

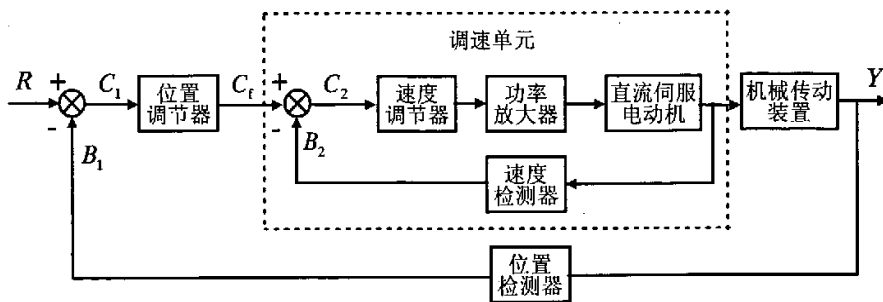


图 2.3 进给伺服系统结构图

1. 组成环节的传递函数

(1) 比较环节

依据自动控制理论，在比较环节(或称相加器)将完成给定信号和反馈信号的比较，两者相减，获得偏差信号。

速度环的比较运算为： $C_2 = C_1 - B_2$ ；

位置环的比较运算为： $C_1 = R - B_1$ 。

(2) 位置调节器

位置调节器的种类很多，最常见的为 PID 控制器，也可以采用比例调节器(P)或比例-积分调节器(PI)。为分析问题方便，这里取位置调节器的传递函数为 $G_c(s)$ ，这是我们后面要进行设计的部分。

(3) 速度调节器和功率放大器

速度调节器同位置调节器一样，一般采用 PID 型控制器(P, PI, PID)，在这里，设其采用比例控制器。功率放大器通常是一个延迟环节，即 $G_s(s) = K_s e^{-Ts}$ ，由于滞后时间很短，因此，可将其视为比例环节^[23]。因此，速度调节器和功率放大器可以合在一起，以一个环节——速度放大器来表示，其传递函数为 $G_s(s) = K_s$ 。

(4) 检测环节

在闭环控制系统中，检测环节起到两个作用，一个是检测出被测信号的大小，另一个作用是把被测信号转换成可与指令信号进行比较的物理量，形成反馈通道^[14]。通常测量转换作用可以看成是一个比例环节，其比例系数就是转换系数。设：

速度反馈环节的传递函数为 $H_v(s) = K_v$ 。

位置反馈环节的传递函数为 $H_p(s) = K_p$ 。

(5) 直流伺服电动机

电枢控制式直流伺服电动机的结构如图 2.4 所示。电动机转速的控制是通过对电枢电压的调节来实现的。求取直流伺服电动机的传递函数时，设输入信号为电枢电压 $u_A(t)$ ，输出信号为电动机转角 $\theta(t)$ 。

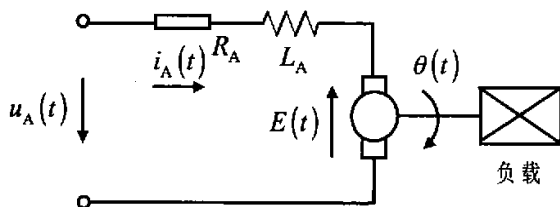


图 2.4 直流伺服电动机工作原理图

根据基尔霍夫定律，建立电枢回路的微分方程

$$L_A \frac{di_A(t)}{dt} + R_A i_A(t) + E(t) = u_A(t) \quad (2.1)$$

$$E(t) = K_E \frac{d\theta(t)}{dt} \quad (2.2)$$

式中， L_A ——电枢回路总电感，单位为 H；

R_A ——电枢回路总电阻，单位为 Ω ；

$i_A(t)$ ——电枢回路电流，单位为 A；

$E(t)$ ——电动机的反电动势，单位为 V；

K_E ——电动机的反电动势系数，单位为 $V \cdot s / \text{rad}$ 。

假设不考虑电动机轴上的负载转矩，则其转矩平衡方程为

$$J_M \frac{d^2\theta(t)}{dt^2} + f_M \frac{d\theta(t)}{dt} = M_A(t) \quad (2.3)$$

式中， J_M ——电动机轴上的转动惯量，单位为 $s^2 \cdot N \cdot m$ ；

f_M ——电动机粘性阻尼系数，单位为 $s \cdot N \cdot m / \text{rad}$ ；

$M_A(t)$ ——电动机的电磁转矩。

又
$$M_A(t) = K_M i_A(t) \quad (2.4)$$

式中， K_M ——电动机的转矩系数。

将式 (2.4) 代入式 (2.3) 得

$$i_A(t) = \frac{1}{K_M} \left(J_M \frac{d^2\theta(t)}{dt^2} + f_M \frac{d\theta(t)}{dt} \right) \quad (2.5)$$

$$\frac{di_A(t)}{dt} = \frac{1}{K_M} \left(J_M \frac{d^3\theta(t)}{dt^3} + f_M \frac{d^2\theta(t)}{dt^2} \right) \quad (2.6)$$

将式 (2.2)、(2.5)、(2.6) 分别代入式 (2.1)，消去中间变量，得

$$J_M L_A \frac{d^3\theta(t)}{dt^3} + (J_M R_A + f_M L_A) \frac{d^2\theta(t)}{dt^2} + (R_A f_M + K_E K_M) \frac{d\theta(t)}{dt} = K_M u_A(t) \quad (2.7)$$

假设初始条件为零，对两边进行拉普拉斯变换得

$$J_M L_A s^3 \theta(s) + (J_M R_A + f_M L_A) s^2 \theta(s) + (R_A f_M + K_E K_M) s \theta(s) = K_M U_A(s) \quad (2.8)$$

设电动机的传递函数为 $G_A(s)$ ，则

$$\begin{aligned}
 G_A(s) &= \frac{\theta(s)}{U_A(s)} \\
 &= \frac{K_M}{s [J_M L_A s^2 + (J_M R_A + f_M L_A) s + (R_A f_M + K_E K_M)]} \\
 &= \frac{K_A}{s (T_A s^2 + T_M s + 1)}
 \end{aligned} \tag{2.9}$$

式中, $K_A = \frac{K_M}{R_A f_M + K_E K_M}$, $T_A = \frac{J_M L_A}{R_A f_M + K_E K_M}$, $T_M = \frac{J_M R_A + f_M L_A}{R_A f_M + K_E K_M}$

设电动机转速为 $\omega(t)$, 则

$$\omega(t) = \frac{d\theta(t)}{dt} \tag{2.10}$$

设初始条件为零, 对两边进行拉普拉斯变换, 得

$$\theta(s) = \frac{\omega(s)}{s} \tag{2.11}$$

因此, 直流伺服电动机的传递函数可以用图 2.5 所示的方框图来表示。

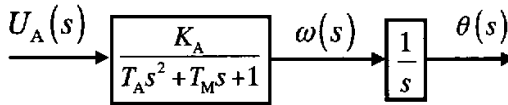


图 2.5 直流伺服电机的传递函数

(6) 机械传动装置

图 2.6 为机械传动装置的原理图。

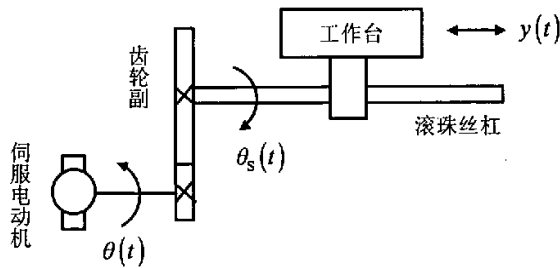


图 2.6 机械传动装置原理图

将整个机械传动装置的刚度、惯量、阻尼折算到丝杠上, 设丝杠的输出转角 $\theta_s(t)$ 为输出信号, 电动机转角 $\theta(t)$ 为输入信号, 其微分方程为

$$J_s \frac{d^2\theta_s(t)}{dt^2} + f_s \frac{d\theta_s(t)}{dt} = M_s(t) \tag{2.12}$$

式中, $M_s(t)$ ——丝杠的驱动转矩, 单位为 $N \cdot m$;

J_s ——折算到丝杠轴上的总转动惯量, 单位为 $s^2 \cdot N \cdot m$;

f_s ——折算到丝杠轴上的粘性阻尼系数，单位为 $s \cdot N \cdot m / rad$ 。

$$\text{又} \quad M_s(t) = K_s [i\theta(t) - \theta_s(t)] \quad (2.13)$$

式中， i ——电动机轴到丝杠的传动比；

K_s ——折算到丝杠轴上的扭转刚度，单位 $N \cdot m / rad$ 。

将式 (2.13) 代入式 (2.12) 得

$$iK_s\theta(t) = J_s \frac{d^2\theta_s(t)}{dt^2} + f_s \frac{d\theta_s(t)}{dt} + K_s\theta_s(t) \quad (2.14)$$

设初始条件为零，两端取拉普拉斯变换得

$$\frac{\theta_s(s)}{\theta(s)} = \frac{iK_s}{J_s s^2 + f_s s + K_s} \quad (2.15)$$

设工作台的位移为 $y(t)$ ，则

$$y(t) = \frac{L_s}{2\pi} \theta_s(t) \quad (2.16)$$

式中， $\frac{L_s}{2\pi}$ ——丝杠螺母副的传动比， L_s 为丝杠的导程。

设初始条件为零，两端取拉普拉斯变换得

$$Y(s) = \frac{L_s}{2\pi} \theta_s(s) \quad (2.17)$$

将式 (2.17) 代入式 (2.15)，即可得机械传动链的传递函数 $G_1(s)$ 为

$$G_1(s) = \frac{Y(s)}{\theta(s)} = \frac{iK_s \frac{L_s}{2\pi}}{J_s s^2 + f_s s + K_s} = \frac{K_0 \omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (2.18)$$

其中， $K_0 = \frac{iL_s}{2\pi}$ ， $\xi = \frac{f_s}{2\sqrt{J_s K_s}}$ ， $\omega_n = \sqrt{\frac{K_s}{J_s}}$ 。

可见，对于机械传动装置，当输入为电机转角，输出为工作台位移时可将其视为一个二阶振荡环节。

2. 建立控制对象的传递函数

将上述几个环节按照系统的工作原理有机地组合起来，即可得到位置伺服系统的控制模型，如图 2.7 所示

如果我们把位置控制器之后的部分看成一个广义的控制对象，则可得到被控对象的传递函数为

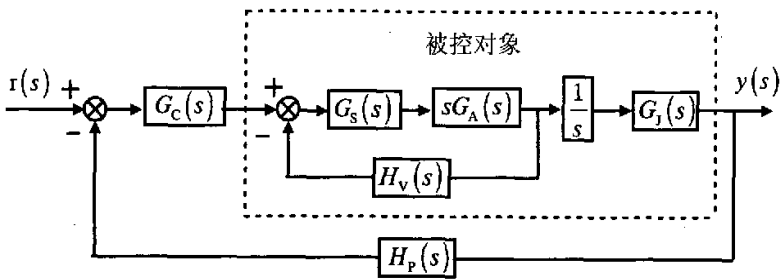


图 2.7 进给伺服系统的控制模型

$$G_0(s) = \frac{G_v(s)G_s(s)G_A(s)G_j(s)}{1+sH_v(s)G_v(s)G_s(s)G_A(s)} \quad (2.19)$$

$$= \frac{K_1}{s(T_1s^4 + T_2s^3 + T_3s^2 + T_4s + T_5)}$$

其中, $K_1 = K_vK_sK_AK_0\omega_n^2$;

$$T_1 = T_A;$$

$$T_2 = 2\xi\omega_n T_A + T_M;$$

$$T_3 = T_A\omega_n^2 + 2\xi\omega_n T_M + 1;$$

$$T_4 = T_M\omega_n^2 + 2\xi\omega_n;$$

$$T_5 = \omega_n^2 (K_{fv}K_vK_sK_AK_0 + 1);$$

由此可见, 这是一个五阶的控制对象。

由于对于容量较小的直流伺服电动机, 可取电枢回路总电感 $L_A = 0$, 即 $T_A = 0$; 忽略电动机的粘性阻尼系数 f_M , 则直流伺服电动机的传递函数可简化为^[14]

$$G_A(s) = \frac{K_A}{s(T_Ms + 1)} \quad (2.20)$$

其中, $K_A = \frac{1}{K_E}$, $T_M = \frac{J_MR_A}{K_EK_M}$ 。

对于机械传动装置, 如果忽略丝杠的折算惯量 J_s 和折算粘性阻尼 f_s ^[14], 则其传递函数可简化为

$$G_j(s) = K_0 = \frac{iL_s}{2\pi} \quad (2.21)$$

由此, 控制对象的模型可简化为

$$G_0(s) = \frac{K_sK_AK_0}{s[T_Ms + (K_vK_sK_AK_0 + 1)]} \quad (2.22)$$

即控制对象被简化成了二阶系统。

2.6 位置调节器设计方法分析

由于数控机床的最高移动速度、运动精度和定位精度等重要指标均取决于伺服系统的动态性能和静态性能，因此，研究和开发高性能的伺服系统一直是现代数控机床的关键技术之一。目前，数控机床的位置调节器仍然是采用一些传统的 PID 型控制器，如 P 型（比例型）、PI 型（比例—积分型）、PID 型（比例—积分—微分型）等^[16]。

2.6.1 PID 控制原理

PID 控制器是一种线性控制器，因其算法简单、鲁棒性好、可靠性高，在工业控制领域得到了广泛的应用，尤其适于能建立精确数学模型的系统。其基本原理如图 2.8 所示

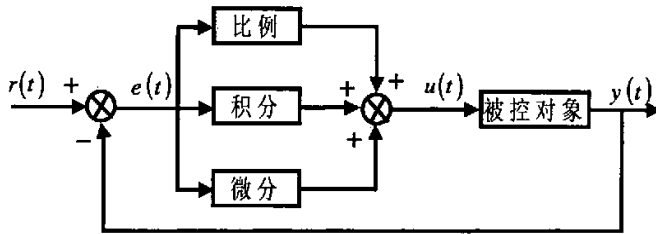


图 2.8 PID 控制器系统原理图

它根据给定值 $r(t)$ 与实际输出值 $y(t)$ 构成控制偏差 $e(t) = r(t) - y(t)$ ，将偏差的比例 (P)、积分 (I) 和微分 (D) 通过线性组合构成控制量，对被控对象进行控制，故称 PID 控制器。其控制规律为：

$$u(t) = K_p \left[e(t) + \frac{1}{T_I} \int_0^t e(t) dt + \frac{T_D}{dt} de(t) \right] \quad (2.23)$$

式中， K_p ——比例系数；

T_I ——积分时间常数；

T_D ——微分时间常数；

式(2.23)称为模拟 PID 控制算法。

由于现代的工业生产一般都采用计算机控制，而计算机控制是一种采用控制，它只能根据采样时刻的偏差值计算控制量，因此，式(2.23)中的积分项和微分项不能直接使用，必须对其进行离散化，由此便产生了数字 PID 控制算法。数字 PID 控制算法有两种表示形式：

$$u(k) = K_p e(k) + K_I \sum_{j=0}^k e(j) + K_D [e(k) - e(k-1)] \quad (2.24)$$

或

$$\Delta u(k) = K_p \Delta e(k) + K_i e(k) + K_D [\Delta e(k) - \Delta e(k-1)] \quad (2.25)$$

式中, k ——采样序号, $k=0,1,2,\dots$, 采样周期为 T ;

$u(k)$ ——第 k 次采样时刻的计算机输出值;

$e(k)$ ——第 k 次采样时刻输入的偏差值;

$e(k-1)$ ——第 $(k-1)$ 次采样时刻输入的偏差值;

K_p ——比例系数;

K_i ——积分系数, $K_i = K_p T / T_i$;

K_D ——微分系数, $K_D = K_p T_D / T$ 。

$\Delta e(k)$ ——误差的增量, $\Delta e(k) = e(k) - e(k-1)$ 。

其中, 式 (2.24) 称为位置式 PID 控制算法; 式 (2.25) 称为增量式 PID 控制算法, 对应的实际控制量 $u(k) = u(k-1) + \Delta u(k)$ 。

由于一般计算机控制系统采用固定的采样周期 T , 所以, 只要确定了 K_p 、 K_i 、 K_D , 即可根据采样得到的值, 构造出相应的控制量或控制量的增量。

因此, 简单说来, PID 控制器共包括三个环节, 即比例环节、积分环节和微分环节, 各环节的作用如下:

1、比例环节。即时成比例地反映控制系统的偏差信号, 偏差一旦产生, 控制器立即产生控制作用, 以减小偏差, 控制作用的强弱取决于比例系数 K_p 。但对于具有自平衡性 (即系统阶跃响应终值为一有限值) 的被控对象, 存在静差。

2、积分环节。它能对误差进行记忆并积分, 主要用于消除系统的静态误差, 提高系统的无差度。但它的不足之处是积分作用具有滞后特性, 积分控制作用太强会使系统的动态性能变差, 使系统变的不稳定。积分作用的强弱取决于积分时间常数 T_i , T_i 越大, 积分作用越弱, 反之越强。

3、微分环节。它能对误差进行微分, 敏感出误差的变化趋势, 在偏差信号值变得太大之前, 在系统中引入有效的早期修正信号, 从而加快系统的动作速度, 减少调节时间, 使系统尽快达到稳定。但它的缺点是对干扰同样敏感, 使系统抑制干扰的能力降低。

2.6.2 PID 控制参数的整定方法

1. 试凑法

试凑法是通过给模拟或实际的闭环系统施加某种典型的控制作用 (通常为阶跃信号), 然后观察系统的响应曲线, 根据各调节参数对系统响应的大致影响, 反复试凑参数, 以达到满意的响应, 从而确定 PID

控制器中的三个参数。人们在实践中总结出如下的规律：

(1) 增大比例系数 K_p 一般将加快系统的响应，在有静差的情况下，有利于减小静差。但过大的比例系数 K_p 会使系统出现较大的超调，并出现振荡，使系统的稳定性变坏。

(2) 增大积分系数 K_I 一般有利于消除系统的静差，使静差的消除速度加快。但过大的 K_I 会使系统产生积分饱和，从而导致系统出现较大的超调，甚至动荡。

(3) 增大微分系数 K_D 有利于加快系统的响应，减小振荡，使系统稳定性增加，但系统对干扰的抑制能力也随之减弱，对扰动有较敏感的反应；另外，过大的增大微分系数 K_D 也将使系统的稳定性变坏。

试凑时，可以参考以上的一般规律，对参数的调整顺序为先比例，后积分，再微分，即

(1) 先整定比例环节：将比例系数 K_p 由小逐渐增大，并观察相应的系统响应趋势，直到得到响应快、超调小的响应曲线。如果系统没有静差或静差已达到允许范围内，同时响应曲线已较令人满意，那么只需用比例控制器即可，最优比例系数也由此确定。

(2) 如果在比例调节的基础上系统的静差不能满足设计要求，则需加入比例环节。整定时，一般先设置一个较小的积分系数 K_I ，同时将第一步整定得到的比例系数 K_p 缩小一些（比如去原来值的 80%），然后再逐步增大积分系数，在保持系统有较好的动态性能的基础上使系统的静差得到消除。在此过程中，可以根据响应曲线的变化趋势反复地改变比例系数和积分系数，直到得到满意的控制效果。

(3) 如果使用比例积分控制器消除了静差，但系统的动态过程仍不满足，则可以加入微分环节，构成 PID 控制器。整定时，可先置微分系数 K_D 为零，然后逐步的增大微分系数，同时相应地改变比例系数和积分系数，反复试凑，直至获得满意的控制效果。

值得一提的是，PID 控制器的三个参数可以相互补偿，即某一个参数的减小可以由其它参数的增大或减小来补偿。因此，用不同的整定参数可以得到相同的控制效果，这也决定了 PID 控制器参数的非唯一性。另外，对于无自平衡能力的对象，则不应包括积分环节。在实时控制过程中，只要控制对象的主要性能指标达到了设计要求，就可以选定相应的控制器参数为最终参数^[15]。

2. Ziegler—Nichols 方法

ZN(Ziegler—Nichols)法是传统整定方法的典型，它是一种根据受控过程(或受控对象)的阶跃响应特性结合经验公式来计算控制器整定参

数的方法。它的实现是在系统闭环情况下，去除积分和微分作用（即 $K_I = K_D = 0$ ），使系统仅受纯比例控制器的作用，然后逐渐地增大比例系数 K_P ，直至系统产生等幅振荡，测出系统的临界振荡频率 K_C 和临界振荡周期 T_C ，然后按照如表 2.1 所示的 ZN 经验公式整定出相应的参数。

表 2.1 ZN 法整定公式

控制器类型	阶跃响应整定		
	K_P	T_I	T_D
P	$0.5 K_C$	—	—
PI	$0.45 K_C$	$0.8 T_C$	—
PID	$0.6 K_C$	$0.5 T_C$	$0.125 T_C$

2.6.3 常规 PID 控制的局限性

常规 PID 控制之所以具有强大的生命力，在于它具有以下几个优点：

第一，PID 控制对于大多数过程都具有良好的控制效果和鲁棒性；

第二，PID 控制算法原理简明，参数物理意义明确，理论分析体系完整且应用经验丰富；

第三，对于工业过程，人们的认识还远远不够。过程的动态特性大都具有高阶、非线性、大延迟及时变等特性，这给以精确数学模型为基础的现代控制理论的应用带来了困难。

然而，随着数控加工精度要求的提高，对控制器的要求也越来越高，传统的 PID 控制已显得越来越不适应，主要表现在以下几个方面：

首先，传统的 PID 器中参数的整定需要建立控制对象的精确数学模型。而现代高精度数控机床的加工过程往往具有非线性和时变性，难以建立精确的数学模型。

其次，传统的 PID 控制器缺乏自适应能力和自学习能力。传统的 PID 控制器的参数通常是针对某一具体工况进行整定的，一旦整定完毕，便固定不变。而随着加工精度要求的逐步提高，数控加工工艺日趋复杂，加工过程的特性在不断地发生变化，要保证加工精度就必须使控制器随着工况的变化进行的适当的调整，即具备自适应和自学习能力，采用固定参数的常规 PID 控制器显然无法满足这种要求。

再者，对于传统的 PID 控制器，参数的整定通常只能满足生产过程控制目标某一个方面的要求，例如快速性、准确性或稳定性等，为了尽量满足多方面的要求不得不采用折中的方案，因此造成控制器的性能未能达到最佳。而现代的高精度数控机床则要求控制器满足多方面的综合要求，例如既要满足响应的快速性，又要保证跟踪、定位的准确性，尤

其不允许出现超调，同时又要保证系统的稳定性；既要满足系统的“设定值跟踪特性”，又要提高系统的“干扰抑制特性”等等。

由此看来，仅采用常规的 PID 控制策略已越来越不能满足高精度数控机床的控制要求，必须对其进行适当的改进，以加快控制系统的响应速度，增强控制器的自适应、自学习能力和抗干扰能力，提高控制系统的定位、跟踪精度，从而保证加工出高精度的合格产品。

2.6.4 论文中控制思想的提出

长期以来，虽然人们为了提高伺服系统的性能而在不断地努力，也提出了一些新的控制算法，如前馈控制（Feedforward Control）、预测控制（Predictive Control）和学习控制（Learning Control）等，取得了一定的效果。但这些算法本身存在一定的不足之处，例如，前馈控制虽然从理论上来说可以完全消除系统的静态位置误差以及由外界扰动所引起的误差，但它的前提是把进给控制系统看成一个比例环节，而这事实上是很难实现的；预测控制则是通过预测整个机床的伺服传递函数，再改变伺服系统的输入量，以产生符合要求的输出，但这种方法无法克服扰动即传递函数误差对精度的影响；而学习控制需要在控制过程中产生高频反馈，但实际的伺服系统由于存在齿轮间隙和机械振动等因素，不会产生高频反馈^[3]。因此，这些方法虽然在一定程度上改善了伺服系统的性能，提高了数控机床的加工精度，但其自身存在的缺陷始终制约着它的进一步应用，成为影响伺服系统控制性能进一步提高的桎梏。

随着微处理机技术、智能控制理论（如模糊逻辑、神经网络等）和新型优化算法（如遗传算法等）的发展与应用，为控制复杂无规则系统开辟了新途径^[2]。模糊逻辑控制是一种仿人智能控制，它不需要建立被控对象的精确数学模型，而是将控制专家的经验知识转化成相应的控制规则，通过模糊推理，得出相应的控制量进行控制；神经网络也是一种不依赖控制对象精确数学模型的控制方法，它是模仿人的大脑的结构和功能组建起来的，具有较强的自适应、自学习适应能力、容错能力和泛化能力；而遗传算法则是一种基于生物进化过程的随机搜索的全局优化方法，它通过对个体编码串进行复制、交叉、变异等操作，使效果较差的个体逐步地被淘汰出局，而效果较好的个体则被保留下来，数量逐步增多，直至最终找到最佳的个体。由此联想到，如果能将这些新的控制策略与常规的 PID 控制相结合，或许可以找到更好的设计 PID 控制器的方法。正是基于这种思想，本论文提出将模糊逻辑、神经网络和遗传算法引入 PID 控制参数的整定过程中，希望能找到一种新的 PID 控制参数

整定方法，为设计新型的 PID 控制器，提高伺服系统的控制性能做一些理论上的探讨。

第 3 章 模糊神经网络与遗传算法

3.1 模糊逻辑控制

3.1.1 概述

模糊逻辑控制 (Fuzzy Logic Control, FLC) 是以模糊集合论、模糊语言变量及模糊逻辑推理为基础的计算机智能控制, 它在一定程度上模仿了人的控制, 其基本概念是由美国加利福尼亚大学著名教授扎德 (L. A. Zadeh) 首先提出的^[17]。它不需要控制对象的精确数学模型, 而是将控制专家或操作人员总结出来的描述各相关因素之间关系的经验知识转换成相应的模糊语言控制规则, 并将这些规则上升为简单的数值运算, 然后通过模糊推理得到控制量。这些经验规则的生成, 往往不是基于对各种因素之间关系的定量而严格的数学分析, 而是基于它们所进行的定性、大致精确地观察和总结。正因为如此, 实现这些语言性经验规则的数值运算也就无需严格准确地反映出各因素之间的精确的数学关系, 无需基于它们的精确数学模型。

模糊逻辑控制技术已在许多领域成功解决了传统控制方法无法解决或难以解决的问题, 取得了良好的效果。如家用电器、航空航天、图像处理、智能机器人等。概括起来, 模糊控制技术之所以得到广泛应用是由于它具有以下一些优点:

1. 直接采用语言型控制规则, 不需要被控对象的精确数学模型, 只需控制专家或操作工人的经验知识, 使得控制机理与策略易于接受, 便于应用。

2. 由过程的定性而非定量知识出发, 很容易建立语言变量的模糊语言规则, 并可不断地进行修改。

3. 系统的模糊语言控制规则具有相对的独立性, 利用这些控制规律间的模糊连接, 容易找到折中的选择, 使控制效果优于常规的控制。

4. 有利于模拟人工控制的过程和方法, 增强控制系统的适应能力, 使之具有一定的自学习和自适应能力。

5. 系统的鲁棒性强, 干扰和参数变化对控制效果的影响被大大减弱, 尤其适合于非线性、时变和纯滞后系统的控制。

3.1.2 模糊控制器的基本结构

按照输入变量的个数, 模糊控制器可以分为一维模糊控制器、二维

模糊控制器和三维模糊控制器。其中，一维模糊控制器的输入量为受控量的偏差，由于仅仅采用偏差值，因此很难反映受控过程的动态特性品质；二维模糊控制器的输入量通常为受控量的偏差和偏差的变化量或变化率，由于它们能够严格地反映受控过程中输出变量的动态特性，因此，在控制效果上要比一维模糊控制器好得多，也是目前采用较广泛的一种模糊控制器；而三维模糊控制器通常采用受控量的偏差、偏差的变化量或变化率以及偏差的二次变化率或偏差的累积作为输入量，由于这类模糊控制器结构较复杂，推理运算时间较长，因此，除非对系统的动态特性要求特别高的场合，一般较少选用三维模糊控制器^[2]。

二维模糊控制器的基本结构如图 3.1 所示

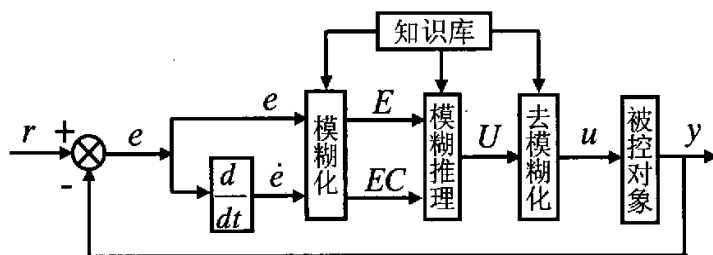


图 3.1 模糊控制器结构图

由图中可知，模糊控制器主要由以下四部分组成：

1. 模糊化

模糊化的作用是将输入的精确观测量映射为模糊控制器输入论域上的模糊集合。过程如下：

(1) 将输入的精确观测量进行尺度变换，使其变换到各自要求的论域范围。变换的方法可以是线性的也可以是非线性的。例如，若实际的输入量为 x_0 ，其变化范围为 $[x_{\min}^*, x_{\max}^*]$ ，要求的量化论域为 $[x_{\min}, x_{\max}]$ ，若采用线性变换，则

$$x_0 = \frac{x_{\min} + x_{\max}}{2} + k \left(x_0^* - \frac{x_{\min}^* + x_{\max}^*}{2} \right) \quad (3.1)$$

$$k = \frac{x_{\max} - x_{\min}}{x_{\max}^* - x_{\min}^*} \quad (3.2)$$

其中， k 称为量化因子。

量化论域可以是连续的也可以是离散的。如果要求离散的论域，则需要将连续的论域离散化或量化，量化可以是均匀的，也可以是非均匀的。

(2) 将已经变换到要求论域范围的输入量进行模糊化处理，使原先精确的输入量变成模糊量，并用相应的模糊集合来表示。实际上也就是确定输入量和输出量的隶属度函数。模糊化的方法有多种，其中较常用的有三种，分别是单点模糊集合法、三角形函数模糊集合法和铃形隶属度函数法。

a. 单点模糊集合

如果输入量数据 x_0 是精确的，则通常将其模糊化为单点模糊集合。设该模糊集合用 A 所示，则有

$$\mu(x) = \begin{cases} 1 & x = x_0 \\ 0 & x \neq x_0 \end{cases} \quad (3.3)$$

其隶属度函数如图 3.3 所示。

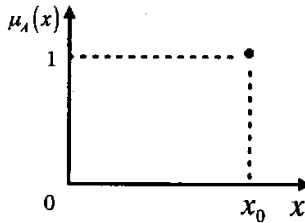


图 3.3 单点模糊集合的隶属度函数

b. 三角形模糊集合

如果输入量数据存在随机测量噪声，这时模糊化运算相当于将随机量变换为模糊量。对于这种情况，可以去模糊量的隶属度函数为等腰三角形，三角形的顶点相应于该随机数的均值，底边长度等于 2σ ， σ 表示该随机数据的标准差。其隶属度函数如图 3.4 所示

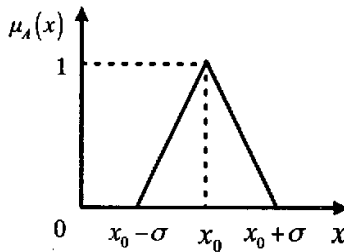


图 3.4 三角形模糊集合的隶属度函数

c. 另一种常用的方法是取隶属度函数为铃形函数，即

$$\mu_A(x) = e^{-\left(\frac{x-x_0}{\sigma}\right)^2} \quad (3.4)$$

其中， x_0 为隶属度函数的中心值， σ 为隶属度函数的标准差。其隶属度函数如图 3.5 所示

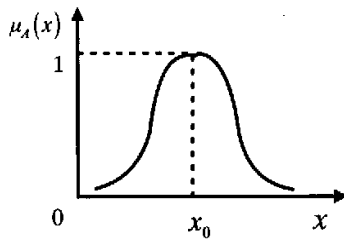


图 3.5 铃形模糊集合的隶属度函数

在模糊控制中，表示偏差、偏差的变化和控制量的模糊集合通常有：NB（负大）、NM（负中）、NS（负小）、ZE（零）、PS（正小）、PM（正中）、PB（正大）等。模糊集合的隶属度函数可以采用数值描述方法，也可以采用函数描述方法。对于论域为离散，且元素个数为有限时，模糊集合的隶属度函数可以用向量或者表格的形式来表示。而对于论域为连续的情况，隶属度函数常常用函数的形式来描述，最常见的有三角形函数、铃形函数、梯形函数等。

隶属度函数的形状对模糊控制器的性能有很大影响。当隶属度函数比较窄瘦时，控制较灵敏，反之，控制较粗略和平稳。通常当误差较小时，隶属度函数可取得较为窄瘦，当误差较大时，隶属度函数可取得宽胖些。

2. 知识库

知识库中包含了具体应用领域中的知识和要求的控制目标。它通常由数据库和模糊控制规则库两部分组成。

规则库包括了用模糊语言变量表示的一系列模糊控制规则，它们反映了控制专家的经验 and 知识。规则通常用“如果—则”式的模糊条件语句来表示，如：

R: 如果 x 是 A and y 是 B , 则 z 是 C 。

其中， x 、 y 、 z 均为模糊语言变量， A 、 B 、 C 为表示模糊语言变量取值的模糊集合。 x 、 y 称为前提的语言变量， z 称为结论的语言变量。 A 、 B 称为规则的前件， C 称为规则的后件。规则库就是由一系列这样的规则组成的，规则与规则之间用连接词“also”连接。

数据库主要包括各语言变量的隶属度函数、量化因子、比例因子以及输入输出模糊空间的分割数等。所谓模糊分割是要确定对于每个语言变量取值的个数，每个语言变量值就是一个模糊集合，如前面所述的 NB、NM、NS、ZE、PS、PM、PB 等，它们通常都具有一定的含义。模糊分割的个数决定了最大可能的模糊规则的个数，也就决定了控制精细化的程度。模糊分割数越多，控制规则数也越多，控制越精细，但太多的控制规会

加重机器的运算量，影响决策速度，所以模糊分割不可太细，否则需要确定太多的控制规则。当然，模糊分割数太小将导致控制太粗略，难以对控制性能进行精心的调整。目前尚没有一个确定模糊分割数的指导性的方法和步骤，主要靠经验和试凑。图 3.6 表示了两个模糊分割的例子，这里采用的是对称分布，也可以采用非对称分布，由具体问题决定。

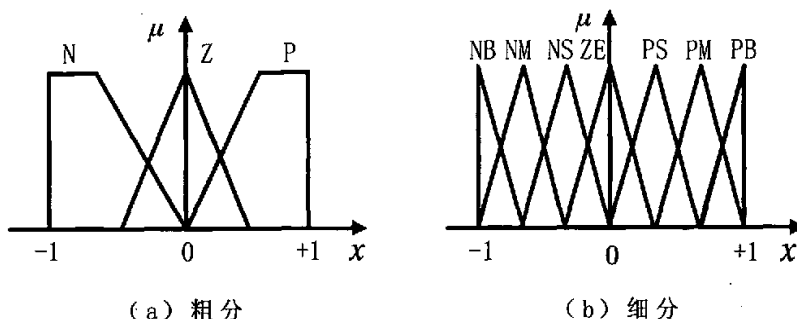


图 3.6 模糊分割的图形表示

3. 模糊推理

模糊推理是模糊控制器的核心，它具有模拟人的基于模糊概念的推理能力。该推理过程是基于模糊逻辑中的模糊蕴含关系运算法则及推理规则来进行的。其中模糊蕴含关系的运算方法有多种，可以证明，当模糊蕴含运算采用模糊蕴含最小运算或模糊蕴含积运算以及“also”采用求并运算给出了最好的控制效果，而且实现起来也最简单。

(1) 模糊蕴含最小运算 (Mamdani)

$$R_c = A \rightarrow B = A \times B = \int_{x \times y} \mu_A(x) \wedge \mu_B(y) / (x, y) \quad (3.5)$$

(2) 模糊蕴含积运算 (Mamdani)

$$R_p = A \rightarrow B = A \times B = \int_{x \times y} \mu_A(x) \mu_B(y) / (x, y) \quad (3.6)$$

(3) 模糊集合的求并运算

$$\mu_c(x) = \mu_A(x) \vee \mu_B(x) = \max[\mu_A(x), \mu_B(x)] \quad (3.7)$$

4. 去模糊化

去模糊化的作用是将模糊推理得到的模糊控制量变换为实际用于控制的清晰量。它包含以下两部分内容：

(1) 将模糊控制量经清晰化变换成表示在量化论域的清晰量。

(2) 将表示在论域范围的清晰量经比例因子作用后转换成能用于实际控制的量。

常用的清晰化方法有三种：最大隶属度函数法、中位数法和加权平均法（重心法）。

a. 最大隶属度函数法

若输出量模糊集合 C 的隶属度函数只有一个峰值，则取隶属度函数的最大值为清晰值，即

$$\mu_C(z_0) \geq \mu_C(z) \quad z \in Z \quad (3.8)$$

其中， z_0 表示清晰量。若输出量的隶属度函数有多个极值，则取这些极值的平均值为清晰值。

b. 中位数法

取 $\mu_C(z)$ 的中位数作为 z 的清晰值，即 z_0 满足

$$\int_a^{z_0} \mu_C(z) dz = \int_{z_0}^b \mu_C(z) dz \quad (3.9)$$

c. 加权平均法

这种方法取 $\mu_C(z)$ 的加权平均值为 z_0 ，即

$$z_0 = df(z) = \frac{\int_a^b z \mu_C(z) dz}{\int_a^b \mu_C(z) dz} \quad (3.10)$$

对于论域为离散的情况则有

$$z_0 = \frac{\sum_{i=1}^n z_i \mu_C(z_i)}{\sum_{i=1}^n \mu_C(z_i)} \quad (3.11)$$

在求得清晰值 z_0 后，还需要经尺度变换变为实际的控制量。

3.2 神经网络控制

3.2.1 概述

神经网络控制 (Neural Network Control, NNC) 也是一种基本上不依赖于模型的控制方法，它主要根据所提供的数据样本，通过学习和训练，找出输入与输出之间的内在联系，从而求得问题的解答。它采用高度并行处理的方法，大大加快了处理信息的速度，具有很好的逼近非线性映射的能力、自适应能力、泛化能力和良好的容错能力，比较适合于那些具有不确性或高度非线性的控制对象。

1. 人工神经元模型

人脑是由大量的神经细胞组合而成的，它们之间互相连接。每个神经细胞（也叫神经元）由细胞体、树突和轴突构成。细胞体是神经元的

中心，它一般由细胞核、细胞膜等组成。树突是神经元的主要接受器，它主要用来接受信息。轴突的作用主要是传导信息，它将信息从轴突起点传到轴突末梢，轴突末梢与另外一神经元的树突或细胞体构成一种突触的机构。通过突触实现神经元之间的信息传递。

人工神经元模型是利用物理器件来模拟生物神经网络的某些结构和功能，图 3.7 是最典型的人工神经元模型。

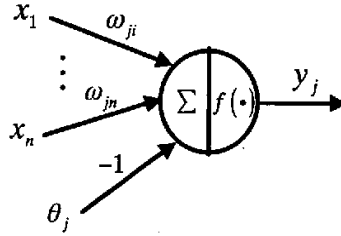


图 3.7 人工神经元模型

该神经元的输入输出关系为

$$s_j = \sum_{i=1}^n \omega_{ji} x_i - \theta_j = \sum_{i=0}^n \omega_{ji} x_i \quad (x_0 = \theta_j, \omega_{j0} = -1) \quad (3.12)$$

$$y_j = f(s_j) \quad (3.13)$$

其中， θ_j 称为阈值， ω_{ji} 称为连接权系数， s_j 称为神经元状态， $f(\cdot)$ 称为输出变换函数。

常用的输出变换函数有以下几种：

(a) 比例函数 $y = f(s) = s \quad (3.14)$

(b) 符号函数 $y = f(s) = \begin{cases} 1 & s \geq 0 \\ -1 & s < 0 \end{cases} \quad (3.15)$

(c) 双曲函数 $y = f(s) = \frac{1 - e^{-\mu s}}{1 + e^{-\mu s}} \quad (3.16)$

(d) s 型函数 $y = f(s) = \frac{1}{1 + e^{-\mu s}} \quad (3.17)$

(e) 阶跃函数 $y = f(s) = \begin{cases} 1 & s \geq 0 \\ 0 & s < 0 \end{cases} \quad (3.18)$

(f) 饱和函数

$$y = f(s) = \begin{cases} 1 & s \geq \frac{1}{k} \\ ks & -\frac{1}{k} \leq s < \frac{1}{k} \\ -1 & s < -\frac{1}{k} \end{cases} \quad (3.19)$$

其图形如图 3.8 所示。

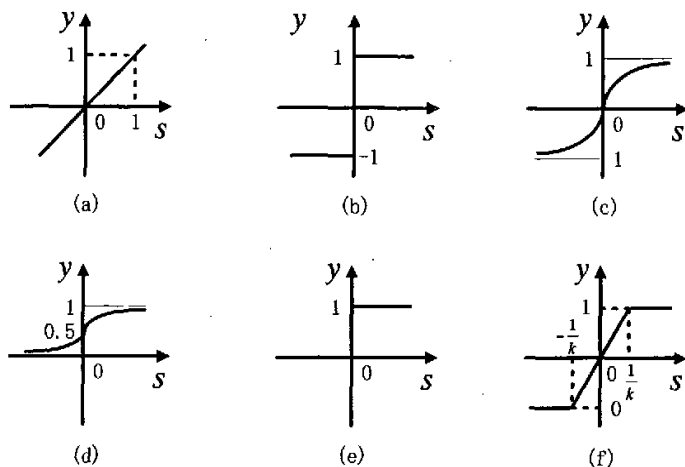


图 3.8 常用的输出变换函数

2. 人工神经网络

人工神经网络是一个并行和分布式的信息处理网络结构，该网络结构一般由许多个神经元组成，每个神经元有一个单一的输出，它可以连接到很多其他的神经元，其输入有多个连接通路，每个连接通路对应一个连接权系数。

按照结构的不同，人工神经网络可分为前馈型神经网络和反馈型神经网络。其结构如图 3.9 所示。其中，(a)为前馈型，(b)为反馈型。

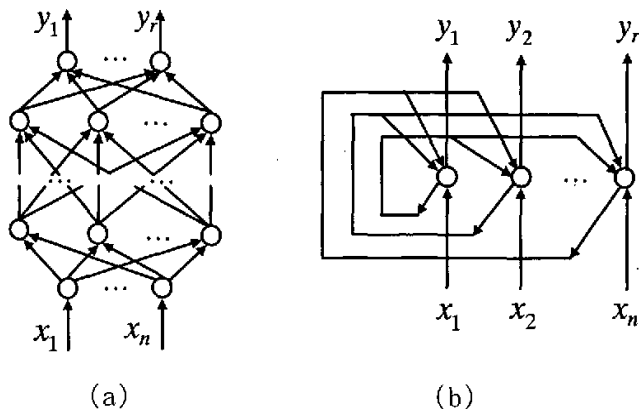


图 3.9 典型的神经网络结构

3.2.2 BP 网络

在神经网络控制中，最常用的网络结构是前馈型网络，其结构如图所示，它具有分层的结构，最下面一层是输入层，中间层是隐层，最上面一层是输出层。信息从输入层依次向上传递，直至输出层。其中，输入层不起变换作用，它只是把系统的输入按原值送给其它神经元结点。

而前馈型神经网络中最简单也是最常用的是 BP 网络，其神经元结点的变换函数采用 s 型函数，因此输出量是 0 到 1 之间的连续量，它可以实现从输入到输出的任意的非线性映射。之所以称其为 BP 网络，是因为其连接权系数和阈值的调整采用的是反向传播 (Back Propagation, BP) 的学习算法，它是由 D. E. Rumerlhart 和 J. L. McClelland 等人于 1986 年提出的。

3.2.2.1 BP 学习算法

设网络共有 Q 层，其中第 q 层的神经元结点数为 n_q ，第 $q-1$ 层的第 i 个结点到第 q 层的第 j 个结点的连接权系数为 $w_{ji}^{(q)} (i=1,2,\dots,n_{q-1}; j=1,2,\dots,n_q)$ ，网络的输入输出变换关系为

$$s_j^{(q)} = \sum_{i=0}^{n_{q-1}} w_{ji}^{(q)} x_i^{(q-1)} \quad (x_0^{(q-1)} = -1, w_{j0}^{(q)} = \theta_j^{(q)}) \quad (3.20)$$

$$x_j^{(q)} = f(s_j^{(q)}) = \frac{1}{1 + e^{-\mu s_j^{(q)}}} \quad (3.21)$$

$$i=1,2,\dots,n_{q-1}; j=1,2,\dots,n_q; q=1,2,\dots,Q$$

设给定 N 组输入输出样本

$$X(k) = [x_1^{(0)}(k) x_2^{(0)}(k) x_3^{(0)}(k) \dots x_{n_0}^{(0)}(k)]^T \quad (3.22)$$

$$d(k) = [d_1(k) d_2(k) d_3(k) \dots d_{n_0}(k)]^T \quad (3.23)$$

$$k=1,2,\dots,N$$

设取拟合误差的代价函数为

$$E = \frac{1}{2} \sum_{j=1}^{n_0} e_j^2(k) \quad (3.24)$$

现在的问题是如何调整连接权系数以使代价函数 E 最小。优化的方法很多，比较典型的是采用一阶梯度法，即最速下降法。由此，连接权系数的调整规律为

$$\Delta w_{ji} \propto -\frac{\partial E}{\partial w_{ji}} \quad (3.25)$$

经过推理，最终可得连接权系数的学习算法如下

$$w_{ji}^{(q)}(k+1) = w_{ji}^{(q)}(k) + \eta \left(-\frac{\partial E(k)}{\partial w_{ji}^{(q)}} \right) = w_{ji}^{(q)}(k) + \eta \delta_j^{(q)}(k) x_j^{(q-1)}(k) \quad (3.26)$$

$$\delta_j^{(q)}(k) = f' [s_j^{(q)}(k)] \sum_{i=1}^{n_{q+1}} \delta_i^{(q+1)}(k) w_{ij}^{(q+1)}(k) \quad (3.27)$$

$$\delta_j^{(0)}(k) = f' [s_j^{(0)}(k)] e_j(k) \quad (3.28)$$

$$i = 1, 2, \dots, n_{q-1} \quad j = 1, 2, \dots, n_q \quad q = Q, Q-1, \dots, 1$$

其中 η 称为学习率或学习步长， $\eta > 0$ 。

概括起来，BP 网络的主要优点是：

(1) 只要隐层数目足够多，BP 网络可以以任意精度逼近任意的非线性映射关系。

(2) 具有自适应功能。它主要根据所提供的数据，通过学习和训练，找出输入与输出之间的内在联系，从而求得问题的解答，而不是依靠对问题的先验知识和规则，因而它具有很好的适应能力。

(3) 具有较好的泛化能力，即对于不是样本集中的输入也能给出合适的输出，从函数拟合的角度，它说明 BP 网络具有插值功能；

(4) 由于是将输入与输出之间的关联信息分布地存储于连接权中，而连接权的数量又较多，个别连接权的损坏只对输入输出关系有较小的影响，因此，BP 网络具有较好的容错能力。

但它也有不足之处，主要有以下几个方面：

(1) 由于 BP 算法只有在所有的连接权系数都调整完毕时，网络的误差代价函数才能达到最小，也就是说它采用的是全局逼近的方法，因此网络的收敛速度较慢。

(2) 难以确定隐层和隐结点的数目，只能靠经验和试凑。

(3) 由于在函数的局部极值点处梯度为零，搜索停止，因此，网络易收敛于局部极值，而不是全局极值。

(4) 学习步长难以确定，取得太小会导致收敛太慢，而太大又可能导致一步跳过极值点，出现在极值点附近反复搜索，导致系统出现振荡。

3.2.2.2 BP 学习算法的改进

为了加快算法的收敛速度，人们提出了改进的 BP 算法，常用的主要有以下两种：

1. 引入动量项

$$\omega(k+1) = \omega(k) + \eta [(1-\alpha)D(k) + \alpha D(k-1)] \quad (3.29)$$

其中, η ——学习率或学习步长, $\eta > 0$

α ——动量项因子, $0 \leq \alpha < 1$

$$D(k) = -\frac{\partial E(k)}{\partial \omega_{\mu}}$$

该方法所加入的动量项实质上相当于阻尼项, 它减小了学习过程的振荡趋势, 改善了收敛性, 是目前应用比较广泛的一种改进方法。

2. 变步长法

$$w(k+1) = w(k) + \eta(k)D(k) \quad (3.30)$$

$$\hat{\eta}(k) = 2^{\lambda} \eta(k-1) \quad (3.31)$$

$$\lambda = \text{sgn}[D(k)D(k-1)] \quad (3.32)$$

即当连续两次迭代其梯度方向相同时, 说明下降太慢, 这时可使步长加倍; 当连续两次迭代其梯度方向相反时, 说明下降过头, 这时可使步长减半。

也可将上述两种方法结合起来:

$$w(k+1) = w(k) + \eta(k)[(1-\alpha)D(k) + \alpha D(k-1)] \quad (3.33)$$

3.3 模糊神经网络

由前面的介绍可知, 模糊逻辑是一种处理不确定性和非线性问题的有力工具, 比较适合于表达那些模糊或定性的知识, 其推理方式比较类似于人的思维模式。但它也存在一定的局限性, 一方面关于模糊控制器的隶属度函数和控制规则的确定目前还没有一套严谨而精确的指导理论, 主要依靠控制专家的经验知识产生, 带有较大的主观性; 另一方面就是一旦模糊模型建立以后, 很难适应变化了的情况, 即缺乏自学习和自适应能力。

而神经网络具有并行计算、分布式信息存贮、容错能力较强以及具备自适应学习功能等一系列优点。但却不适于表达基于规则的知识, 因此在对神经网络进行训练时, 由于不能很好地利用已有的经验知识, 常常只能将初始权值取为零或随机数, 从而增加了网络的训练时间或容易陷入局部极值, 这是神经网络的一个不足^[12]。

由此想到, 若能将两者适当地结合起来, 吸取两者的长处, 则可组成比单独的神经网络或者单独的模糊逻辑控制性能更好的系统。

3.3.1 模糊逻辑与神经网络的结合方式

所谓模糊神经网络 (Fuzzy Neural Network, FNN) 就是将模糊逻辑控制与神经网络相结合。结合的方法有很多种, 概括起来有以下四种:

1. 模糊逻辑与神经网络的简单结合

在这种结合方式中, 模糊逻辑与神经网络各自以其独立的方式存在, 并各自起着一定的作用。

a. 松散型结合。对于系统中能用“如果—则”规则表达的部分, 用模糊逻辑系统来描述; 而对于很难用“如果—则”规则表达的部分, 用神经网络来描述, 两者之间没有直接联系, 各自完成各自的任务, 利用各自的优点和长处, 取长补短, 共同完成对系统的控制。

b. 并联型结合。即模糊逻辑与神经网络按并联方式连接, 享用共同的系统输入, 然后对各自产生的结果进行组合, 得出实际所需的控制量。

c. 串联型结合。即模糊逻辑与神经网络按串联方式连接, 其中一方的输出同时也是另一方的输入。这种情况可看成是两段处理或者是串联中的前者是后者的预处理部分。例如, 应用神经网络预处理模糊系统的输入数据, 可实现数据整理和抑制噪声, 使模糊逻辑中规则的获取变得更容易。

2. 用模糊逻辑增强的神经网络

这种结合方式的主要目的是用模糊逻辑作为辅助工具, 增强神经网络的学习和适应能力, 克服神经网络容易陷入局部极值的弱点。另外, 为提高神经网络的训练速度, 还可用模糊逻辑设计神经网络的初始连接权系数。

3. 用神经网络增强的模糊逻辑

这种类型的模糊神经网络是用神经网络作为辅助工具, 以便更好地设计模糊逻辑控制器。

a. 网络学习型的结合。模糊系统设计的关键是如何获取专家知识, 传统方法难以有效地获取规则和调整隶属度函数, 难以实现自适应学习。而神经网络具有自学习和适应功能, 可以克服这一缺点。

b. 基于知识扩展型的结合。要实现模糊逻辑系统的自学习能力就需要两者进行知识交换, 但问题是随着推理路径的增长, 模糊程度会加大, 这是我们所不希望的, 而神经网络的联想记忆功能可以解决此问题。这种结合的目的是为了扩展知识库, 并对知识库进行不断地修正, 以增强

系统的学习能力。这种学习能力是靠模糊逻辑与神经网络之间不断地进行知识交换而实现的。

4. 模糊逻辑与神经网络增强的完全融合

自 1990 年以来,这种结合形式的模糊神经网络一直是众多学者研究的热点。即借用神经网络的结构来实现模糊逻辑推理,也就是借鉴模糊逻辑控制的思路设计一些特殊结构的神经网络,从而使神经网络的连接权系数具有模糊逻辑推理中参数的物理含义。

3.3.2 基于标准模糊模型的模糊神经网络

3.3.2.1 模糊系统的标准模型

设描述输入输出关系的模糊规则为:

R_i : 如果 x_1 是 A_1^i and x_2 是 $A_2^i \cdots$ and x_n 是 A_n^i , 则 y 是 B^i

其中 $i=1,2,\dots,m$, m 表示规则总数, $m \leq m_1 m_2 \cdots m_n$ 。

若输入量采用单点模糊集合的模糊化方法,则对于给定的输入 \bar{x} , 可以求得对于每条规则的适用度为

$$\alpha_i = \mu_{A_1^i}(x_1) \wedge \mu_{A_2^i}(x_2) \wedge \cdots \wedge \mu_{A_n^i}(x_n) \quad (3.34)$$

或

$$\alpha_i = \mu_{A_1^i}(x_1) \mu_{A_2^i}(x_2) \cdots \mu_{A_n^i}(x_n) \quad (3.35)$$

通过模糊推理可得对于每一条模糊规则的输出量模糊集合 B_i 的隶属度函数为

$$\mu_{B_i}(y) = \alpha_i \wedge \mu_{B^i}(y) \quad (3.36)$$

或

$$\mu_{B_i}(y) = \alpha_i \mu_{B^i}(y) \quad (3.37)$$

从而输出量总的模糊集合为

$$B = \bigcup_{i=1}^m B_i \quad \mu_B(y) = \bigvee_{i=1}^m \mu_{B_i}(y) \quad (3.38)$$

若采用加权平均的清晰化方法,则可求得输出的清晰化量为

$$y = \frac{\int_{\underline{U}_y}^{\overline{U}_y} y \mu_B(y) dy}{\int_{\underline{U}_y}^{\overline{U}_y} \mu_B(y) dy} \quad (3.39)$$

由于计算上式的积分比较麻烦,实际计算时通常采用如下的近似公式

$$y = \frac{\sum_{i=1}^m y_{c_i} \mu_{B_i}(y_{c_i})}{\sum_{i=1}^m \mu_{B_i}(y_{c_i})} \quad (3.40)$$

其中 y_{c_i} 是使 $\mu_{B_i}(y)$ 取最大值的点，它一般也就是隶属度函数的中心点。显然

$$\mu_{B_i}(y_{c_i}) = \max_y \mu_{B_i}(y) = \alpha_i \quad (3.41)$$

从而输出量的表达式可变为

$$y = \frac{\sum_{i=1}^m y_{c_i} \alpha_i}{\sum_{i=1}^m \alpha_i} = \sum_{i=1}^m \left(\frac{\alpha_i}{\sum_{i=1}^m \alpha_i} \right) y_{c_i} = \sum_{i=1}^m \bar{\alpha}_i y_{c_i} \quad (3.42)$$

3.3.2.2 网络结构

根据前面给出的模糊系统的模型，可以设计出如图 3.10 所示基于标准模糊模型的模糊神经网络，图中所示为多输入多输出系统。

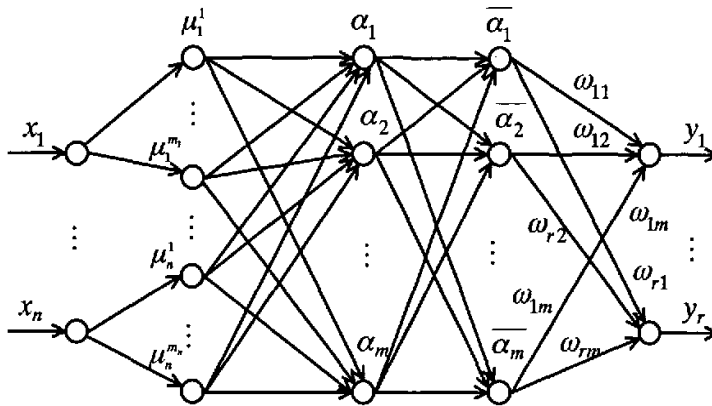


图 3.10 基于标准模型的模糊神经网络

图中，第一层为输入层。该层的各个结点直接与输入向量的各个分量 x_i 连接，它起着将输入值 $\bar{x} = [x_1 x_2 \dots x_n]^T$ 传送到下一层的作用。该层的结点数 $N_1 = n$ 。

第二层的每个结点代表一个语言变量值，如 NB, PS 等。它的作用是计算各输入分量属于各语言变量值模糊集合的隶属度函数 μ_i^j ，其中

$$\mu_i^j = \mu_{A_i^j}(x_i) \quad (3.43)$$

$i=1,2,\dots,n;m=1,2,\dots,m_i$ 。 n 是输入向量的维数， m_i 是 x_i 的模糊分割数。例如，若隶属度函数采用高斯型函数表示的铃形函数，则

$$\mu_i^j = e^{-\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}} \quad (3.44)$$

其中 c_{ij} 和 σ_{ij} 分别表示隶属度函数的中心和宽度。该层的结点数 $N_2 = \sum_{i=1}^n m_i$ 。

第三层的每个结点代表一条模糊控制规则，它的作用是用来匹配模糊控制规则的前件，计算出每条规则的适用度。即

$$\alpha_j = \min \{ \mu_1^{i_1}, \mu_2^{i_2}, \dots, \mu_n^{i_n} \} \quad (3.45)$$

或者

$$\alpha_j = \mu_1^{i_1} \mu_2^{i_2} \dots \mu_n^{i_n} \quad (3.46)$$

其中 $i_1 \in (1, 2, \dots, m_1), i_2 \in (1, 2, \dots, m_2), \dots, i_n \in (1, 2, \dots, m_n), j=1, 2, \dots, m, m = \prod_{i=1}^n m_i$ 。该层的结点数 $N_3 = m$ 。

对于给定的输入，只有在输入点附近的那些语言变量值才有较大的隶属度值，远离输入点的语言变量值的隶属度或者很小（高斯隶属度函数），或者为0（三角形隶属度函数）。当隶属度函数很小（例如小于0.05）时近似取为0。因此，在网络第三层中只有少量结点的输出为非0，而多数结点的输出为0。故模糊神经网络属于局部逼近网络的一种。

第四层的结点数与第三层相同，即 $N_4 = N_3 = m$ ，它的作用是实现归一化计算，即

$$\bar{\alpha}_j = \alpha_j / \sum_{i=1}^m \alpha_i \quad j=1, 2, \dots, m \quad (3.47)$$

第五层是输出层，它所实现的是清晰化计算，即

$$y_i = \sum_{j=1}^r \omega_{ij} \bar{\alpha}_j \quad i=1, 2, \dots, r \quad (3.48)$$

这里的 ω_{ij} 相当于 y_i 的第 j 个语言变量值隶属度函数的中心值。该层的结点数 $N_5 = r$ 。

3.3.2.3 学习算法

假设各输入分量的模糊分割数是预先确定的，那么需要学习的参数主要是最后一层的连接权 $\omega_{ij} (i=1, 2, \dots, r; j=1, 2, \dots, m)$ 以及第二层的隶属度函数的中心值 c_{ij} 和宽度 $\sigma_{ij} (i=1, 2, \dots, n; j=1, 2, \dots, m_i)$ 。由于上面所给出的模糊神

神经网络本质上也是一种多层前馈网络，所以可以仿照 BP 网络用误差反向传播的方法来设计调整参数的学习算法。

模糊神经网络虽然属于局部逼近网络，但它是按照模糊控制系统的模型建立的，网络中的各个结点及所有参数均有明显的物理意义，因此，这些参数的初值可以根据系统的模糊或定性的知识来加以确定，然后利用上面所述的算法进行训练，可以很快地收敛到要求的输入输出关系，这是模糊神经网络比单纯的神经网络的优点所在。同时又由于它具有神经网络的结构，因而参数的学习和调整比较容易，这是它比单纯的模糊逻辑系统的优点所在。

3.4 遗传算法

3.4.1 概述

遗传算法 (Genetic Algorithm, GA) 是一种基于自然选择和基因遗传学原理的随机搜索算法。它将“适者生存”这一基本的达尔文进化理论引入串结构，并且在串与串之间进行有组织但又随机的信息交换。伴随着算法的运行，优良的品质被逐渐保留并加以组合，从而不断产生出更佳的个体。这一过程就如生物进化过程一样，好的特征被不断地继承下来，坏的特性被逐渐淘汰。新一代个体中包含有上一代个体的大量信息，新一代的个体不断地在总体特性上胜过旧的一代，从而使整个群体向前进化发展^[18]。

首先我们来考察常规的寻优方法，按照一般文献的介绍，寻优的方法主要有三种类型：解析法、枚举法和随机搜索法。

解析法是研究的最多的一种，它一般又分为间接法和直接法。间接法是通过让目标函数的梯度为零，进而求解一组非线性方程来寻求极值。而直接法则是按照梯度信息沿最陡的方向逐次运动来寻求极值。这两种方法的主要缺点是：一方面它们只能寻找局部极值而非全局极值；另一方面它们要求目标函数必须是连续光滑的、可导的。这就使得其鲁棒性能较差。

而枚举法虽可以克服上述解析法的两个缺点，即它可以找到全局极值，而且也不需要目标函数必须是连续光滑的，但其计算效率太低，对于一个实际问题，常常由于太大的搜索空间而不可能将所有的情况都搜索到。

鉴于以上两种寻优方法有严重缺陷，随机搜索法受到人们的青睐。它通过在搜索空间中随机地漫游并随时记录下所取得的最好结果。但出

于效率的考虑，搜索到一定程度便终止，因此，所得结果一般并不是最优值。正是由于这些缺点，使得这些常规的寻优方法的应用前景受到了限制。

遗传算法虽然也用到了随机搜索技术，但它不同于上述的随机搜索，它通过对寻优参数进行编码并用随机选择作为工具引导搜索过程向着更高效的方向发展。对于待寻优的函数基本无限制（如连续、导数存在、可微及单峰等），而且可以有效地避免搜索过程收敛于局部最优解，并且有较大的可能求得全部最优解，同时由于采用了并行计算，使得计算速度大大提高。正是由于它的这些优点使得其鲁棒性能远远优于常规的寻优方法，从而在工程上得到了广泛的应用，应用前景十分广阔。

总的说来，与其他寻优算法相比，它具有计算简单、功能强大、鲁棒性好的特点，其主要的本质差别有以下几点：

（1）遗传算法是对参数的编码进行操作，而不是参数本身。

（2）遗传算法是从许多初始点开始并行操作，而不是从一个点开始，因而可以有效地防止搜索过程收敛于局部最优解，而且有较大可能收敛于全局最优解。

（3）遗传算法通过目标函数来计算适应度，而不需要其他的推导和附属信息，从而对问题的依赖性小。

（4）遗传算法使用概率的转变规则，而不是确定性的规则。

（5）遗传算法在解空间内不是盲目地穷举或完全随机测试，而是一种启发式搜索，其搜索效率往往优于其它方法。

（6）遗传算法对于待寻优的函数基本无限制（如连续、导数存在、可微及单峰等），既可以是数学解析式所表达的显函数，也可以是映射矩阵甚至是神经网络等隐函数，因而应用范围较广。

（7）遗传算法具有并行计算的特点，因而可通过大规模并行计算来提高计算速度。

（8）遗传算法更适合大规模复杂问题的优化。

3.4.2 遗传算法的工作原理及基本操作

遗传算法是一种基于生物进化过程的随机搜索的全局优化方法。它通过对寻优参数进行编码并用随机选择作为工具引导搜索过程向着更高效的方向发展。一个标准的遗传算法一般包括以下三个步骤：

第一步，初始化。包括两方面的内容，一是确定寻优参数的编码方案，并根据编码方案构造合适的适应度函数，以便对个体的性能作出评价；二是产生初始种群。

所谓编码就是将一个问题的可行性解从解空间转换到遗传算法能够处理的搜索空间的一种转换方法。编码方法因优化问题的不同而不同。目前人们已提出了许多种编码方法。总的说来，可以分为以下几类：二进制编码方法、格雷码编码方法、浮点数编码方法、符号编码方法和多参数级联定点映射编码方法等。其中最常用的有二进制编码方法、浮点数编码方法和多参数级联定点映射编码方法^[19]。

其中，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 组成的二值符号集 $\{0, 1\}$ ，所构成的个体基因型是一个二进制编码符号串。编码长度与问题所要求的求解精度有关。具体内容后面叙述。

所谓浮点数编码方法是指个体的每个基因值用某一范围内的一个浮点数表示，个体的编码长度等于决策变量的个数。因为这种编码方法使用的是决策变量的真实值，所以又叫真值编码方法。它是针对一些多维、高精度要求的连续函数优化问题提出的。因为此时使用二进制编码将会有一些不利之处。首先，二进制编码存在着连续函数离散化时的映射误差；其次，二进制编码不便于反映所求问题的特定知识；再者，参数较多时，编码长度较长，搜索空间将急剧扩大，导致搜索时间过长，收敛过慢。

遗传算法的一个重要特点就是它仅使用所求问题的目标函数值就可得到下一步的有关搜索信息。而对目标函数值的使用是通过评价个体的适应度来体现的。因此，就存在目标函数到适应度函数的映射问题。而映射方式则取决于最优化问题的类型。由于适应度是非负的，任何情况下总是希望越大越好。而目标函数值有正有负，甚至可能是复数值，并且适应度函数与目标函数值之间的关系也是多种多样的。因此映射应保证两条：首先，应保证适应度是非负的；其次目标函数的优化方向应对应于适应度的增大方向。同时，为了防止因初始种群中含有少数几个非凡个体（即适应度较大）而导致算法早熟收敛，或者在算法的结束阶段因种群的平均适应度接近最大适应度而导致选择算子变成随机的步骤，有必要对适应度进行适当的调整。调整的方法有三种：线性调整法、乘幂调整法和指数调整法。

初始种群的产生方法视编码方法而定。对于采用二进制编码的遗传算法，初始种群的产生可以采用掷硬币的方法，正面代表 1，反面代表 0，每掷一次产生个体串中的一位，连续掷 m 次即可产生一个个体。若用计算机，可考虑首先产生 0-1 之间均匀分布的随机数，然后规定产生的随机数在 0-0.5 之间代表 0，在 0.5-1 之间代表 1。而对于采用浮点数编码

方法的遗传算法，则可以通过计算机中的随机数发生函数来产生。

第二步，进行遗传操作。即运用复制、交叉和变异等遗传算子对种群中的个体进行运算。

第三步，终止条件判断。当算法满足一定的终止条件时，即可认为算法已收敛，终止算法运行，当前群体中的最优个体即为最终的寻优结果。否则，返回步骤二继续执行，直到满足终止条件为止。

1. 复制

遗传算法中的复制 (reproduction) 操作是指从父代种群中选择生命力强的个体产生新种群的过程。其主要目的是为了避开基因缺失、提高算法的全局收敛性和提高运算效率。复制操作是按照个体适应度的大小来进行的，适应度越大的串在下一代中将有更多的机会提供一个或多个子孙，即被复制的概率或次数会更多。产生的新个体放入匹配池中。

根据优化问题的不同，人们提出了许多种选择算子，如比例选择、最优保留策略、确定式采样、无回放随机选择、无回放余数随机、排序选择和联赛选择等。其中最常用的有比例选择、最优保留策略。下面具体阐述一下最优保留策略。

在遗传算法的运行过程中，虽然随着群体的进化过程会产生越来越多的优良个体，但由于交叉、变异等操作的随机性，它们也有可能破坏掉适应度最高的个体，而这会降低群的平均适应度，并对遗传算法的收敛性和运行效率产生不利的影响。为此，人们提出了最优保留策略，它可以保证当前代中适应度最高的个体不受交叉和变异等操作的影响而直接保留到下一代群体中。具体操作过程是：

(1) 找出当前群体中适应度最高的个体和适应度最低的个体。

(2) 若当前群体中最佳个体的适应度比迄今为止所有代中的最佳个体的适应度还要高，则以当前群体中的最佳个体作为新的迄今为止的最佳个体。

(3) 用迄今为止的最佳个体替换当前群体中的最差个体。

另外，最优保留策略还可加以推广，即在每一代的进化过程中，保留多个最优个体不参加交叉和变异等遗传操作，而将它们直接复制到下一代。这种选择方法也称为稳态复制。

2. 交叉

遗传算法中的交叉运算是指相互配对的两个个体按某种方式相互交换其部分基因，从而形成两个新个体的过程。它是遗传算法区别于其它进化算法的重要特征，在遗传算法中起着关键作用，是产生新个体的主要方法。

具体地讲，交叉操作可分为两个步骤，第一步是将新复制产生的匹配池中的成员随机地两两配对；第二步是进行交叉繁殖。

交叉算子的设计和实现与所研究的问题密切相关，一般要求它既不要太多地破坏个体编码串中表示优良性状的优良模式，又要能够有效地产生出一些较好的新个体模式。另外，交叉算子的设计要和个体编码设计统一考虑。目前，人们已发展出多种交叉算子，如单点交叉、两点交叉、多点交叉、均匀交叉和算术交叉等。其中较常用的是单点交叉、两点交叉。对于采用浮点数编码的个体，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。

3. 变异

遗传算法中的变异运算是指以很小的概率随机地将个体编码串的某些基因座上的基因值用该基因座的其它等位基因替换，从而形成一个新个体的过程。变异的概率通常是很小的，一般只有千分之几。

变异算子的设计包含两方面内容，一是如何确定变异点的位置；二是如何进行基因值替换。

目前变异算子有基本位变异、均匀变异、边界变异、非均匀变异和告斯变异等，其中较常用的是基本位变异。对于采用浮点数编码的个体，变异运算用决策变量取值范围内的随机数来替换原基因值。

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，其作用是第一位的；而变异运算只是产生新个体的辅助方法，其作用是第二位的，但也是必不可少的一个步骤。一方面它可以改善遗传算法的局部搜索能力，使个体更加逼近最优解。因为仅使用交叉算子虽可以接近最优解，但无法对搜索空间的细节进行局部搜索，有可能无法到达最优解；另一方面变异算子用新的基因值替换原有基因值，从而可以改变个体编码串结构，防止因遗传算法的进行而丢失一些有用的遗传因子，维持群体的多样性，这样就有助于防止出现早熟现象。交叉算子和变异算子相互配合，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以良好的搜索性能完成最优化问题的寻优过程。

3.4.3 遗传算法的有效性及其运行机理分析

3.4.3.1 遗传算法的模式理论

一个模式就是一个描述种群中在位串的某些确定位置上具有相似性的位串子集的相似模板。这些相似性与个体的适应度高低之间存在某种因果关系。例如，凡是以 1 开头的位串，其适应度就高；以 0 开头的位

串，其适应度就低。而这种相似性正是遗传算法有效工作的因素。根据对种群中高适应度位串之间的相似性的分析，Holland 提出了模式理论。

所谓模式就是有三个元符号 $\{0, 1, *\}$ 所组成的一个字符串，其中“*”代表通配符，可以是 0 或 1。如 $A=011*0$ 。模式通常用 H 来表示。我们称一个模式与一个特定的串匹配是指该模式中的 1 与串中的 1 相匹配，模式中的 0 与串中的 0 相匹配，模式中的“*”可以匹配串中的 0 或 1。如上面的模式 A 匹配两个串： $\{01100, 01110\}$ 。

一般地，假定字符表的基数为 k ，例如 $\{0, 1\}$ 的基数为 2，则定义在该字符表上的长度为 l 的串可能包含的最大模式数为 $(k+1)^l$ ，原因是在 l 个位置中的任何一个位置上都可以取 k 个字符中的一个及通配符“*”，即共有 $k+1$ 个不同的表示，因此 l 个位置的全排列数为 $(k+1)^l$ 。而位串的数量仅为 k^l ，所以模式的数量要大于位串的数量。例如，对于二进制，其基数为 2（对应 0，1），则长度为 4 的串可能代表的最大模式数为 $3^4=81$ ，而位串的数量仅为 $2^4=16$ 。

对于任一长度为 l 的给定位串，其中所包含的模式数为 k^l 个。因为在 l 个位置中的任一位置上除了其确定值外，还可以取通配符“*”，即任一位置上都有两种不同表示。因此对于规模为 n 的种群，其所包含的模式数为 $2^l \square n \cdot 2^l$ 个。

为论述方便，首先定义一些名词术语。

模式的阶数：指模式中确定位置（对于二进制，即 0 或 1 所在的位置）的个数，通常用符号 $O(H)$ 来表示。

模式的定义长度：指模式中第一个确定位置与最后一个确定位置之间的距离，通常用符号 $\delta(H)$ 来表示。

下面我们来分析遗传算法的几个重要操作对模式的影响。

1. 复制对模式的影响

设在给定的时间 t ，种群 $A(t)$ 中包含 m 个特定模式 H ，记为 $m=m(H, t)$ 。在复制过程中， $A(t)$ 中的任何一个串 A_j 以概率 $f_j/\sum f_i$ 被选中进行复制。因此，我们可以期望在复制完成后，在 $t+1$ 时刻，特定模式 H 的数量将变为

$$m(H, t+1) = m(H, t) n f(H) / \sum_{i=1}^n f_i = m(H, t) f(H) / \bar{f} \quad (3.49)$$

其中， $f(H)$ 表示在时刻 t 时对应于模式 H 的串的平均适应度。 $\bar{f} = \sum_{i=1}^n f_i / n$

为整个群体的平均适应度。

可见，经过复制操作以后，特定模式的数量将按照该模式的平均适应度与整个种群平均适应度的比值成比例地改变。换言之，适应度高于种群平均适应度的模式在下一代中的数量将增加，而低于种群平均适应度的模式在一代中的数量将减少。另外，种群 $A(t)$ 中的所有模式 H 的处理是并行进行的，即所有模式经复制操作后，均同时按照其平均适应度占种群平均适应度的比例进行增减。所以，可以概括地说，复制操作对模式的影响是使得平均适应度高于种群平均适应度的模式的数量增加，低于种群平均适应度的模式的数量减少。

为了进一步分析高于平均适应度得的模式的数量增长，设

$$f(H) = (1+c)\bar{f} \quad (3.50)$$

则上面的方程可以改成如下的差分方程

$$m(H, t+1) = m(H, t)(1+c) \quad (3.51)$$

假定 c 为常数时，可得

$$m(H, t) = m(H, 0)(1+c)^t \quad (3.52)$$

可见，对于平均适应度高于种群平均适应度的模式的数量将呈指数形式增长。

对复制过程的分析可以看到，虽然复制过程成功地以并行方式控制着模式数量以指数形式增减，但由于复制只是将某些高适应度个体全盘复制，或是丢弃某些低适应度个体，而决不会产生新的模式结构，因而对性能的改进是有限的。

2. 交叉对模式的影响

交叉过程是串与串之间的有组织的而又随机的信息交换，它在创建新结构的同时，最低限度地破坏复制过程所选择的高适应度模式。为了观察交叉对模式的影响，下面考察一个长度为 7 的串以及此串所包含的两个代表模式。

$$A = 0111000$$

$$H_1 = *1****0$$

$$H_2 = ***10**$$

交叉的过程是随机选择一个匹配伙伴，再随即选择一个交叉位置，然后互换相应的片断。假定对上面给定的串，选择交叉点为 3，则很容易看出它对两个模式 H_1 和 H_2 的影响。

$$A = 011:1000$$

$$H_1 = *1*:***0$$

$$H_2 = ***:10**$$

除非串 A 的匹配伙伴在模式的固定位置与 A 相同(我们忽略这种可能性), 模式 H_1 将被破坏, 因为在位置 2 的“1”和在位置 7 的“0”将被分配到不同的后代个体中。同样可以明显地看出, 模式 H_2 将继续存在, 因为位置 4 的“1”和位置 5 的“0”原封不动地进入下一代个体中。虽然该例中的交叉点位置是随机选取的, 但不难看出, 模式 H_1 比模式 H_2 更容易破坏。因为平均看来, 交叉点更容易落在两个头尾确定位置之间。若定量的分析, 模式 H_1 的定义长度为 5, 如果交叉点始终是随机地从 6 个可能的位置选取, 那么很显然模式 H_1 被破坏的概率为

$$p_d = \delta(H_1)/(l-1) = 5/6 \quad (3.53)$$

存活概率为

$$p_s = 1 - p_d = 1/6 \quad (3.54)$$

类似地, 模式 H_2 的定义长度为 1, 它被破坏的概率为 $p_d = 1/6$, 存活的概率为 $p_s = 1 - p_d = 5/6$ 。推广到一般情况, 可以计算出任何模式的交叉存活概率的下限为

$$p_s \geq 1 - \frac{\delta(H)}{l-1} \quad (3.55)$$

其中大于号表示当交叉点落入定义长度内时也存在模式不被破坏的可能性。

在前面的讨论中, 我们均假定交叉的概率为 1, 一般情况若假设交叉的概率为 p_c , 则上式变为

$$p_s \geq 1 - p_c \frac{\delta(H)}{l-1} \quad (3.56)$$

若综合考虑复制和交叉的影响, 特定模式 H 在下一代中的数量可用下式来估计

$$m(H, t+1) \geq m(H, t) \frac{f(H)}{f} \left[1 - p_c \frac{\delta(H)}{l-1} \right] \quad (3.57)$$

可见, 对于那些高于平均适应度且具有短的定义长度的模式将更多出现在下一代中。

3. 变异对模式的影响

变异是对串中的单个位置以概率 p_m 进行随机替换, 因而它可能破坏特定的模式。一个模式要存活意味着它所有的确定位置都存活。由于单个位置的基因值存活的概率为 $(1 - p_m)$, 而每个变异的发生是统计独立, 所

以一个特定模式仅当它所有的 $O(H)$ 个确定位置都存活时才存活。由此可得，经变异后特定模式的存活概率为

$$(1-p_m)^{O(H)} \quad (3.58)$$

由于 $p_m \ll 1$ ，所以上式也可近似表示为

$$(1-p_m)^{O(H)} \approx 1-O(H)p_m \quad (3.59)$$

综合考虑上述复制、交叉和变异操作，可得特定模式 H 的数量改变为

$$m(H,t+1) \geq m(H,t) \frac{f(H)}{f} \left[1 - p_c \frac{\delta(H)}{l-1} \right] [1-O(H)p_m] \quad (3.60)$$

由此可知，对于那些具有低阶、短的定义长度且适应度高于种群平均适应度的模式在后代中的数量将呈指数级增长。这就是遗传算法的模式理论。

根据模式理论，随着遗传算法一代一代地进行，那些短定义长度、低阶数、高适应度的模式将越来越多，因而可期望最后得到的串（即这些模式的组合），性能越来越得到改善，并最终趋向全局的最优点。

3.4.3.2 积木块假设

由模式定理可知，对于那些具有低阶、短的定义长度且适应度高于种群平均适应度的模式在后代中的数量将呈指数级增长。这种具有低阶、短的定义长度且适应度高于种群平均适应度的模式就称为基因块或积木块。之所以称为积木块，是由于遗传算法的求解过程并不是在搜索空间中逐一地测试各个基因的枚举组合，而是通过一些较好的模式，象搭积木一样，将它们拼接在一起，从而逐渐地构造出适应度越来越高的个体编码串。

模式定理说明了遗传算法寻求最优解的可能性，但它并未指明遗传算法一定能找到最优解。而积木块假设却说明了遗传算法的这种能力。

积木块假设：个体的基因块通过选择、交叉、变异等遗传算子的作用，能够相互拼接在一起，形成适应度更高的个体编码串。

积木块假设说明了运用遗传算法求解各类问题的基本思想，即通过基因块之间的相互拼接能够产生出问题更好的解。基于模式定理和积木块假设，就使得我们能够在很多应用问题中广泛地使用遗传算法的思想。

需要说明的是，虽然积木块假设并未得到完整而精密的数学证明，但大量的应用实践说明了其有效性^[20]。

3.4.3.3 隐含并行性

在遗传算法的运行过程中，每一代都处理了 n 个个体，但由于一个个体编码串中隐含有多种不同的模式，所以算法实质上处理了许多模式。

以二进制编码串为例，有前面所述可知，个体编码长度为 l 、规模大小为 n 的种群可能隐含有 $2^l \sim n2^l$ 种不同的模式。随着进化过程的进行，一些定义长度较长的模式被破坏掉，而另一些定义长度较短的模式却生存下来。下面对每代群体中按指数级增长的模式个数的下界进行估算，这个数据也就是遗传算法能够有效处理的模式个数。

由于定义长度较长的模式很容易被交叉运算破坏掉，所以这里估算群体中定义长度在 $l_s - 1$ 以下的模式个数，其中 l_s 为常数，且 $l_s < l$ 。假设群体中的某一个体 A 和某一模式 H 如下：

$$A = a_1 a_2 \cdots a_i : a_{i+1} a_{i+2} \cdots a_{i+l_s} : a_{i+l_s+1} \cdots a_l$$

$$H = * * \cdots * : s_1 s_2 \cdots s_{l_s} : * \cdots *$$

这样的子串 $s_1 s_2 \cdots s_{l_s}$ 的起始位置共有 $l - l_s + 1$ 个，并且每个 $s_1 s_2 \cdots s_{l_s}$ 表示定义长度在 $l_s - 1$ 以下的模式，则其中至少有一个基因值是 0 或 1，最多全部都是固定的。显然，这种定义长度在 $l_s - 1$ 以下的模式共有 $2^{l_s - 1}$ 种，由此可知，与一个个体相对应的模式数应该为：

$$(l - l_s + 1) 2^{l_s - 1} \tag{3.61}$$

这样，在群体的全部 n 个体中所隐含的模式数为：

$$n(l - l_s + 1) 2^{l_s - 1} \tag{3.62}$$

若群体规模 n 较大，则对一些低阶的模式肯定会一些重复，为排除这些重复部分，可取群体的规模为 $n = 2^{l/2}$ 。这时，阶数高于或等于 $l_s/2$ 的模式最多只重复计数一次。由此可估计出排除重复模式后的模式数量 n_s 约为：

$$n_s \geq \frac{n(l - l_s + 1) 2^{l_s - 1}}{2} \approx \frac{(l - l_s + 1)}{4} n^3 \tag{3.63}$$

即有：

$$n_s = cn^3 = o(n^3) \tag{3.64}$$

由此，我们可以得出如下结论：遗传算法所处理的有效模式总数约与群体规模的立方成正比。

也就是说，虽然在进化过程的每一代中只处理了 n 个个体，但实际上我们并行处理了与 n^3 成正比的模式数。这种并行处理过程有别于一般意义下的并行算法的运行过程，是包含在处理过程内部的一种隐含并行性，通过这种隐含并行性，使得我们可以快速地搜索出一些较好的模式。

3.4.4 遗传算法的运行参数

遗传算法中需要选择的运行参数主要有个体编码串长度 l 、群体规模 n 、交叉概率 p_c 、变异概率 p_m 及终止代数 T 等。对这些参数的取值，目前尚没有一个确定的准则，只能依据经验推理确定，而缺乏严格的理论证明。

1. 编码串长度 l

有前面叙述可知，采用二进制编码方法对个体编码时，编码串长度 l 与问题所要求的求解精度有关；采用浮点数编码方法时，编码串长度 l 与决策变量的个数相等；采用符号编码方法时，编码串长度 l 由问题的编码方式来决定；另外也可使用变长度的编码串来表示个体。

对于二进制编码，假设参数 θ 的取值范围为 $[\theta_{\min}, \theta_{\max}]$ ，编码长度为 l ，则它总共能产生 2^l 种不同的编码，若使参数编码时的对应关系如下

$$\begin{array}{lll} 00 \cdots 000 = 0 & \rightarrow & \theta_{\min} \\ 00 \cdots 001 = 1 & \rightarrow & \theta_{\min} + \delta \\ \vdots & & \vdots \\ 11 \cdots 111 = 2^l - 1 & \rightarrow & \theta_{\max} \end{array}$$

则二进制编码的精度为

$$\delta = \frac{\theta_{\max} - \theta_{\min}}{2^l - 1} \quad (3.65)$$

对应的解码公式为

$$\theta = \theta_{\min} + \frac{\text{binrep}}{2^l - 1} (\theta_{\max} - \theta_{\min}) \quad (3.66)$$

其中， binrep 是由 l 位字符串所表示的十进制数。

2. 群体规模 n

群体规模是指群体中包含个体的数量。当 n 取值较小时，可提高遗传算法的运算速度，但却降低了种群的多样性，有可能会产生早熟收敛；而当 n 较大时，又会导致遗传算法的运行效率降低。一般建议的取值范围为 $n = 20 \sim 100$ 。

3. 交叉概率 p_c

交叉操作是遗传算法中产生新个体的主要方法，因此交叉概率 p_c 一般应选的大一些。 p_c 越大，新个体产生的速度就越快，然而 p_c 过大，遗传模式被破坏的可能性就越大，使得高适应度的个体结构很快就会被破坏；但是 p_c 过小，产生新个体的速度较慢，以致搜索过程缓慢，搜索周期变长。一般建议的取值范围为 $p_c = 0.4 \sim 1.0$ 。

4. 变异概率 p_m

对于变异概率 p_m ，若 p_m 取值较大，虽然能够产生较多的新个体，使得算法能不断地搜索新的解空间，增加模式的多样性，但也有可能破坏掉较好的模式，当 p_m 取得过大时，会使得遗传算法变成纯粹的随机搜索算法；若 p_m 取值较小，则不易产生新的个体结构，使算法产生早熟收敛，无法找到全局最优解。一般建议的取值范围为 $p_m = 0.001 \sim 0.1$ 。也可使用自适应的思想来确定变异概率。

Srinivas 等曾提出了一种自适应遗传算法，其基本思想是针对每一对进行交叉的个体和每一个进行变异的个体分别设计一个交叉概率和变异概率，具体表达式如下：

$$p_c = \begin{cases} \frac{k_1(f_{\max} - f')}{f_{\max} - f_{avg}} & f' \geq f_{avg} \\ k_2 & f' < f_{avg} \end{cases} \quad (3.67)$$

$$p_m = \begin{cases} \frac{k_3(f_{\max} - f)}{f_{\max} - f_{avg}} & f' \geq f_{avg} \\ k_4 & f' < f_{avg} \end{cases} \quad (3.68)$$

式中， f_{\max} 为群体中最大的适应值， f_{avg} 为群体的平均适应值， f' 为进行交叉的两个个体中较大的适应值， f 为要变异的个体的适应值， k_1, k_2, k_3, k_4 为常数，且 $0 < k_1, k_2, k_3, k_4 \leq 1$ 。

王耀南也提出了一种自适应遗传算法，其基本思想是遗传进化每进行一代，就根据种群中个体的分布情况设计一个交叉概率和变异概率，具体表达式如下：

$$p_c = k_1 / (f_{\max} - f_{avg}) \quad (3.69)$$

$$p_m = k_2 / (f_{\max} - f_{avg}) \quad (3.70)$$

其中， f_{\max} 为群体中最大的适应值， f_{avg} 为群体的平均适应值， k_1, k_2 为常数，且 $0 < k_1, k_2 \leq 1$ [21]。

5. 终止代数 T

终止代数是表示遗传算法运行结束条件的一个参数，它表示遗传算法运行到指定的进化代数后就停止运行，并将当前群体中的最佳个体作为问题的最优解输出。一般建议的取值范围为 $T = 100 \sim 1000$ 。

另外，遗传算法的终止条件，还可以利用某种判定准则，当判定出群体已经进化成熟且不再有进化趋势时就可终止算法的运行。常用的判

定准则有以下两种：

- (1) 连续几代群体平均适应度的差异小于某一个极小的阈值。
- (2) 群体中所有个体适应度的方差小于某一个极小的阈值。

3.4.5 遗传算法的收敛性分析

3.4.5.1 收敛性判断

常规的数学规划方法都有比较严格确定的收敛判据，但遗传算法的收敛判据基本上启发式的，目前常用的判据有：

(1) 根据计算时间和所采用的计算机容量限制确定，即给定迭代次数和每一代解群体个体串的数目来判断是否收敛。

(2) 从解的质量方面来确定的判据：例如连续几代得到的解群体中的最优解没有变化则可认为遗传算法收敛；或解群体中最佳个体的适应度与群体的平均适应度之差占平均适应度的百分数小于某一给定的评估值则可认为遗传算法收敛。

(3) 状态估计的模糊判据准则：这是一种将模糊目标与遗传算法的实时运算表现结合起来的自调节判定方法，由用户根据问题设定一定的目标与可信度，将先期遗传算法产生的结果作为参考，不断调节进程结果，直到达到规定的目标。

3.4.5.2 早熟收敛性分析

早熟收敛是遗传算法中不可忽视的问题，主要表现在：

- (1) 群体中所有个体都陷于同一极值（局部）而停止进化。
- (2) 接近最优解的个体总是被淘汰，进化过程不收敛。

针对早熟问题，需要在个体编码、适应度函数和遗传操作等设计中综合考虑，主要做法有：

- (1) 提高变异概率，在进化初期可以加强遗传算法的随机搜索能力。
- (2) 调整选择概率，可以把选择概率本身也作为个体进行优化。
- (3) 对适应度函数进行恰当地调整标定。
- (4) 维持群体的多样性。
- (5) 将遗传算法与其它优化算法相结合。

3.4.5.3 收敛性分析

可以证明，标准的遗传算法不是依概率全局收敛的。而采用最优保留策略的遗传算法是依概率全局收敛的^[20]。

第 4 章 位置调节器的设计

4.1 综合智能控制思想的提出

综合前面所述可知, 尽管 PID 控制算法简单、鲁棒性强, 但其参数整定过程繁琐, 整定时需要控制对象的精确数学模型, 而且整定往往是针对某一种具体工况进行的, 缺乏自学习和自适应能力。所有这些决定了它对于具有高度非线性、大延迟以及时变等特性的现代控制系统已越来越不适应。

而模糊神经网络则兼备了模糊逻辑和神经网络的优点, 同时又摒弃了两者各自的一些缺点。首先, 它不再依赖控制对象的精确数学模型, 只需控制专家或操作工人的经验知识, 比较适合于表达那些模糊或定性的知识。其次, 网络的各个参数具有明显的物理意义, 因此其初值可以依据一些先验的知识来确定, 而不再是随机的, 这有助于加快算法的运行, 缩短运行时间。再者, 它具有函数逼近功能, 具有较强的自适应、自学习能力、容错能力和泛化能力。

但同时模糊神经网络也存在一些不足之处, 如隶属度函数和模糊控制规则的确定目前还没有一套可行的指导理论, 只能依靠控制专家的经验知识进行不断的试凑, 因此具有较大的主观性。而且一旦确定以后便无法进行修改, 学习和适应能力较差。而试凑的过程实际上就是一个寻优过程, 因此可借助遗传算法来进行优化。

考虑到以上几点, 本论文提出将模糊神经网络和遗传算法引入 PID 控制参数的整定过程, 从而构造出一种基于模糊神经网络和遗传算法的智能 PID 控制器。

4.2 课题的技术方案

虽然一般模糊神经网络的 BP 学习算法是一种建立和调整模糊推理控制系统的良好方法, 但这种方法本质上是一种只考虑局部区域的梯度法, 缺乏全局性, 有可能仅优化到局部极值部分, 其调整的收敛性依赖于初始状态的选择。而遗传算法是一种基于生物进化过程的随机搜索的全局优化方法, 它通过交叉和变异大大减少了初始状态的影响, 使搜索得到最优结果而不停留在局部极值处。为了发挥 GA 算法和 BP 算法的长处, 可以用 GA 算法优化具有全局性的参数和网络结构, 用 BP 算法调节和优化具有局部性的参数。这两种方法综合使用, 可以大大提高模糊神经网络

络控制系统的自学习性能和鲁棒性^[21]。

具体地讲，由于模糊神经网络的网络层数相对比较固定，可以进行调整的主要是第三层的隐结点数（由模糊控制规则数决定）、输入语言变量的隶属度函数的中心值和底部宽度或均方差（三角形或铃形隶属度函数）以及网络输出层所对应的连接权系数。其中，隐结点数、隶属度函数的中心值和底部宽度或均方差属于全局性参数，借助于遗传算法对其进行优化设计；而连接权系数则属于局部性参数，借助于 BP 算法对其进行优化。同时考虑到量化因子和比例因子对模糊逻辑控制器的性能有较大影响，而交叉概率和变异概率对遗传算法的收敛性和收敛速度有较大影响，因此也决定采用特殊策略进行优化。

具体方案如下：

首先，控制器采用两输入三输出结构，输入分别为偏差和偏差的变化率，输出分别为比例系数 K_p 、积分系数 K_I 和微分系数 K_D ；

其次，用遗传算法同时优化量化因子、比例因子、隶属度函数、控制规则和初始连接权系数；

最后，在遗传算法的运行过程中，根据种群结构的变化，借助模糊逻辑确定适当的交叉概率和变异概率。

相应的控制器结构如图 4.1 所示。

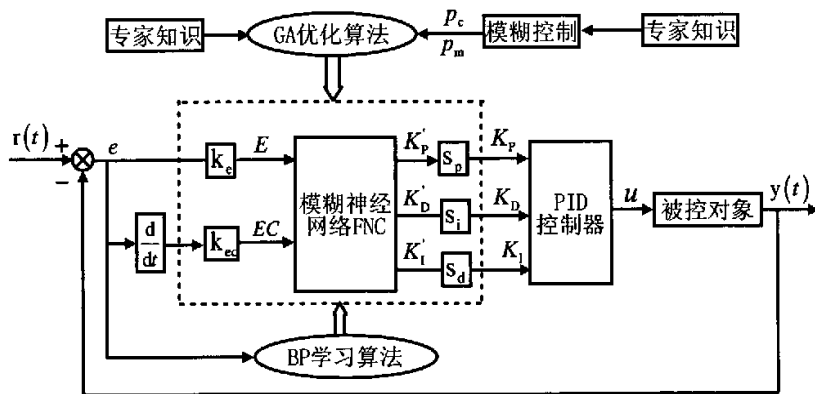


图 4.1 基于 GA 和 FNN 的智能 PID 控制器结构图

采用这种智能 PID 控制器的位置伺服系统结构如图 4.2 所示

4.2.1 用模糊逻辑优化交叉概率和变异概率

交叉概率和变异概率对遗传算法的收敛速度有很大影响，如果选择不当，可能会造成算法收敛过程缓慢，甚至无法收敛，也可能造成早熟收敛。目前关于这两个参数的确定还没有一套严格而精确的指导理论，只能针对不同的优化问题，通过反复试验来确定，因此过程非常繁琐，

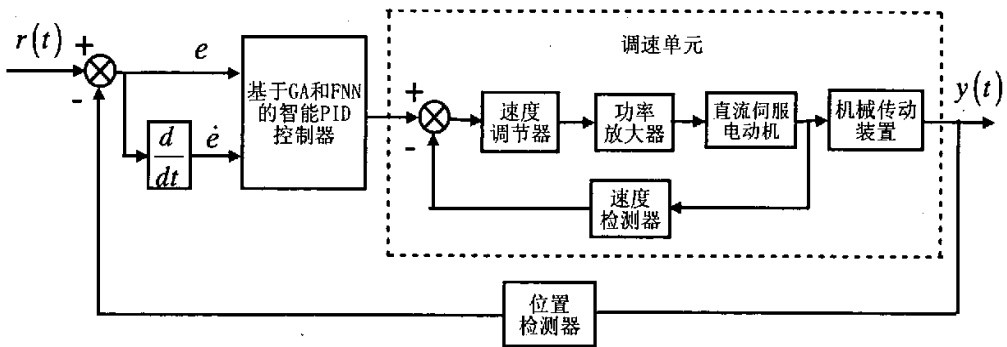


图 4.2 采用智能 PID 控制器的位置伺服系统

而且很难找到适应于每个问题的最佳值。虽然自适应遗传算法中引入了自适应交叉和变异算子，使得交叉和变异算子随着种群结构的变化有了一定的学习和适应能力，取得了一定的效果。但这种方法同时又引入了具有不确定性的系数 k_1, k_2, k_3, k_4 或 k_1, k_2 ，它们的取值没有指导依据，仍然是靠试凑。而模糊控制不需要对象的数学模型，只需控制专家的控制经验就可进行控制，因此，本论文提出借助模糊逻辑来确定交叉概率和变异概率。

4.2.1.1 算法的基本流程

1. 输入输出语言变量的确定

由于 p_c 、 p_m 的确定主要取决于群体中个体适应值的差异，当群体中个体的适应值差别较大时，说明群体的多样性较高，为保证算法的稳定性和快速收敛，这时就需要增大 p_c ，同时减小 p_m ；当群体中个体的适应值差别较小时，说明群体的多样性较差，为防止个别个体在群体中急剧增加，导致算法早熟收敛，这时就需要减小 p_c ，同时增大 p_m ，以加快群体中新个体结构的产生。鉴于此，按下式构造模糊控制器的输入量

$$x_1 = \frac{f_{\max} - f_{\text{avg}}}{f_{\max} - f_{\min}}, x_2 = \frac{dx_1}{dt} \quad (4.1)$$

式中， f_{\max} 、 f_{\min} 、 f_{avg} 分别表示群体中最大的适应值、最小的适应值和平均适应值^[22]。

2. 模糊分割及隶属度函数的确定

取输入量 x_1 、 x_2 和输出量 p_c 、 p_m 的量化论域均为 $[-6, 6]$ ，模糊分割数均取为七，模糊语言值分别为：NB（负大）、NM（负中）、NS（负小）、ZE（零）、PS（正小）、PM（正中）、PB（正大），隶属度函数采用铃形函数。即

$$\mu(x) = \exp\left[-\left(\frac{x-x_0}{\sigma}\right)^2\right] \quad (4.2)$$

3. 模糊控制规则的建立

根据交叉概率和变异概率与种群多样性之间的关系，我们建立如表 4.1 和 4.2 所示的控制规则。

表 4.1 交叉概率 p_c 的模糊控制规则

p_c		偏差变化率 EC						
		NB	NM	NS	ZE	PS	PM	PB
偏差 E	NB	NB	NB	NM	NM	NS	ZE	ZE
	NM	NB	NB	NM	NS	NS	ZE	ZE
	NS	NB	NM	NS	NS	ZE	PS	PS
	ZE	NM	NM	NS	ZE	PS	PM	PM
	PS	NM	NS	ZE	PS	PS	PM	PB
	PM	ZE	ZE	PS	PS	PM	PB	PB
	PB	ZE	ZE	PS	PM	PM	PB	PB

表 4.2 变异概率 p_m 的模糊控制规则

p_m		偏差变化率 EC						
		NB	NM	NS	ZE	PS	PM	PB
偏差 E	NB	PB	PB	PM	PM	PS	ZE	ZE
	NM	PB	PB	PM	PS	PS	ZE	NS
	NS	PM	PM	PM	PS	ZE	NS	NS
	ZE	PM	PM	PS	ZE	NS	NM	NM
	PS	PS	PS	ZE	NS	NS	NM	NM
	PM	PS	ZE	NS	NM	NM	NM	NB
	PB	ZE	ZE	NM	NM	NM	NB	NB

4. 模糊推理

由第三章的叙述知，模糊蕴含关系采用最小运算或积运算以及“also”采用求并运算时可以给出最好的控制效果，而且实现起来也最简单。因此，在这里模糊蕴含运算采用最小运算，句子连接词“also”采用求并运算。

5. 去模糊化

采用最常用的加权平均法（即重心法）。

4.2.1.2 实例验证

由于数学函数优化问题不需要专门的领域知识，且能较好地反映算法本身的实际效能，所以常用于遗传算法的测试。这里我们选用 Shaffer 的 F_6 函数：

$$f_6(x,y) = 0.5 - \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{(1 + 0.001(x^2 + y^2))^2} \quad -100 \leq x, y \leq 100 \quad (4.3)$$

分析函数特性可知，此函数在其定义域内无限多个局部极大值点，但仅有一个全局极大值点 $(0, 0)$ ，最大值为 1。在此函数峰值周围有一个圈脊，它们的取值均为 0.990283，因此，很容易陷入局部极值点^[23]。

为了验证算法的有效性，我们还分别进行了常规遗传算法和王耀南提出的自适应遗传算法的仿真。其中，编码方案均采用二进制编码方法，编码精度为 10^{-6} ，则编码长度为 42；种群规模均取为 100；常规遗传算法中的交叉概率 $p_c = 0.5$ ，变异概率 $p_m = 0.1$ ；自适应遗传算法中的常数 $k_1 = 0.7, k_2 = 0.4$ ；模糊逻辑优化时交叉概率的取值范围为 $[0.4, 1]$ ，变异概率的取值范围为 $[0.01, 0.1]$ ；最大进化代数为 200；适应度函数采用如下形式：

$$f = 10f_6(x, y) \quad (4.4)$$

经多次运算取均值，最终可得如下表 4.3 所示的运算结果。其中，表中所列数据为目标函数值，目标函数值后括号内的数据为首次优化到该值时的进化代数。

表 4.3 遗传算法优化后的数据

算法类型 进化代数	常规遗传算法	自适应 遗传算法	模糊逻辑优化 的遗传算法
20	0.9902028(18)	0.9902605(16)	0.9625413(19)
40	0.9902760(37)	0.9999849(37)	0.9890980
60	0.9993861(59)	0.9999851(57)	0.9960908
80	0.9997334	0.9999879(72)	0.9999967(77)
100	0.9998298(93)	0.9999975(99)	0.9999996(97)
120	0.999987(116)	0.9999993(118)	1.000000(109)
140	0.9999995(136)	1.000000(124)	1.000000
162	1.000000	1.000000	1.000000

由表中数据可以看出，常规遗传算法在进化到第 162 代时收敛，自适应遗传算法在进化到第 124 代时收敛，而模糊遗传算法则在第 109 代

时已收敛。由此可见，采用模糊逻辑优化的遗传算法的收敛速度明显优于另外两种。

4.2.2 用改进的遗传算法优化全局性参数

这里的全局性参数分别是指模糊控制器的量化因子、比例因子、隶属度函数、模糊控制规则和初始连接权系数。

量化因子和比例因子的大小对模糊控制系统的动态性能有较大影响。 k_e 选的较大时，可以减小系统的静态误差，但系统的超调也较大，过渡过程较长，因为 k_e 增大，相当于缩小了误差的基本论域，增大了误差的控制作用，因此导致上升时间变短，但由于超调量较大，使得系统的过渡过程变长，甚至出现振荡； k_{cc} 选择较大时，系统的超调量减小， k_{cc} 越大，系统的超调量就越小，但系统的响应速度会变慢； s_d 选的较大时，可以减小系统的静态误差，但选择过大，会使系统变的不稳定。目前，这些因子的确定还没有一套可行的指导理论，主要是靠控制专家的经验知识不断地进行试凑，过程比较繁琐。同时考虑到量化因子和比例因子的选择不同，隶属度函数参数的选择也会有所不同，因此，决定与隶属度函数和模糊控制规则一起由遗传算法进行优化。

1. 参数编码

这里重点说的是控制规则的编码。对于模糊神经网络，优化控制规则的主要目的是去掉不必要的或对控制效果影响较小的规则，以达到简化规则库的目的。假定控制器的输入量 E 和 EC 均划分为七个模糊子集，即 NB、NM、NS、ZE、PS、PM、PB，则控制器的规则总数最多为 $7*7=49$ 条，优化就是在这些规则的基础上进行的。为此，我们可以给每条模糊规则分别赋予一个二进制整数 0 或 1，其中，1 代表该条规则存在，即起作用；而 0 则表示该条规则不存在。那么整个规则库只需要 49 个这样的二进制数即可完整表示。如果输入量和输出量的隶属度函数均采用铃形函数，则最终需要优化的参数共有 229 个。由于需要优化的参数较多，如果采用二进制编码，会使搜索空间巨大，搜索时间过长。因此，在这里采用浮点数多参数级联定点影射编码方法，编码长度就等于待优化参数的个数。为方便后面进行交叉、变异等遗传操作，编码时同一类型的参数放在一起，按以下顺序排列：量化因子、比例因子、偏差 E 的隶属度函数的中心值及均方差值、偏差 EC 的隶属度函数的中心值及均方差值、模糊控制规则、连接权系数。这样，种群中的每一个个体代表一个完整的控制器。

2. 初始种群的产生

这里重点说的是关于隶属度函数中心值的产生方法。

在产生初始种群之前，首先要确定控制器各输入输出量的量化论域。由于数控机床对位置伺服控制有特殊的要求，即在数控机床位置进给控制中，为了加工出光滑的零件表面，绝对不允许出现位置超调。因此，为了便于理解，同时也为了更接近实际情况，我们取偏差 E 的量化论域为 $[0, 6]$ ，偏差变化率 EC 的量化论域为 $[-6, 6]$ ，输出量 K_p 、 K_i 、 K_d 的量化论域均为 $[0, 6]$ 。同时，对各模糊子集隶属度函数的均方差的取值范围作出一定的限制，对偏差 E ，设定其各模糊子集隶属度函数的均方差值的取值区间为 $[0.3, 0.8]$ ；对偏差变化率 EC ，设定其各模糊子集隶属度函数均方差的取值范围为 $[0.5, 1.5]$ ，之所以这样做，是为了降低产生不合理控制规则的概率。

产生初始种群时，如果仍采用随机的方法在 $[0, 6]$ 或 $[-6, 6]$ 之间随机产生初始中心值，产生的中心值依次赋给 NB、NM、NS、ZE、PS、PM、PB，则一方面产生的隶属度函数有可能过于集中，无法覆盖整个论域，而使算法的收敛速度变慢；另一方面有可能会造成混乱和不必要的麻烦。例如，对于输入量偏差，假设随机产生的中心值分别为： $-5.5, -2, -4.3, -0.5, 1.8, 3.6, 6$ ，我们显然不能把 -2 赋给 NM，把 -4.3 赋给 NS，这是错误的。为避免这种错误，就必须先对产生的随机数进行排序，然后再依次将其赋给每个模糊子集，因此比较繁琐。

在这里，我们采用如下的初始中心值产生策略：对输入量 E ，将模糊子集 NB 的初始中心值限定在 $[0, 1]$ 范围内，首先在 $[0, 1]$ 范围内产生一个随机数作为 NB 的初始中心值，然后再在该中心值的基础上每次加上 $5/6$ ，分别作为模糊子集 NM、NS、ZE、PS、PM、PB 的中心值；对于输入量 EC ，则将模糊子集 NB 的初始中心值限定在 $[-6, -4]$ 范围内，首先在 $[-6, -4]$ 范围内产生一个随机数作为 NB 的初始中心值，然后再在该中心值的基础上每次加上 $5/3$ ，分别作为模糊子集 NM、NS、ZE、PS、PM、PB 的中心值。这样，不仅可以避免繁琐的排序过程，而且可以使所产生的规则尽可能的覆盖整个论域，提高算法的收敛速度。

3. 适应度函数的构造

为综合反映各参数对控制器性能的影响，采用文献 [24] 提出的根据系统性能指标来构造归一化适应度函数的方法：

$$f = \alpha \frac{\sigma_m - \sigma}{\sigma_m} + \beta \frac{t_{sm} - t_s}{t_{sm}} + \gamma \frac{e_{sm} - e}{e_{sm}} \quad (4.5)$$

其中， σ, t_s, e_s 分别为控制系统的超调量、调节时间和稳态误差； σ_m, t_{sm}, e_{sm} 分别为控制系统超调量、调节时间和稳态误差的上限值； α, β, γ 为各性能指

标的加权值, 且 $\alpha + \beta + \gamma = 1$ 。如果 $\sigma_m - \sigma$ (或 $t_{sm} - t_s$, $e_{sm} - e_s$) 为负, 则将其置零。

4. 遗传操作

(1) 复制

由于标准的遗传算法不是依概率全局收敛的。而采用最优保留策略的遗传算法是依概率全局收敛的。因此, 在这里复制操作采用带最优保留策略的比例选择法。

(2) 交叉

由于编码串中包含多个类型的参数, 如果仍采用常规的一些交叉方法, 如单点交叉、两点交叉或多点交叉, 则有可能会出现这样的现象, 即每次进行交叉操作之后, 只有某一个或几个类型的参数进行了基因交换, 另外剩余的参数并没有进行基因交换, 而且每次交换的是代表参数的整个基因串, 也就是说交换后参数值并没有变化, 只是从一个个体交换到了另一个个体, 不能像二进制编码串那样生成新的参数值, 这将降低遗传算法的运算效率, 有可能导致算法在规定的进化代数内仅优化到局部极值点, 而非全局极值点; 如果采用均匀交叉法, 则又会使新个体的产生速度过快, 使一些优良的基因结构也被破坏掉, 同样会降低算法的收敛性和稳定性。因此, 在这里交叉算子采用算术交叉。

所谓算术交叉是指通过两个个体的线性组合而产生出两个新的个体。例如, 假设在两个个体 X_A 和 X_B 之间进行算术交叉, 则算术交叉后产生的两个新个体为:

$$\begin{aligned} X'_A &= \alpha X_B + (1-\alpha) X_A \\ X'_B &= \alpha X_A + (1-\alpha) X_B \end{aligned} \quad (4.6)$$

式中, α 为一参数, 它可以是一个常数, 此时所进行的交叉运算称为均匀算术交叉。它也可以是一个由进化代数所决定的变量, 此时所进行的交叉运算称为非均匀交叉运算。

同时, 又考虑到代表控制规则的基因段相当于二进制编码串, 如果进行算术交叉, 一方面需要对交叉后的结果进行取整; 另一方面有可能使规则变化较大, 不利算法的稳定。因此, 决定将其单独分离出来, 按照二进制编码串分别进行单点交叉。

交叉概率则是根据种群结构的变化通过模糊逻辑来确定, 具体方法如上文所述。

(3) 变异

对于变异, 同样的问题, 如果对整个编码串进行基本位变异, 则有可能每次进行变异操作之后, 只有某一个类型的参数发生了变异, 而其

余大部分的参数并没有发生变化，这也将降低遗传算法的运算效率，导致算法在规定的进化代数内仅优化到局部极值点，而非全局极值点；而均匀变异又会使新个体的产生速度过快，使一些优良的基因结构也被破坏掉，同样会降低算法的收敛性和稳定性。因此，变异算子需要进行改进。

本文提出如下的变异策略：即将个体基因编码串按参数类型进行分段，共分为七段，其中量化因子和变异因子为一段，偏差 E 的隶属度函数的中心值及均方差值、偏差 EC 的隶属度函数的中心值及均方差值、模糊控制规则、 K_p 对应的连接权系数、 K_I 对应的连接权系数、 K_D 对应的连接权系数各为一段，然后分别对每一个基因段运用变异算子。其中，控制规则段仿照二进制编码串的变异方法进行变异，其余按浮点数编码串的变异方法进行变异，即用待寻优参数取值范围内的随机数替换原基因值，变异概率根据种群结构的变化通过模糊逻辑来确定，具体方法如上文所述。

5. 终止条件

这里取最大进化代数作为算法的终止条件。取最大进化代数 $T=100$ 。

4.2.3 用改进的 BP 算法在线调整连接权系数

常规的模糊神经网络，通常是以网络输出层各参数的输出值与参考值之差的平方和作为误差代价函数，然后利用事先采集好的训练样本，通过误差反向传播，进行连接权的调整。但是，对于本论文的网络结构而言，网络的输出层所对应的参数分别是比例系数 K_p 、积分系数 K_I 和微分系数 K_D ，而对于一个实际的被控对象，要想得到关于这三个参数的训练样本具有一定的困难。因此，在这里，我们对误差代价函数进行一些改进，不再利用 K_p 、 K_I 和 K_D 构造误差代价函数，而是改用整个控制系统最终的输出值来构造误差代价函数，从而得到一种新的训练连接权的 BP 算法。

设取误差代价函数为

$$E = \frac{1}{2} [y_d(k) - y(k)]^2 \quad (4.7)$$

其中， $y(k)$ 和 $y_d(k)$ 分别为 k 时刻被控对象的实际输出和期望输出。

设网络输出层神经元结点与网络第四层各结点的连接权系数分别记为： $w_{p1}, w_{p2}, \dots, w_{p49}$ ， $w_{i1}, w_{i2}, \dots, w_{i49}$ ， $w_{d1}, w_{d2}, \dots, w_{d49}$ ，则根据最速下降法的原理可知：

$$\Delta w_{ji}(k) \propto -\frac{\partial E}{\partial w_{ji}(k)} \quad (j = P, I, D; i = 1, 2, \dots, 49) \quad (4.8)$$

又

$$\frac{\partial E}{\partial w_{ji}(k)} = \frac{\partial E}{\partial y(k)} \frac{\partial y(k)}{\partial w_{ji}(k)} = \frac{\partial E}{\partial y(k)} \frac{\partial y(k)}{\partial u(k)} \frac{\partial u(k)}{\partial w_{ji}(k)} \quad (4.9)$$

而

$$\frac{\partial E}{\partial y(k)} = -y(k)[y_d(k) - y(k)] \quad (4.10)$$

$$u(k) = s_p K_p e(k) + s_i K_i \sum_{j=0}^k e(j)T + s_d K_d \frac{e(k) - e(k-1)}{T} \quad (4.11)$$

所以

$$\frac{\partial u(k)}{\partial w_{pi}(k)} = \frac{\partial (s_p K_p e(k))}{\partial w_{pi}(k)} = s_p e(k) \frac{\partial (K_p)}{\partial w_{pi}(k)} = s_p e(k) \bar{\alpha}_i \quad (4.12)$$

$$\frac{\partial u(k)}{\partial w_{ii}(k)} = s_i \bar{\alpha}_i \sum_{j=0}^k e(j)T \quad (4.13)$$

$$\frac{\partial u(k)}{\partial w_{di}(k)} = s_d \bar{\alpha}_i \frac{-e(k) - e(k-1)}{T} \quad (4.14)$$

其中, T 为采集样本时的采样周期, s_p 、 s_i 、 s_d 分别为 K_p 、 K_i 、 K_d 的比例因子, $\bar{\alpha}_i (i=1, 2, \dots, 49)$ 为第 i 条规则的适应度经归一化的值。

由参考文献 [25] 可知, $\partial y(k)/\partial u(k)$ 可由符号函数 $\text{sign}(\partial y(k)/\partial u(k))$ 来代替, 由此所带来的计算不精确影响可以通过调整学习率来补偿。综上所述, 可得连接权系数的调整规律为:

$$\begin{aligned} \Delta w_{ji}(k+1) &= w_{ji}(k) + \eta \left(-\frac{\partial E}{\partial w_{ji}(k)} \right) \\ &= w_{ji}(k) + \eta y(k) [y_d(k) - y(k)] \frac{\partial u(k)}{\partial w_{ji}(k)} \text{sign} \left(\frac{\partial y(k)}{\partial u(k)} \right) \end{aligned} \quad (4.15)$$

其中, η 为学习率或学习步长, 且 $\eta > 0$ 。

为了加快算法的收敛速度, 对上述算加以改进, 加入动量项。即

$$w_{ji}(k+1) = w_{ji}(k) + \eta \left[(1-\alpha) \left(-\frac{\partial E(k)}{\partial w_{ji}(k)} \right) + \alpha \left(-\frac{\partial E(k-1)}{\partial w_{ji}(k-1)} \right) \right] \quad (4.16)$$

其中, η —— 学习率或学习步长, $\eta > 0$;

α —— 动量项因子, $0 \leq \alpha < 1$ 。

综上所述，相应的算法流程图如图 4.3 所示。

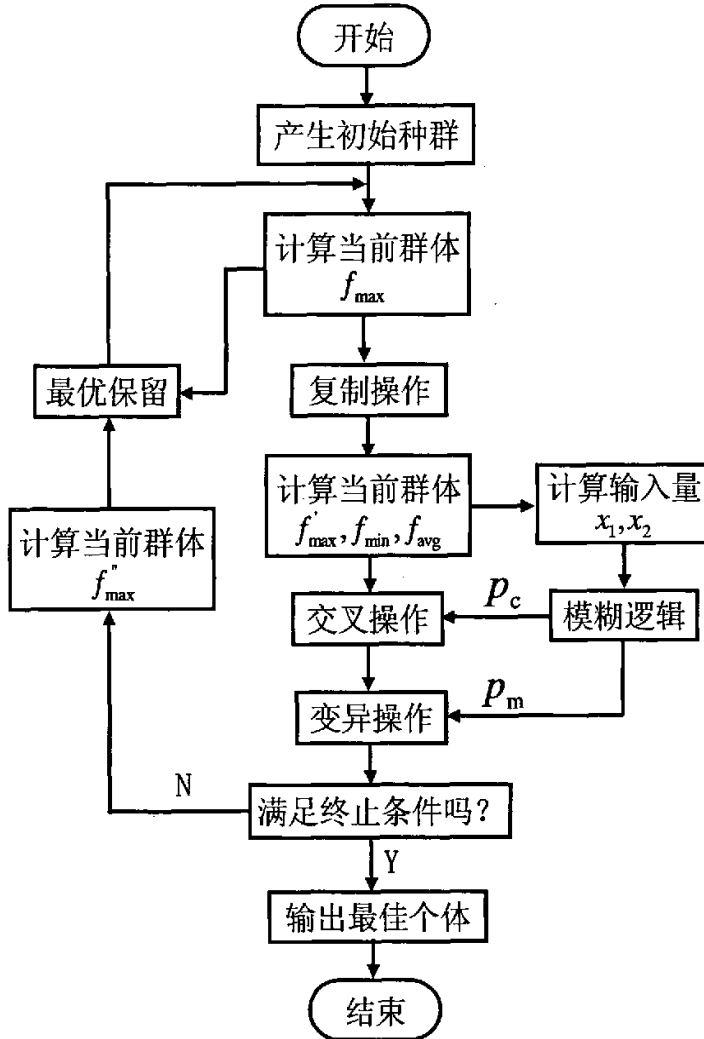


图 4.3 算法流程图

4.3 设计实例及仿真

以某数控机床的位置伺服系统为例，其系统参数如下：电动机电枢回路总电阻 $R_A = 5.6\Omega$ ，反电动势系数 $K_E = 0.703\text{V}\cdot\text{s}/\text{rad}$ ，电动机轴上的等效转动惯量 $J_M = 0.0113\text{kg}\cdot\text{m}^2$ ，转矩系数 $K_M = 0.703\text{N}\cdot\text{m}/\text{A}$ ，速度放大器比例系数 $K_S = 20$ ，速度反馈比例系数 $K_V = 0.03188\text{V}/\text{rad}$ ，位置反馈比例系数 $K_P = 1\text{1}/\text{m}$ ，滚珠丝杠总长度 $L = 0.963\text{m}$ ，导程 $L_S = 0.012\text{m}$ ，采用一级齿轮传动，传动比 $i = 2$ 。

前面第二章我们已经推导出位置伺服系统中广义被控对象的简化数学模型为：

$$G_0(s) = \frac{K_S K_A K_0}{s [T_M s + (K_V K_S K_A K_0 + 1)]} \quad (4.17)$$

代入上述数据可得：

$$G_0(s) = \frac{0.3374}{s(0.3974s + 3.1144)} \quad (4.18)$$

取采样周期 $T = 0.01s$ ，参考输入为单位阶跃信号，下面设计控制器。

1. 设计常规 PID 控制器

由于数控机床对位置伺服系统的最基本要求是不允许出现超调，因此，整定参数时我们就以这个作为首要目标，经过仔细选择，取 $K_p = 40$ ， $K_i = 0.01$ ， $K_d = 2.8$ 。

2. 设计智能 PID 控制器

首先，确定各优化参数的取值范围，经过仔细分析，取 $6 \leq k_c \leq 15$ ， $2 \leq k_{cc} \leq 8$ ， $35 \leq s_p \leq 50$ ， $0.01 \leq s_i \leq 0.1$ ， $400 \leq s_d \leq 500$ ；遗传算法中的交叉运算采用算术交叉，参数 $\alpha = 0.6$ ；适应度函数中的 $\sigma_m = 0.05$ ， $t_{sm} = 0.8s$ ， $e_{sm} = 0.02$ ，相应的加权系数为 $\alpha = 0.4, \beta = 0.3, \gamma = 0.3$ ；BP 算法中的学习步长 $\eta = 0.45$ ，动量因子 $\alpha = 0.02$ 。

经 MATLAB^[26, 27] 程序运行后，可得到如下的结果：

(1) 量化因子和比例因子分别为 $k_c = 9.9613$ ， $k_{cc} = 5.8004$ ， $s_p = 45.431$ ， $s_i = 0.0154$ ， $s_d = 468.01$ 。

(2) 偏差 E 和偏差变化率 EC 的各个模糊子集的隶属度函数的中心值和均方差值如下表 4.4 所示，相应的隶属度函数形状如图 4.4 所示。

(3) 优化后的糊控制规则共有 34 条，由于模糊神经网络只需对控制规则的前件进行匹配，因此，在这里只列出控制规则的前件，如表 4.5 所示。其中，1 代表该条规则存在，0 代表该条规则不存在。

(4) 经 BP 算法训练后，网络输出层的连接权系数分别如表 4.6，表 4.7 和表 4.8 所示，其中，空格处表示该条规则不存在。

表 4.4 优化后 E 和 EC 的隶属度函数的中心值和均方差

类型		模糊子集						
		NB	NM	NS	ZE	PS	PM	PB
偏差 E	中心值	0.6663	1.2996	2.3330	3.2663	4.0004	4.6670	5.7337
	均方差	0.6228	0.5843	0.7146	0.5791	0.6284	0.5307	0.6894
偏差变化率 EC	中心值	-4.6322	-3.6663	-1.9996	-0.3330	1.3337	3.2004	4.6670
	均方差	1.0381	0.8163	1.2262	0.8697	1.2677	1.2437	0.8498

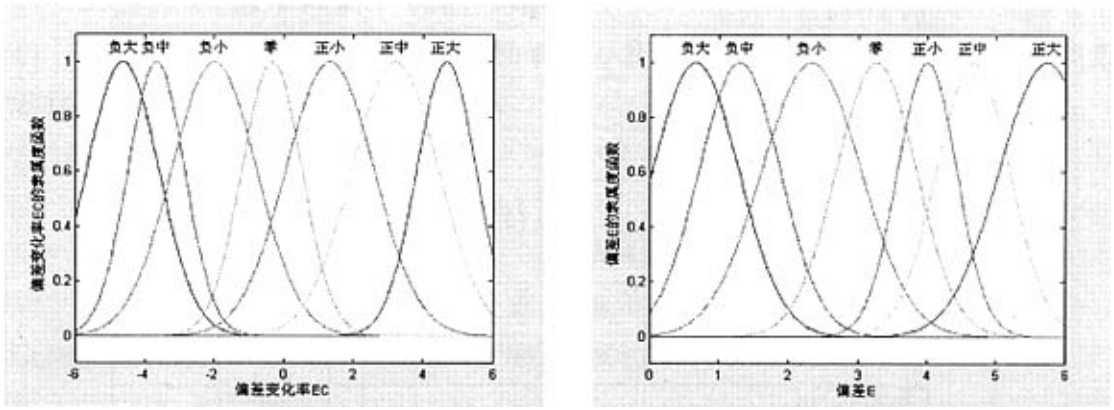


图 4.4 优化后 E 和 EC 的隶属度函数

表 4.5 优化后的模糊控制规则

类型		偏差变化率 EC						
		NB	NM	NS	ZE	PS	PM	PB
偏差 E	NB	0	1	1	0	1	1	0
	NM	1	1	0	1	1	0	1
	NS	1	1	1	0	1	1	0
	ZE	1	0	1	1	0	0	1
	PS	1	1	1	0	1	0	1
	PM	1	1	0	1	1	1	1
	PB	1	0	1	1	0	1	1

表 4.6 训练后比例系数 K_p 所对应的连接权系数

K_p		偏差变化率 EC						
		NB	NM	NS	ZE	PS	PM	PB
偏差 E	NB		5.4915	1.5153		0.2379	4.6980	
	NM	0.8661	3.1813		2.5890	5.9230	0	1.5430
	NS	5.7426	5.9745	5.8946		3.7223	4.6141	
	ZE	4.7142		0.4518	1.1879			0.7271
	PS	4.6500	4.1967	4.0216		0.2363		2.1429
	PM	3.7653	0.9801		3.8627	4.1246	2.4966	0.5084
	PB	5.2149		3.1961	2.9753		4.7228	2.2935

表 4.7 训练后比例系数 K_I 所对应的连接权系数

K_I		偏差变化率 EC						
		NB	NM	NS	ZE	PS	PM	PB
偏差 E	NB		2.1014	5.0516		5.8136	5.2518	
	NM	0.5052	4.2356		5.8858	2.1201		3.1336
	NS	4.3349	2.9496	0.7865		3.9257	5.2884	
	ZE	3.4308		0.0266	3.2080			3.6022
	PS	4.1918	4.3244	5.9992		0.1015		3.4443
	PM	1.8893	0.6714		3.0698	5.4434	4.3318	2.6940
	PB	1.7715		1.8981	2.2674		5.4947	5.7629

表 4.8 训练后比例系数 K_D 所对应的连接权系数

K_D		偏差变化率 EC						
		NB	NM	NS	ZE	PS	PM	PB
偏差 E	NB		3.8783	4.6940		1.5195	2.9743	
	NM	1.6403	4.6353		4.4916	2.1639		2.4382
	NS	4.3018	0.3047	3.5000		2.9971	3.7614	
	ZE	1.6831		3.8823	2.8392			1.1342
	PS	4.1762	4.7489	4.0409		5.5121		1.3159
	PM	2.8197	5.7503		5.7387	2.9496	4.9462	4.8124
	PB	5.6469		1.1167	4.3234		1.7214	4.6851

最终得到的单位阶跃响应曲线如图 4.5 所示

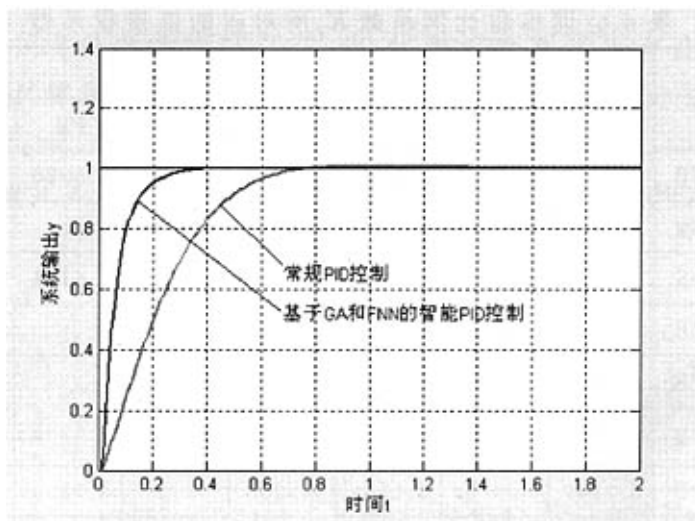


图 4.5 智能 PID 控制的单位阶跃响应曲线

3. 自适应能力和抗干扰能力验证

为了验证新型控制器的自适应能力和抗干扰能力，我们对模型参数进行适当改变，观察其输出曲线。

假设新模型为

$$G_0(s) = \frac{0.7}{s(0.3974s + 2.8)}$$

则最终得到的单位阶跃响应曲线如图 4.6 所示。

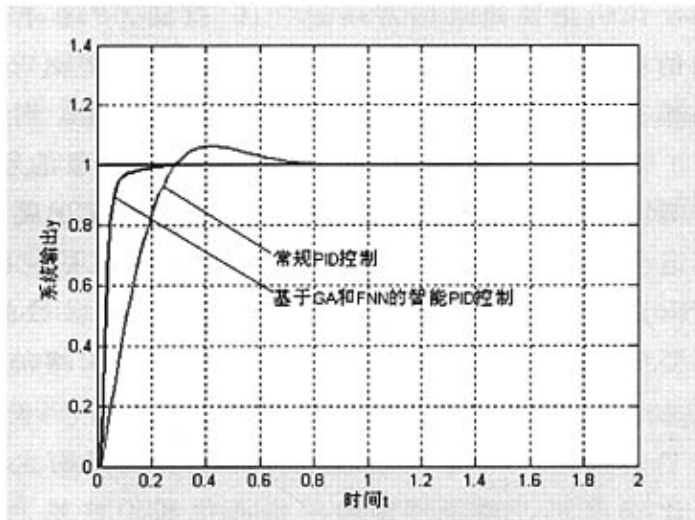


图 4.6 智能 PID 控制器对新模型的响应曲线

从图 4.5 和图 4.6 的仿真曲线可以看出，与常规的 PID 控制器相比，采用遗传算法和模糊神经网络优化的智能 PID 控制器不但实现了无超调控制，响应速度明显加快，过渡过程时间大大缩短，而且具有较强的自适应、自学习能力和抗干扰能力；另外，同标准的模糊神经网络相比，新型的控制器还使得规则库大大简化，网络结构更为简单。

由此可以看出，新型的控制策略对于提高数控机床位置伺服系统的定位精度和跟踪精度，从而提高数控机床的加工精度具有明显的效果。

第 5 章 结论

在数控机床中，位置伺服系统作为数控装置与机床传动部件之间的中间连接环节，它的性能对于数控机床的整体性能具有较大影响。如工作台的最高运动速度、跟踪及定位精度、加工表面质量、生产率及工作可靠性等指标均取决于位置伺服系统的动态和静态性能。目前，位置伺服控制器的设计仍主要是采用常规的 PID 控制。PID 控制具有算法简单、鲁棒性强的特点。然而，随着数控机床加工精度要求的提高，数控加工工艺变得越来越复杂，加工过程往往具有非线性、时变性，常规的 PID 控制因其自身存在的一些局限性，如需要控制对象的精确数学模型、自适应能力差等，已越来越不能满足高精度数控机床的需要。因此，有必要对其进行适当的改进，以提高数控机床位置伺服系统的性能。

本文将模糊逻辑、神经网络和遗传算法引入 PID 参数的整定过程中，设计了一种新型的智能 PID 控制器。借助模糊神经网络的结构来优化设计 PID 控制器的三个参数，同时又对模糊神经网络中相关参数的训练学习方法进行适当的改进，一方面借助遗传算法优化具有全局性的参数，如量化因子、比例因子、隶属度函数的中心值和均方差、模糊控制规则数等；另一方面运用 BP 算法训练具局部性的参数。同时，在遗传算法的运行过程中，又借助模糊逻辑优化对算法运行性能有较大影响的交叉概率和变异概率，使交叉概率和变异概率不再固定不变，而是根据种群结构的变化不断地进行调整，加快了算法的收敛速度，降低了算法陷入局部极值点的概率。另外，虽然连接权系数属于局部参数，但考虑到其初值的选取对网络的训练速度有较大影响，以往一般是采用随机方法产生，算法的训练时间较长，且容易陷入局部极值，因此，决定借助遗传算法进行适当的先期优化，以加快网络的训练速度，缩短训练时间。

基于 MATLAB 的仿真显示，采用这种新型智能 PID 控制器的位置伺服系统大大改善了数控机床的各项性能。与采用常规 PID 控制的位置伺服系统相比，新型控制器不但实现了无超调控制，而且大大加快了系统的响应速度，缩短了系统的调节时间，同时还使系统具备了较强的自适应能力和抗干扰能力，而这些对于改善数控机床的加工精度，提高加工水平都具有很强的现实意义。

参考文献

- [1] 白恩远.现代数控机床伺服及检测技术[M].北京:国防工业出版社,2002
- [2] 陶永华.新型PID控制及其应用[M].第2版.机械工业出版社,2002
- [3] 吴祖育,史鹏飞.数控机床[M].上海:上海科学技术出版社,2000
- [4] Karr C L, gentry E J. fuzzy control of PH using a genetic algorithm[J].IEEE Trans.Fuzzy system,1993,1(1):46-53
- [5] 王耀南.基于遗传算法的模糊神经控制及其应用[J].系统工程与电子技术.1999,21(6):54-56
- [6] 李琳.基于GA的模糊神经网络及其在故障诊断中的应用:[武汉理工大学硕士学位论文].武汉:武汉理工大学,2001
- [7] Buckley J.Fuzzy genetic algorithms and application.Fuzzy sets and syetems. 1994:61-70
- [8] Thrift p. Fuzzy logic synthesis with genetic algorithms. Proc. 4th. int. Conf. on GA. San diego,1991:509-513
- [9] 金耀初,蒋静坪.基于进化计算的模糊控制规则优化.控制与决策.1996,11(6):672-676
- [10] Lee M. A,Takagi H. Integerating design stage of fuzzy systems using genetic algorithms. Proc.2nd.IEEE Int. Conf. Fuzzy Syst. San Francisco, CA. 1993: 612-617
- [11] Homaifar A, Me Connick E. Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms. IEEE Trans. Fuzzy Systems.1995,3(2):129-139
- [12] 王晶,李玉兰,蔡自兴.基于遗传算法的模糊系统优化设计[J].控制理论与应用.1999,16(5):699-704
- [13] 周志坚,毛宗源.一种基于遗传算法的模糊神经网络最优控制[J].控制理论与应用.2000,17(5):784-788
- [14] 杨有君.数字控制技术与数控机床[M].北京:机械工业出版社,2001
- [15] 李元春.微型计算机控制技术[M].长春:吉林大学出版社.1998
- [16] 徐宇馨.数控机床伺服系统智能测控算法研究:[燕山大学硕士学位论文].河北:燕山大学,2005
- [17] 孙增圻,张再兴.邓志东.智能控制理论与技术[M].北京:清华大学出版社,1997

- [18] 李士勇.模糊控制·神经控制和智能控制论[M].哈尔滨工业大学出版社,1996
- [19] 周明,孙树栋.遗传算法原理及应用[M].国防工业出版社,1999
- [20] 毕义明,李景文.遗传算法及其军事应用[M].解放军出版社,1998
- [21] 王耀南.智能控制系统——模糊逻辑·专家系统·神经网络控制[M].湖南大学出版社,1996
- [22] 戚志东.基于遗传算法的神经模糊控制器的设计及应用:[江苏大学硕士学位论文].江苏:江苏大学,2002
- [23] 王小平,曹立明.遗传算法——理论、应用与软件实现[M].西安交通大学出版社,2002
- [24] 刘镇,李程远.用遗传算法优化模糊控制规则[J].贵州工业大学学报 1999,28(5):7-11
- [25] 廖芳芳,肖建.基于 BP 神经网络 PID 参数自整定的研究[J].系统仿真学报,2005,17(7):1711-1713
- [26] 楼顺天,胡昌华,张伟.基于 MATLAB 的系统分析与设计——模糊系统 [M].西安:西安电子科技大学出版社.2001
- [27] 张威.MATLAB 基础与编程入门[M].西安:西安电子科技大学出版社,2004
- [28] 李人厚.智能控制理论与方法[M].西安电子科技大学出版社,1999
- [29] 张乃尧,阎平凡.神经网络与模糊控制[M].清华大学出版社,1998
- [30] 冯冬青,谢宋和等.模糊智能控制[M].化学工业出版社,1998
- [31] 易继锴,侯媛彬.智能控制技术[M].北京工业大学出版社,1999
- [32] 杨辉,王金章.模糊控制技术及其应用[M].江西科学技术出版社,1997
- [33] 李友善,李军.模糊控制理论及其在过程控制中的应用[M].国防工业出版社,1993
- [34] 李士同.模糊系统、模糊神经网络及应用程序设计[M].上海科学技术文献出版社,1998
- [35] 王顺晔,舒迪前.智能控制系统及其应用[M].机械工业出版社,1995
- [36] 陈国良,王煦法,庄镇泉,王东生.遗传算法及其应用[M].人民邮电出版社,1996
- [37] 张其光,王执铨.基于遗传算法的模糊神经网络控制器设计及其稳定性分析[J].控制理论与应用.1999,16(5):767-769
- [38] 李琳.基于 GA 的模糊神经网络及其在故障诊断中的应用[J].武汉理工大学硕士学位论文,2001
- [39] 程启明,陈熙源,万德钧.基于遗传算法的模糊神经网络智能控制

- 器及其应用[J].系统工程与电子技术,1999,21(8)41-44
- [40] 田华.基于遗传算法的模糊神经网络控制器设计及应用:[哈尔滨理工大学硕士学位论文]. 哈尔滨:哈尔滨理工大学,2003
- [41] 樊慧丽.基于遗传算法的模糊神经网络控制:[太原理工大学硕士学位论文].太原:太原理工大学,2003
- [42] 王鹤曾.工业过程 PID 控制器的设计与性能评价研究[浙江大学硕士学位论文].杭州:浙江大学,2004
- [43] 曹刚.PID 控制器参数整定方法及其应用研究[浙江大学硕士学位论文]. 杭州:浙江大学,2004
- [44] 徐建明.PID 控制器及其设计方法研究. [浙江工业大学硕士学位论文]. 杭州:浙江工业大学,2004
- [45] 张木军,司徒忠.基于模糊神经网络和遗传算法的仿人智能 PID 控制器设计[J].现代制造工程, 2002,23(3)23-25
- [46] 贺素良,王湘中,喻寿益.遗传算法及模糊、神经网络融合技术的研究[J].计算机工程,2003,29(7)
- [47] 王耀南.An Adaptive Control Using Fuzzy Logic Neural Network and Its Application[J].控制理论与应用,1995,12(4)
- [48] Zhi-Wei Woo, Hung-Yuan Chung, Jin-Jye Lin. A PID type fuzzy controller with self-tuning scaling factors[J]. Fuzzy Sets and Systems.115(2000):321-326
- [49] Astrom K J, Hagglund T. PID Controllers:Theory, Design and Tuning[J]. New York: ISA, 1995
- [50] Astrom K J, Hagglund T.PID Controllers:Theory, Design and Tuning, 2nd Edition[J]. Research Triangle Park, North Carolina: Instrument Society of America, 1995
- [51] King P. J. and Mamdani E. H. The Application of Fuzzy Control System to Industrial Processes[J]. Special Interest Discussion on Fuzzy Automata and Decision Processes, 6th IFAC World Cong, Boston, Mass, 1975
- [52] Karr C. Applying Genetics to Fuzzy Logic[J]. AI Expert, 1991,3
- [53] Harris C J, Moore C G, Brown M, Intelligent Control: Aspects of Fuzzy logic and Neural Nets[J]. World Scientific Publishing Co. Pte. Ltd. Singapore, 1993

致 谢

值此论文完稿之际，谨向所有关心、支持和帮助过我的老师和同学表示真诚地感谢，正是有了他们的帮助和鼓励，才使我能够顺利地完成课题的全部工作。

特别感谢我的导师阎树田教授，他严谨、务实的治学态度和积极、负责的敬业精神无时不给我极大的鼓舞，让我终生受益。三年来，无论是在学业上还是在生活上，阎老师都给了我莫大的鼓舞和帮助，让我不断地丰富自己，充实自己，从他那里我学到了作为一个科技工作者应具有的品质和修养，也使我更加有信心面对人生道路上的任何挑战。三年的时光虽然短暂，但足以使我一生受益。

感谢机电研究所的各位老师和同学，感谢他们对我的悉心教导和无私帮助！

感谢我的家人对我学业的支持和鼓励，正是他们的默默支持和帮助，使我能够顺利地完成学业！

附录 A 攻读硕士学位期间发表的学术论文

- [1] S.T.Yan,P.J.Liu,Z.L.Tang.Mathematics Analysis of the Influence of Metal Cutting Machine SpindleMovement Error on Machining Precision. The 11th International Manufacturing Conference in China, Jinan, China. September 18-20, 2004.
- [2] S.T.Yan,Z.L.Tang,P.J.Liu.Mathematics Analysis of the Influence of Axis Error of Gear with Circular Tooth Profile on Transmission Precision. The 11th International Manufacturing Conference in China, Jinan, China. September 18-20, 2004.
- [3] 阎树田, 刘鹏军, 苏玉瑞, 乔伟峰. 一种基于模糊神经网络和遗传算法的智能 PID 控制器. 兰州理工大学学报 (已录用, 拟发表于 2006 年第 4 期)