

第3章

专家控制系统

专家系统(expert system, ES)是人工智能另一个最早的和主要的应用研究领域。20 世纪 70 年代中期,专家系统的开发获得成功。正如专家系统的先驱费根鲍姆(Feigenbaum)所说:专家系统的力量是从它处理的知识中产生的,而不是从某种形式主义及其使用的参考模式中产生的。这正符合一句名言:知识就是力量。到 80 年代,专家系统在全世界得到迅速发展和广泛应用。现在,专家系统并不过时,而是不断更新,被称为“21 世纪知识管理和决策的技术”。

顾名思义,专家控制系统是一个应用专家系统技术的控制系统,也是一个典型的和广泛应用的基于知识的控制系统。海斯·罗思(Hayes Roth)等在 1983 年提出专家控制系统。他们指出,专家控制系统的全部行为能被自适应地支配;为此,该控制系统必须能够重复解释当前状况,预测未来行为,诊断出现问题的原因,制定补救(校正)规划,并监控规划的执行,确保成功。关于专家控制系统应用的第一次报道是在 1984 年,它是一个用于炼油的分分布式实时过程控制系统。奥斯特洛姆(Åström)等在 1986 年发表他们的题为“专家控制”(Expert Control)的论文。从此之后,更多的专家控制系统获得开发与应用。专家系统和智能控制两者都是以模仿人类智能为基础的,而且都涉及某些不确定性问题。专家控制既包括高层控制(决策与规划),又可涉及低层控制(动作与实现)。

本章主要讨论如下 5 个问题,即专家系统基本原理、专家系统的主要类型及其结构、专家控制系统的结构与类型、专家控制器的设计以及专家控制系统的应用实例等。下面我们将逐一对它们加以介绍。

3.1 专家系统的基本概念

自从 1965 年第一个专家系统 DENDRAL 在美国斯坦福大学问世以来,经过 20 年的研究开发,到 80 年代中期,各种专家系统已遍布各个专业领域,取得很大的成功。现在,专家系统得到更为广泛的应用,并在应用开发中得到进一步发展。

3.1.1 专家系统的定义与一般结构

1. 专家系统的定义

定义 3.1 专家系统

专家系统是一个智能计算机程序系统,其内部含有大量的某个领域专家水平的知识与

经验,能够利用人类专家的知识 and 解决问题的方法来处理该领域问题,以人类专家的水平完成特别困难的某一专业领域的任务。

也就是说,专家系统是一个具有大量的专门知识与经验的程序系统,它应用人工智能技术和计算机技术,根据某领域一个或多个专家提供的知识和经验,进行推理和判断,模拟人类专家的决策过程,即模仿人类专家如何运用他们的知识和经验来解决所面临问题的方法、技巧和步骤,以便解决那些需要人类专家处理的复杂问题。简而言之,专家系统是一种模拟人类专家解决领域问题的计算机程序系统。

定义 3.2 专家系统(Weiss 和 Kulikowski,1984)

专家系统能够处理现实世界中需要专家做出解释的复杂问题,并使用专家推理的计算机模型解决这些问题,得出与专家相同的结论。

而专家就是那些擅长解决特定问题的专门人才。

定义 3.3 专家系统(Durkin,1994)

专家系统是一个设计用于建立人类专家问题求解能力模型的计算机程序。

定义 3.4 专家系统(Brachman,Amarel 和 Feigenbaum 等,1988)

对专家系统的定义使用了下列 7 个相对独立的方面:

- (1) 获取专门知识,使用高级规则,避免盲目搜索,有效解决问题。
- (2) 采用符号表示和推理。
- (3) 具有智能,注重领域原理,使用弱推理法。
- (4) 问题具有较大的复杂度和求解难度。
- (5) 进行重新描述,把术语的描述转化为适于专家规则形式的描述。
- (6) 具有不同形式的解释推理的能力,尤其是对过程自身的推理的解释能力。
- (7) 建立系统要完成的总任务和功能。

定义 3.5 基于知识的专家系统(Giarratano 和 Riley,1998)

专家系统是广泛应用专门知识以解决人类专家水平问题的人工智能的一个分支。专家系统有时又称为基于知识的系统或基于知识的专家系统。

专家系统的另两种定义如下:

定义 3.6 专家系统

专家系统是利用存储在计算机内的某一特定领域的人类专家知识,来解决需要人类专家才能解决的现实问题的计算机系统。

定义 3.7 专家系统

专家系统是一种具有大量专门知识和经验的智能计算机系统,通常主要指计算机软件系统。

2. 专家系统的一般结构

专家系统的结构是指专家系统各组成部分的构造方法和组织形式。系统结构选择恰当与否,是与专家系统的适用性和有效性密切相关的。选择什么结构最为恰当,要根据系统的应用环境和所执行任务的特点而定。例如,MYCIN 系统的任务是疾病诊断与解释,其问题的特点是需要较小的可能空间、可靠的数据及比较可靠的知识,这就决定了它可采用穷尽检索解空间和单链推理等较简单的控制方法和系统结构。与此不同,HEARSAY-II 系统的任务是进行口语理解,需要检索巨大的可能解空间,数据和知识都不可靠,缺少问题的比较固

定的路线,经常需要猜测才能继续推理等。这些特点决定了 HEARSAY-II 必须采用比 MYCIN 更为复杂的系统结构。

图 3.1 表示专家系统的简化结构图。图 3.2 则为理想专家系统的结构图。由于每个专家系统所需要完成的任务和特点不相同,其系统结构也不尽相同,一般只具有图中部分模块。

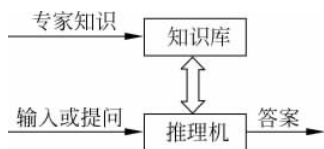


图 3.1 专家系统简化结构图

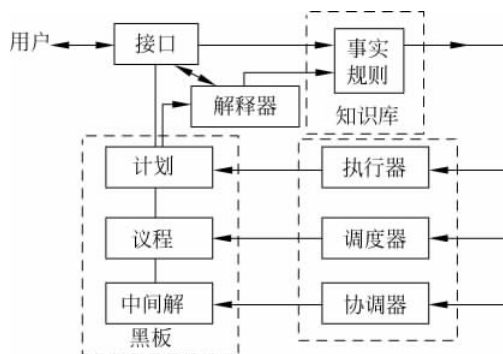


图 3.2 理想专家系统结构图

接口是人与系统进行信息交流的媒介,它为用户提供了直观方便的交互作用手段。接口的功能是识别与解释用户向系统提供的命令、问题和数据等信息,并把这些信息转化为系统的内部表示形式。另一方面,接口也将系统向用户提出的问题、得出的结果和作出的解释以用户易于理解的形式提供给用户。

黑板是用来记录系统推理过程中用到的控制信息、中间假设和中间结果的数据库,它包括计划、议程和中间解 3 部分。计划记录了当前问题总的处理计划、目标、问题的当前状态和问题背景。议程记录了一些待执行的动作,这些动作大多是由黑板中已有结果与知识库中的规则作用而得到的。中间解区域中存放当前系统已产生的结果和候选假设。

知识库包括两部分内容:一部分是已知的同当前问题有关的数据信息;另一部分是进行推理时要用到的一般知识和领域知识。这些知识大多以规则、网络 and 过程等形式表示。

调度器按照系统建造者所给的控制知识(通常使用优先权办法),从议程中选择一个项作为系统下一步要执行的动作。执行器应用知识库中的及黑板中记录的信息,执行调度器所选定的动作。协调器的主要作用就是当得到新数据或新假设时,对已得到的结果进行修正,以保持结果前后的一致性。

解释器的功能是向用户解释系统的行为,包括解释结论的正确性及系统输出其他候选解的原因。为完成这一功能,通常需要利用黑板中记录的中间结果、中间假设和知识库中的知识。

专家系统程序与常规的应用程序之间有何不同呢?一般应用程序与专家系统的区别在于:前者把问题求解的知识隐含地编入程序,而后者则把其应用领域的问题求解知识单独组成一个实体,即为知识库。知识库的处理是通过与知识库分开的控制策略进行的。更明确地说,一般应用程序把知识组织为两级:数据级和程序级;大多数专家系统则将知识组织成三级:数据、知识库和控制。

在数据级上,是已经解决了的特定问题的说明性知识以及需要求解问题的有关事件的当前状态。在知识库级,是专家系统的专门知识与经验。是否拥有大量知识是专家系统成功与否的关键,因而知识表示就成为设计专家系统的关键。在控制程序级,根据既定的控制策略和所求解问题的性质来决定应用知识库中的哪些知识,这里的控制策略是指推理方式。按照是否需要概率信息来决定采用非精确推理或精确推理。推理方式还取决于所需搜索的程度。

下面把专家系统的主要组成部分归纳如下。

(1) 知识库(knowledge base)

知识库用于存储某领域专家系统的专门知识,包括事实、可行操作与规则等。为了建立知识库,要解决知识获取和知识表示问题。知识获取涉及知识工程师(knowledge engineer)如何从专家那里获得专门知识的问题;知识表示则要解决如何用计算机能够理解的形式表达和存储知识的问题。

(2) 综合数据库(global database)

综合数据库又称全局数据库或总数据库,它用于存储领域或问题的初始数据和推理过程中得到的中间数据(信息),即被处理对象的一些当前事实。

(3) 推理机(reasoning machine)

推理机用于记忆所采用的规则和控制策略的程序,使整个专家系统能够以逻辑方式协调地工作。推理机能够根据知识进行推理和导出结论,而不是简单地搜索现成的答案。

(4) 解释器(explanator)

解释器能够向用户解释专家系统的行为,包括解释推理结论的正确性以及系统输出其他候选解的原因。

(5) 接口(interface)

接口又称界面,它能够使系统与用户进行对话,使用户能够输入必要的数据、提出问题和了解推理过程及推理结果等。系统则通过接口,要求用户回答提问,并回答用户提出的问题,进行必要的解释。

3.1.2 专家系统的建造步骤

成功地建立专家系统的关键在于尽可能早地着手建立系统,从一个比较小的系统开始,逐步扩充为一个具有相当规模和日臻完善的试验系统。

建立专家系统的一般步骤如下:

(1) 设计初始知识库。知识库的设计是建立专家系统最重要和最艰巨的任务。初始知识库的设计包括:

① 问题知识化,即辨别所研究问题的实质,如要解决的任务是什么,它是如何定义的,可否把它分解为子问题或子任务,它包含哪些典型数据等。

② 知识概念化,即概括知识表示所需要的关键概念及其关系,如数据类型、已知条件(状态)和目标(状态)、提出的假设以及控制策略等。

③ 概念形式化,即确定用来组织知识的数据结构形式,应用人工智能中各种知识表示方法把与概念化过程有关的关键概念、子问题及信息流特性等变换为比较正式的表达,它包括假设空间、过程模型和数据特性等。

④ 形式规则化,即编制规则,把形式化了的知识变换为由编程语言表示的可供计算机执行的语句和程序。

⑤ 规则合法化,即确认规则化了知识的合理性,检验规则的有效性。

(2) 原型机(prototype)的开发与试验。在选定知识表达方法之后,即可着手建立整个系统所需要的实验子集,它包括整个模型的典型知识,而且只涉及与试验有关的足够简单的任务和推理过程。

(3) 知识库的改进与归纳。反复对知识库及推理规则进行改进试验,归纳出更完善的结果。经过相当长时间(例如数月至两三年)的努力,使系统在一定范围内达到人类专家的水平。

这种设计与建立步骤,如图 3.3 所示。

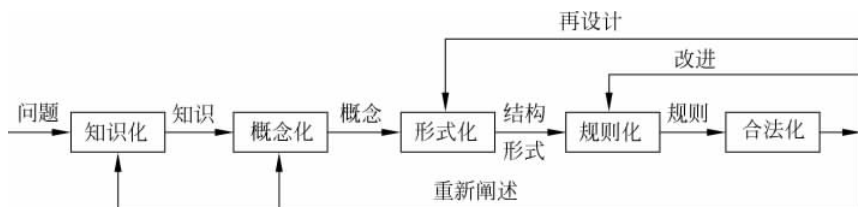


图 3.3 建立专家系统的步骤

3.2 专家系统的主要类型与结构

本节将根据专家系统的工作机理,逐一讨论基于规则的专家系统、基于框架的专家系统和基于模型的专家系统(可分别简称为规则专家系统、框架专家系统和模型专家系统)的工作机理及结构。

3.2.1 基于规则的专家系统

1. 基于规则专家系统的工作模型

产生式系统的思想比较简单,然而却十分有效。产生式系统是专家系统的基础,专家系统就是从产生式系统发展而成的。基于规则的专家系统是个计算机程序,该程序使用一套包含在知识库内的规则对工作存储器内的具体问题信息(事实)进行处理,通过推理机推断出新的信息。其工作模型如图 3.4 所示。

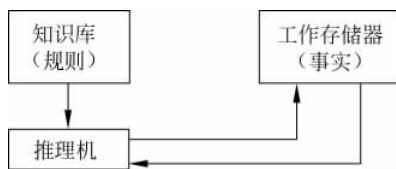


图 3.4 基于规则专家系统的工作模型

从图 3.4 可见,一个基于规则的专家系统采用下列模块来建立产生式系统的模型:

- (1) 知识库。以一套规则建立人的长期存储器模型。
- (2) 工作存储器。建立人的短期存储器模型,存放问题事实和由规则激发而推断出的新事实。
- (3) 推理机。借助于把存放在工作存储器内的问题事实和存放在知识库内的规则结合

起来,建立人的推理模型,以推断出新的信息。推理机作为产生式系统模型的推理模块,对事实与规则的先决条件(前项)进行比较,看看哪条规则能够被激活。通过这些激活规则,推理机把结论加进工作存储器,并进行处理,直到再没有其他规则的先决条件能与工作存储器内的事实相匹配为止。

基于规则的专家系统不需要一个人类问题求解的精确匹配,而能够通过计算机提供一个复制问题求解的合理模型。

2. 基于规则专家系统的结构

一个基于规则专家系统的完整结构示于图 3.5。其中,知识库、推理机和工作存储器是构成本专家系统的核心,已在上面叙述过。其他组成部分或子系统如下:

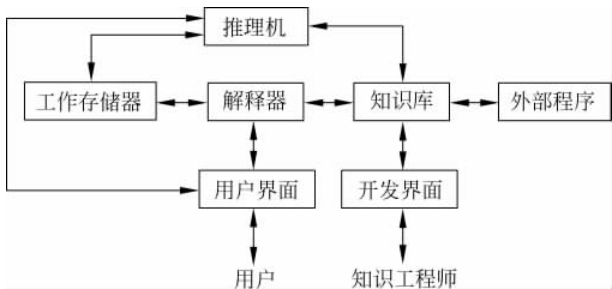


图 3.5 基于规则专家系统的结构

- (1) 用户界面(接口)。用户通过该界面来观察系统,并与系统对话(交互)。
- (2) 开发(者)界面。知识工程师通过该界面对专家系统进行开发。
- (3) 解释器。对系统的推理提供解释。
- (4) 外部程序。如数据库、扩展盘和算法等,对专家系统的工作起支持作用。它们应易于为专家系统所访问和使用。

所有专家系统的开发软件,包括外壳和库语言,都将为系统的用户和开发者提供不同的界面。用户可能使用简单的逐字逐句的指示或交互图示。在系统开发过程中,开发者可以采用原码方法或被引导至一个灵巧的编辑器。

解释器的性质取决于所选择的开发软件。大多数专家系统外壳(工具)只提供有限的解释能力,诸如,为什么提这些问题以及如何得到某些结论。库语言方法对系统解释器有更好的控制能力。

基于规则的专家系统已有数十年的开发和应用历史,并已被证明是一种有效的技术。专家系统开发工具的灵活性可以极大地减少基于规则专家系统的开发时间。尽管在 20 世纪 90 年代,专家系统已向面向目标的设计发展,但是基于规则的专家系统仍然继续发挥重要的作用。基于规则的专家系统具有许多优点和不足之处,在设计开发专家系统时,使开发工具与求解问题匹配是十分重要的。

3.2.2 基于框架的专家系统

框架是一种结构化表示方法,它由若干个描述相关事物各方面及其概念的槽构成,每个槽拥有若干侧面,每个侧面又可拥有若干个值。

1. 面向目标编程与基于框架设计

基于框架的专家系统就是建立在框架的基础之上的。一般概念存放在框架内,而该概念的一些特例则表示在其他框架内并含有实际的特征值。基于框架的专家系统采用面向目标编程技术,以提高系统的能力和灵活性。现在,基于框架的设计和面向目标的编程共享许多特征,以致在应用“目标”和“框架”这两个术语时,往往引起某些混淆。

面向目标编程涉及的所有数据结构均以目标形式出现。每个目标含有两种基本信息,即描述目标的信息和说明目标能够做些什么的信息。应用专家系统的术语来说,每个目标具有陈述知识和过程知识。面向目标编程为表示实际世界目标提供了一种自然的方法。现实中观察到的世界,一般都是由物体组成的,如小车、鲜花和蜜蜂等。

在设计基于框架系统时,专家系统的设计者们把目标叫做框架。现在,从事专家系统开发研究和应用者,已交替使用这两个术语而不产生混淆。

2. 基于框架专家系统的结构

与基于规则的专家系统的定义类似,基于框架的专家系统也是个计算机程序,该程序使用一组包含在知识库内的框架对工作存储器内的具体问题信息进行处理,通过推理机推断出新的信息。这里是采用框架而不是采用规则来表示知识。框架提供一种比规则更为丰富的获取问题知识的方法,不仅提供某些目标的包描述,而且还规定该目标如何工作。

为了说明设计和表示框架中的某些知识值,考虑图 3.6 所示的人类框架结构。图中,每个圆看作面向目标系统中的一个子目标,而在基于框架系统中看作某个框架。用基于框架系统的术语来说,存在孩子对父母的特征,以表示框架间的自然关系。例如,约翰是父辈“男人”的孩子,而“男人”又是“人类”的孩子。

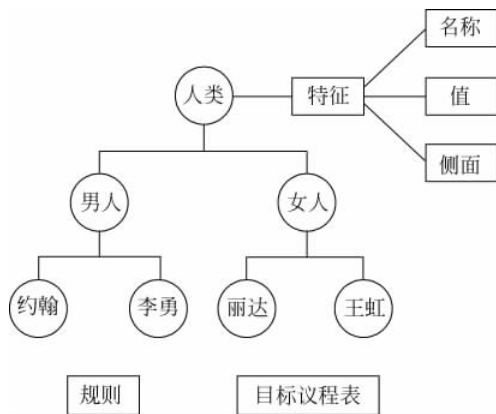


图 3.6 人类的框架分层结构

在图 3.6 中,最顶部的框架表示“人类”这个抽象的概念,通常称之为类(class)。附于这个类框架的是“特征”,有时称为槽(slots),是某个这类物体一般属性的表列。附于该类的所有下层框架将继续继承所有特征,每个特征有它的名称和值,还可能有一组侧面,以提供更进一步的特征信息。一个侧面可用于规定对特征的约束,或者用于执行获取特征值的过程,或者说明在特征值改变时应该做些什么。

图 3.6 的中层,是两个表示“男人”和“女人”这种不太抽象概念的框架,它们自然地附属于其前辈框架“人类”。这两个框架也是类框架,但附属于其上层类框架,所以称为子类

(subclass)。底层的框架隶属于其适当的中层框架,表示具体的物体,通常称为例子(instances),它们是其前辈框架的具体事物或例子。

这些术语,类、子类和例子(物体)用于表示对基于框架系统的组织。从图 3.6 还可以看到,某些基于框架的专家系统还采用一个目标议程表(goal agenda)和一套规则。该议程表仅仅提供要执行的任务表列。规则集合则包括强有力的模式匹配规则,它能够通过搜索所有框架,寻找支持信息,从整个框架世界进行推理。

更详细地说,“人类”这个类的名称为“人类”,其子类为“男人”和“女人”,其特征有年龄、居住地、期望寿命、职业和受教育情况等。子类和例子也有相似的特征。这些特征,都可以用框架表示。

3. 基于框架专家系统的一般设计方法

基于框架专家系统的主要设计步骤与基于规则的专家系统相似。它们都依赖于对相关问题的理解,从而能够提供对问题的洞察,采用最好的系统结构。对于基于规则的系统,需要得到组织规则和结构以求解问题的基本思想和方法;而对于基于框架的系统,需要了解各种物体是如何相互关联并用于求解问题的。在设计初期,就要为课题选好正确的编程语言或支撑工具(外壳等)。

对于任何类型的专家系统,其设计是个高度交互的过程。开始时,开发一个小的有代表性的原型(prototype),以证明它对相关课题的可行性。然后对这个原型进行试验,获得课题进行的思想,涉及系统的扩展、存在知识的深化和对系统的改进,使系统变得更聪明。

设计上述两种专家系统的主要差别在于如何看待和使用知识。对于基于规则的专家系统,把整个问题看做是被简练地表示的规则,每条规则获得问题的一些启发信息,这些规则的集合概括和体现了专家对问题的全面理解。设计者的工作就是编写每条规则,使它们在逻辑上抓住专家的理解和推理。在设计基于框架的专家系统时,对问题的看法截然不同。要把整个问题和每件事想象为编织起来的事物。在第一次会见专家之后,要采用一些非正式方法(如黑板、记事本等),列出与问题有关的事物。这些事物可能是有形的实物(如汽车、风扇、电视机等),也可能是抽象的东西(如观点、故事、印象等),它们代表了专家所描述的主要问题及其相关内容。

在辨识事物之后,下一步是寻找把这些事物组织起来的方法。这一步包括:把相似的物体一起收集进类-例关系中,规定事物相互通信的各种方法等。然后,就应该能够选择一种框架结构以适合问题的需求。这种框架不仅应提供对问题的自然描述,而且应能够提供系统的实现方法。

开发基于框架的专家系统的主要任务如下:

- (1) 定义问题,包括对问题和结论的考察与综述。
- (2) 分析领域,包括定义事物、事物特征、事件和框架结构。
- (3) 定义类及其特征。
- (4) 定义例及其框架结构。
- (5) 确定模式匹配规则。
- (6) 规定事物通信方法。
- (7) 设计系统界面。
- (8) 对系统进行评价。

(9) 对系统进行扩展,深化和扩宽知识。

基于框架的专家系统能够提供基于规则专家系统所没有的特征,如继承、侧面、信息通信和模式匹配规则等,因而也就提供了一种更加强大的开发复杂系统的工具。也就是说,基于框架的专家系统具有比基于规则的专家系统更强的功能,适用于求解更复杂的问题。

3.2.3 基于模型的专家系统

1. 基于模型专家系统的提出

对人工智能的研究内容有着各种不同的看法。有一种观点认为:人工智能是对各种定性模型(物理的、感知的、认识的和社会的系统模型)的获得、表达及使用的计算方法进行研究的学问。根据这一观点,一个知识系统中的知识库是由各种模型综合而成的,而这些模型又往往是定性的模型。由于模型的建立与知识密切相关,所以有关模型的获取、表达和使用自然地包括了知识获取、知识表达和知识使用。所说的模型概括了定性的物理模型和心理模型等。以这种观点看待专家系统的设计,可以认为一个专家系统是由一些原理与运行方式不同的模型综合而成的。

采用各种定性模型来设计专家系统,其优点是显而易见的。一方面,它增加了系统的功能,提高了性能指标;另一方面,可独立地深入研究各种模型及其相关问题,把求得的结果用于改进系统设计。专家系统开发工具 PESS(purity expert system)共利用四种模型,即基于逻辑的心理模型、神经网络模型、定性物理模型以及可视知识模型。这四种模型不是孤立的,PESS 支持用户将这些模型进行综合使用。基于这些观点,已完成了基于神经网络的核反应堆故障诊断专家系统及中医医疗诊断专家系统,为克服专家系统中知识获取这一瓶颈问题提供了一种解决途径。定性物理模型则提供了对深层知识及推理的描述功能,从而提高了系统的问题求解与解释能力。至于可视知识模型,既可有效地利用视觉知识,又可在系统中利用图形来表达人类知识,并完成人机交互任务。

前面讨论过的基于规则的专家系统和基于框架的专家系统都是以逻辑心理模型为基础的,是采用规则逻辑或框架逻辑,并以逻辑作为描述启发式知识的工具而建立的计算机程序系统。综合各种模型的专家系统无论在知识表示、知识获取还是知识应用上都比那些基于逻辑心理模型的系统具有更强的功能,从而有可能显著改进专家系统的设计。

在诸多模型中,人工神经网络模型的应用最为广泛。早在 1988 年,就把神经网络应用于专家系统,使传统的专家系统得到发展。

2. 基于神经网络的专家系统

神经网络模型从知识表示、推理机制到控制方式,都与目前专家系统中的基于逻辑的心理模型有本质的区别。知识从显式表示变为隐式表示,这种知识不是通过人员的加工转换成规则,而是通过学习算法自动获取的。推理机制从检索和验证过程变为网络上隐含模式对输入的竞争。这种竞争是并行的和针对特定特征的,并把特定论域输入模式中各个抽象概念转化为神经网络的输入数据以及根据论域特点适当地解释神经网络的输出数据。

如何将神经网络模型与基于逻辑的心理模型结合是值得进一步研究的课题。从人类求解问题来看,知识存储与低层信息处理是并行分布的,而高层信息处理则是顺序的。演绎与归纳是不可少的逻辑推理,两者结合起来能够更好地表现人类的智能行为。从综合两种模型的专家系统的设计来看,知识库由一些知识元构成,知识元可为一神经网络模块,也可以

是一组规则或框架的逻辑模块。只要对神经网络的输入转换规则和输出解释规则给予形式化表达,使之与外界接口及系统所用的知识表达结构相似,则传统的推理机制和调度机制都可以直接应用到专家系统中。神经网络与传统专家系统的集成,协同工作,优势互补。根据侧重点不同,其集成有三种模式:

(1) 神经网络支持专家系统。以传统的专家系统为主,以神经网络的有关技术为辅。例如对专家提供的知识和案例,通过神经网络自动获取知识;又如运用神经网络的并行推理技术以提高推理效率。

(2) 专家系统支持神经网络。以神经网络的有关技术为核心,建立相应领域的专家系统,采用专家系统的相关技术完成解释等方面的工作。

(3) 协同式的神经网络专家系统。针对大型的复杂问题,将其分解为若干子问题,针对每个子问题的特点,选择用神经网络或专家系统加以实现,在神经网络和专家系统之间建立一种耦合关系。

图 3.7 表示一种神经网络专家系统的基本结构。其中,自动获取模块输入、组织并存储专家提供的学习实例、选定神经网络的结构、调用神经网络的学习算法,为知识库实现知识获取。当新的学习实例输入后,知识获取模块通过对新实例的学习,自动获得新的网络权值分布,从而更新了知识库。

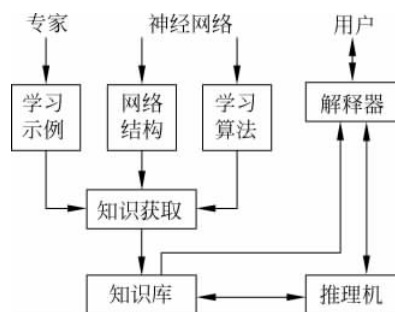


图 3.7 神经网络专家系统的基本结构

下面讨论神经网络专家系统的几个问题。

(1) 神经网络的知识表示是一种隐式表示,是把某个问题领域的若干知识彼此关联地表示在一个神经网络中。对于组合式专家系统,同时采用知识的显式表示和隐式表示。

(2) 神经网络通过实例学习实现知识自动获取。领域专家提供学习实例及其期望解,神经网络学习算法不断修改网络的权值分布。经过学习纠错而达到稳定权值分布的神经网络,就是神经网络专家系统的知识库。

(3) 神经网络的推理是个正向非线性数值计算过程,同时也是一种并行推理机制。由于神经网络各输出节点的输出是数值,因而需要一个解释器对输出模式进行解释。

(4) 一个神经网络专家系统可用加权有向图表示,或用邻接权矩阵表示,因此,可把同一知识领域的几个独立的专家系统组合成更大的神经网络专家系统,只要把各个子系统间有连接关系的节点连接起来即可。组合神经网络专家系统能够提供更多的学习实例,经过学习训练能够获得更可靠更丰富的知识库。与此相反,若把几个基于规则的专家系统组合成更大的专家系统,由于各知识库中的规则是各自确定的,因而组合知识库中的规则冗余度和不一致性都较大;也就是说,各子系统的规则越多,组合的大系统知识库就越不可靠。

3.3 专家控制系统的结构与与设计

定义 3.8 专家控制系统

应用专家系统概念和技术,模拟人类专家的控制知识与经验而建造的控制系统,称为专家控制系统。

专家系统与专家控制系统之间有一些重要的差别:

(1) 专家系统只对专门领域的问题完成咨询作用,协助用户进行工作。专家系统的推理是以知识为基础的,其推理结果为知识项、新知识项或对原知识项的变更知识项。然而,专家控制系统需要独立和自动地对控制作用做出决策,其推理结果可为变更的知识项,或者为启动(执行)某些解析算法。

(2) 专家系统通常以离线方式工作,而专家控制系统需要获取在线动态信息,并对系统进行实时控制。实时控制要求遇到下列一些难题:非单调推理、异步事件、基于时间的推理以及其他实时问题。

跻身自动控制领域的专家控制被视为求解控制问题队伍的重要方面军,而且在过去 20 多年中在各种领域进行许多开发与应用。工作在不同领域和具有不同专业背景的人们已对专家控制系统表现出巨大兴趣,热情不减。

本节首先介绍专家控制系统的结构,然后提出专家控制系统的控制要求和设计原则,最后讨论专家控制器的设计问题。

3.3.1 专家控制系统的结构

1. 专家控制系统的一般结构

图 3.8 给出专家控制系统的原理图。从图 3.8 可见,以专家控制器取代传统控制,如反馈控制系统中的 PID 控制器,即可构成专家控制系统。而如同专家系统一样,知识库和推理机是专家控制器的核心组成部分。

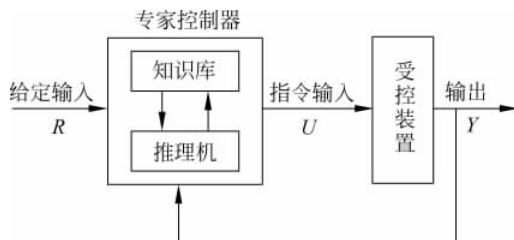


图 3.8 专家控制系统原理图

专家控制系统随着应用场合和控制要求的不同,其结构也可能不一样。然而,几乎所有的专家控制系统(控制器)都包含知识库、推理机、控制规则集和/或控制算法等。

图 3.9 画出专家控制系统的基本结构。从性能指标的观点看,专家控制系统应当为控制目标提供同师傅或专家操作时得到的一样或十分相似的性能指标。

下面举例讨论一种专家控制系统,即黑板专家控制系统的具体结构,如图 3.10 所示。

黑板结构是一种强功能的专家系统结构和问题求解模型,它能够处理大量不同的、错误的和不完全的知识,以求解问题。基本黑板结构是由一个黑板(BB)、一套独立的知识源(KSs)和一个调度器组成。黑板为一共享数据区;知识源存储各种相关知识;调度器起控制作用。黑板系统提供了一种用于组织知识应用和知识源之间合作的工具。黑板系统的最大优点在于它能够控制提供控制的灵活性及综合各种不同的知识表示和推理技术。黑板控制系统由三个部分组成。

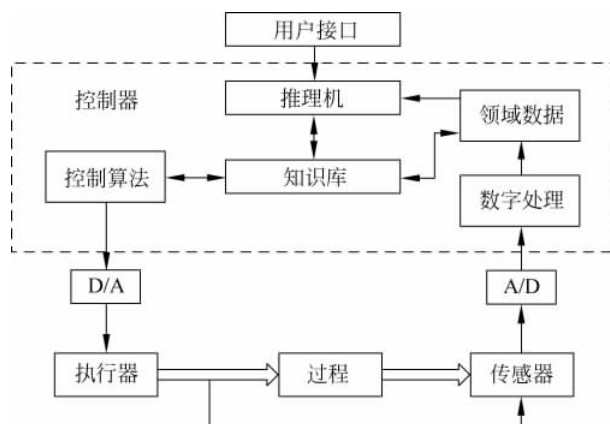


图 3.9 专家控制器的典型结构

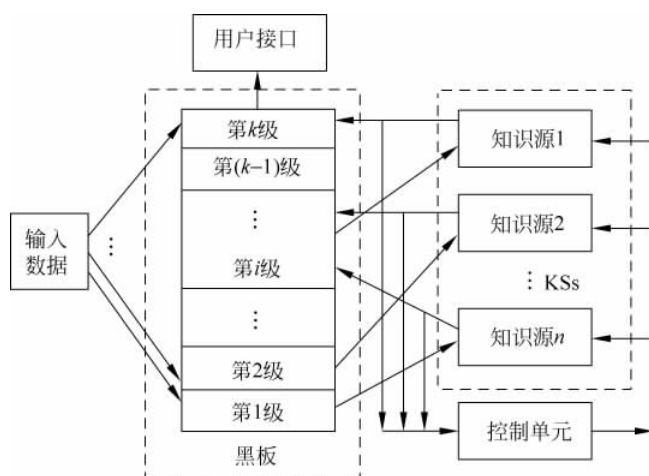


图 3.10 黑板专家控制系统的结构

(1) 黑板(BB)

黑板用于存储所有知识源可访问的知识,它的全局数据结构被用于组织问题求解数据,并处理各知识源之间的通信问题。放在黑板上的对象可以是输入数据、局部结果、假设、选择方案和最后结果等。各知识源之间的交互作用是通过黑板执行的。一个黑板可被分割为无数个子黑板;也就是说,按照求解问题的不同方面,可把黑板分为几个黑板层,如图 3.10 中的第 1 层至第 k 层。因此,各种对象可被递阶地组织进不同的分析层级。

在黑板上的每一个记录条目可有个相关的置信因子,这是系统处理知识不确定性的一种方法。黑板的机理能够保证在每个知识源与已求得的局部解之间存在一个统一的接口。

(2) 知识源(KSs)

知识源是领域知识的自选模块;每个知识源可视为专门用于处理一定类型的较窄领域信息或知识的独立程序,而且具有决定是否应当把自身信息提供给问题求解过程的能力。黑板系统中的知识源是独立分开的,每个知识源具有自己的工作过程或规则集合和自有的数据结构,包含知识源正确运行所必需的信息。知识源的动作部分执行实际的问题求解,并

产生黑板的变化。知识源能够遵循各种不同的知识表示方法和推理机制。因此,知识源的动作部分可为一个含有正向/逆向搜索的产生式规则系统,或者是一个具有填槽过程的基于框架的系统。

(3) 控制器

黑板系统的主要求解机制是由某个知识源向黑板增添新的信息开始的。然后,这一事件触发其他对新送来的信息感兴趣的知識源。接着,对这些被触发的知识源执行某些测试过程,以决定它们是否能够被合法执行。最后,一个被触发了的知识源被选中,执行向黑板增添信息的任务。这个循环不断进行下去。

控制黑板是一个含有控制数据项的数据库,控制器应用这些数据项从一组潜在可执行的知识源中挑选出一个供执行用的知识源。高层规划和策略应在程序执行前以最适合问题状况的方式决定和选择。一组控制知识源,能够不断建构规划以达到系统性能;这些规划描述了求解控制问题所需的作用。规划执行后,控制黑板上的信息得以增补或修改。然后,控制器应用任何一个记录在控制黑板上的启发性控制方法,实现控制作用。

黑板的控制结构使得系统能够对那些与当前挑选的中心问题相匹配的知识源,给予较高的优先权。这些注意的中心可在控制黑板上变化。因此,该系统能够探索和决定各种问题求解策略,并把注意力集中到最有希望的可能解答上。

2. 直接专家控制系统

曾根据系统结构的复杂性把专家控制系统分为两种形式,即专家控制系统和专家控制器。现在我们将按照系统的作用机理来讨论专家控制系统的结构类型。

(1) 直接专家控制系统的结构

专家控制器有时又称为基于知识控制器,以基于知识控制器在整个系统中的作用为基础,可把专家控制系统分为直接专家控制系统和间接专家控制系统两种。在直接专家控制系统中,控制器向系统受控过程直接提供控制信号,产生控制作用,如图 3.11(a)所示。图 3.12 给出直接专家控制系统的一个例子。

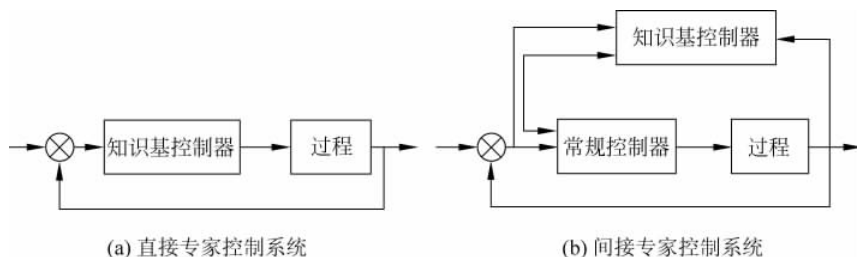


图 3.11 两种专家控制系统原理图

直接专家控制系统的基于知识控制器直接模仿人类专家或人类的认知能力,并为控制器设计两种规则:训练规则和机器规则。训练规则由一系列产生式规则组成,它们把控制误差直接映射为受控对象的作用。机器规则是由积累和学习人类专家/师傅的控制经验得到的动态规则,并用于实现机器的学习过程。在间接专家系统中,智能(基于知识)控制器用于调整常规控制器的参数,监控受控对象的某些特征,如超调、上升时间和稳定时间等,然后拟定校正 PID 参数的规则,以保证控制系统处于稳定的和高质量的运行状态。

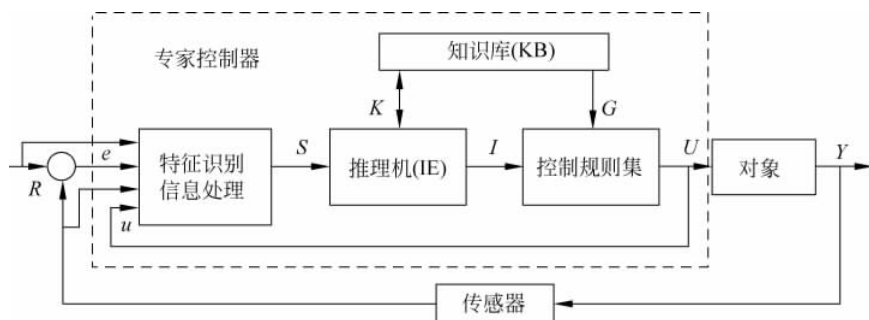


图 3.12 工业专家控制器简化结构图

专家控制器(EC)的基础是知识库(KB),KB 存放工业过程控制的领域知识,由经验数据库(DB)和学习与适应装置(LA)组成。经验数据库主要存储经验和事实。学习与适应装置的功能就是根据在线获取的信息,补充或修改知识库内容,改进系统性能,以便提高问题求解能力。图 3.12 所示的工业专家控制器各部分的作用说明如下。

① 知识库 建立知识库的主要问题是表达已获取的知识。EC 的知识库用产生式规则来建立,这种表达方式具有较高的灵活性,每条产生式规则都可独立地增删、修改,使知识库的内容便于更新。

② 控制规则集(CRS) 控制规则集是对受控过程的各种控制模式和经验的归纳和总结。由于规则条数不多,搜索空间很小,推理机构(IE)就十分简单,采用向前推理方法逐次判别各种规则的条件,满足则执行,否则继续搜索。

③ 特征识别与信息处理(FR&IP) 这个部分的作用是实现对信息的提取与加工,为控制决策和学习适应提供依据。它主要包括抽取动态过程的特征信息,识别系统的特征状态,并对这些特征信息进行必要的加工。

④ 推理机(同前专家系统)。

⑤ 控制规则集(同前专家系统)。

(2) 直接专家控制器的输入输出关系

专家控制器的输入集为

$$E = (R, e, Y, U) \quad (3.1)$$

$$e = R - Y \quad (3.2)$$

式中, R 为参考控制输入, e 为误差信号, Y 为受控输出, U 为控制器的输出集。

I 、 G 、 U 、 K 和 E 之间的关系可由下式表示,即

$$U = f(E, K, I, G) \quad (3.3)$$

其中, E 为专家控制器的误差输入; K 为知识库对推理机的输出; I 为推理机的输出; G 为知识库对控制规则单元的输出; f 为智能算子,是几个算子的复合运算

$$f = g \cdot h \cdot p \quad (3.4)$$

其中, g , h , p 也是智能算子,而且有

$$\left. \begin{aligned} g: E &\rightarrow S \\ h: S \times K &\rightarrow I \\ p: I \times G &\rightarrow U \end{aligned} \right\} \quad (3.5)$$

式中, S 为特征信息输出集, G 为规则修改指令。这些算子具有下列形式

$$\text{IFA} \quad \text{THEN} \quad B \tag{3.6}$$

其中, A 为前提或条件, B 为结论, A 与 B 之间的关系也可以包括解析表达式、模糊关系、因果关系和经验规则等多种形式。 B 还可以是一个规则子集。

3. 间接专家控制系统

专家控制器既含有算法(如基于规则或模型算法)和逻辑,可以根据算法与逻辑分离构建控制系统。控制底层可能是 PID、逻辑或神经网络算法,再与自校正、自适应、自调度、自监控等环节相连。系统按照规则提供的启发式知识,使各功能算法得以正常实现。这种专家系统只间接对控制系统的受控过程发生作用,因而称为间接专家控制系统。

(1) 间接专家控制系统的结构

间接专家控制系统的工作原理如图 3.11(b)所示。间接专家控制系统又可称为监控式专家控制系统。

图 3.13 表示一种间接专家控制系统的作用原理框图,图中的专家控制器能够协调各种算法,利用专家经验规则决定何时采用何种参数和启动什么算法。控制系统工作时,就像一个颇有经验的控制专家能够自在地调度系统参数与结构,并及时回答用户的咨询。

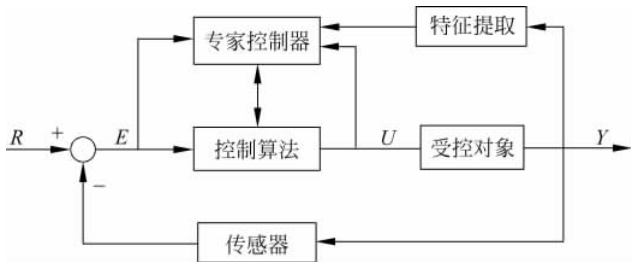


图 3.13 间接专家控制系统的作用原理框图

专家整定 PID 控制系统是一种比较有代表性的间接专家控制系统,其结构如图 3.14 所示。其中,PID 控制器的参数整定过程由推理机和知识库等组成的专家系统实现。

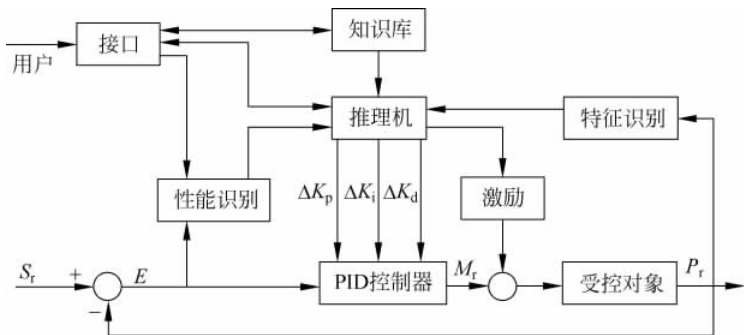


图 3.14 专家整定 PID 控制系统结构图

(2) 专家整定 PID 控制系统工作原理

由图 3.14 可见,专家控制系统的控制信号由 PID 控制器提供给受控对象,而专家系统

通过对 PID 控制器的参数整定起到间接的控制作用。受控对象的输出信号 Y , 经传感器反馈到系统输入端与给定信号 R 进行比较, 得到偏差信号 E 并作用于 PID 控制器。

① 特征识别 控制系统的输出随控制输入、系统参数和工作环境的变化而变化, 响应形式有平稳收敛、振荡发散、振荡收敛、等幅振荡等, 而响应特性可由调节时间、超调量、峰值时间和衰减比等描述。专家系统能够在线识别特征参数。

② 知识表示与知识获取 整定专家通过实践总结出来的参数调整规程实现对知识点整定, 并采用产生式规则来表示。规则的前项表示调整规程的适用条件, 即控制过程响应曲线的特征描述; 而规则的后项则表示调用该调整规程时进行的操作, 即对 PID 控制器的参数调整。

③ 知识推理过程 专家系统的推理机按照响应曲线的当前特征描述, 选择合适的规则, 进行相应的操作, 确定 PID 控制器参数的调整方向和调整量, 改善系统的控制性能, 获得满意的控制结果。推理机采用前向推理方式。首先根据用户给定的性能指标函数计算系统性能, 当系统性能不能满足要求时, 启动推理过程, 对 PID 参数进行整定。推理过程如下:

启动特征识别程序, 获得控制系统的当前状态描述。

推理求解。根据状态描述, 应用知识库中的调整规程确定 PID 参数的调整方向与调整量。

参数整定。推理得出的 PID 参数经过用户或专家认可后, 修改 PID 控制器参数, 而控制器以更新后的参数运行。

结束。当系统的控制性能满足要求时, 参数整定过程结束; 否则, 进入新的下一轮推理过程。

3.3.2 专家控制系统的控制要求与设计原则

至今为止的自适应控制存在两个显著缺点, 即要求具有准确的装置模型以及不能为自适应机理设定有意义的目标。专家控制器不存在这些缺点, 因为它避开了装置的数学模型, 并为自适应设计提供有意义的时域目标。

1. 专家控制系统的控制要求

一般说来, 对专家控制系统没有统一的和固定的要求, 这种要求是由具体应用决定的。不过, 可以对专家控制系统提出一些综合要求:

(1) 运行可靠性高

对于某些特别的装置或系统, 如果不采用专家控制器来取代常规控制器, 那么, 整个控制系统将变得非常复杂, 尤其是其硬件结构, 其结果使系统的可靠性大为下降。因此, 对专家控制器提出较高的运行可靠性要求, 通常具有方便的监控能力。

(2) 决策能力强

决策是基于知识的控制系统的关键能力之一。大多数专家控制系统要求具有不同水平的决策能力。专家控制系统能够处理不确定性、不完全性和不精确性之类的问题, 这些问题难以用常规控制方法解决。

(3) 应用通用性好

应用的通用性包括易于开发、示例多样性、便于混合知识表示、全局数据库的活动维数

和基本硬件的机动性、多种推理机制(如假想推理、非单调推理和近似推理)以及开放式的可扩充结构等。

(4) 控制与处理的灵活性

包括控制策略的灵活性、数据管理的灵活性、经验表示的灵活性、解释说明的灵活性、模式匹配的灵活性以及过程连接的灵活性等。

(5) 拟人能力

专家控制系统的控制水平必须达到人类专家的水平。

专家控制系统的控制要求是根据应用情况指定的。例如,有个过程控制,对其专家控制器的具体要求与下列情况有关:连续操作,对不同工作档采用多重专家操作,输入材料质量的不相容性,随时间逐渐改变的过程,非常复杂的装置结构,多传感器,对不同的控制任务采用适当的与不同的装置描述级别以及装置的模型可能具有不同的形式等。

2. 专家控制器的设计原则

根据上述讨论,可以进一步提出专家控制器的设计原则如下:

(1) 模型描述的多样性

在设计过程中,对被控对象和控制器的模型应采用多样化的描述形式,而不应拘泥于单纯的解析模型。现有的控制理论对控制系统的设计都唯一依赖于受控对象的数学解析模型。在专家式控制器的设计中,由于采用了专家系统技术,能够处理各种定性的与定量的、精确的与模糊的信息,因而允许对模型采用多种形式的描述。这些描述形式主要有:

① 解析模型。主要表达方式有微分方程、差分方程、传递函数、状态空间表达式和脉冲传递函数等。

② 离散事件模型。用于离散系统,并在复杂系统的设计和分析方面找到更多的应用。

③ 模糊模型。在不知道对象的准确数学模型而只掌握了受控过程的一些定性知识时,用模糊数学的方法建立系统的输入和输出模糊集以及它们之间的模糊关系则较为方便。

④ 规则模型。产生式规则的基本形式为

$$\text{IF (条件) THEN (操作或结论)} \quad (3.7)$$

这种基于规则的符号化模型特别适于描述过程的因果关系和非解析的映射关系等。它具有较强的灵活性,可方便地对规则加以补充或修改。

⑤ 基于模型的模型。对于基于模型的专家系统,其知识库含有不同的模型,其中包括物理模型和心理模型(如神经网络模型和视觉知识模型等),而且通常是定性模型。这种方法能够通过离线预计算来减少在线计算,产生简化模型使之与所执行的任务逐一匹配。

此外,还可根据不同情况采用其他类型的描述方式。例如,用谓词逻辑来建立系统的因果模型,用符号矩阵来建立系统的联想记忆模型等。

总之,在专家式控制器的设计过程中,应根据不同情况选择一种或几种恰当的描述方式,以求更好地反映过程特性,增强系统的信息处理能力。

专家式控制器一般模型可用式(3.3)表示,即

$$U = f(E, K, I, G)$$

其中 f 为智能算子,其基本形式为

$$\text{IF } E \text{ AND } K \text{ THEN (IF } I \text{ THEN } U) \quad (3.8)$$

其中, $E = \{e_1, e_2, \dots, e_m\}$ 为控制器输入集; $K = \{k_1, k_2, \dots, k_n\}$ 为知识库中的经验数据与事实集; $I = \{i_1, i_2, \dots, i_p\}$ 为推理机构的输出集; $U = \{u_1, u_2, \dots, u_q\}$ 为控制器输出集。

智能算子的基本含义是: 根据输入信息 E 和知识库中的经验数据 K 与规则进行推理, 然后根据推理结果 I , 输出相应的控制行为 U 。智能算子的具体实现方式可采用前面介绍的各种方式(包括解析型和非解析型)。图 3.12 中给出这些参量的位置。

(2) 在线处理的灵巧性

智能控制系统的重要特征之一就是能够以有用的方式来划分和构造信息, 在设计专家式控制器时应十分注意对过程在线信息的处理与利用。在信息存储方面, 应对做出控制决策有意义的特征信息进行记忆, 对于过时的信息则应加以遗忘; 在信息处理方面, 应把数值计算与符号运算结合起来; 在信息利用方面, 应对各种反映过程特性的特征信息加以抽取和利用, 不要仅限于误差和误差的一阶导数。灵活地处理与利用在线信息将提高系统的信息处理能力和决策水平。

(3) 控制策略的灵活性

控制策略的灵活性是设计专家式控制器所应遵循的一条重要原则。工业对象本身的时变性与不确定性以及现场干扰的随机性, 要求控制器采用不同形式的开环与闭环控制策略, 并能通过在线获取的信息灵活地修改控制策略或控制参数, 以保证获得优良的控制品质。此外, 专家式控制器中还应设计异常情况处理的适应性策略, 以增强系统的应变能力。

(4) 决策机构的递阶性

人的神经系统是由大脑、小脑、脑干、脊髓组成的一个递阶决策系统。以仿智为核心的智能控制, 其控制器的设计必然要体现递阶原则, 即根据智能水平的不同层次构成分级递阶的决策机构。

(5) 推理与决策的实时性

对于设计用于工业过程的专家式控制器, 这一原则必不可少。这就要求知识库的规模不宜过大, 推理机构应尽可能简单, 以满足工业过程的实时性要求。

由于专家式控制器在模型的描述上采用多种形式, 就必然导致其实现方法的多样性。虽然构造专家式控制器的具体方法各不相同, 但归纳起来, 其实现方法可分为两类: 一类是保留控制专家系统的结构特征, 但其知识库的规模小, 推理机构简单; 另一类是以某种控制算法(例如 PID 算法)为基础, 引入专家系统技术, 以提高原控制器的决策水平。专家式控制器虽然功能不如专家控制系统完善, 但结构较简单, 研制周期短, 实时性好, 具有广阔的应用前景。

3.3.3 专家控制系统的设计问题

专家系统是专家控制器的主要组成部分和重要基础, 因此, 有关专家系统的各种知识, 包括专家系统的结构、建造、知识表示与推理等, 对专家控制器的设计也是十分有用的。不过, 本节讨论专家控制器的设计, 主要涉及冲突消解策略设计和用于控制的知识获取问题。

1. 冲突消解策略设计

推理机制往往由知识库的先验和独立的知识所说明。一般地, 设计者应当对各种不同的受控对象采用同样的推理机制, 而仅仅改变知识库以反映如何适当地控制被考虑的具体

受控对象。可以采用替代方案来设计专家控制器的推理机制,特别是选择冲突消解策略的次序和类型。例如,对于某些应用,如果有的冲突消解策略不能真正模仿手头控制问题的正确决策方法,那么就可以忽略这些策略。

设计者还可以修改前面讨论过的冲突消解策略。譬如说,可以这样修改它们,使得每次一条规则激活时,允许重新考虑一定的其他规则而不管该激活规则是否影响其他规则的匹配。这类冲突消解策略可独立应用上述讨论过的同样机理;不过,当激活规则 i 允许(不允许)在冲突消解策略中考虑规则 j 时,必须重新定义矩阵 A 使得 a_{ij} 为 1(0)。这类冲突消解策略可能具有较小的内存和较有效的计算,但是,由于这类冲突消解策略的设计需要设计者采用 ad-hoc 方法对其进行调整直到满足设计目标为止,因此该设计方法很难开发。建议采用某些冲突消解策略或者它们的子集,适当地调整知识库。可能找出一些好理由(如它不能适当模仿专家的决策等)而不使用某一具体的冲突消解策略。在有些情况下,可设计一种只实现特殊性、优先权和任意性 3 种冲突消解策略的推理机。这种推理机具有较小的计算要求并易于实现,因为该推理机没有状态($X^c = X^b$)。一些有代表性的冲突消解策略(步骤)如下:

(1) 折射(refraction) 移去冲突集合中所有已被激活了的规则。不过,如果激活一条规则影响了其他规则前项的匹配数据,那么要对这些规则进行冲突消解。

(2) 修正(recency) 根据知识库中匹配每条规则前项的“年龄”信息,为激活规则分配一个优先权。把匹配某条规则前项的数据“年龄”规定为冲突集合中的规则上一次被激活以来的规则激活数字。

(3) 特殊性(distinctiveness) 激活与规则库中的大多数(或最重要)数据相匹配的规则(在专家系统中应用许多不同种类的特殊性检测)。因此,计算用于规则前项的不同前项函数 P_i 的数字,并以此数字作为对特殊性的量测。

(4) 优先权配置(priority schemes) 为规则指定一个优先权等级,然后从冲突集合中选出具有最高优先权的规则作为激活规则。

(5) 任意性(arbitrary) 随机地从冲突集合中取出一条待激活的规则。

2. 控制知识获取

下面说明如何把控制受控对象的知识加载到知识库。有两种知识库设计方法。第一种为标准的专家系统设计方法,第二种是把受控对象模型并入知识库,成为知识库的一部分,即应用基于模型的控制来协助控制决策。

对于专家系统的知识库设计方法,通过把那些与给定输入和对象输出(规则的左项,即前项)及对象输入(规则的右项,即后项)直接相关的规则,加入知识库,实现控制知识。首先,必须设计一种具有 m 元变量的状态 x^b ,以便与受观察的对象输出和给定输入(及其他可能的中间变量)的 m 个条件相对应。这些条件可以表示对象输出、输出变化率、给定输入等。其次,必须规定管理知识库状态 x^b 更新过程的规则 r 的集合,状态 x^b 由给定输入和对象输出决定。

值得指出,规则的后项表示规则库内的信息状态随输入事件的出现如何变化,不过,它们不直接对受控对象提供控制作用,受控对象的控制作用是由专家控制器的状态对受控对象可实现指令输入事件的映射决定的。可以预定状态 x^b 的一个元素(变量),用于表示受控对象的可实现指令输入事件。换句话说,可以把知识库状态内的一个元素 x_i^b 与控制器的当

前输入联系起来,使 $x_i^b \in E_o$,例如,令控制器的全部输出事件标上数字($j=1,2,\dots,n$);然后,可定义 $\delta^c(x_k^c)$,譬如说为 x_{5k}^b (状态 x^b 的第 5 个元素); $x_{5k}^b=j$ 意味着在时间 k 可实现的对象指令输入为指令输入事件 j 。

对于一些受控对象,需要设计一种把对象输出和给定输入直接映射至控制作用的控制器。这种设计方法与模糊控制方法相似或采用“IF P_i THEN C_i ”形式的规则,其中, P_i 和 C_i 为仅对推理环的指令输入函数。对于这些情况,状态 x^b 具有一维, x^b 表示控制器的输出事件,而且 $\delta^c(x^c)=x^b$ 。

另一种设计方法是把受控对象模型并入知识库,这与传统控制系统中采用的基于模型的方法相似。受控对象的状态被包括为知识库状态 x^b 的一部分,知识库中受控对象的状态必须用对象的输出来更新。因此,必须细心地说明知识库中对象的模型,以便应用可得到的信息准确地反映对象的动态行为。

除受控对象的模型状态外,状态 x^b 还可包含事实真值 $T(a_i)$,它对应于受控对象和控制器行为的历史状况(受控对象和控制器的过去条件)和其他变量。必须指定知识库规则集合 $r \in R$ 以表示知识库的状态 x^b 的变换函数和对象的可实现指令输入。也可以与上面的设计方法一样,预定状态 x^b 的一个元素(变量),用于表示受控对象的可实现指令输入事件。对于这类基于模型设计来说,后项公式要比先前的公式复杂,因为它们要同时更新用于控制器的受控对象模型。

3.4 专家控制系统的建模

专家控制系统也是一个反馈控制系统。因此,在本节中我们首先可以借鉴传统控制的理论和方法来讨论受控对象(装置、过程)的数学模型,然后建立专家控制器的模型(包括规则库和推理机的动力学模型),最后讨论专家控制系统的模型。

3.4.1 受控对象模型

假设受控对象可由模型 G 表示,即能够实际表示一定类型的离散事件系统

$$G = (X, E, f_e, \delta_e, g, E_v) \quad (3.9)$$

式中, X 为受控对象状态 x 的集合; $E=E_u \cup E_d \cup E_o$ 为事件的集合,而 E_u 为对象指令输入事件的集合, E_d 为对象扰动输入事件的集合, E_o 为对象输出事件的集合; $g: X \rightarrow P(E_u \cup E_d) - \{\emptyset\}$ 为可实现函数(enable function),而 P 表示能力集合; $f_e: X \rightarrow X, e \in P(E_u \cup E_d) - \{\emptyset\}$,为状态转换映射; $\delta_e: X \rightarrow E_o, e \in P(E_u \cup E_d) - \{\emptyset\}$,为输出映射; E_v 为全部有效事件轨迹的集合。

值得指出,为方便起见,我们始终采用“状态”这一术语,但是对于建模的某些系统来说, x 并非传统意义上的状态。在讨论时间 k 的状态和事件时, $x_k \in X$ 为对象状态, $e_{u,k} \in E_u$ 为对象的指令输入事件, $e_{d,k} \in E_d$ 为对象的扰动输入事件, $e_{o,k} \in E_o$ 为对象的输出事件。我们称 $e_k \subset g(x_k)$ 为时间 k 的可实现事件,是上述指令和扰动输入事件的集合。为了方便,假定 e_k 包含每种 E_u 和 E_d 的大多数事件。如果一个事件是可实现的,那么该事件就可能出现。如果一个事件 e_k 在时间 k 出现,而且当前状态为 x_k ,那么下一个状态为 $x_{k+1} = f_{e,k}(x_k)$,而输出为 $e_{o,k} = \delta_{e,k}(x_k)$ 。要求当 $e \subset g(x)$ 时 f_e 和 δ_e 是确定的,而且 E_v 允许 e 出现。

对于所有 k , 任何有限或无限长度序列 $\{x_k\}$ 使得 $x_{k+1} = f_{e,k}(x_k)$, 就称 $e_k \subset g(x_k)$ 为状态轨迹。全部事件轨迹的集合由序列 $\{e_k\}$ 组成, 使得存在一个状态轨迹 $\{x_k\}$ 对于所有 k 满足 $e_k \subset g(x_k)$ 。因此, 对于每个指定函数 f_e 应用次序(事件激发次序)的事件轨迹, 对应某个唯一的状态轨迹(不过, 一般上反之不然)。定义 E 为所有无限和有限长度事件轨迹(事件 $e \in P(E_u \cup E_d) - \{\emptyset\}$ 的序列)的集合, 该轨迹可由 g 和 f_e 生成。集合 $E_v \subset E$ 为实际上可能有的全部事件轨迹; 因此, 即使 $x_k \in X$ 和 $e_k \subset g(x_k)$, 仍然不可能出现 e_k 的情况, 除非它位于这样的一条有效事件轨迹上, 该轨迹结束于 x_{k+1} 且 $x_{k+1} = f_{e,k}(x_k)$ 。该有效事件轨迹 E_v 允许对实际系统建模有一定的灵活性。

3.4.2 专家控制器的模型

式(3.5)给出专家控制的一种模型, 表示控制器各参量(集合)间的一种定性关系。下面介绍另一种如图 3.15 所示的专家控制器的建模方法。

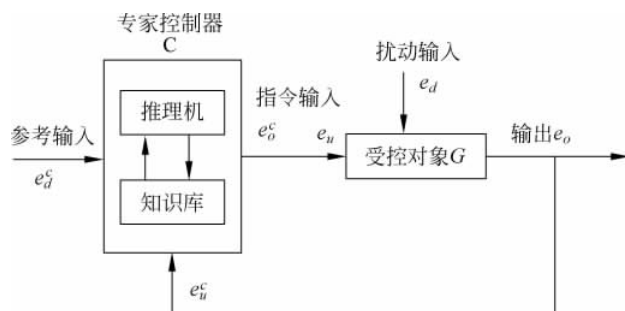


图 3.15 另一种专家控制系统

1. 专家控制器建模

在图 3.15 中, 专家控制器有两个输入, 即给定(参考)输入事件 $R = e_d^c \in E_d^c$ 和受控对象的输出事件 $Y = e_u^c \in E_u^c$, 因此, 专家控制器模型说明在控制环内的人如何协调应用受控对象的反馈信息、给定输入(建立当前控制目标模型)和控制器存储的信息(控制状态)。此外, 专家控制器还建立认知过程的模型, 用于决定如何应用这全部信息做出执行哪个控制动作的决策。专家控制器中推理机和知识库间的交互作用, 常常称为“推理环”。该推理环构成专家控制器的核心; 在该核心, 知识库内的信息由推理机进行解释, 产生作用, 知识库被更新, 而这一过程不断重复进行(所以称为环)。

图 3.15 所示的专家控制器具有如下模型

$$C = (X^c, E^c, f_e^c, \delta_e^c, g^c, E_v^c) \quad (3.10)$$

式中, $X^c = X^b \times X^i$ 为专家控制器状态 x^c 的集合, X^b 为知识库状态 x^b 的集合, X^i 为推理机状态 x^i 的集合;

$E^c = E_u^c \cup R \cup E_o^c$ 为专家控制器事件的集合, 而 $E_u^c \subset P(E_o \cup E_d^c) - \{\emptyset\}$ 为给定输入集合(E_d^c)和对对象输出事件(E_o)的集合, R 为专家控制器知识库内的规则集合, $E_o^c \subset P(E_u) - \{\emptyset\}$ 为专家控制器输出事件的集合;

$g^c: X^b \times X^i \rightarrow P(E_u^c \cup R) - \{\emptyset\}$ 为可实现函数;

$f_e^c: X^b \times X^i \rightarrow X^b \times X^i, e \in P(E_u^c \cup R) - \{\emptyset\}$, 为状态转换映射;

$\delta_o: X^b \times X^i \rightarrow E_o^c$ 为输出映射;

$E_v^c \subset E^c$ 为有效推理环轨迹(控制器事件轨迹)的集合。

在上述框架中,假设一个指令输入事件的出现总是伴随着一条使能规则 $r \in R$ 的激活,使得推理环能够相应更新。与此相似,一条规则 $r \in R$ 不可能独自激活,因为仅当存在受控对象输出变化或给定输入事件变化时,推理环才能被更新。因此,专家控制器的输入事件应当严格地包含一条规则 $r \in R$ 和一个给定输入事件 $e_u^c \in E_u^c$,每个 e_u^c 最多只有一个对象输出事件 $e_o \in E_o$ 。而且包含给定输入事件 $e_d^c \in E_d^c$ (可用 E_v^c 来建模)。对于 $e \subset g^c(x^c)$, $f_e^c(x^c)$ 为描述当对象输出、给定输入或两者变化以及规则 r 激活时知识库和推理机状态的更新。专家控制器的输出事件 $\delta_o(x^c)$ 被定义为对受控对象 G 的可实现指令输入。基于规则库和冲突消解策略,有效推理环轨迹 $E_v^c \subset E^c$ 为也可约束输入事件轨迹。值得指出,当控制器的状态变化时,对象的使能指令输入事件不可能同时出现;不过,这些事件可在以后的任何时间出现。这个条件是闭环专家控制系统适当定时所必需的。控制器能够控制对象(装置或过程)的指令输入事件能力,但并没有控制对象扰动输入事件的任何能力。对本专家控制器 C 的完全说明是通过规定专家控制器的规则库和推理机,即规定推理环的组成部分来达到的。

2. 规则库建模

尽管本章着重基于规则的系统,然而并不把建模仅限于基于规则的系统。指出这一点是很重要的,人工智能中许多知识表示形式也易于表达。此外,在知识推理方面,也有很多方法可用于专家系统和专家控制系统。下面建立规则库的模型。

令 $A = \{a_1, a_2, \dots, a_n\}$ 为可真或假的事实(它们的真值可随时间变化)的集合。假设

$$T: A \rightarrow \{0, 1\} \quad (3.11)$$

式中, $T(a_i) = 1 (=0)$ 表明 a_i 为真(假)。又令 R 表示实数, $V \subset R^m$, 而 $\mathbf{v} \in V$ 表示变量的一个 m 维列矢量。这里所说的事实和变量存放在“工作存储器”内。令 $X^b = R^{m+n}$, 其中 $\mathbf{x}^b \in X^b$, $\mathbf{x}^b = [v^T T(a_1) T(a_2) \dots T(a_n)]^T = [x_1^b, x_2^b, \dots, x_{m+n}^b]$ (其中 T 表示矩阵的转置), 又令 $x_{i,k}^b$ 表示 $\mathbf{x}^b$ 在时刻 k 的第 i 个分量。设 $P_i, i=1, 2, \dots, p$ 表示前提函数 p 的集合, 即

$$P_i: X^b \times E_u^c \rightarrow \{0, 1\} \quad (3.12)$$

而且 $P_i(x_k^b, e_{u,k}^c) = 1$ (或 0) 指明 $P_i(x_k^b, e_{u,k}^c)$ 在 k 时刻为真(假)。 P_i 将用作为规则的前提(先决条件)以规定能动规则的条件, 即建立规则左部的模型。假设前项公式 (antecedent formulas) Φ 由下列递归方法来定义:

(1) 对于所有 $a \in A$ 和 $P_i, i=1, 2, \dots, p, T(a)$ 是前项公式。

(2) 如果 Φ 和 Φ' 都是前项公式, 那么 $\sim \Phi, \Phi \wedge \Phi', \Phi \vee \Phi', \Phi \rightarrow \Phi'$ 均为标准的布尔 (Boolean) 连词。

(3) 有限次地应用上述两项规定而得到的公式都是前项公式。

例如, 如果 $m=3, n=2, A = \{a_1, a_2\}, V \subset R^3$, 而 P_1 检验“ $x_{2,k}^b < 5.23$ ”, P_2 检验“ $x_{3,k}^b = 1.89$ ”, P_3 检验“ $e_{d,k}^c < 5 \vee e_{o,k}^c \geq 2$ ”, 其中 $e_{d,k}^c$ 和 $e_{o,k}^c$ 为实数, 那么 $\Phi' = P_1 \wedge P_2 \wedge P_3 \wedge (T(a_1) \vee \sim T(a_2))$ 为一个有效前项公式。令 $C_i, i=1, 2, \dots, q$ 表示后项函数 (consequent function), 其中,

$$C_i: X^b \times E_u^c \rightarrow X^b \quad (3.13)$$

用于表示规则的后项, 即规则的右部, 也就是表示当某条规则被激活时对知识库采取哪个作用。令后项公式 Ψ 由下列递归方法来定义:

(1) 对于任何 $C_i, i=1, 2, \dots, q, C_i$ 为一后项公式。

(2) 对于任何 $C_i, C_j, C_i \wedge C_j$ 为一个后项公式。

(3) 有限次地应用上述两项规定而得到的公式都是后项公式。

规则库中的规则形式如下

$$r = \text{IF } \Phi \text{ THEN } \Psi \quad (3.14)$$

式中,只有当 Φ 估计为真时,才能采取 Ψ 作用。例如,对于规则 $r' = \text{IF } \Phi' \text{ THEN } \Psi'$,如果 Φ' 在时刻 k 对知识库状态 $\mathbf{x}^b$ 和指令输入 $e_{u,k}^c$ 的估计为真,那么 $r' = \text{IF } \Phi' \text{ THEN } \Psi'$ 就是一条在时刻 k 可实现的有效知识库规则。如果 $e'_k = \{r', e_{u,k}^c\}$ 是可实现的而且出现了,那么控制器的下一状态 $\mathbf{x}_{k+1}^b$ 的知识库分量是在时刻 k 把 Ψ' 应用至 $\mathbf{x}^b$ 而生成的。推理机状态的更新是以所有可实现规则 $r \in R$ 和被激活规则 r' 为基础的。把输入事件包含进规则库,允许专家控制器的设计者把受控对象的输出反馈和给定输入变量直接作为规则的一部分进行合并。这是与传统的基于规则专家系统中应用变量相似的。

3. 推理机建模

要建立推理机制模型,必须表示下列 3 个一般函数分量:

(1) 匹配阶段 规则的前提与存储在知识库内的当前事实和数据以及给定输入和受控对象输出相匹配。

(2) 选择阶段 选择一条准备被激活的规则。

(3) 动作阶段 包括在激活规则后项中的动作,作用于知识库,更新推理机状态,并对受控对象生成输入。

推理机制匹配阶段的特性被表示于知识库中。用人工智能术语来说

$$\Gamma_k = \{r: \{r, e_{u,k}^c\} \subset g^c(x_k^c) \text{ 使得规则 } r \in R \text{ 的 } \Phi \text{ 对于 } e_{u,k}^c \text{ 估计为真}\} \quad (3.15)$$

实际上为时刻 k 的知识库“冲突集合”。选择阶段(从 Γ_k 挑选出一条激活规则)组成“冲突消解策略”(即启发式推理策略)。

值得理解的是,特殊性冲突消解策略实际上是优先权配置的特例,之所以把它们分别介绍是由于在以往开发专家系统时发现特殊性很有用。对于一个具体的专家系统,任何数字(固定的、变化的或顺序的数字)的上述冲突消解策略都可用于决定冲突集合中的哪条规则将被激活。这些冲突消解策略通常用于“修剪”知识库冲突集合 Γ_k 的大小,直到获得一个可实现规则的较小集合止。在进行冲突消解修剪之后,组合知识库和推理机的控制器模型 C 中的这些规则为可实现规则。如果应用了全部冲突消解策略,而且有一条以上的规则得以保留,那么第(5)步(任意性)被随机地用于激活一条剩下的规则。作用阶段的建模由算子 f_e^c 来进行;如果某条含有相应输入事件的规则出现于推理环,那么 f_e^c 表示对知识库和推理机采取的作用。

知识库中规则的优先权和特殊性对所有时间都是固定的,但折射和修正随时间变化。因此,推理机状态 $\mathbf{x}^i$ 必须含有折射和修正两者信息。设知识库有 n_r 条规则,并对规则从 1 至 n_r 编号。定义一函数 $\Pi(i)=1$,如果规则 i 从冲突集合中删除; $\Pi(i)=0$,如果规则 i 被考虑进冲突消解。这个函数用于实现选择阶段的折射作用。令 $p=[\Pi(1) \Pi(2) \Pi(3) \cdots \Pi(n_r)]^t$ 为 n_r -矢量,其分量表示当某条规则在状态 $\mathbf{x}^b$ 为有效时该规则是否能被包含在冲突集合中。令 n_r -矢量 $s=[s_1 s_2 s_3 \cdots s_{n_r}]^T$ 在,其中 s_i 表示知识库内与规则 i 的前项匹配的信息年龄的整

数。用 s 来协助表示修剪冲突消解策略。定义推理机的状态为 $\mathbf{x} = [\mathbf{p}^T \mathbf{s}^T]^T \in X^i$ 。

要建立专家控制器的模型,需要充分地定义 g^e 和 f^e 。对 p 和 s 进行更新的状态变换,是以由分量 $\mathbf{x}^i$ 表示的信息的折射和修正为基础的。采用矩阵 A 来说明如何对 p 和 s 进行更新,并被定义具有 $n_r \times n_r$ 维及其第 ij -个分量 $a_{ij} = 1$ (或 0),如果激活规则 i 影响或不影响对规则 j 的匹配。本质上,矩阵 A 包含关于知识库互连结构的静态信息;在动态推理过程开始之前,一旦规则被加进知识库,该知识库就被自动指定。

对于 $i, 1 \leq i \leq n_r$, 采用变量 $\tilde{e}_i, d_i$ 和 p_i 来定义 $\mathbf{x}^b, p$ 和 s 的更新过程;其中, $\tilde{e}_i = 1(0)$ 表示规则 i 为可实现(不可实现)的, d_i 维持规则 i 的特殊性等级(优先权与 p_i 值成正比), p_i 维持规则 i 的优先权等级(优先权与 p_i 值成正比)。当知识库确定之后, d_i 和 p_i 分量就被指定,而且维持不变。 $s_i, \tilde{e}_i$ 和 $\Pi(i)$ 值随时间 k 变化,因此,用 $s_i, \tilde{e}_i$ 和 $\Pi(i)$ 分别表示它们在 k 时的值。

专家控制器中的推理环可用下列方法执行。首先,通过知识获取规定知识库;然后 p, s 和 $\tilde{e}_i$ 被初始化为 0。从时间 k 至 $k+1$ 的推理是通过执行:(1)匹配阶段,(2)选择阶段,(3)动作阶段这 3 个步骤得到的:

(1) 匹配阶段(match phase)

FOR rule $r=1$ TO rule $r=n$, DO:

IF $r \in \Gamma_k$ THEN $\tilde{e}_r^k := 1$ {找到可实现规则}

IF there is just one r' such that: $\tilde{e}_{r'}^k = 1$ THEN GOTO the Act Phase

IF there are no r' such that $\tilde{e}_{r'}^k = 1$ THEN STOP {专家控制器未适当规定,即专家控制器不能作用于所有可能的对象输出和给定输入条件}

(2) 选择阶段(select phase)

FOR rule $r=1$ TO rule $r=n$, DO: {以折射为基础进行修剪}

IF $\tilde{e}_r^k = 1$ THEN

IF $\Pi_k(r) = 1$ THEN $\tilde{e}_r^k := 0$

IF there is just one r' such that $\tilde{e}_{r'}^k = 1$ THEN GOTO the Act Phase

IF there are no r' such that $\tilde{e}_{r'}^k = 1$ THEN STOP {专家控制器未适当规定}

LET $s = -\infty$ {以修正为基础进行修剪}

FOR $j=1$ TO 2 DO: {寻找具有最小年龄值的规则}

FOR rule $r=1$ TO rule $r=n$, DO:

IF $\tilde{e}_r^k = 1$ THEN

IF $-s_r^k < s$ THEN $\tilde{e}_r^k := 0$

ELSE $s := -s_r^k$

IF there is just one r' such that $\tilde{e}_{r'}^k = 1$ THEN GOTO the Act Phase

LET $d = 0$ {以特殊性为基础进行修剪}

FOR $j=1$ TO 2 DO: {寻找具有最大特殊性值的规则}

FOR rule $r=1$ TO rule $r=n$, DO:

IF $\tilde{e}_r^k = 1$ THEN

IF $d_r < d$ THEN $\tilde{e}_r^k := 0$

ELSE $d := d_r$

IF there is just one r' such that $\tilde{e}_{r'}^k = 1$ THEN GOTO the Act Phase

LET $p = 0$ {以优先权为基础进行修剪}

FOR $j=1$ TO 2 DO: {寻找具有最高优先权值的规则}
 FOR rule $r=1$ TO rule $r=n$, DO:
 IF $\tilde{e}_r^k=1$ THEN
 IF $p_r^k < p$ THEN $\tilde{e}_r^k := 0$
 ELSE $p := p_r$
 LET r' be any r such that $\tilde{e}_r^k=1$ {以任意性为基础进行修剪}

(3) 动作阶段(act phase)

LET $e' = \{r', e_{u,k}^c\}$
 LET $(x_{k+1}^b, x_{k+1}^i) = f_v^c(x_k^c)$ {更新知识库状态, 该状态 x_{k+1}^i 定义于下}
 $\Pi_{k+1}(r') := 1$ {以折射为基础把规则 r 从冲突集合中移出}
 FOR rule $r=1$ TO rule $r=n$, DO:
 IF $r \in \Gamma_k$ THEN $s_r^{k+1} := s_r^k + 1$ {对所有在冲突集合中用于修正的规则增大其匹配年龄}
 FOR $r=1$ TO $r=n$, DO:
 IF $a_{r'}=1$ THEN $\Pi_{k+1}(r) := 0$ and $s_r^{k+1} := 0$ {允许把那些受到规则 r' 激活的规则放入冲突集, 并重置这些规则的年龄为 0}

总之, 专家控制器的操作由下列步骤进行:

- (1) 获取时间 k 时的受控对象输出 $e_{u,k}^c$ 和给定输入事件。
- (2) 在匹配阶段, 根据知识库内的规则集, 以 $e_{u,k}^c$ 、各种事实的当前真值状态和知识库内变量的当前值为基础, 建立冲突集合 Γ_k 。
- (3) 在选择阶段, 应用冲突消解策略(折射、修正、特殊性、优先权、任意), 找到一条待激活的规则 $r' \in \Gamma_k$ 。
- (4) 在动作阶段, 执行由规则 r' 的后项规定的作用, 包括更新知识库和推理机状态以及产生受控对象的输入。

3.4.3 专家控制系统模型

下面讨论闭环专家控制系统(ECS)的模型, 突出专家控制系统的一些定时特性。

1. 闭环专家控制系统的模型

闭环专家控制系统的模型 S 由下式给出

$$S = (X^s, E^s, f_e^s, \delta_e^s, g^s, x_o^s, E_v^s) \quad (3.16)$$

式中, $X^s = X \times X^c$ 为闭环系统状态 x^s 的集合;

$E^s = E_u \cup E_d^s \cup E_o$, $E_d^s = E_d \cup E_d^c$, E_u , E_d 和 E_d^c 为闭环系统的输入事件, E_o 包含受控对象的输出事件;

$g^s: X \times X^c \rightarrow P(E_u \cup E_d^s) - \{\emptyset\}$ 为闭环系统的可实现函数;

$f_e^s: X \times X^c \rightarrow X \times X^c$, $e \in P(E_u \cup E_d^s) - \{\emptyset\}$, 为闭环状态转换映射;

$\delta_e^s: X \times X^c \rightarrow E_o$, $e \in P(E_u \cup E_d^s) - \{\emptyset\}$, 为闭环输出映射;

$x_o^s \in X^s$ 为闭环系统的初始状态;

E_v^s 为全部有效闭环系统事件轨迹的集合(事件轨迹的一个子集, 它能够导致有效的受控对象和控制器的状态轨迹, 并使受控对象和控制器连接在一起)。

$E_v^s \in E^s$, E^s 为所有无限和无限长闭环输入事件轨迹的集合, 该轨迹可由闭环系统 g^s 和

f_e^s 生成。必须仔细设计控制器使得控制器能够消除不需要的闭环系统行为。 E_v^s 表示闭环系统的可能行为。显然, E_v 和 E_v^c 设置了对闭环系统可允许的事件序列, 即对 E^s 的约束。此外, 闭环系统中对事件序列的进一步约束可由 E^s 来表示。有必要指定闭环系统的初始状态以减少无意义的闭环状态组合, 这种组合可能增加不必要的建模复杂性。如果知道闭环系统的初始状态, 那么就能够把状态变换限制在那些可能的闭环状态, 因而就能够消除不可能的闭环状态组合。

下面规定 g^s 和 f_e^s 。

给出闭环系统的初始状态, 于是在时刻 k 状态 $x_o^s = (x_o, x_o^c)$ 的闭环系统可实现函数为

$$g^s(x_k^s) = [g^c(x_k^c) \cap E_d^c] \cup [\delta^c(x_k^c) \cap g(x_k)] \cup [g(x_k) \cap E_d] \quad (3.17)$$

式中, $g^c(x_k^c) \cap E_d^c$ 是控制器在状态 x_k^c 可实现的给定输入事件集合, $\delta^c(x_k^c) \cap g(x_k)$ 是在控制器当前状态 x_k^c 和对对象当前状态 x_k 可实现的受控对象指令输入事件集合, 而 $g(x_k) \cap E_d$ 是在对象当前状态 x_k 可实现的受控对象扰动输入事件集合。闭环系统中可实现的对象输入事件包括可由当前对象状态 x_k 和控制器状态 x_k^c 两者实现的对象事件以及可在对象状态 x_k 实现的扰动输入事件。因此, 控制器仅能控制 E_u 类事件的实现。假定控制器已被适当规定, 使它能够过渡至下一个状态以反映出现的任何对象输出事件。这是规定好闭环系统动力学所需要的。

再来规定 f_e^s 。令 $e_k \subset g^s(x_k^s)$ 为闭环系统中某个在状态 $x_k^s = (x_k, x_k^c)$ 可实现的事件。那么, 如果

$$\begin{aligned} e_{d,k}^c &\in g^c(x_k^c) \cap E_d^c \\ e_{u,k} &\in \delta^c(x_k^c) \cap g(x_k) \\ e_{d,k} &\in g(x_k) \cap E_d \end{aligned}$$

e_k^i 可由下列情况之一给定(取决于同时出现何种指令和扰动输入事件): ① $e_k^1 = \{e_{u,k}, e_{d,k}, e_{d,k}^c\}$; ② $e_k^2 = \{e_{u,k}, e_{d,k}\}$; ③ $e_k^3 = \{e_{u,k}, e_{d,k}^c\}$; ④ $e_k^4 = \{e_{d,k}, e_{d,k}^c\}$; ⑤ $e_k^5 = \{e_{u,k}\}$; ⑥ $e_k^6 = \{e_{d,k}\}$; ⑦ $e_k^7 = \{e_{d,k}^c\}$ 。称 e_k^i 为 i 类事件, 其中, $1 \leq i \leq 7$ 。对于 $i=1, 3, 4, e_{p,k} = \{e_{u,k}, e_{d,k}\}$, $\tilde{e}_k^i = e_k^i - \{e_{d,k}^c\}$, 而 $e_{c,k}^i = \{\delta \text{ 接下标 } e_k^i(x_k), e_{d,k}^c\}$ 。于是对于 e_k^i 类, 有 $f_{ek}^s(x_k^s)$ 及 $x_{k+1}^s = (x_{k+1}, x_{k+1}^c)$, 式中:

- (1) 如果出现 e_k^1 , 那么 $x_{k+1} = f_{cpk}(x_k)$ 和 $x_{k+1}^c = f_z^c(x_k^c)$, 其中 $z = e_{c,k}^1$;
- (2) 如果出现 e_k^2 , 那么 $x_{k+1} = f_{cpk}(x_k)$ 和 $x_{k+1}^c = f_z^c(x_k^c)$, 其中 $z = \{\delta_{c,k}^2(x_k)\}$;
- (3) 如果出现 e_k^3 , 那么 $x_{k+1} = f_{cuk}(x_k)$ 和 $x_{k+1}^c = f_z^c(x_k^c)$, 其中 $z = e_{c,k}^3$;
- (4) 如果出现 e_k^4 , 那么 $x_{k+1} = f_{cdk}(x_k)$ 和 $x_{k+1}^c = f_z^c(x_k^c)$, 其中 $z = e_{c,k}^4$;
- (5) 如果出现 e_k^5 , 那么 $x_{k+1} = f_{cuk}(x_k)$ 和 $x_{k+1}^c = f_z^c(x_k^c)$, 其中 $z = \{\delta_{c,k}^5(x_k)\}$;
- (6) 如果出现 e_k^6 , 那么 $x_{k+1} = f_{cdk}(x_k)$ 和 $x_{k+1}^c = f_z^c(x_k^c)$, 其中 $z = \{\delta_{c,k}^6(x_k)\}$;
- (7) 如果出现 e_k^7 , 那么 $x_{k+1} = x_k$ 和 $x_{k+1}^c = f_z^c(x_k^c)$, 其中 $z = \{e_{d,k}^7\}$ 。

至此, 我们完成了对受控对象(过程或装置)、专家控制器和闭环专家控制系统数学模型的定义与说明。

2. 闭环专家控制系统的相关定时问题

下面让我们讨论闭环专家控制系统的相关定时问题。

假定事件是瞬时出现的,这意味着,如果出现某一事件,那么该事件是在 0 时刻出现的,系统状态也是在 0 时刻发生变化的。闭环系统事件是与受控对象和控制器事件同时出现的。因此,在对象输入事件出现的时刻,对象输出事件引起控制器状态变化为它的下一状态。与这一新的控制器状态相应的是,控制器输出成为下一个可实现的对象指令输入。不过,只可能出现在下一个对象状态可实现的事件。

尽管受控对象的输入事件的每次出现总是影响控制器的状态,但是,控制器输入事件的出现并不需要立即影响受控对象的状态。例如,某个控制器的给定输入事件, $e_d^c \in E_d^c$, 能够自己出现,并引起控制器状态变化至它的下一状态而对受控对象的状态产生影响。这可从上面讨论的第 7 类输入事件的数学表达式看到,只有闭环状态 $x^s = (x, x^c)$ 的控制器状态部分受到影响。

值得指出,闭环系统的可实现输入事件能够与它们在 E_v^s 上输入事件轨迹同时出现。如果对受控对象和控制器的有效事件轨迹施加某些约束,那么就能够把 E_v^s 用于强迫对象指令输入(forced plant command input)事件的建模。两种可能的约束如下:

(1) 每次闭环系统状态发生变化时,下一个可能出现的对象输入事件必须包含由当前控制器状态激活的对象指令输入。

(2) 每次闭环系统状态发生变化时,下一个可能出现的输入事件必须仅是由当前控制器状态激活的受控对象的指令输入事件。

3.5 专家控制系统应用举例

近 20 年来,在过程(流程)工业中开发和应用专家系统的兴趣与日俱增,其中,大部分涉及监控和故障诊断,而且越来越多的专家系统被用于实时过程控制。

3.5.1 实时控制系统的特点与要求

定义 3.9 实时控制系统

如果一个控制系统:①对受控过程表现出预定的足够快的实时行为;②具有严格的响应时间限制而与所用算法无关;那么这种系统称为实时控制系统。

实时系统与非实时系统(如医疗诊断系统)的根本区别在于,实时系统具有与外部环境及时交互作用的能力。换句话说,实时系统得出结论要比装置(对象、过程)快。如果一个系统在组成部件发生爆炸后 3 分钟才报告其灾祸即将出现,那就太糟了!某些常见的实时控制系统包括简单的控制器(如家用电器)和监控系统(如报警系统)等。在飞行模拟、导弹制导、机器人控制和工业过程等系统中,已经应用许多比较复杂的实时系统。这些系统都具有一个共性,即当它们与变化的外部环境交互作用时,都受到处理(控制)时间的约束。实时约束意味着专家控制系统应当自动适应受控过程。

专家系统与实时系统在控制上的集成是开发专家系统技术和实时系统技术的一个合乎逻辑的步骤。实时专家控制系统能够在广泛范围内代替或帮助操作人员进行工作。支持开发实时专家控制系统的一个理由是能够减轻操作者识别负担,从而提高生产效率。

为了提高实时专家控制系统的执行速度,需要采用特别技术。要实现实时推理与决策。专家控制系统的知识库的规模不应太大,推理机制应尽可能简单,一些关键规则可用较低级语言(如C语言或汇编语言)编写。对某些软件包采用调试监督程序。知识库可被分区使得不同类型的知识能分别由单独的处理器执行处理,这就是已介绍过的黑板技术;每一单独处理器可看作独立专家,各处理器之间通过把各自的推理过程结果置于黑板来实现通信;在黑板上,另一专家系统能够获得与应用这些结果。

实时专家控制系统的具体要求和设计特点如下:

- (1) 准确地表示知识与时间的关系。
- (2) 具有快速和灵敏的上下文激活规则。
- (3) 能够控制任意时变非线性过程。
- (4) 能够进行时序推理、并行推理和非单调推理。
- (5) 修正序列的基本控制知识。
- (6) 具有中断过程和异步事件处理能力。
- (7) 及时获取动态和静态过程信息,以便对控制系统进行实时序列诊断。
- (8) 有效回收不再需要的存储元件,并保持传感器的数据。
- (9) 接受来自操作者的交互指令序列。
- (10) 连接常规控制器和其他应用软件。
- (11) 能够进行多专家系统之间以及专家系统与用户之间的通信。

下面以高炉监控专家系统为例,讨论实时专家控制系统的设计和应用问题。

3.5.2 高炉监控专家系统

1. 高炉控制概况

高炉生产过程的操作是一个十分复杂的过程。铁矿和焦炭从炉顶加入,而鼓风机则由底部吹风。为保证生铁冶炼的质量,高炉安装了几百个传感器,从采集的数据中观察高炉内的状况。早已采用计算机对炼铁的高炉进行控制和管理;这种管理控制系统往往采用复杂的数学模型,具有以下三个主要功能,如图3.16所示。

- (1) 数据分析:分析和采集传感器的数据。
- (2) 炉内静态状况分析:当操作约束条件改变很大时,要根据分析结果来寻求最合适的操作方法。
- (3) 炉况诊断:控制操作过程基本上是基于传感器数据的采集、分析和过程模型的建立。

当炉况比较稳定时,这种操作是比较有效的。但是,当炉内状况非常复杂,发生不正常工况而严重干扰炉子运行时,许多操作还是要依靠有经验操作员或专家的知识 and 经验。因此有必要引入专家系统或智能控制系统来改善高炉运行条件,以求提高生铁的质量。开发和建立专家控制系统对高炉进行控制,其主要目的有以下三种:

- (1) 利用人工智能技术,建立准确的控制系统。
- (2) 将高炉操作技术标准化和规范化。
- (3) 灵活处理经常性的系统变化要求。

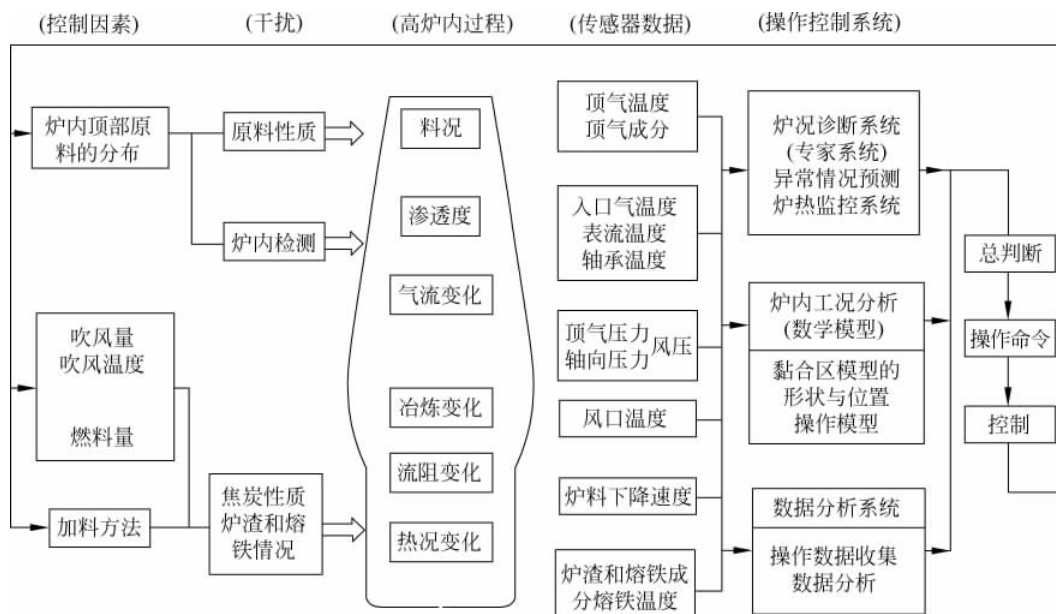


图 3.16 高炉监控的操作及功能

2. 高炉监控专家系统的结构与功能

该监控系统由两部分组成。一是异常炉况预测系统(AFS),用于预测炉内炉料滑动和沟道的产生情况;二是高炉熔炼监控系统(HCS),用于判断炉内熔炼过程并指导操作员对高炉进行合理的操作。

这是一个观察和控制型的专家系统,能够处理时间序列数据,具有实时性。为了实现这些特性,系统应具有两部分功能,见图 3.17。其一是推理的预处理部分,它用常规的方法在过程计算机上执行;其二是推理部分,它用知识工程技术在 AI(人工智能)处理器上实现。前者采集传感器的数据,并把它们寄存在时间序列数据库中,经预处理后形成推理所需的事实数据,且显示推理结果。后者利用从前者所产生的事实数据和知识库的规则,对高炉的状况进行推理。

3. 监控专家系统开发过程

本专家系统的开发工具基于 LISP 语言,常规算法的开发采用 FORTRAN 语言。

对于高炉控制与诊断这样具体的专家系统,其开发过程大体如图 3.18 所示。图中各阶段的工作内容说明如下:

(1) 决定目标。明确系统的功能与所涉及的范围。

(2) 获取知识。研究有关高炉领域的技术文献资料,研究高炉操作员手册;从领域专家搜集知识。

(3) 知识的汇编与系统化。把专家的思维过程进行归纳整理分类;检查其合理性和存在的矛盾;传感器数据模式整理和分类、数据滤波、分级和求导(差分);知识模糊性(不确定性)的表示。

(4) 规则结构的设计。将规则分组和结构化,考虑推理的速度。

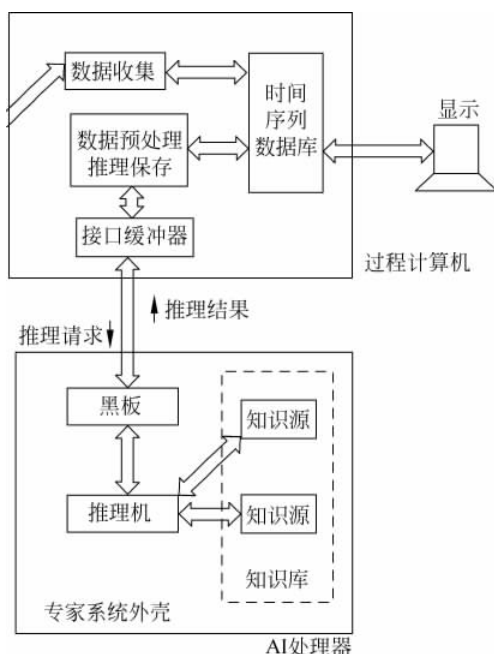


图 3.17 高炉监控专家系统的结构

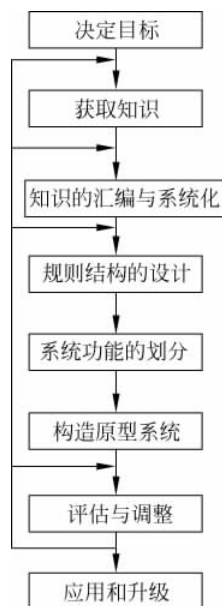


图 3.18 系统开发过程框图

- (5) 系统功能的划分。实现在线实时处理；将系统功能划分为预处理和推理两部分。
- (6) 构造原型系统。描述规则和黑板模型；将实际系统和测试系统形式化。
- (7) 评估与调整。利用离线测试系统调试系统；检查系统的有效性；调节确定性因子的值。
- (8) 应用和升级。增加和校正规则。
- 上述各个步骤中,知识的获取是关键,它要解决的问题涉及:
- (1) 如何表达知识库和规则库中的经验知识的不确定性以便构成高度准确的系统。
 - (2) 如何获取专家自己意识不到或不很明确的知识(对专家而言这种知识也许是常识性的)。
 - (3) 利用某些条件,对密集性知识进行分解。
- 在专家系统开发中必须得到专家(包括操作员和工厂职员)的全力协助。

4. 专家系统的知识表示与知识库结构

由高炉工程师和操作员采集到的领域专门知识和启发性规则都存放在 AI 计算机的知识库中,基本上采用产生式规则来表示知识。在高炉监控中,主要问题是根据传感器的信息来判断炉内工况,预测异常炉况的发生并采取合适的操作。因此,专家系统应构成特征分析型的系统。根据对给定数据的分析结果,选择几种假设中排列在前面的最合适的假设。此外,在高炉熔炼监控系统中采用了基于框架的模型,以表示有关高炉各部分温度和压力的静态知识。用黑板模型来存储时间序列数据,进行知识单元之间的知识传递,并记录推理方法和推理结果。

为了进行实时在线推理,根据传感器的功能属性,将知识库划分为由若干规则集合组成的几个知识单元即子知识库。整个知识库采用递阶式结构,如图 3.19 所示。

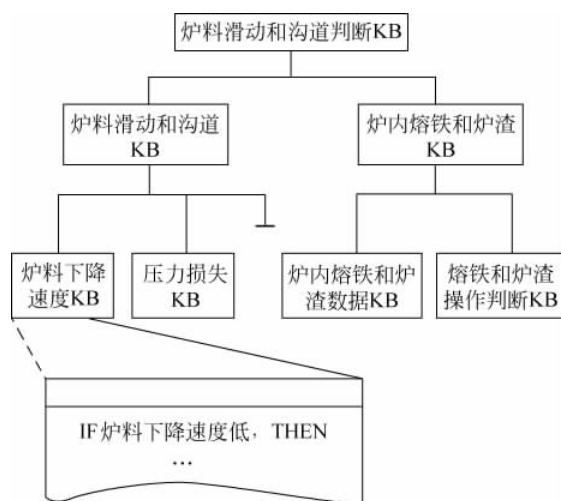


图 3.19 AFS 中知识库的构成

这种设计具有下列优点：

- (1) 知识库按递阶形式构成,便于以不同的方式来表示各子库的专家知识。
- (2) 将规则集合分到各子知识库,便于检查各规则集的有效性。
- (3) 采用递阶式多知识库结构来划分规则,可以改善推理效率。

在异常炉况预测系统中,实用的规则具有如下的形式：

IF 炉料下降的速度低于 XX, THEN 有可能发生沟道(确定因子为: X. XX)

IF 炉料下降速度的积分小于 YY, THEN 有可能发生沟道(确定因子为: Y. YY)

在异常炉况预测系统知识库中,建立了约 100 条规则；在高炉熔炼监控系统的知识库中,约有 300 条规则。

5. 传感数据的预处理

时间序列数据的模式识别以前是由人类专家来完成的,现在可在计算机上来执行,并用人工智能方法来处理。由于高炉的数据量大,且受到实时性的约束,至今还没有合适的工具来处理,所以实际上采用以下两个步骤对传感器数据进行推理前的预处理。

第一步是对传感器数据进行平滑。鼓风机有 200 多个数据。不同的数据,例如,炉子的压力和温度,往往受到扰动和由于加料所造成的非周期性变化的影响。为了去除这种影响,采用统计方法做平滑处理(线性回归过程)。炉顶气体温度如图 3.20 所示,为了对它进行预处理,采用了平滑处理过程。

在这个回归过程中,利用每分钟测量一次的 N 个数据 $T_c(t), t=1, 2, \dots, N$, 并以此拟合成以下的线性方程

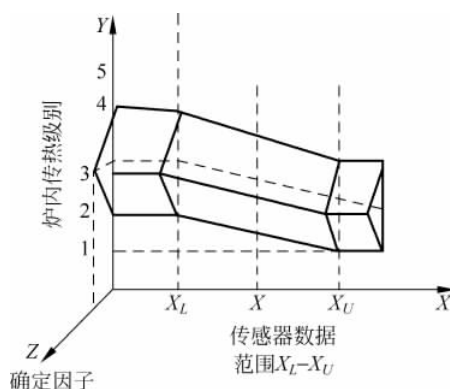


图 3.20 数据预处理第一步例子

$$f(t) = C_0 + C_1 t \quad (3.18)$$

使

$$S = \sum_{t=1}^N \{T_c(t) - f(t)\}^2 \quad (3.19)$$

为最小,由此决定系统 C_0 和 C_1 ,从 $T_c(N)$ 中计算 $f(t)$ 。这也就是最小平方法。

第二步则是利用第一步的结果,抽取传感器数据特定的交变模式(这些数据造成炉况变化)。抽取过程如下:

(1) 比较数据变化的趋向。例如,计算气体入口温度、炉子轴向温度的变化率;计算气体利用率等。

(2) 计算数据的级别。即计算顶部气体压力、吹风压力损失及炉料下降速度等数据所处的范围,看它们是否大于(或小于)极限值。

(3) 计算方差。

(4) 计算数据的积分值。

(5) 预测(拟合)典型的变化模式。

(6) 计算变化的量。

在鼓风炉运行的复杂过程中,由于操作员和专家本身所具有的经验不同,即使观察到的数据相同,判断炉况的结果也会有所差别。因此判断总带有某种程度的不确定性。在有些情况下,即使传感器的数据变化,炉况也并不一定不正常。相反,在发生不正常情况之前,传感器的数据也不一定变化。为了改善系统运行品质,就需要采用确定性因子(CF 值)和隶属函数来处理其模糊性。在高炉控制过程中完全可以用模糊集合的原理和方法来确定或选择控制的规则。

以炉内传热过程为例。炉内传热可以分为 5 级,由于级别本身具有含糊性,也有必要引入隶属函数。对不同的传感器数据,可用图 3.21 所示的三维隶属函数。通过对隶属函数的复合运算和多种输入的组合,经过推理,系统会采取最合适的动作(或规则)来对高炉进行操作。

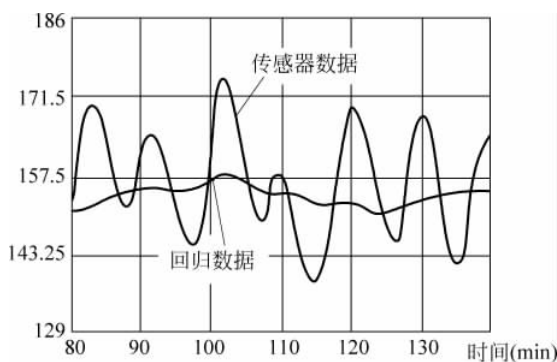


图 3.21 高炉内传热隶属函数

高炉采用智能控制以后,不但可以减轻操作工人的劳动强度,在发生故障能及时、正确进行处理;而且能够提高生产率 5%~8%。

3.6 小结

专家系统的定义、类型、结构和建造步骤等内容是本章首先研究的专家系统基本问题。接着,3.2节讨论了基于不同技术建立的专家系统,即基于规则的专家系统、基于框架的专家系统和基于模型的专家系统。从这些系统的工作原理和模型可以看出,人工智能的各种技术和方法在专家系统中得到很好的结合和应用。

3.3节阐述专家控制系统的结构与设计,包括专家控制系统的结构、控制要求、设计原则和设计时要处理的问题。这些问题涉及冲突消解策略设计和用于控制的知识获取问题。有关专家系统的各种知识,包括专家系统的结构、建造、知识表示与推理等,对专家控制器的设计也是十分有用的,值得设计专家控制器和专家控制系统时参考。根据系统的复杂性,可把专家控制系统分为两类,即专家控制器和专家控制系统。按照系统的控制机理,又可将专家控制系统分为直接专家系统和间接专家控制系统两种。前者由控制器直接向受控过程提供控制信号;而后者由控制器间接对受控制过程发生作用。

3.4节探讨专家控制系统的建模问题。首先借鉴传统控制的理论和方法来讨论受控对象(装置、过程)的数学模型,然后建立专家控制器的模型(包括规则库和推理机的动力学模型),最后讨论专家控制系统的建模。尽管这些设计思想和方法并不规范和成熟,但它们能为读者提供一种发现问题、提出问题和解决问题的思路,仍不失为一种有意义的工作。目前对专家控制的深层研究仍然较少,有些专著和论文结合具体应用对专家控制器和专家控制系统的模型有更详细和更深入的分析与探讨;感兴趣的读者可参阅相关文献。

3.5节举例说明了专家控制系统应用,即用于控制炼铁高炉温度的实时监控专家系统。通过这个应用例子,读者对专家控制系统的结构、设计方法与实现会有更多和更好的了解。仿真和应用结果已经表明,专家控制系统(控制器)具有优良的性能,并具有广泛的应用领域。

习题 3

- 3-1 怎样定义专家系统? 你是如何理解专家系统的?
- 3-2 试就专家系统的构成和各部分的作用说明专家系统是如何工作的。
- 3-3 结合结构说明基于规则的专家系统的工作原理。
- 3-4 基于框架的专家系统与面向目标编程有何关系? 其结构特点和设计任务是什么?
- 3-5 为什么要提出基于模型的专家系统? 试述神经网络专家系统的一般结构。
- 3-6 在设计专家系统时,应考虑哪些技术?
- 3-7 什么是专家控制的理论基础?
- 3-8 什么是专家控制和专家控制系统?
- 3-9 专家控制系统的控制要求和设计时应遵循的原则有哪些?
- 3-10 专家控制系统设计涉及哪些问题?

- 3-11 试从专家控制系统的一般结构和各部分作用,举例说明专家控制系统的工作原理。
- 3-12 专家控制系统有哪几种类型?它们有何区别?
- 3-13 建造专家控制系统的关键步骤是哪些?
- 3-14 为什么要研究专家控制系统的建模?专家控制系统的建模包括哪些问题?
- 3-15 举例说明实时专家控制系统的工作原理及其实现。
- 3-16 专家控制系统的应用和发展前景如何?