

## 摘 要

制动器是保证机构或机器安全正常工作的重要部件，应用范围非常广泛。其中电力液压推杆制动器由于使用维护简单可靠、节能、无噪音、使用寿命长等优点，在运输、冶金、矿山、港口码头等机械得到广泛的应用。但是因为电力液压推杆制动器的工作原理是弹簧上闸，推杆松闸，所以弹簧长度一旦确定，制动力矩随之确定，不能根据不同的负载或工况改变。而随着社会快速发展，制动器实际使用单位希望获得更加稳定、可靠的制动效果，并且制动力矩可调，即实现制动器的智能控制。

本文就是在电力液压推杆制动器的基础上，以龙门起重机大车运行机构为对象，研究了模糊控制这种智能控制技术在制动领域中的应用。论文首先介绍了相关课题的研究现状、智能控制技术和模糊控制技术。然后针对电力液压推杆制动器设计了模糊控制与变频调速技术相结合的控制方案，建立了模糊制动控制器的开环控制系统，研究其模糊控制算法和决策准则。然后将模糊控制与传统 PID 技术相结合，建立了模糊 PID 闭环制动控制系统，并与传统 PID 控制技术相比较。最后进行系统仿真和结果分析，得出模糊控制能有效地改善电力液压推杆制动器的制动性能的结论。

关键词：电力液压推杆制动器

模糊控制

模糊 PID 控制

## ABSTRACT

The arrester is the important part to guarantee the machine work safely, and it is extensively applied in many aspects of industry. One of them is the electric hydraulic push rod brake which is widely used in many fields, such as transporting , metallurgy , mine and port quay ,because it maintains such advantages as reliable , energy-conservation , no noise , having long performance time ,etc. But the operation principle of the electric hydraulic push rod brake is that spring push the brake and the push rod loses the brake, once the length of the spring is fixed, the moment is also fixed and can not be changed according to the different load and operating modes. With the rapid development of society, the real applying unit of the brake hopes to get more steady and more reliable result when applying the brake, and the moment can be adjusted, that is to say that brake can be controlled intelligently.

Based on the electric hydraulic push rod brake and regarded the cart operation organization of the gantry crane as the target, this paper studies the application of the fuzzy control in the brake field. At first, the thesis introduces the current research situation, intellectual control technology and fuzzy control technology of relevant subjects. Then we design the control scheme which combining fuzzy control and speed-frequency technology according to the electric hydraulic push brake, and set up the open loop system of fuzzy control, and study the control algorithms and decision criterion. Then we combine fuzzy control and traditional PID control technology and set up the closed loop system of fuzzy PID control, and compare fuzzy PID control and traditional PID control technology. Finally we carry on the system emulation and results analysis, and draw the conclusion that fuzzy control can improve the performance of the electric hydraulic push rod brake effectively.

Key work: the electric hydraulic push rod brake, Fuzzy control, Fuzzy PID control

## 独 创 声 明

本人所呈交的学位论文是本人在导师的指导下进行的研究工作及取得的研究成果。据我所知，论文中除特别加以标注和致谢的地方外，不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京工商大学或其他教育机构的学位证书所使用过的材料。其他人对本研究成果所做的贡献均已在论文中做了明确的说明，并表示谢意。

特此声明

作者签名：\_\_\_\_\_

日期      年    月    日

# 第一章 绪 论

## 1.1 课题的提出

制动器具有减速、停止和支持重物等功能，是保证机构或机器安全正常工作的重要部件，应用范围相当广泛，大至起重运输、建筑、矿山、石油、港口等机械设备，小至全自动洗衣机等家用电器，具有较大的市场需求量。

随着社会的进步和现代工业的高速发展，对生产制造过程和设备的自动化程度、工作平稳性、安全可靠性等诸多方面都提出了更高的要求，尤其在装卸易燃、易爆、易碎物品和停放位置要求准确的起重运输机械上。传统机械式的常闭式制动器一般靠松闸装置松闸，靠上闸弹簧力产生制动力，一旦弹簧长度调节好，弹簧力大小和制动力矩便确定了，设备工作时如载荷小、速度低会引起制动过猛，造成冲击；载荷大、速度高时又会造成制动时间过长，因此设备或物品不易达到准确的停放位置。常开式制动器可以弥补常闭式制动器的不足，但要完全依赖于人的操作，司机的技术熟练程度、经验、身体状况、疲劳程度直接影响制动结果，存在着随机性和潜在危险。因此上述两种形式的制动器的性能显然已难以适应现代工业生产制造过程的要求，制动器的使用单位迫切希望能够研制一种能按预定规律控制制动力矩变化或达到自适应载荷、速度变化的制动过程、制动平稳性更高、使停车位置更准的“智能型”制动技术，来取代传统机械式制动器工作。

目前工业上常用的制动器按松闸装置的类型分有电磁铁、液压电磁、电力液压推杆制动器等几种。国内外使用最多的、技术最成熟的则是电力液压推杆制动器，和其它几种制动器相比，它具有许多优点：①制动平稳。由于液压推杆制动器的工作特性，其制动力是逐渐释放的，故制动平稳、振动较小。②噪音小。因为其工作原理是通过电机带动叶轮旋转，使缸内液体形成压力推动活塞向上运动，所以噪音小。推动器工作时的噪声一般在 75 分贝以下，而电磁铁则达到 85 分贝以上。③寿命长，经济性好。电力液压推杆采用机、电、液一体全封闭结构，工作油路循环与无压的封闭钢筒里，体积小，便于安装、维修。在恶劣的工作环境下，不吸尘、不进水，内部不锈蚀。其使用寿命约为交流电磁铁的 3~5 倍。④推动器的允许操作频率远高于电磁铁。⑤省电节能，仅以正常工作电流比较，MYT3-40 的耗电仅为 MZD1-200 的 60%，YT3-70 的耗电是 MZD1-300 的 41.7%。此外，此种推动器还有一大优点，就是推动器电机的负载是被它驱动的离心泵叶

轮,这种负载匹配是在设计时予以确定的,所以电机不会因为推动器的外负载增大而过载,即电机总不会过载。

而智能控制是从解决工程和技术问题的实践中产生和发展起来的。随着自动化程度的提高和普及,受控对象的日趋复杂,对于许多难以获得数学模型或默许复杂的过程,应用经典和现代控制理论往往不能取得令人满意的控制效果。计算机技术的发展为智能技术的实现提供了可能,如今人工智能、模糊控制、神经网络等的研究都取得了可喜的进展。

综合以上内容,针对使用广泛的电力液压推杆制动器开展智能制动技术和智能制动系统的研究,具有一定的现实意义,研究结果对提高现有制动器的制动性能,提高设备运行的自动化、智能化程度有着积极的促进、指导作用。

## 1.2 课题相关研究的历史、现状及主要成果

制动装置是保证机械设备安全、正常工作的重要部件。制动器的分类按工作状态可分为常闭式/常开式;按结构状态可分为块(闸瓦)式/带式/盘式;按制动驱动方式可分为电磁铁式/液压推动式/液压电磁式/惯性式等。

### 1.2.1 国内制动器的现状

我国制动器标准的制定原则是:70年代参照苏联,80年代参照德国,90年代等效采用先进国家的标准.但由于苏联与德国标准体系不同,造成我国标准和产品种类繁多,修订频繁.由于德国制动器在世界处于领先地位,其起重运输机械出口占主导地位,再加上其制动器标准体系比较完善,所以我国现行的制动器标准基本参照德国 DIN 标准。

国内目前实际生产制动器的企业大约有 200 家,多数属于乡镇企业或私营企业,开发能力差,设备比较陈旧,大多生产 YWZ\TJ2\YT1 等老产品.但随着起重机和运输机等主机生产许可证的发放,制动器的重要性正逐渐被人们认识,1998年被列为国家安全认证产品。常用的制动器有液压推杆制动器、电磁块式制动器、盘式制动器等。

#### (1) 液压推杆式制动器

根据推动器的结构形式(单杆或双杆)和制动器的构造(直臂或弯臂,单臂或双臂)组合成不同系列的制动器,见表 1.1。

| 序号 | 代号   | 结构特征        | 符合标准         | 规格范围(mm)                 | 制动力矩(Nm) |
|----|------|-------------|--------------|--------------------------|----------|
| 1  | YWZ  | 直臂, 双推杆, 横簧 | JB/ZQ4386-86 | $\Phi 100 \sim \Phi 300$ | 40~12500 |
| 2  | YWZ2 | 弯臂, 单推杆, 横簧 | GB6333-86    | $\Phi 160 \sim \Phi 800$ | 80~12500 |
| 3  | YWZ3 | 弯臂, 双推杆, 横簧 | GB6333-86    | $\Phi 160 \sim \Phi 800$ | 80~12500 |
| 4  | YWZ4 | 直臂, 单推杆, 横簧 | JB/ZQ4386-86 | $\Phi 150 \sim \Phi 700$ | 100~8000 |
| 5  | YWZ5 | 弯臂, 单推杆, 竖簧 | GB6333-86    | $\Phi 200 \sim \Phi 710$ | 112~8000 |
| 6  | YW   | 弯臂, 单推杆, 竖簧 | JB/T6406-92  | $\Phi 160 \sim \Phi 800$ | 63~8000  |
| 7  | YWP  | 弯臂, 单推杆, 横簧 | JB/T6406-92  | $\Phi 160 \sim \Phi 800$ | 63~8000  |

表 1.1 液压推杆式制动器系列

电力液压推杆制动器是由电力液压推杆作为推动器的制动器。我国推动器的生产和应用是从双推杆开始的, 型号为“MYT1”, 以其为驱动元件的是 YWZ 系列制动器。80 年代以来, 研制了换代产品 MYT2、MYT3、MYT4 等系列产品, 还从德国引进了 Ed 系列产品。与此同时我国各制动器生产厂家也先后推出了 YT2、YT3、YT4、YT5、YT6 等各型单推杆系列产品。1996 年, 我国在 Ed 等单推杆系列产品基础上, 又设计出结构、性能更好的 Yd 单推杆系列产品。这里对各种推杆制动器的结构和性能作一比较:

YT1 型双推杆电力液压推动器(如图 1.1)的基本结构是电动机置于液压缸上方, 两根推杆对称地固定在支持件两边, 杆体穿过缸盖从电动机两侧伸出, 顶部用中间装有连接快的横梁连接。共用一个后盖板的上下两个半开式叶轮, 分别由支持件的中部通孔和活塞上的 4 个暗孔吸入油液, 从活塞与支持构成的油液导向室中活塞的 4 个暗孔壁外的孔隙间压出。



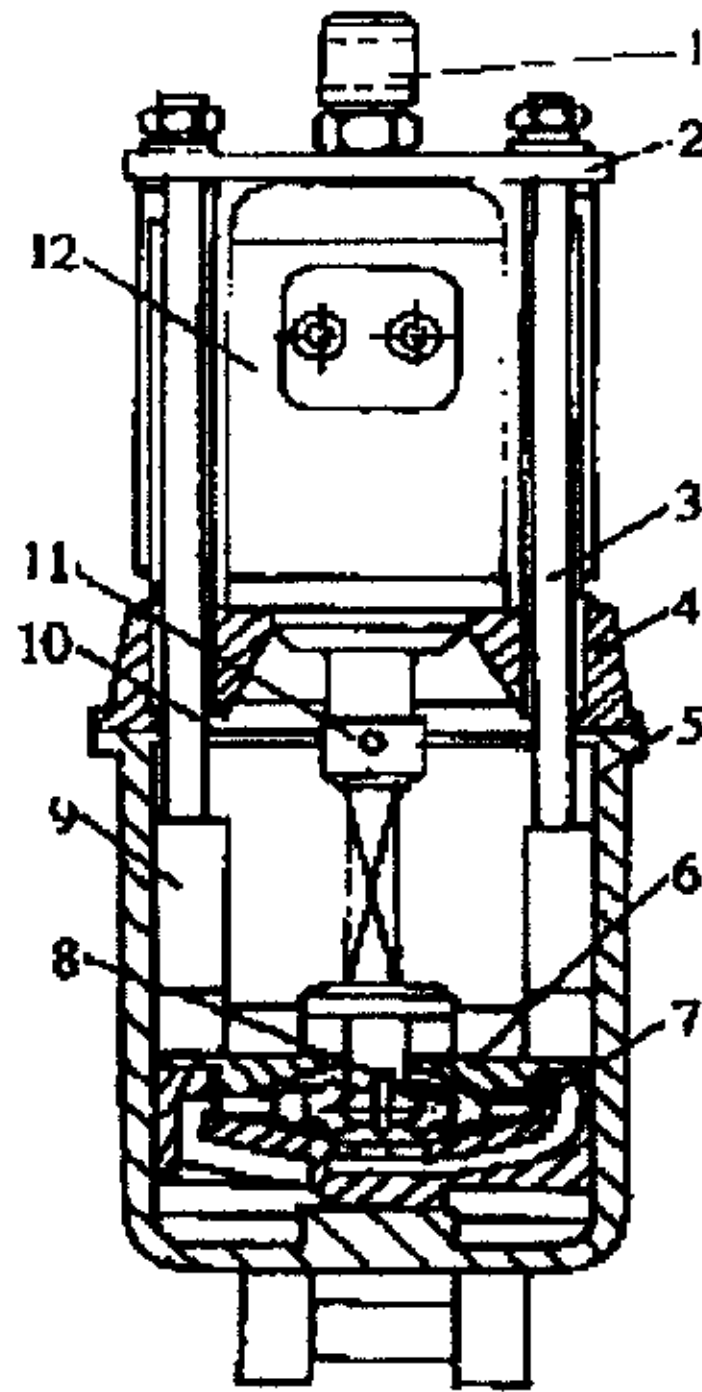


图 1.1 YT1 型双推杆电力液压推动器

这样的结构使 YT1 型双推杆电力液压推动器不可避免存在下列缺陷：油液压出流道有近半被活塞的 4 个暗孔外壁阻塞，导致能量容积损失增大，机械效率低；整个结构复杂，特别是双吸式的叶轮结构使活塞结构复杂化。

Ed 型单推杆电力液压推动器（如图 1.2）的基本结构是电动机位于液压缸下部，叶轮在电动机的轴伸处悬臂安装，液压缸为有套缸的双层结构，活塞或

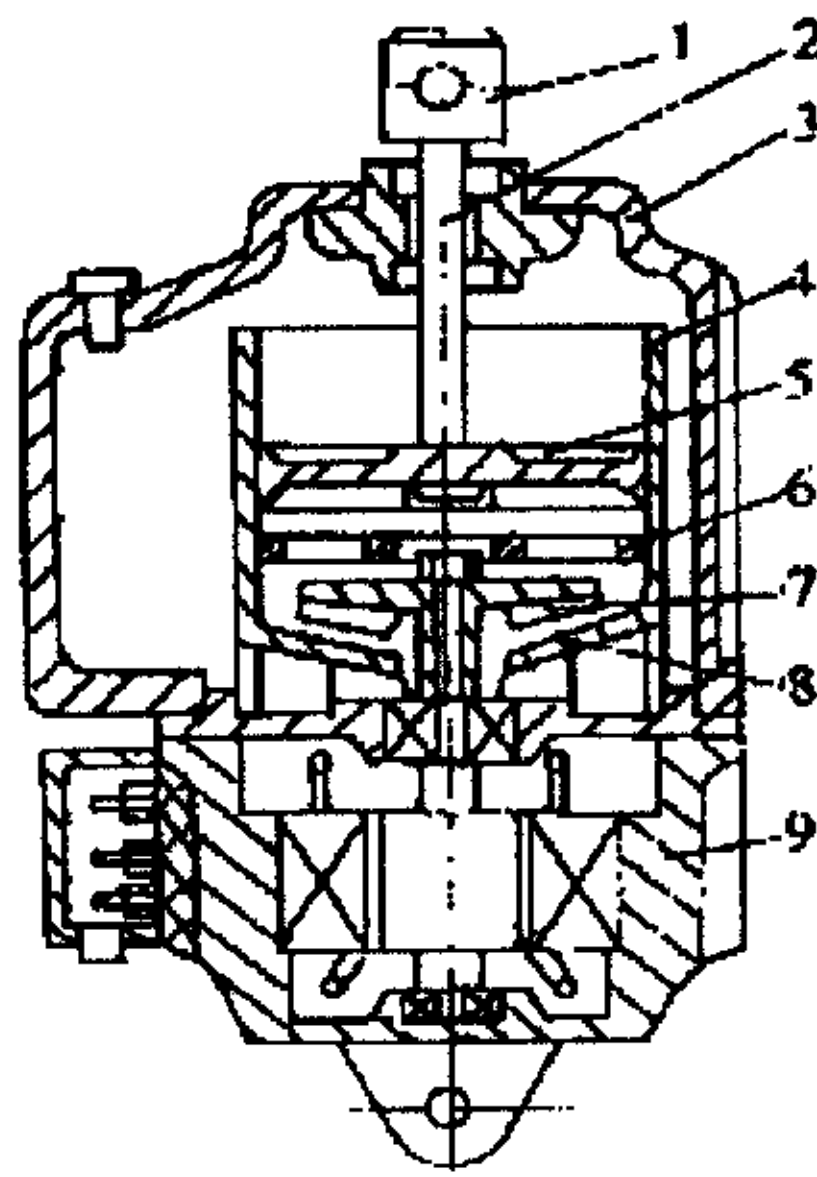


图 1.2 Ed 型单推杆电力液压推动器

部，叶轮在电动机的轴伸处悬臂安装，液压缸为有套缸的双层结构，活塞或

为圆形装在套缸内，或为环形套在套缸外。高速旋转的叶轮和油液导向盘可使油液在套缸内外往复流动。这样的结构使 Ed 型单推杆电力液压推动器存在以下缺陷：带套缸的双层缸体结构比较复杂；由于活塞外径小，单位面积承压大，油液内压高，为防止渗漏，不得不选用昂贵的铝合金和工艺难度大的压力铸造成型。Ed 型单推杆电力液压推动器虽然存在上述缺欠，但与 YT1 型双推杆产品相比，其推杆压力的稳定性、每小时可操作次数及其使用寿命都要好些，不过成本与售价也相应的要比 YT1 型产品高。

Yd 型单推杆电力液压推动器（见图 1.3）是针对我国现产的 YT1 型推杆及 Ed 型等各单推杆电力液压推动器所存在的上述缺欠而研制的专利新产品。其基

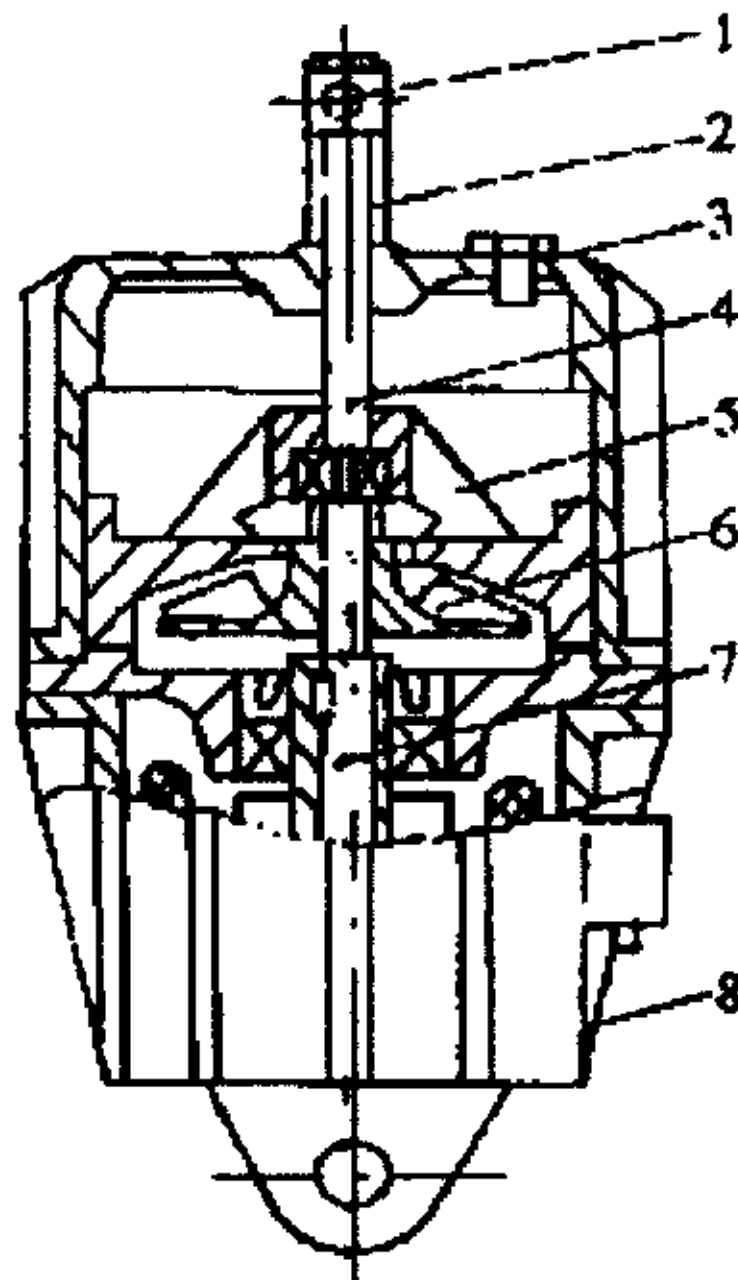


图 1.3 Yd 型单推杆电力液压推动器

本结构由电动机、传动轴、叶轮、活塞、推杆、连接块等六大件构成。工作原理是：通电时，伸进电机轴盲孔中的传动轴及固定在传动轴上的叶轮随电机以高速旋转，将油缸内活塞上部的油液吸到活塞与叶轮下部形成压差，迫使活塞及固定在活塞上的推杆随传动轴和叶轮以  $10\text{mm}/0.1\text{s}$  左右的速度一同上升，举起负荷；断电后，推杆在与之固连的各零件的自重及负荷压力下复位。叶轮与活塞下部的油液则通过叶轮径向叶片间的流道被重新压到活塞上部。该产品与 YT1 型双推杆及 Ed 型单推杆电力液压推动器相比优越性如下：结构精巧简单，只有一根长径比很小的推杆传动推力非常平稳，规格全、相近规格零件通用率高。



(2) 电磁块式制动器

电磁铁按电源种类分为交流型和直流型，按构造分为长行程（螺管式）和短行程（拍合式）如表 1.2。早期我国生产的电磁块式制动器主要是仿制苏联的产品，如 JWZ、JCZ 等。近年来，在原来的直流短行程电磁铁基础上，研制了多种新型电磁铁。如电磁铁采用直流构造，外接交流电源通过整流电路转变为直流控制电磁线圈，并采用强励磁启动、弱励磁维持的原理，达到节能节材的目的。但由于设计上的原因和整流元件可靠性等因素，这种电磁铁有待于进一步研究。

| 序号 | 代号   | 结构特征          | 符合标准         | 规格范围(mm)    | 制动力矩(Nm)  |
|----|------|---------------|--------------|-------------|-----------|
| 1  | Tj2  | 交流,直臂,电磁铁在上部  |              | Φ 100~Φ 300 | 10~500    |
| 2  | TZ2  | 直流,直臂,电磁铁在上部  |              | Φ 100~Φ 300 | 10~500    |
| 3  | TJ2A | 交流,直臂,电磁铁在上部  |              | Φ 100~Φ 300 | 10~500    |
| 4  | ZWZ  | 直流,直臂,电磁铁在侧面  | JB/ZQ4386-86 | Φ 400~Φ 800 | 550~12500 |
| 5  | ZWZ3 | 直流,弯臂,电磁铁在侧面  | GB6334-86    | Φ 160~Φ 800 | 23~10000  |
| 6  | MW   | 交直流,弯臂,电磁铁在侧面 | JB/T7685-95  | Φ 160~Φ 800 | 80~10000  |
| 7  | JCZ  | 交流,直臂,长行程     | JB/ZQ4387-86 | Φ 200~Φ 600 | 200~5000  |
| 8  | MWT  | 直流,弯臂,电磁铁在侧面  | 美国日本标准       | Φ 112~Φ 762 | 80~13000  |

表 1.2 电磁块式制动器系列

近年来，北起所开发了替代液压推动器的长行程电磁铁（电磁推动器 MT 系列）和替代欧美国家英制单位电磁块式制动器 MWT 系列。后者已有好几种规格的产品成功应用于宝钢集团梅山钢厂。

(3) 盘式制动器

起重运输机械用盘式制动器发展较慢，1987 年太原重机学院与焦作制动器厂研制成功利用 Ed 推动器的制动臂盘式制动器，并于 1992 年发布了部标 JB/T7019 制动盘，JB/T7020 制动臂盘式制动器，但由于该结构盘式制动器沿制动盘轴向尺寸较大，而使产品的推广应用受到一定的限制，目前只有港口起重机上得以使用。1995 年，焦作制动器厂开发了气动钳盘式制动器，并在包钢应用成功。1997 年，太原重机学院研制的脚踏常开液压盘式制动器通过部级鉴定，并应用于秦皇岛门座起重机回转机构上，效果良好。同年焦作制动器厂与法国公

司组建合资的瑞塞尔盘式制动器有限公司，生产气动盘式制动器、液压盘式制动器、电磁盘式制动器等产品，已在武钢、邯郸钢厂应用了数十套。上海东屋电气公司开发成功了 DPB 直流电磁盘式制动器，通过控制箱可直接接入 380 交流电路中。产品如表 1.3:

| 序号 | 代号     | 结构特征           | 符合标准        | 规格范围(mm)     | 制动力矩(Nm)   |
|----|--------|----------------|-------------|--------------|------------|
| 1  | YPB I  | 液压推杆, 臂式钳盘     | JB/T7020-93 | Φ 315~Φ 1000 | 450~1000   |
| 2  | YPB II | 液压推杆, 楔块, 臂式钳盘 | JB/T7020-93 | Φ 315~Φ 800  | 355~8000   |
| 3  | CYPZ   | 脚踏常开式, 液压, 钳盘  | JB/T7020-93 | Φ 315~Φ 630  | 300~1600   |
| 4  | GP12.7 | 气动, 钳盘 (常闭式)   | JB/T7020-93 | Φ 250~Φ 610  | 100~2000   |
| 5  | DPB    | 电磁, 钳盘 (常闭式)   | JB/T7020-93 | Φ 400~Φ 600  | 2989~12500 |

表 1.3 盘式制动器系列

## 1.2.2 国外制动器的现状

在工业发达国家中，德国、法国等国家的制动器基本代表当今世界的制动器发展水平。法国各制动器企业主要生产盘式制动器，我国的邯钢、宝钢、水口升船机等许多重点工程上都有应用，与德国 Bubenzer 公司的制动臂盘式制动器各具特色。只是前者使用电磁铁和液压作动力，后者使用推动器。

### (1) 德国 EMG 公司

德国专业生产电力液压推动器的企业，其生产的 Ed 推动器在相当长一段时期代表着国际先进水平。我国许多企业在参考 Ed 推动器的基础上，开发了单杆推动器。EMG 东德推动器公司生产的 EB 系列推动器比 Ed 系列具有更低的驱动电机功率，更快的速度，更轻的自重，更合理的结构，成为推动器发展的新趋势。

### (2) 德国 Bubenzer 公司

德国专业生产制动器的公司，主要生产制动臂盘式制动器、电磁盘式制动器、液压推杆块式制动器及制动器监控装置等产品。代表产品如下：

SB 系列制动臂盘式制动器：动力源为液压推动器，国内开发的 YPB1 型盘式制动器参照了该公司早期的 SB14 系列。其新一代产品代号为 SB23/SB28 系列，其结构呈全对称布置，外形安装尺寸也有所减小，增加了退距均等装置，磨损补偿性能有了大幅度提高，广泛应用于世界各大港口的起重机上。

BSC/SF 液压盘式制动器:自带液压泵站,BSC 系列 3 种规格的夹持力为 7~50KN,SF 系列 3 种规格的夹持力为 50~400KN。

EB 系列液压推杆块式制动器:符合 DIN 标准,动力源为 EMG 公司的液压推动器.EBN 系列与我国 YWZ5 系列相同.EBH 系列将推动器和弹簧水平布置在制动轮上方,安装空间尺寸小。

EBN-2St 两级制动的制动器:采用双制动弹簧,弹簧 1 正常上闸,弹簧 2 上装有阻尼液压缸,在延时 1~7 秒之后上闸,延时时间可调节,制动力矩等于两个弹簧产生的力矩总和,其中弹簧 1 产生总力矩的 20%~50%,弹簧 2 产生 30%~70%。

### (3) Mannesmann Demag 公司

该公司生产符合 DIN 标准的 DOBA 系列液压块式制动器,配套使用 AEG 推动器。DOBA 系列除了基本型以外,还有卧式结构、液压或气动双动力源、两级制动及常开液压制动型等多种结构。此外,该公司还生产 SSB 液压浮钳盘式制动器。

### (4) 法国 SIME 公司

SIME 公司是世界上最大的重工业盘式制动器生产企业之一,主要生产电磁铁和液压盘式制动器及块式制动器。其产品应用于我国包括鞍钢、宝钢、秦山核电站等许多主机上。

电磁钳盘式制动器:采用碟形弹簧上闸,电磁松闸和强励启动、弱励维持的原理,带有磨损自动补偿和磨损限位报警功能。S、C、600 等三大系列制动器,适用盘径  $\Phi 175\sim\Phi 795$ ,单钳制动力矩 55~4000Nm,每盘双钳制动力矩为 10~8000Nm。OSA、OOSA、2SA 等三大系列安全制动器单钳夹持力 26.5~100KN,可适应任意盘径,用于低速轴的安全制动。

电动液压钳盘式制动器:T 系列电动液压钳盘式制动器采用碟形弹簧上闸,电动液压松闸的原理,带有磨损自动补偿和磨损限位报警功能,适用盘径  $\Phi 445\sim\Phi 995$ ,单钳制动力矩 775~14600Nm。

液压钳盘式制动器:TH、SH 和 BCH 系列液压钳盘式制动器采用碟形弹簧上闸,液力松闸的原理,与相应的液压站配套,单钳夹持力 4~180KN,可适应任意盘径。

块式制动器,SIME 公司生产各种结构形式的块式制动器,包括电动液压推杆制

动器和电磁制动器，符合法国钢铁工程师协会 SSF 标准的规格为  $\Phi 100\sim\Phi 750$ ，制动力矩 11~5400Nm。符合德国 DIN 标准的规格为  $\Phi 160\sim\Phi 710$ ，制动力矩 21~5520Nm。

#### (5) 法国 ATV 公司

ATV 公司主要生产电磁铁和液压钳盘式制动器，多数产品与 SIME 公司相似，1997 年与焦作制动器集团组建合资的瑞塞尔盘式制动器有限公司，生产和组装各种电磁和液压钳盘式制动器。

### 1.2.3 智能控制技术的分类及发展概况

控制理论的形成和发展经历了经典控制理论、现代控制理论和智能控制力量三个阶段。其中，经典控制理论和现代控制理论是建立在精确的数学模型基础之上的，而智能控制理论适合用来解决系统模型和环境本身均不确定的问题。1987 年智能控制正式成为一门独立的学科，它是人工智能和自动控制理论等多学科相结合的交叉学科。

目前国内外最新智能控制方法主要有：神经网络控制、模糊控制、自适应控制、集成智能控制等。智能控制系统的基本结构由六部分组成，即执行器、感知处理器、环境模型、判值部件、传感器和行为发生器，如图 1.4 所示：

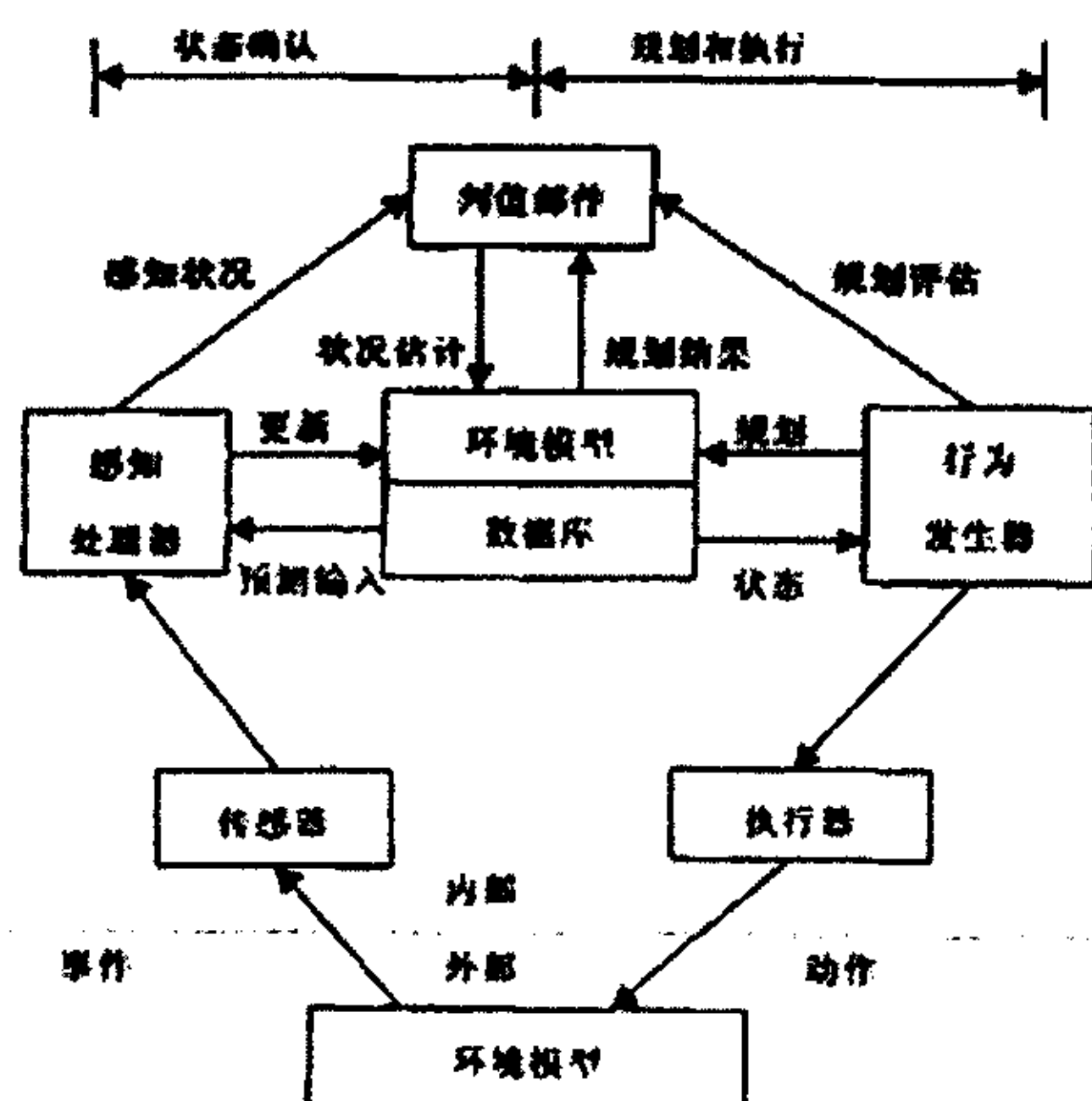


图 1.4 智能控制系统的基本结构

#### ●神经网络控制

1943年，心理学家 McCulloch 和数学家 Pitts 合作提出形式神经元的数学模型 MP 模型，从而开创了神经科学理论研究的时代。而网络系统的研究则是在 50 年代和 60 年代初开始的。到 80 年代，Rumelhart 和 McClelland 等人发展了多层网络的 BP 算法（反向传播学习算法）。神经网络是由大量的处理单元广泛互连而成的网络，它是一个具有高度非线性的超大规模连续时间动力系统，它具有大规模并行分布处理能力、学习联想能力、数据泛化以及多输入多输出的能力。

●模糊控制

自从 1965 年 Zadeh 教授首先提出模糊集理论并且 1974 年 Mamdani 成功地将模糊理论应用于锅炉和汽轮机的过程控制以来，模糊控制迅速发展并得到广泛应用。模糊控制是基于模糊集、模糊逻辑，同控制理论相结合，模拟人的思维方式，对难建模的对象实施的一种控制方法。其主要优点在于：(1)不需知道被控对象精确的数学模型；(2)鲁棒性强，适于解决过程控制中的非线性、强耦合、时变、滞后等问题；(3)为“语言型”控制，易于形成知识库。

●自适应控制

自适应控制系统是 50 年代提出来的，是一种特殊形式的非线性控制系统，其在运行中能自动的获取改善系统品质的有关信息并能修正控制系统的结构或参数，使系统达到所要求的状态。它包括基本的调节控制反馈回路、系统的准则给定、实时在线辨识和实时修正调整机构等四部分。系统框图如图 1.5：

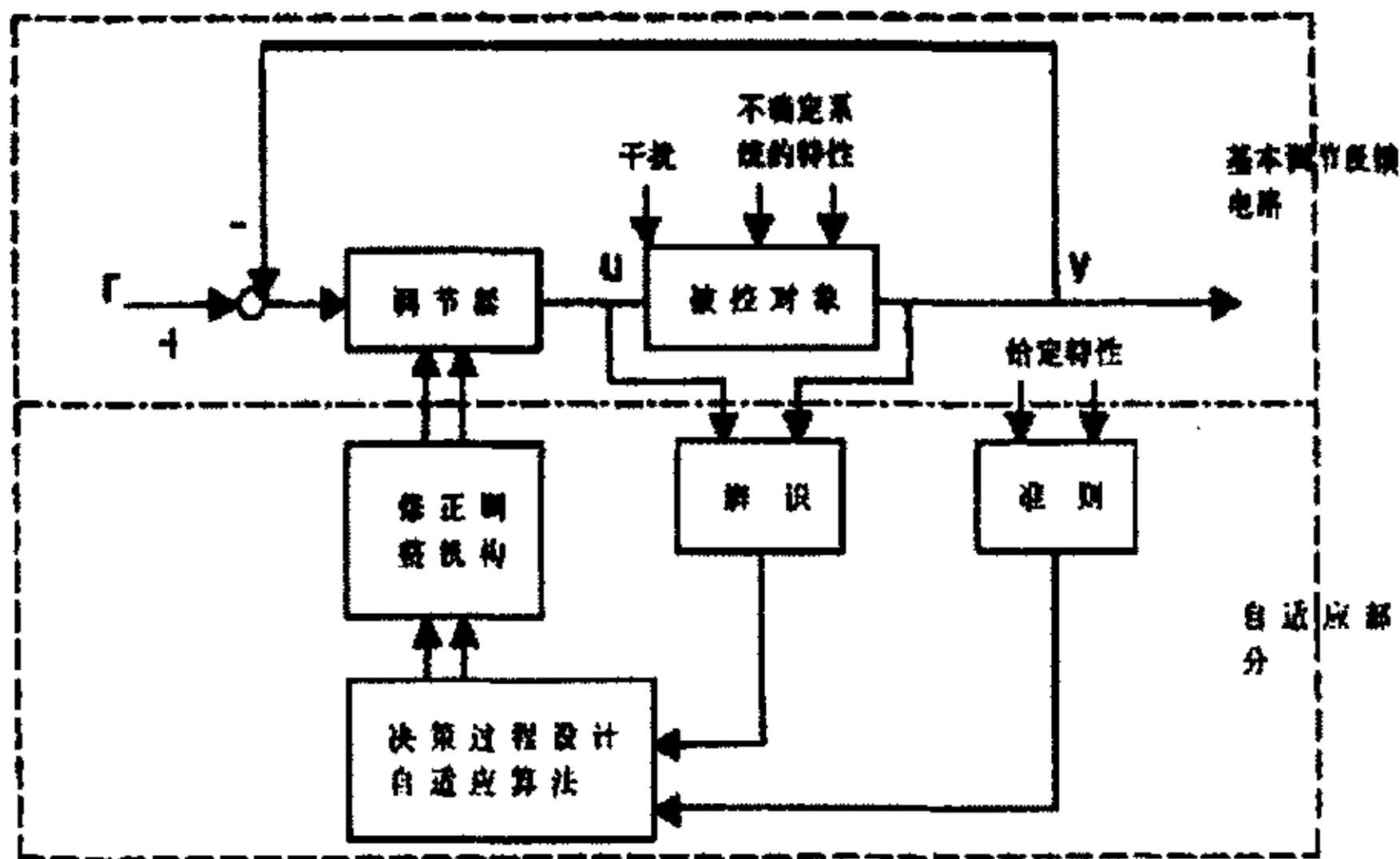


图 1.5 自适应控制系统典型结构

●集成智能控制



集成智能控制是将几种智能控制方法或机理融合在一起而构成的智能控制方法。近年来形成了模糊神经网络控制、专家模糊控制等多个发展方向。

### 1.3 本课题研究的主要内容及预期达到的效果

制动器有诸多类型，经过分析比较，本课题选用目前应用广泛、具有代表性的电力液压推杆制动器作为研究对象。

电力液压推杆制动器的核心部件是驱动装置电力液压推杆，其性能的好坏很大程度上决定了制动器的性能。这种制动器是常闭式制动器，依靠推杆的推力松闸，利用弹簧力产生制动力，一旦弹簧长度调节好，弹簧力大小和制动力矩便确定了，设备工作时如载荷小、速度低会引起制动过猛，造成冲击；载荷大、速度高时又会造成制动时间过长，因此设备或物品不易达到准确的停放位置。为了解决上述问题，使制动器达到更好的制动效果，本文选取智能控制中的模糊逻辑对制动器进行控制。主要研究内容包括：

1. 研究电力液压推杆制动器的制动控制方案、系统组成结构；
2. 结合制动器的特性和载荷工况，建立电力液压推杆制动器智能控制器的模型，研究模糊控制算法、控制参数和决策准则。
3. 在电力液压推杆制动器的基础上，进行不同负载和不同转速的模糊控制器开环系统构建，并进行仿真；
4. 在电力液压推杆制动器的基础上，将模糊控制与 PID 相结合，构建制动器的闭环控制系统，并与传统 PID 控制进行比较，同时进行仿真。
5. 分析研究结果，并为今后的制动器控制系统提出建议方案。



## 第二章 制动控制系统

为了让研究具有普遍性，本课题选取具有代表意义，应用较广泛的电力液压推杆制动器作为分析对象。它的工作原理是：通电时，叶轮随电机以高速旋转，将油缸内活塞上部的油液吸到活塞与叶轮下部形成压差，迫使活塞及固定在活塞上的推杆随传动轴和叶轮一起上升，释放弹簧压力使制动力矩减小；断电后，推杆在与之固连的各零件的自重及弹簧压力下复位，通过弹簧压住制动瓦进行制动。

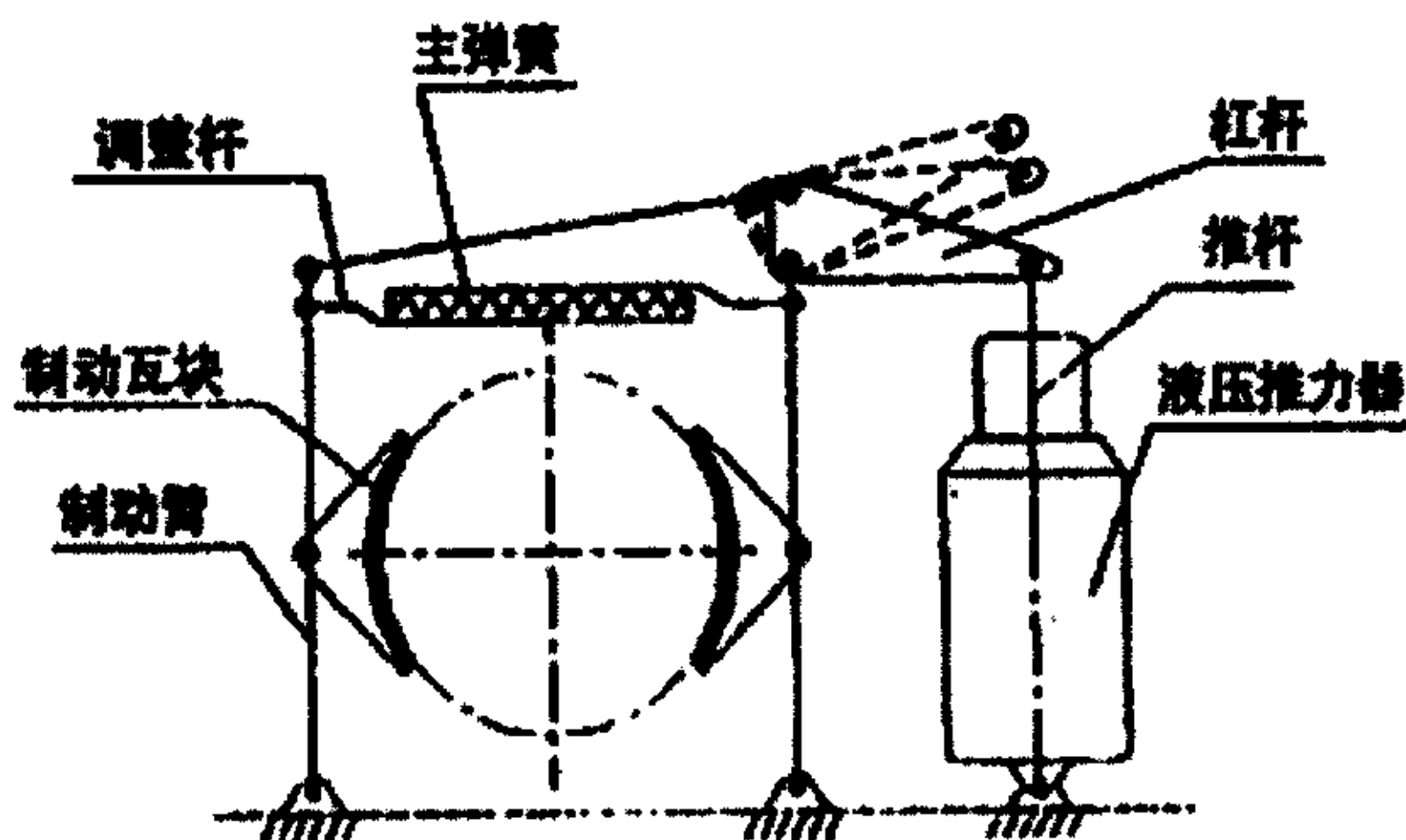


图 2.1 电力液压推杆制动器结构简图

本控制系统希望对制动器推杆推力进行控制，由上图可以看出电力液压推杆制动器是由弹簧产生制动力，当制动瓦块与制动轮接触时，改变推杆推力就可以改变制动力矩，这是因为此时制动力是由弹簧拉力与推杆推力之差所产生，推力大则制动力矩小；推力小则制动力矩大，由此我们可以设计出电力液压推杆制动器的控制系统。根据制动器的工作原理我们知道推杆是由电机带动的，电机转速改变则推力相应发生变化，所以我们只需要对制动器电机转速进行控制。本课题采用变频调速的方法来实现：通过调节变频器的输出频率控制制动器电机转速，转速改变推力就会相应改变，则弹簧压力与推力所产生的制动力矩随之变化，从而达到制动力矩可调。

### 2.1 传统制动控制技术

制动器制动包括机械制动、反接制动、能耗制动、液压调压制动等。其中反接制动的优点是没有抱闸机构，制动转矩大且迅速，实现制动比较容易；缺点是制

动时冲击大,对机构的传动部件损坏大,且制动电流是电机额定电流的 3~5 倍,对定子绕组、接触器主触头和配电线路的危害很大,增加维护量并严重缩短电器设备使用寿命,频繁制动对位时能量损耗也相当大。能耗制动的优点是制动转矩平滑,能随时改变制动转矩,可以使生产机械可靠停止,最适合用于经常起动、频繁逆转并要求迅速停车的生产机械;缺点是能量不能回馈电网,还需增加一直流电源。液压调压制动的优点是制动力矩正比于调定压力,与转速无关,故可将转速制动到零,无需设机械闸,控制容易,系统简单,虽然油泵工作寿命不如液力制动器,但由于油泵经常空载运行,油在液压系统内循环流动,散热良好,因而使用寿命也较长。其缺点是制动力为恒定值,而被制动的机械设备的负载是随机变化的,致使减速度为变量。液压调速制动系统是通过调整与泵串联的调速阀开度来控制被制动机械设备的减速度,易于实现加减速度的控制。恒减速制动是当前性能最优的液压制动系统,它极大的改善了安全制动时对机械设备的冲击。以中信重型机械公司为代表的提升机液压制动系统基本反映了国内情况。1980 年后,以二级制动性能的液压制动系统为代表,定型的产品有:TE002/TE003(液压延时型);130/TE131(电器延时型);90 年代中期国内第一台恒减速液压制动系统投入运行,经过改进定型为 E128。国外以 SIEMENS、ABB 公司为代表的产品都是基于恒减速控制的思想。SIEMENS 公司的代表产品叫做可控安全制动或最佳安全制动(Controlled Safety Brake 或 Optimum Safety Brake)。ABB 公司的代表产品叫做可控制动力矩或可控减速度(Controlled braking torque 或 Controlled deceleration)。国外产品的最显著优点是高性能、高稳定、高可靠和智能化。

本课题以电力液压推杆制动器为研究对象,主要是通过控制电机转速来改变推力,进而改变制动力矩,就此而言上述各种制动技术并不适合,我们发现变频器调速较简单、方便,适合用来控制电力液压推杆制动器的电机转速。因此作者将变频调速与智能控制技术结合起来,实现制动器制动系统的智能化。

## 2.2 智能控制与模糊控制

智能控制技术有许多种,在 1.2.3 节中我们已经简单介绍过目前较新或流行的几种。本课题最终选用了模糊智能控制,这是因为模糊控制与传统的控制方法相比有以下十分明显的特点:

(1) 模糊控制无须知道被控系统的数学模型。精确理论一般需要用数学微分方程

来描述自然科学的某些基本规律或系统，但在实际中有的系统所涉及的因素很多，而每个因素之间还存在着耦合的关系；同时，系统所处的条件也千变万化，要用微分方程来描述这种系统，要么根本无法实现，要么设定大量的约束条件，最终无法得到结果。模糊控制是以人对被控系统的控制经验为依据而设计控制器的，故无须知道被控系统的数学模型。

(2) 模糊控制是一种反映人类智慧思维的智能控制，它采用人类思维中的模糊量，如“高”、“中”、“低”、“大”、“小”等，控制量由模糊推理导出。这些模糊量和模糊推理是人类智能活动的体现。

(3) 模糊控制易为人们所接受，它的核心是控制规则，这些规则是以人类语言表达的，对数据的精确性要求不高，例如“衣服较脏，则投入洗涤剂较多，洗涤时间较长”，这是洗衣服时的控制规则。很明显，易为一般人所接受和理解。

(4) 容易构造，如果用微型机系统或单片机来构造模糊控制系统，其构造和一般数学控制系统无异，模糊控制方法是用软件实现的。如果用模糊单片机构造模糊控制系统则更为方便，基本上只要给模糊单片机设置好数据就可以了。

(5) 适应性好，模糊控制系统无论被控对象是线性还是非线性的，都能执行有效的控制，具有良好的鲁棒性和适应性。

(6) 模糊控制能充分利用专家信息，相比于神经网络，模糊控制建立在已经熟悉相关系统的专家基础上，能够充分利用现有的专家知识。

(7) 模糊控制是柔性的。对于给定的系统，很容易处理以及直接增加新的功能，而不需要从头做起。

下面一节我们将简单介绍一下模糊控制技术。

### 2.2.1 模糊控制技术

模糊控制是以模糊集合论、模糊语言变量及模糊逻辑推理为基础的计算机智能控制。基本概念是由美国加利福尼亚大学著名教授查德提出，此后在短短 20 年中取得了令人瞩目的成果。

尽管“模糊”这个词在这里容易使人产生误解，实际上在模糊逻辑控制中的每一个特定的输入都对应着一个实际的输出，并且这个输出值是完全可以预测的。所以模糊控制本身并不模糊，模糊控制并不是“模糊的”控制，而是用来对“模糊”进行处理以达到精确的控制，它是一种精确解决不精确、不完全信息的

方法，其最大特点就是用它可以比较自然地处理人的概念，是一种更人性化的方法，模糊控制越来越为更多的科技工作者接受。模糊控制系统由于采用类似于人脑的模糊推理方法，遵循一定的控制规则，结合实际经验，对系统进行动态调整，具有不依赖对象的数学模型，便于利用人的经验知识，鲁棒性好，简单实用等优点。

模糊理论主要包括模糊集合理论、模糊逻辑、模糊推理和模糊控制等方面内容。

### 2.2.1.1 模糊集合

经典集合是具有精确边界的集合。经典集合对集合中的对象关系进行严格划分，一个对象要么是完全属于这个集合，要么就完全不属于这个集合，不存在介于两者之间的情况。模糊集合是没有精确边界的集合。这意味着，从“属于一个集合”到“不属于一个集合”之间的转变是逐渐的，这个平滑的转变是由隶属函数来表征的。

给定论域  $U$ ， $U$  到  $[0, 1]$  闭区间的任一映射  $u_A$ ：

$$u_A : \begin{array}{l} U \rightarrow [0,1] \\ u \rightarrow u_A(u) \end{array}$$

都确定  $U$  的一个模糊子集，映射  $u_A(u)$  称为模糊子集  $A$  的隶属度函数， $u_A(u)$  称为  $u$  对于模糊集合  $A$  的隶属度。在不引起混淆的情况下，模糊子集也可称为模糊集合<sup>[16]</sup>。

上述定义表明，论域  $U$  上的模糊子集  $A$  由隶属度函数  $u_A(u)$  来表征， $u_A(u)$  的取值范围为闭区间  $[0, 1]$ ， $u_A(u)$  的大小反映了  $u$  对于模糊子集  $A$  的隶属程度。若  $u_A(u)$  接近 1，表示  $u$  属于  $A$  的程度高，若  $u_A(u)$  接近 0，表示  $u$  属于  $A$  的程度低。模糊集合具有灵活的隶属关系，它允许在一个集合中部分隶属。对象在模糊集合中的隶属度可以从 0 到 1 之间的任何值，而不像在经典集合中非得是严格的 0 或 1。这样模糊集合就可以从“不隶属”到“隶属”逐渐的过渡。

### 2.2.1.2 隶属度函数

隶属度函数类型有如下几种：三角形、梯形、高斯型、钟型、Sigmoid 型、 $\Pi$  型以及  $Z$  型。利用 MATLAB 工具箱中提供的函数可以建立和计算上述各种类型



隶属度函数。

最简单的隶属函数是三角形函数 (trimf) 和梯形函数 (trapmf); 高斯型函数 gaussmf、双边高斯型函数 gauss2mf 和钟型函数 gbellmf。三角形函数和梯形函数) 具有很好的光滑性, 没有零点而且有比较清晰的物理意义, 是描述模糊集合最为常用的隶属度函数, 具有非常重要的地位; 高斯和钟型函数具有很好的光滑性及对称性, 但他们不适合表示那些具有非对称性的事物特征, sigmoid 型隶属度函数 sigmf 就可以很好的描述那些具有非对称性的事物, 它的曲线图形同样具有很好的光滑性。

### 2.2.1.3 模糊控制规则

#### (1) . “如果—则” 推理规则

“如果—则” 规则有称为 “if-then” 规则。一个最简单的 “if-then” 规则具有如下形式:

if x is A then y is B (如果 x 是 A, 则 y 是 B)。

其中 A 和 B 是分别定义在论域 X 和 Y 上的模糊语言变量。规则的 “如果部分 (x is A) 被称为前提部分, 规则 “那么部分 (y is B) ” 被称为结论部分。

#### (2) . 基于 “如果—则” 规则的模糊蕴涵关系

描述元素之间是否相关的数学模型称为关系, 描述模糊元素之间相关程度的数学模型称为模糊关系

规则 “如果 x 是 A 则 y 是 B” 表示了 A 与 B 之间的一种关系, 其中 A、B 都是模糊语言变量, 这种关系是一种模糊关系, 这里称之为模糊蕴涵关系, 记为  $A \rightarrow B$ 。在模糊逻辑中  $A \rightarrow B$  没有严格的定义, 也不是普通逻辑中  $A \rightarrow B$  的推广。学者们提出了许多不同的定义方法, 下面是在模糊逻辑推理中经常用到的几种模糊蕴涵关系的定义和运算方法。

假设 A 和 B 是分别定义在 X 和 Y 上的模糊集合, 由  $A \rightarrow B$  所表示的模糊蕴涵关系是定义在  $X \times Y$  上的一个特殊的模糊关系<sup>[16]</sup>, 计算公式如下:

模糊蕴涵最小运算(Mamdani)

$$R_c = A \rightarrow B = \int_{x \times Y} u_A(x) \wedge u_B(y) / (x, y)$$

模糊蕴涵积运算(Larsen)

$$R_p = A \rightarrow B = \int_{x \times Y} u_A(x) u_B(y) / (x, y)$$

模糊蕴涵算术运算(Zadeh)

$$R_z = A \rightarrow B = \int_{x \times Y} 1 \wedge (1 - u_A(x) + u_B(y)) / (x, y)$$

模糊蕴涵布尔运算(Mamdani)

$$R_b = A \rightarrow B = \int_{x \times Y} (1 - u_A(x)) \vee u_B(y) / (x, y)$$

这里的  $\int$ 、 $/ (x, y)$  不是代表积分和除法，只代表一种记号。

在以上几个公式中，最常用的是最小运算  $R_c$ 、 $R_p$ 。

### (3) 基于“如果—则”规则的模糊推理

设  $A$ 、 $A'$  和  $B$  分别是  $X$ 、 $X$  和  $Y$  的模糊集合，模糊蕴含  $A \rightarrow B$  的模糊关系  $R$ ，

输入条件： $x$  是  $A'$

逻辑规则：如果  $x$  是  $A$  则  $y$  是  $B$

推理输出： $y$  是  $B'$

推理是根据模糊集合  $A'$  与模糊蕴含关系  $A \rightarrow B$  的合成计算出结论  $B'$

$$B' = A' \circ (A \rightarrow B) = A' \circ R$$

从上面的公式我们可以看到，模糊推理的结果是由模糊合成以及模糊蕴含的计算决定的，而对于模糊蕴含以及模糊合成计算的定义并不是唯一的，因此正确的决定模糊蕴含以及模糊合成的计算公式对于最终的模糊推理结果的合理性是非常重要的。

MATLAB 模糊逻辑工具箱提供了  $R_c$  和  $R_p$  两种模糊蕴含推理方法；对于合成算法，则默认并唯一采用最大—最小合成算法。

若采用  $R_c$ ，则上式表示为：

$$A' \circ R \leftrightarrow u_{B'}(y) = \bigvee_x (u_{A'}(x) \wedge u_R(x, y))$$

### (4) 基于模糊规则库的模糊推理

1) 连接词“and”、“or”、和“also”



在多输入多输出 (MIMO) 的模糊逻辑系统中, 常常可能会有很多条规则, 而且这些规则的前提条件部分和结论部分也可以是由许多部分组成。前面介绍的 If-then 规则只是最简单的形式, 是最基本的模糊推理系统, 如果满足“输出-输入是解耦的, 各输入间、各输出间分别是独立的”, 就可以用连接词“and”、“or”和“also”连接的一系列简单的模糊规则库来表示。其中, “and”和“or”用来连接同一规则的多个输入或多个输出, “also”用来连接多条不同的模糊规则。由于系统是解耦的, 所以系统的多个输出之间不存在“or”, 只有“and”关系, 而对于同一规则的多个输入间, 则可以用“and”和“or”来连接。因为对于同一条规则中用“or”连接的不同输入前提条件也可以拆分成多条用“also”连接的规则, 所以在很多文献上的模糊规则库只采用“and”和“also”两种连接词。例如规则“如果 x 是 A or y 是 B 则 z 是 C”可以写成如下两条规则: 如果 x 是 A, 则 z 是 C; 如果 y 是 B, 则 z 是 C。MATLAB 中使用“and”、“or”、和“also”三种连接词来构造模糊推理系统。

## 2) 连接词“and”、“or”、和“also”的计算

对于用“and”和“or”连接的同一规则的不同前提条件 (例如“如果 x 是 A and y 是 B”, “如果 x 是 A or y 是 B”), 以及对于模糊系统常用的  $R_c$  和  $R_p$  模糊蕴含算法及最大-最小合成算法的系统中, “and”和“or”分别采用下列算法可以取得较好的效果。

### ①and 计算

交运算  $u_{A \times B}(x, y) = \min\{u_A(x), u_B(y)\}$  (MATLAB 中对应“and”的 min 算法)

积运算  $u_{A \times B}(x, y) = u_A(x)u_B(y)$  (MATLAB 中对应“and”的 prod 算法)

### ②or 计算

并运算  $u_{A \times B}(x, y) = \max\{u_A(x), u_B(y)\}$  (MATLAB 中对应“or”的 max 算法)

代数和  $u_{A \times B}(x, y) = u_A(x) + u_B(y) - u_A(x)u_B(y)$  (MATLAB 中对应“or”的 probor 算法)

对于对个“and”和“or”连接的前提条件, 这些运算可以组合起来, 并且类似算术四则很和运算的一些规律 (例如结合律、交换律、分配律等, 其中“and”具有乘法的性质, “or”具有加法的性质。

在模糊逻辑规则库中，常常给出一系列的模糊控制规则（来源于专家的知识），这些规则之间无先后次序之分，连接这些句子的连接词用“also”表示。MATLAB 的模糊逻辑推理系统先针对每一条规则推理出结果，然后对所有不同规则得到的推理结果进行“also”计算（MATLAB 称为合成 Aggregation 计算）。

### 2.2.2 Mamdani 型模糊推理方法

Mamdani 型的模糊推理方法是最常见的算法，最先将模糊集合的理论用于控制系统。它是由 Ebrahim Mamdani 在 1975 年为了控制蒸汽发动机提出来的，这种方法采用综合一系列有经验的操作者提供的线性控制规律来控制锅炉。

工程应用中，往往期待推理输出的是一个确定的控制量或其他的确定的数值，而应用 Mamdani 型的模糊推理系统，每一条规则推理后得到的输出是变量的分布隶属度函数或离散的模糊集合。在将多条规则的结果合成（Aggregation）以后，对每一个输出变量模糊集合都需要进行模糊化（解模糊化）处理，以得到实际问题期望的输出。

Mamdani 型的模糊推理系统有五个过程：输入变量的模糊化、模糊关系运算、模糊合成运算、不同规则结果的综合、逆模糊化。

#### (1) 输入模糊化

一般来说，对于实际问题输入的模糊化是建立模糊推理系统的第一步，也就是选择系统的输入变量并根据其相应的隶属度函数来确定这些输入分别属于恰当的模糊集合。在 MATLAB 模糊逻辑工具箱中，模糊化过程的输入必须是一个确定的数值，这也对输入变量的广泛性起了一些限制作用，而输入则是一个特定的模糊集合上的隶属程度（总是 0~1 之间）。输入的模糊化相当于一个对应的查表或是函数计算。

#### (2) 输入模糊集合的合成运算（连接词“and”和“or”计算）

当输入已经被模糊化以后，我们就可以知道这些输入满足相应的模糊推理规则的程度。但是如果给定的模糊规则的条件部分不是单一输入，而是多输入，就需要运用模糊合成运算对这些多输入进行综合考虑和分析。经过模糊合成计算，这多个输入可以得到一个数值来表示对该多条件输入规则的综合满足程度，继而被用于输出函数中。模糊合成运算的输入对象是两个或多个经过模糊化后的输入变量的隶属度值，输出是一个唯一确定的数值。任何完善合理的模糊合成方法都

可以用“与操作 (and)”和“或操作 (or)”来实现。

### (3) 模糊蕴涵方法

在进行模糊推理之前,还必须考虑不同模糊规则的权重问题(对于多规则系统,可能各条规则的重要程度是不同的)。因此,每一条规则赋有一个0~1之间的权重值,这个权重值与每条规则的输入发生作用。通常权重相同且为1,所以它对推理的结果并不产生影响。但在某些情况下则有可能修改各条模糊规则的相对权重而不是简单地取为1。

一旦给各条规则分配了恰当的权重,就可以进行蕴涵计算了。模糊蕴涵计算过程的输入是由前面输入模糊集合的合成运算得到的单一数值及模糊集合,输出为根据模糊规则推导的结论模糊集合。模糊蕴涵也就是各条规则的代表问题。在MATLAB中蕴涵有两种方法:最小法 min 和乘积法 prod。

### (4) 输出的合成 Aggregation

由于在模糊推理系统里决策的生成取决于所有的模糊规则,因而上一步所计算出的模糊输出必须用某种方式组合起来以得出结果。输出的合成就是对于所有模糊规则输出的模糊集合进行综合的过程。对于每一个输出变量,合成只进行一次。最终,对于每个输出变量仅给出得到一个模糊输出集合。

合成的方法应该是与顺序无关的,各条规则的结果的合成顺序并不影响结果。MATLAB 提供三种方法:最大值法 max、概率法 probor、求和法 sum。

### (5) 逆模糊化

逆模糊化的输入是模糊集合(即模糊推理系统的总体输出模糊集合),输出是一个数值。最后对实际有用的每一个变量的输出结果通常要求是一个确定的值。由于经过模糊推理后所得到的是输出变量的一个范围上的隶属度函数,因此必须进行去模糊化以将输出变为确定的值。最经常的去模糊化的方法是面积中心法。

Centroid (面积中心法) 又称重心法,即计算隶属度函数曲线包围区域的重心。对于连续论域的情形,设  $U$  是某一变量  $u$  在论域  $U$  上的模糊集合,则去模糊化的结果为:

$$u_c = \frac{\int U(u)u du}{\int U(u) du}$$

基于推理合成规则，由系统误差  $E$  的论域  $X = \{-5, -4, \dots, 0, \dots, +4, +5\}$  和误差变化率  $EC$  的论域  $Y = \{-5, -4, \dots, 0, \dots, +4, +5\}$ ，根据语言变量误差  $E$  和误差变化率  $EC$  的隶属函数曲线，针对论域  $X$ 、 $Y$  全部元素的所有组合，求取相应的输出语言变量  $U$  的语言值，并应用最大隶属度法对语言值进行模糊判决，取得以论域  $Z = \{-5, \dots, 0, \dots, +5\}$  的元素表示的控制量  $u$ 。可建立模糊控制查询表如表 2.1 所示。

| $u$ | -5 | -4 | -3 | -2 | -1 | 0 | +1 | +2 | +3 | +4 | +5 |
|-----|----|----|----|----|----|---|----|----|----|----|----|
| -5  |    |    |    |    |    |   |    |    |    |    |    |
| -4  |    |    |    |    |    |   |    |    |    |    |    |
| -3  |    |    |    |    |    |   |    |    |    |    |    |
| -2  |    |    |    |    |    |   |    |    |    |    |    |
| -1  |    |    |    |    |    |   |    |    |    |    |    |
| 0   |    |    |    |    |    |   |    |    |    |    |    |
| +1  |    |    |    |    |    |   |    |    |    |    |    |
| +2  |    |    |    |    |    |   |    |    |    |    |    |
| +3  |    |    |    |    |    |   |    |    |    |    |    |
| +4  |    |    |    |    |    |   |    |    |    |    |    |
| +5  |    |    |    |    |    |   |    |    |    |    |    |

表 2.1 模糊控制查询表

### 2.3 模糊控制系统

一般而言，模糊控制时建立在人们的经验和常识的基础上，即操作人员对被控系统的了解不是通过精确的数学表达式，而是通过以往的实践经验和常识。由于人的决策过程本身就带有模糊性，因此控制动作并非稳定一致，而带有一定的主观性。但是我们可以用语言把操作人员的控制策略描述出来，用语言表达成一组定性的条件语句和精确的决策规则，然后利用模糊集合作为工具使其量化，设计一个控制器，用人的经验法则模仿人的控制策略，再驱动设备对复杂的过程进行控制，形成模糊控制器。其方框图如下：

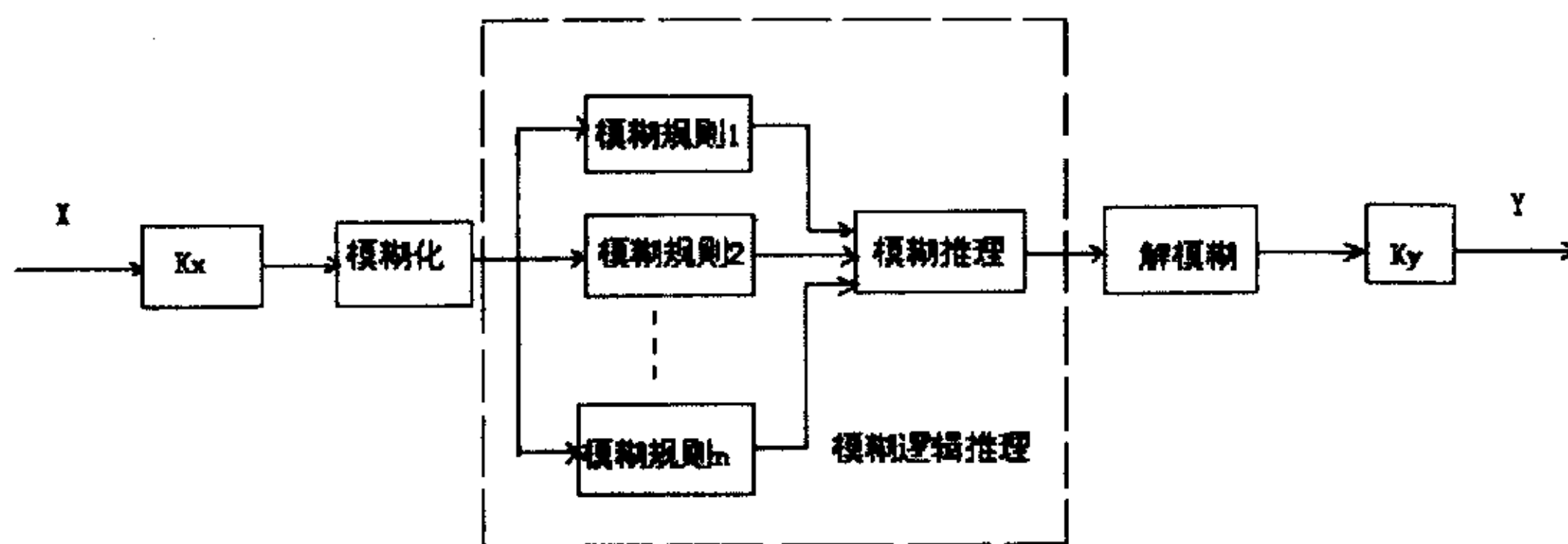


图 2.1 模糊推理控制器方框图

模糊控制系统是以模糊数学、模糊语言形式的知识表示和模糊逻辑的规则推理为理论基础，采用计算机控制技术构成的一种具有反馈通道的闭环结构的数字控制系统。它的组成核心是智能型的模糊控制器。模糊控制系统的基本结构与原理如图 2.2 所示，从图中可见，模糊控制系统一般由以下五个部分组成：

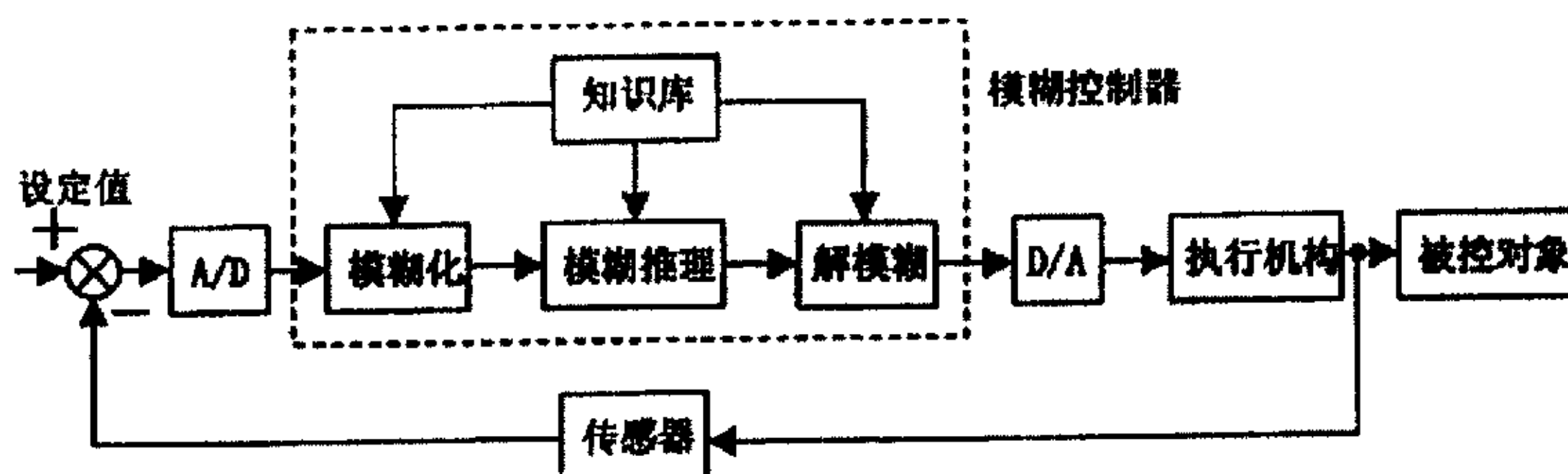


图 2.2 模糊控制系统基本结构

(1)模糊控制器：这是模糊控制系统的核心部分，也是和其它控制系统区别最大的环节。模糊控制器由四个基本部分组成：

①模糊化：即把输入的精确量转化为模糊量。输入信号映射到相应论域上的一个点后，将其转化为该论域上的一个模糊子集。

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目标，通常由数据库和模糊规则库两部分组成。数据库主要包括各语言变量的隶属度函数，尺度变换因子和模糊空间的分级数等；规则库包括了用模糊语言变量表示的一系列控制规则，它们反映了控制专家的经验 and 知识。

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蕴含关系及推理规则来进行的。



④解模糊化：作用是将模糊推理得到的控制量(模糊量)变换为实际可用于控制的精确量，它包括两部分内容：一是将模糊的控制量经解模糊变换变成表示在论域范围的精确量；二是将表示在论域范围的精确量经量程转换变成实际的控制量。

(2)输入/输出接口装置：在实际系统中，多数被控对象的控制量及其可观测状态量是模拟量，因此模糊控制器需要通过模/数(A/D)转换从被控对象获取数字信号量，并将模糊控制器的输出控制量经数/模(D/A)转换为模拟信号，送给执行机构去控制被控对象。

(3)被控对象：可以是线性或非线性的、定常或时变的，也可是单变量或多变量，以及具有强耦合和干扰等多种情况。一般而言，对于那些难以建立精确数学模型的复杂对象，适宜选择模糊控制。

(4)执行机构：可以是电气的，如交、直流电动机，伺服电动机、步进电动机等，也可为气动和液压的，如各类气动调节阀和液压马达、液压阀等。

(5)传感器：由于被控制对象往往为温度、压力、速度、流量等非电量，需要通过传感器将其转化为数字或模拟的电信号。传感器的精度直接影响控制系统的精度，因此要求其精度高、稳定性好。

模糊控制器通常由计算机实现，计算机经过采样和 A/D 转换获取被控制量的精确值，然后将此量与给定值比较得到误差信号  $e$ 。一般选误差信号  $e$  作为模糊控制器的一个输入量，由模糊控制器的输入接口将该精确量转化为相应的模糊量，误差  $e$  的模糊量可用相应的模糊语言子集  $e$  来表示。再由  $e$  和模糊控制规则  $R$ (模糊关系)，根据推理的合成规则进行模糊决策，得到模糊控制量  $u$ 。为将模糊控制量  $u$  转化为精确量，由模糊控制器的输出接口作解模糊处理，得到精确数字控制量后，经 D/A 转换变为精确的模拟量送给执行机构，对被控对象进行进一步控制。由上可知，设计模糊控制系统的核心部分即模糊控制器，必须解决三方面的问题：

(1)精确量的模糊化。将模糊控制器的精确输入量(例如误差  $e$ 、误差变化率  $ec$ )转化为相应语言变量论域上的语言值(模糊子集)。

(2)模糊控制规则的确定及模糊推理。将来自熟练操作者的知识和经验整理成一组模糊条件语句，构成模糊控制规则，用于模糊推理。



(3)输出语言变量(模糊量)的模糊判决。将模糊推理得出的输出模糊量转换为精确量

为保证模糊控制系统设计与仿真工作的可靠性及高效性，选择合适的开发工具非常重要。专用的模糊系统开发环境，如美国国家半导体公司的 Neufuz4，德国 Inform 公司的 Fuzzy Tech3.0 等，这些系统功能强大，但价格不菲；如果自行编程完成模糊系统的设计，则开发时间长，效率低且通用性差。MATLAB 是目前应用广泛且具权威性的控制系计算机辅助设计语言和软件工具。本论文利用 MATLAB 软件，结合其 SIMULINK 图形化仿真环境和模糊逻辑工具箱（Fuzzy Logic Toolbox）来完成制动器模糊控制系统的设计与仿真。

### 第三章 制动器模糊控制系统设计

考虑到开发一种新型的制动系统将花费很长的时间与金钱，本课题于是在比较成熟的产品电力液压推杆制动器上进行改进研究。在电力液压推杆制动器的基础上，以室外龙门起重机大车运行机构（双制动器分别制动）作为研究对象。

我们所选取的电力液压推杆制动器参数如下：

型号：YWZ

制动轮径：200mm

额定制动力矩：200Nm

所选取龙门起重机大车运行机构的参数如下：

额定负载 10t；

L=26m，吊钩伸出距离每侧 5m；

龙门架总重 460kN；

大车运行速度 40m/min；

工作制度 中级 JC25%；

迎风面积 龙门架  $80\text{m}^2$ ，小车  $6\text{m}^2$ ，货物  $7\text{m}^2$

制动力矩计算公式如下式：

$$\begin{aligned} M &= \frac{J \cdot n}{9.55t} + (Wf + Wp - Wm) \frac{Dl \cdot \eta}{2i} \\ &= \frac{3.513 \times 941}{9.55 \times 4} + (27.9 + 0.602 - 2.75) \times 10^3 \frac{0.7 \times 0.9}{2 \times 51.7} \\ &= 86.5 + 153 \\ &= 293.5\text{Nm} \end{aligned}$$

其中  $J$  为转动惯量， $n$  为大车运行机构高速轴电机转速， $Wf$  为大车运行过程中的风阻力， $Wp$  为坡阻力， $Wm$  为摩擦阻力。

由此可以看出，制动力矩主要是由大车运行机构高速轴电机转速  $n$  和所受风阻力  $Wf$  决定的，所以本课题在设计模糊控制器时，将两个输入变量确定为：龙门起重机电机转速和所受风阻力，一个输出变量确定为：变频器频率。控制方式就是当转速和风阻力变化时，改变变频器频率，从而该变电力液压推杆制动器电机转速，使控制力矩相应变大或变小。这里我们只考虑顺风情况，在实际室外龙

门起重机运行过程中也会出现逆风情况，可参照本文编写逆风状态时的模糊规则，这里就不再介绍了。需要声明的是：电力液压推杆制动器是靠弹簧力上闸，靠推杆推力松闸，因此利用频率控制制动器电机转速进而控制推力时，输出频率越大制动力矩反而越小，输出频率越小制动力矩反而越大。

这里将输入变量分别模糊化成三个模糊量：大、中、小和高、中、低；输出变量模糊化成五级：很大、大、中、小、很小。则描述输入输出量的语言值模糊子集记为：{大，中，小}

模糊规则如下：

- (1) If 风力大 and 转速高 then 输出频率很小；
- (2) If 风力大 and 转速中 then 输出频率小；
- (3) If 风力大 and 转速低 then 输出频率中；
- (4) If 风力中 and 转速高 then 输出频率小；
- (5) If 风力中 and 转速中 then 输出频率中；
- (6) If 风力中 and 转速低 then 输出频率大；
- (7) If 风力小 and 转速高 then 输出频率中；
- (8) If 风力小 and 转速中 then 输出频率大；
- (9) If 风力小 and 转速低 then 输出频率很大。

通过查阅手册我们确定龙门起重机制动过程中所受风阻力在 10~20kN 之间为大，在 4~10kN 之间为中，4kN 之下为小；电机转速在 700~940r/min 之间为高，在 300~700r/min 之间为中，低于 300r/min 为低；变频器频率在 45~50hz 之间为很大，在 35~45HZ 之间为大，在 20~35HZ 之间为中，在 5~20HZ 之间为小，低于 5HZ 为小。模糊推理系统有两种建立方法，本章利用命令行来建立实现制动控制的 FIS 系统，第四章则利用各种 GUI 来建立模糊控制系统。

### 3.1 建立制动器模糊推理系统

建立新的模糊推理系统，将 FIS 结构名设为模糊制动制动：

```
Mohul=newfis('模糊制动控制')
```

这样就建立了一个 Mamdani 型新的模糊推理系统，此时输入、输出及规则等都没有定义。如果键入 FIS 结构名就可以得到确认：

```
mohul
```

```

mohul =

    name: '模糊制动控制'
    type: 'mamdani'
    andMethod: 'min'
    orMethod: 'max'
    defuzzMethod: 'centroid'
    impMethod: 'min'
    aggMethod: 'max'
    input: []
    output: []
    rule: []

```

### 3.2 定义输入、输出变量

给系统添加两个输入和一个输出，其输入变量名分别为“风阻力”和“转速”，变量范围为[-5 5]和[-10 10]；输出变量名为“频率”，变量范围为[-5 5]。

```

mohul=addvar(mohul,'input','风阻力',[-5 5]);
mohul=addvar(mohul,'input','转速',[-10 10]);
mohul=addvar(mohul,'output','频率',[-5 5]);

```

这时得到的 FIS 具有两个输入和一个输出，其输入、输出域为非空：

```

mohul.input
ans =

1x2 struct array with fields:

    name
    range
    mf

mohul.output
ans =

    name: '频率'

```

```
range: [-5 5]
mf: [1x5 struct]
```

### 3.3 设计隶属度函数

在模糊控制系统中,对输入精确量进行模糊化的关键是确定其模糊集的隶属函数。隶属函数是人们对客观事物不分明性的定量描述,必然带有人的主观性,不同的人对同一模糊概念会建立完全不同的隶属函数。所以从理论上说,即使根据专家的经验确定的隶属函数,这种没有理论化的方法也不能保证其正确性,因为任何人的经验和知识都是有局限性的。在这个方面,国内外学者已经进行了大量的研究,提出各种各样的确定方法,诸如模糊统计法、函数分段法、二元对比排序法、对比平均法、滤波函数法、示范法和专家经验法等等方法。不过,在实际应用中,虽然用不同方法确定了不同的模糊集合的隶属函数,在一定范围内,尽管实现控制过程的细节和达到目标的过程细节和相应时间可能有差别,但模糊逻辑控制却都能实现控制,达到预期目标。换句话说,隶属函数的确定并不是唯一的,允许有不同的组合。隶属度函数类型有如下几种:三角形、梯形、高斯型、钟型、Sigmoid 型、 $\Pi$ 型以及 Z 型。最简单的隶属函数是三角形函数 (trimf) 和梯形函数 (trapmf); 高斯型函数 gaussmf、双边高斯型函数 gauss2mf 和钟型函数 gbellmf。利用 MATLAB 工具箱中提供的函数可以建立和计算上述各种类型隶属度函数。本论文中控制器的输入输出分别采用几种隶属函数。

```
mohu1=addmf(mohu1,'input',1,'大','gaussmf',[2.1 5]);
mohu1=addmf(mohu1,'input',1,'中','gaussmf',[2.1 0]);
mohu1=addmf(mohu1,'input',1,'小','gaussmf',[2.1 -5]);
```

这样就给第一个输入变量定义了三个高斯型隶属度函数:“大”、“中”、“小”,其参数分别为 [2.1 5], [2.1 0], [2.1 -5]。同样可以给第二个输入变量和输出变量定义隶属度函数。

```
mohu1=addmf(mohu1,'input',2,'高','gaussmf',[3.3 10]);
mohu1=addmf(mohu1,'input',2,'中','gaussmf',[3.3 0]);
mohu1=addmf(mohu1,'input',2,'低','gaussmf',[3.3 -10]);
mohu1=addmf(mohu1,'output',1,'很大','gbellmf',[1.5 6 -5]);
mohu1=addmf(mohu1,'output',1,'大','gbellmf',[1.5 6 -2]);
```

```
mohu1=addmf(mohu1,'output',1,'中','gbellmf',[0.5 6 0]);  
mohu1=addmf(mohu1,'output',1,'小','gbellmf',[1.5 6 2]);  
mohu1=addmf(mohu1,'output',1,'很小','gbellmf',[1.5 6 5]);
```

MATLAB 提供了显示隶属度函数曲线的绘图函数 `plotmf`，借此可以方便地显示出所定义的变量隶属度函数：

```
figure  
subplot(3,1,1)  
plotmf(mohu1,'input',1)  
subplot(3,1,2)  
plotmf(mohu1,'input',2)  
subplot(3,1,3)  
plotmf(mohu1,'output',1)
```

执行后得到如图 3.1 所示隶属度函数曲线：



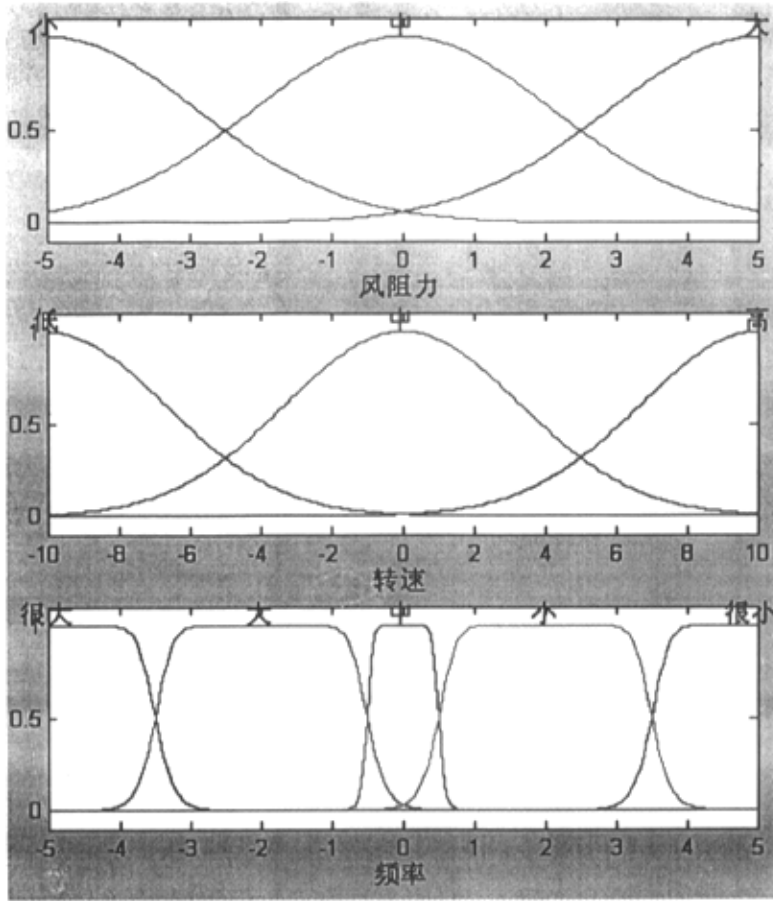


图 3.1 隶属度函数曲线

### 3.4 设计制动器模糊推理规则

按照上面的设计要求，可以得到如表 3.1 所示的模糊推理关系。

|              |    |   |    |
|--------------|----|---|----|
| 频率/转速<br>风阻力 | 高  | 中 | 低  |
| 大            | 很小 | 小 | 中  |
| 中            | 小  | 中 | 大  |
| 小            | 中  | 大 | 很大 |

表 3.1 制动器模糊推理关系

给输入变量“风阻力”的三个模糊量“大”、“中”、“小”依次编号为 1, 2, 3; 给输入变量“转速”编号为 1, 2, 3; 给输出变量“频率”编号为 1, 2, 3, 4, 5, 则根据上表可以产生规则表 rulelist。Addrule 函数有两个变量, 第一个变量为 FIS 的变量名, 第二个变量就是 rulelist, 用来表示规则的矩阵。规则列表矩阵有严格的要求: 当模糊系统有  $m$  个输入、 $n$  个输出时, 规则列表矩阵有  $m+n+2$  列, 其中前  $m$  列表示系统的输入, 每列的数值表示输入变量隶属度函数的编号; 接着的  $n$  列表示系统的输出, 每列的数值表示输出变量隶属度函数的编号; 第  $m+n+1$  列的内容为该条规则的权值 ( $0\sim 1$ ); 第  $m+n+2$  列的值决定模糊操作符的类型: 1 (当模糊操作符为 and 时), 2 (当模糊操作符为 or 时)。

```
rulelist=[1 1 5 1 1;
          1 2 4 1 1;
          1 3 3 1 1;
          2 1 4 1 1;
          2 2 3 1 1;
          2 3 2 1 1;
          3 1 3 1 1;
          3 2 2 1 1;
          3 3 1 1 1];
```

然后添加到系统 mohul 中去:

```
mohul=addrule(mohul,rulelist);
```

我们就建立了有关风阻力、转速与频率关系的模糊控制推理系统 mohu1, 这时在 MATLAB 中输入 mohu1, 可以得到

mohu1 =

```
      name: '模糊制动控制'  
      type: 'mandani'  
 andMethod: 'min'  
      orMethod: 'max'  
 defuzzMethod: 'centroid'  
      impMethod: 'min'  
      aggMethod: 'max'  
      input: [1x2 struct]  
      output: [1x1 struct]  
      rule: [1x9 struct]
```

### 3.5 输出曲线

关于负载、转速与频率关系的模糊控制推理系统建立好后, 我们需要观测其输出结果是否按照我们所要求的条件变化。

首先利用 surfview 命令绘制系统输出曲面:

```
surfview(mohu1);
```

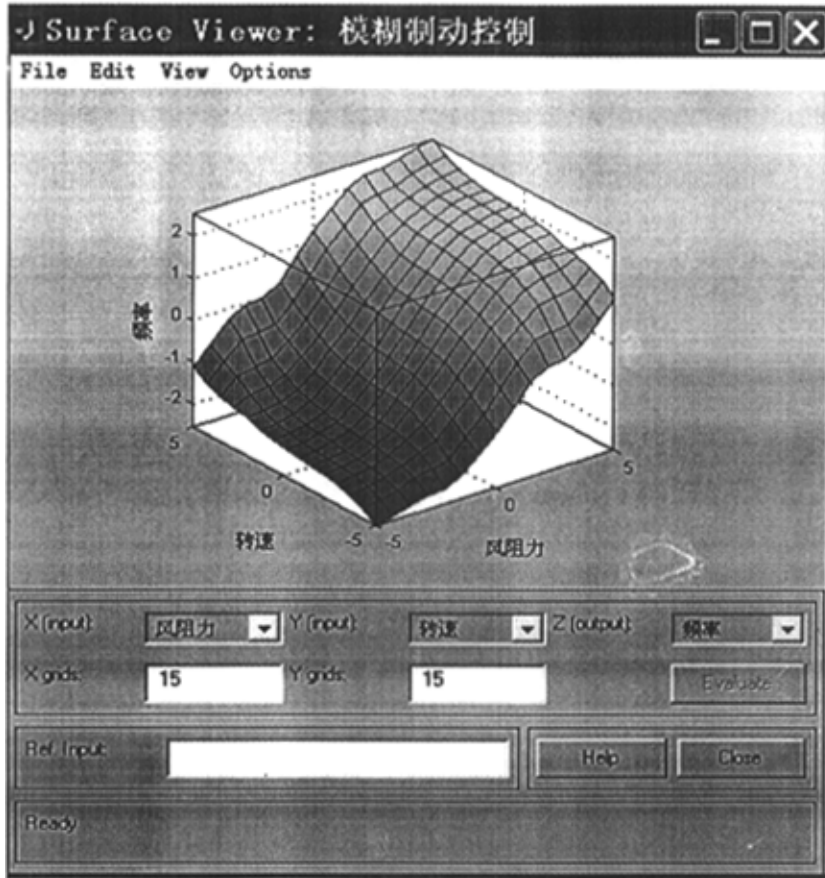


图 3.2 制动器模糊控制系统输出曲面

利用 `plotfis` 命令绘制所建立的 FIS 系统:

```
plotfis(mohul);
```

则可以得到如图 3.3 所示的 FIS 系统。

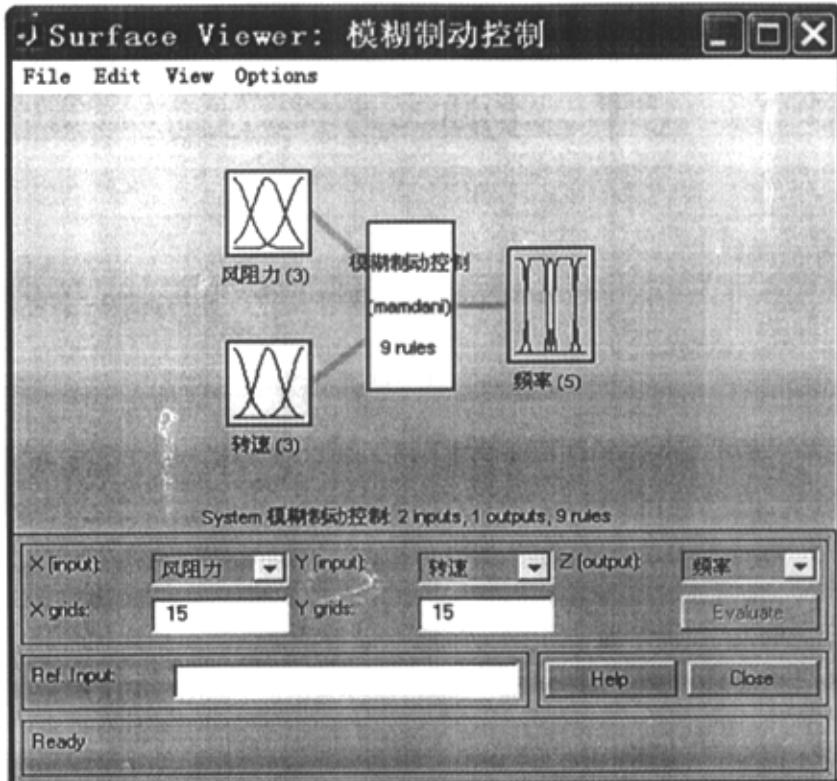


图 3.3 模糊控制推理系统

然后可以利用 evalfis 计算出指定输入的输出值，如当负载为 3，转速为 2 时：

```
evalfis([3 2],mohu1)
ans =
```

```
1.7785
```

接下来绘制输入/输出曲线，当转速不变，负载变化时，输入输出关系可求出如下：

```
x1=-5:0.1:5;
x2=0;
y=evalfis([x1' x2*ones(size(x1'))],mohu1);
figure(2)
plot(x1',y,'k')
```

```
xlabel('风阻力')
```

```
ylabel('转速')
```

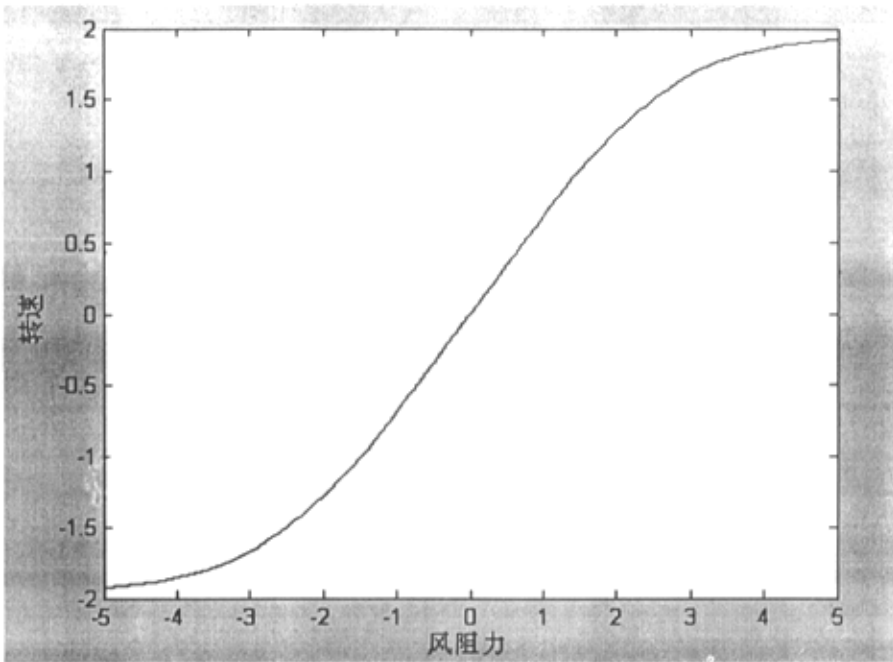


图 3.4 转速不变风阻力变化时的输入/输出曲线

而当负载不变，转速变化时，输入输出关系可求出如下：

```
x2=-10:1:10;  
x1=0;  
y=evalfis([x2' x1*ones(size(x2'))],mohu1);  
figure(3)  
plot(x2',y,'k')  
xlabel('转速')  
ylabel('风阻力')
```



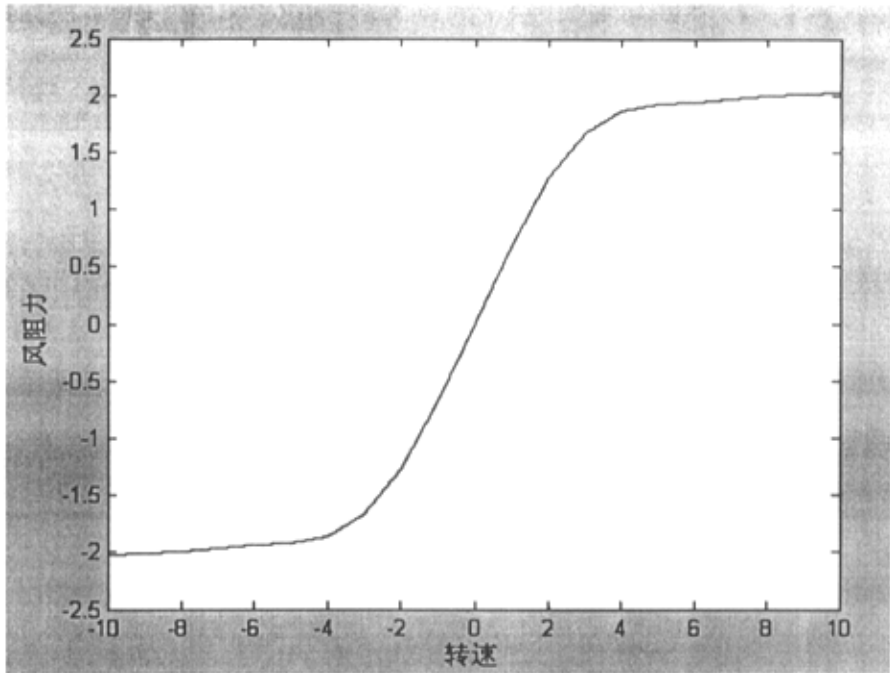


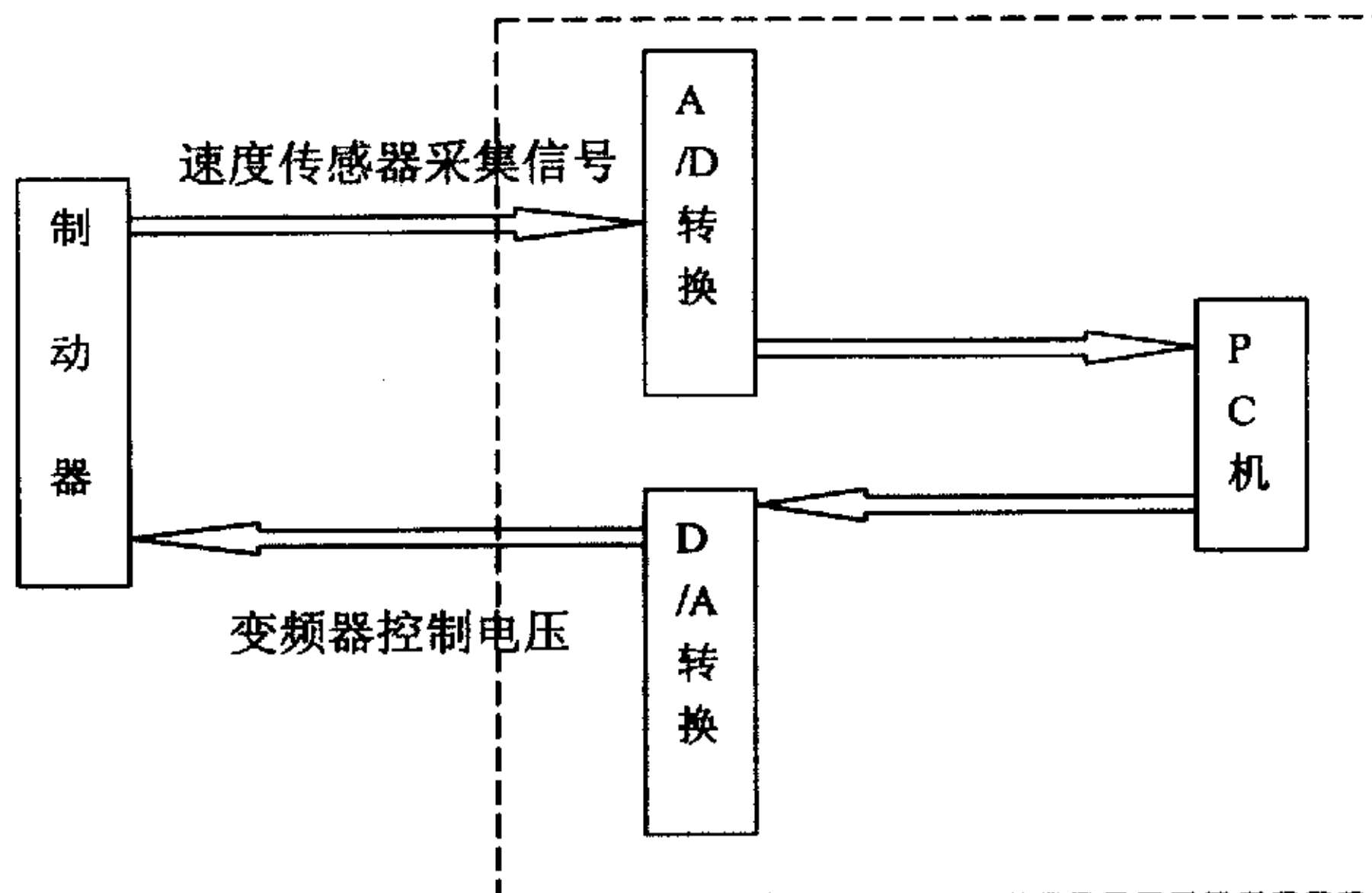
图 3.5 风阻力不变而转速变化时的输入/输出曲线

从图中可以看出：制动器模糊控制可以针对不同的负载和转速调节变频器的频率值，从而实现智能控制，改变了以往不论负载和转速的大小如何制动力矩不变的情况，减小了制动力矩过大对设备造成的冲击影响。说明模糊控制器能根据输入及时调整控制规则，将专家知识或操作人员的经验转换成自动控制策略，提高系统的动态性能。所以模糊控制并不是“模糊的”控制，而是用来对“模糊”进行处理以达到精确的控制，是一种精确解决不精确、不完全信息的方法模糊控制方法。

本章研究了制动器的模糊控制方法，针对龙门起重机利用模糊智能技术进行制动，设计出输出频率随风阻力或转速变化而改变的开环模糊控制，较好的改变了制动器的制动性能。有些制动器使用单位还希望能用闭环控制系统进行控制，提高系统的稳态和动态特性。第四章我们将模糊控制与传统 PID 控制相结合，从闭环控制的角度验证模糊控制的制动效果。

## 第四章 模糊 PID 控制系统

虽然模糊控制具有很多优点，然而模糊控制的精度受到量化等级的制约，另外，对于普通的模糊控制而言，它类似于比例微分的控制方式，还有一个非零的稳态误差，属于有差调节。如果将 PID 控制技术和它接合起来，取长补短，应该能发挥两者的优势从而取得更好的效果。因此在本章，我们建立了模糊 PID 控制系统，并将其与传统的 PID 控制系统相比较，通过仿真结果显示其优劣。这里仍然以应用于龙门起重机的电力液压推杆制动器为例，采用较流行的变频调速方式，通过改变制动器电机的频率来改变制动力矩的大小。速度闭环系统如图 4.1，主要由速度传感器、转换电路、计算机、变频器组成。



根据制动器实际运行情况，我们可以通过计算求取理想的速度曲线（如图 2.2），即随着制动力矩的增大，速度一直衰减到零。在系统运行过程中，将实际转速与理想制动速度曲线相比较，用它们的差值按一定算法求出控制量来调节变频器的给定频率，从而改变电机转速来达到更好的制动效果。

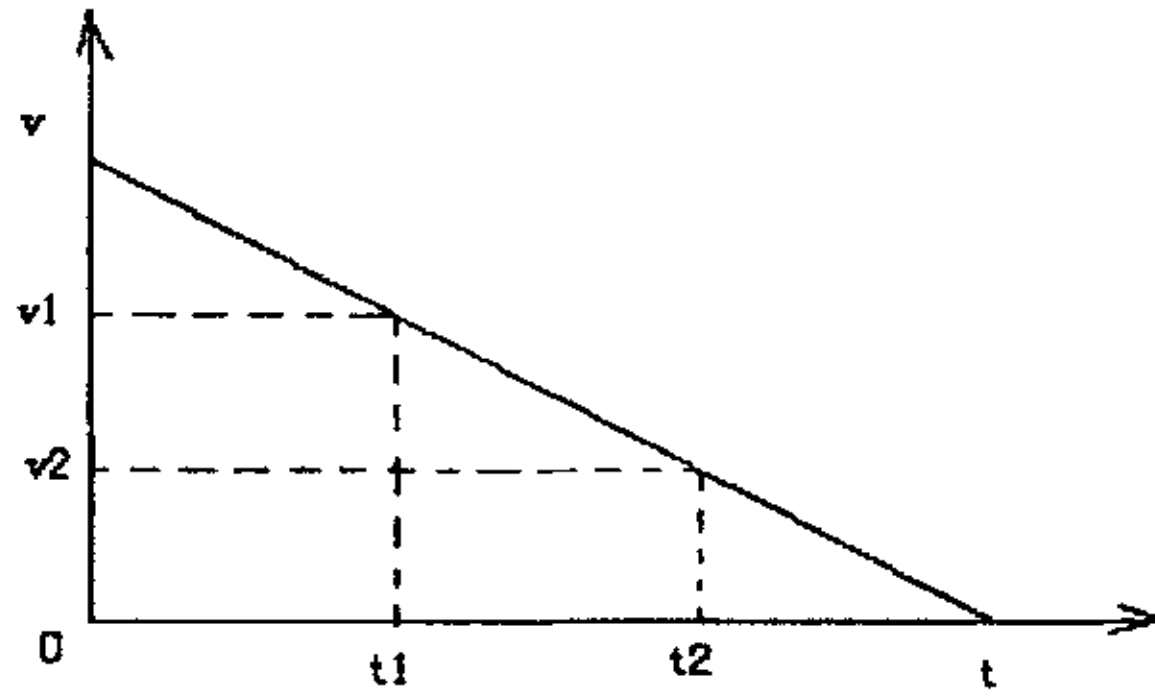


图 4.2 理想速度曲线

### 4.1 常规 PID 控制器

在控制系统中，常规 PID 控制由于原理简单、使用方便而得到广泛应用，其控制系统由 PID 控制器和被控对象组成，原理如图 4.3。

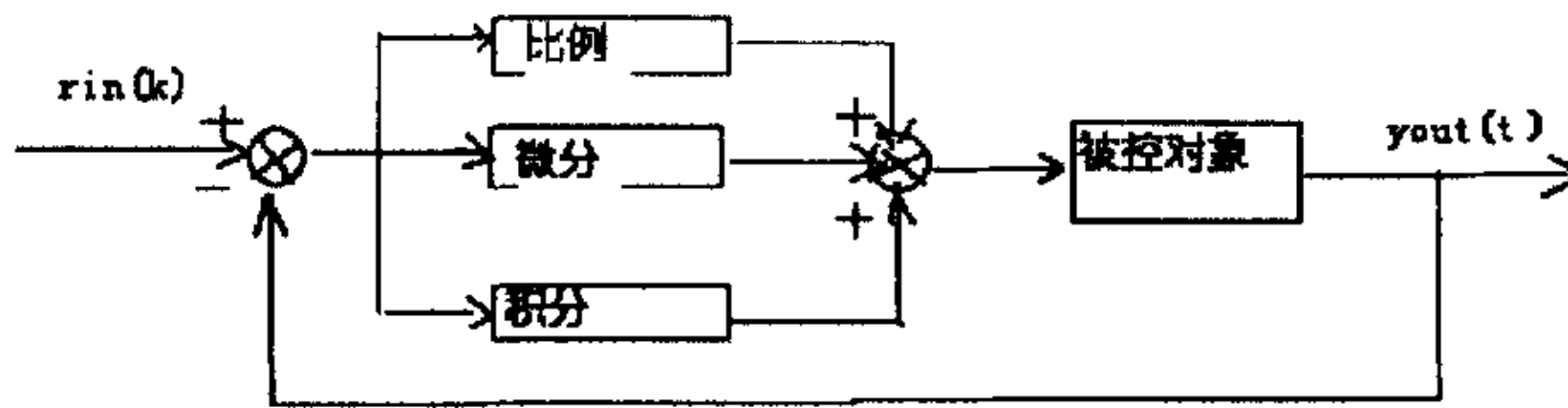


图 4.3 PID 控制系统原理图

PID 控制器是一种线性控制器，根据给定值  $r_{in}(t)$  与实际输出值  $y_{out}(t)$  构成控制偏差：

$$Error(t) = r_{in}(t) - y_{out}(t)$$

PID 的控制规律为：

$$u(t) = K_p \left[ error(t) + \frac{1}{T_i} \int error(t) dt + \frac{T_d \cdot derror(t)}{dt} \right]$$

或写成传递函数的形式：

$$G(s) = \frac{U(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right)$$

式中， $K_p$  为比例系数， $T_i$  为积分时间常数， $T_d$  为微分时间常数。各环节的作用

如下:

比例环节: 成比例地反映控制系统的偏差信号  $error(t)$ , 偏差一旦产生控制器立即产生控制作用, 以减小偏差。

积分环节: 主要用于消除静差, 提高系统的无差度。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, 反之则越强。

微分环节: 反映偏差信号的变化趋势, 并能在偏差信号变得太大之前, 在系统中引入一个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。

常规 PID 主要用于线性控制, 对于时变和非线性系统则显得无能为力。本文应用模糊 PID 推理方法进行控制, 通过仿真结果表明控制效果有明显的改善。

## 4.2 制动器模糊 PID 控制系统

### 4.2.1 模糊控制器原理

模糊控制的基本原理框图如图 2.6 所示。其核心部分为模糊控制器(图中虚线框部分), 模糊控制器的控制规律有计算机编程实现。

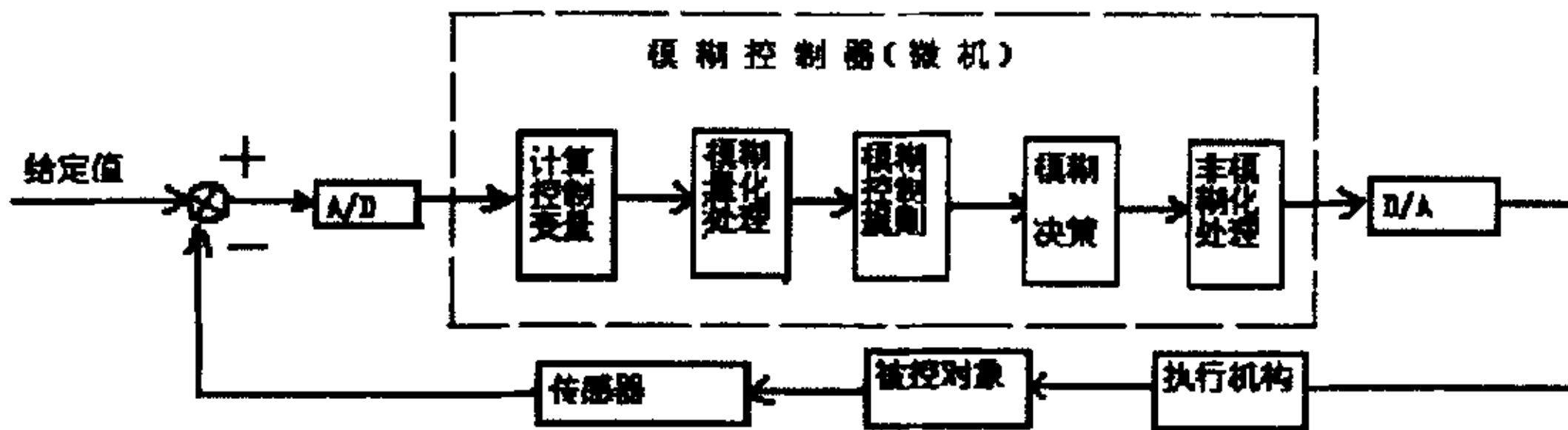


图 4.4 模糊控制原理图

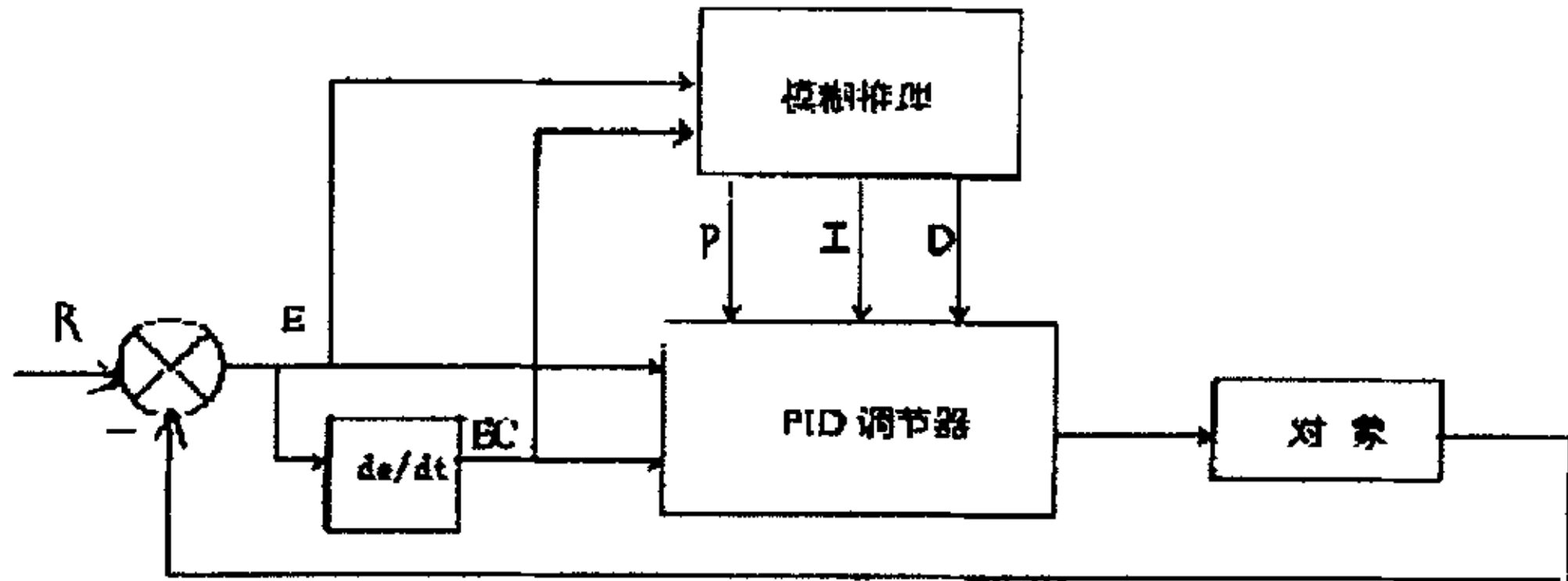
过程描述如下: 计算机采集被控制量的信号值, 将此量与给定值相比较得到误差信号  $E$ , 一般选取误差信号  $E$  作为模糊控制器的一个输入量。把误差信号  $E$  模糊化变成模糊量。  $E$  的模糊量可用相应的模糊语言来表示, 得到误差  $E$  的模糊语言的一个子集  $e$  ( $e$  是一个模糊矢量), 再由  $e$  和模糊控制规则  $R$  (模糊算子) 根据推理的合成规则进行模糊决策, 从而得到模糊控制量  $U$ :

$$U=e \cdot R$$

模糊控制系统与通常的控制系统的主要差别就是采用了模糊控制器。模糊控制器是模糊控制系统的核心, 一个模糊控制系统的性能优劣主要取决于模糊控制器的结构、所采用的规则、推理算法及模糊决策的方法等因素。

#### 4.2.2 制动器模糊控制器算法设计

本课题研究中，我们以常规 PID 控制为基础，采用模糊推理思想对 PID 参数进行在线自整定。所以本控制系统由常规 PID 控制与模糊推理两部分组成。系统框图如图 4.5。



4.5 制动器 PID 模糊控制器系统图

PID 参数模糊自整定是找出 PID 三个参数与  $e$  和  $ec$  之间的模糊关系，在运行中通过不断检测  $e$  和  $ec$ ，根据模糊控制原理来对三个参数进行在线修改，以满足不同  $e$  和  $ec$  时对控制参数的不同要求，而使被控制动器有良好的动、静态性能。

从系统的稳定性、响应速度、超调量和稳态精度等各方面来考虑， $K_p$ ， $K_i$ ， $K_d$  的作用如下：

比例系数  $K_p$  的作用是加快系统的响应速度，提高系统的调节精度。 $K_p$  越大，系统的响应速度越快、调节精度越高，但容易产生超调，甚至会导致系统不稳定。 $K_p$  取值过小又会降低调节精度使响应速度缓慢，从而延长调节时间，使系统静态、动态特性变坏。

积分作用系数  $K_i$  的作用是消除系统的稳态误差。 $K_i$  越大，系统的静态误差消除越快，但  $K_i$  过大会在响应过程的初期产生积分饱和现象，从而引起相应过程的较大超调。若  $K_i$  过小，会使系统静态误差难以消除，影响系统的调节精度。

微分作用系数  $K_d$  的作用是改善系统的动态特性，其主要作用是在响应过程中抑制偏差向任何方向变化，对偏差变化作提前预报。但  $K_d$  过大会使响应过程提前制动，从而延长制动调节时间，而且会降低系统的抗干扰性能。

第三章通过命令行建立 FIS 系统，本节通过 MATLAB 模糊逻辑工具箱的图形用户界面 (GUI) 建立模糊控制系统。这里主要有五个 GUI 工具：FIS 编辑器、

隶属度函数编辑器、规则编辑器、规则观测器和曲面观测器，它们之间是动态连接的，编辑器所作的修改能够及时地体现在观测器上。

#### (1) 确定控制器输入输出的语言变量

选择较多的词汇描述输入、输出变量，可以使制定的控制规则方便，但是控制规则相应变的复杂。选择词汇过少，使得描述变量变得粗糙，导致控制器的性能变坏。本论文中根据系统需要选择七个语言变量。

设：

|              |   |
|--------------|---|
| B=(Big)      | 大 |
| M=(Middle)   | 中 |
| S=(Small)    | 小 |
| P=(Positive) | 正 |
| N=(Negative) | 负 |
| ZO=(Zero)    | 零 |

则描述输入输出量的语言值模糊子集使用英文字头缩写选取为：

{NB, NM, NS, ZO, PS, PM, PB}

#### (2) 确定控制器输入输出的论域

本文把误差、误差变化的速率及控制输出的变化范围设定为 $[-5, +5]$ 区间，即论域范围取 $[-5, +5]$ ，构成含有 11 个整数元素的离散集合：

{-5, -4, -3, -2, -1, 0, +1, +2, +3, +4, +5}

从理论上讲，增加论域中的元素个数，可以提高控制精度，但同时会引起带来计算量增大，占用内存增多等不利因素。经验表明等级分得过细没有太大必要。

关于速度误差  $e$  的设定为：

论域  $E = \{-5, -4, -3, -2, -1, 0, +1, +2, +3, +4, +5\}$

关于系统偏差变化率  $ec$  的设定为：

论域  $EC = \{-5, -4, -3, -2, -1, 0, +1, +2, +3, +4, +5\}$

关于系统输出  $u$  的设定为：

论域  $U = \{-5, -4, -3, -2, -1, 0, +1, +2, +3, +4, +5\}$

#### (3) 确定控制器输入输出的隶属函数

本论文中控制器的输入输出采用梯形和三角形隶属函数。

#### (4) 建立模糊化表



本论文中的两个输入 E 和 EC 以及输出 U 的模糊化具有相同的形式，采用数值方法用表格描述隶属度如表 4.1 所示：

|    |     |    |     |    |     |   |     |    |     |    |     |
|----|-----|----|-----|----|-----|---|-----|----|-----|----|-----|
|    | -5  | -4 | -3  | -2 | -1  | 0 | +1  | +2 | +3  | +4 | +5  |
| NB | 0.5 | 0  | 0   | 0  | 0   | 0 | 0   | 0  | 0   | 0  | 0   |
| NM | 0.5 | 1  | 0.5 | 0  | 0   | 0 | 0   | 0  | 0   | 0  | 0   |
| NS | 0   | 0  | 0.5 | 1  | 0.5 | 0 | 0   | 0  | 0   | 0  | 0   |
| ZO | 0   | 0  | 0   | 0  | 0.5 | 1 | 0.5 | 0  | 0   | 0  | 0   |
| PS | 0   | 0  | 0   | 0  | 0   | 0 | 0.5 | 1  | 0.5 | 0  | 0   |
| PM | 0   | 0  | 0   | 0  | 0   | 0 | 0   | 0  | 0.5 | 1  | 0.5 |
| PB | 0   | 0  | 0   | 0  | 0   | 0 | 0   | 0  | 0   | 0  | 0.5 |

表 4.1 隶属函数表

(5) 建立模糊控制规则表

模糊控制器的最大特点就是只要有了有经验的操作者或专家给出的控制规则，就可以根据所选定的模糊推理方法获得合理的控制量。

模糊决策一般都用“选择从属度大”的规则。在制动过程中，当系统的偏差较大时，系统的快速性是主要矛盾，系统的稳定性和控制精度却是次要的，这时应使系统快速减小偏差；而当系统偏差较小时，则要求设计的控制器应保证系统的稳定性及控制精度。因而模糊控制规律应遵循：速度偏快，则输出频率的控制量应减小；速度偏慢，则频率的控制量应增加；速度变化率增大，则频率的控制量应增加；速度变化率减小，则频率的控制量应减小。

由此给出  $K_p$ 、 $K_i$ 、 $K_d$  的模糊控制规则表：

|              |    |    |    |    |    |    |    |    |
|--------------|----|----|----|----|----|----|----|----|
| $\Delta k_p$ | cc | NB | NM | NS | ZE | PS | PM | PB |
| e            |    |    |    |    |    |    |    |    |
| NB           |    | PB | PB | PM | PM | PS | ZE | ZE |
| NM           |    | PB | PR | 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 |

表 4.2  $K_p$  的模糊控制规则表

| $\Delta k_i$ / $e$ | NB | NM | NS | ZE | PS | PM | PB |
|--------------------|----|----|----|----|----|----|----|
| 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 4.3  $K_i$  的模糊控制规则表

| $\Delta k_d$ / $ec$ | NB | NM | NS | ZE | PS | PM | PB |
|---------------------|----|----|----|----|----|----|----|
| NB                  | PS | NS | NB | NB | NB | NM | PS |
| NM                  | PS | NS | NB | NM | NM | NS | ZE |
| NS                  | ZE | NS | NM | NM | NS | NS | ZE |
| ZE                  | ZE | NS | NS | NS | NS | NS | ZE |
| PS                  | ZE | ZE | ZE | ZE | ZE | ZE | ZE |
| PM                  | PB | PS | PS | PS | PS | PS | PB |
| PB                  | PB | PM | PM | PM | PS | PS | PB |

表 4.4  $K_d$  的模糊控制规则表

PID 控制作用通过下列算式来描述：

$$U(k) = k_p * E(k) + k_i * \sum_{i=0}^k E(i) + k_d * EC(k)$$

式中  $E(k)$ 、 $\sum_{i=0}^k E(i)$ 、 $EC(k)$  分别是系统偏差、偏差和偏差变化率；而  $K_p$ 、 $K_i$ 、

$K_d$  分别是比例系数、积分系数、微分系数，它们可以通过下面公式来计算。

$$K_p = K_p' + \{E_i, EC_i\}_p$$

$$K_i = K_i' + \{E_i, EC_i\}_i$$

$$K_d = K_d' + \{E_i, EC_i\}_d$$

其中  $K_p'$ 、 $K_i'$ 、 $K_d'$  是参数初始值，而  $\{E_i, EC_i\}_p$ 、 $\{E_i, EC_i\}_i$ 、 $\{E_i, EC_i\}_d$  则是参数的校正量。

### 4.3 模糊 PID 控制器的软件实现

#### 4.3.1 程序流程图

本系统程序如图 4.6:

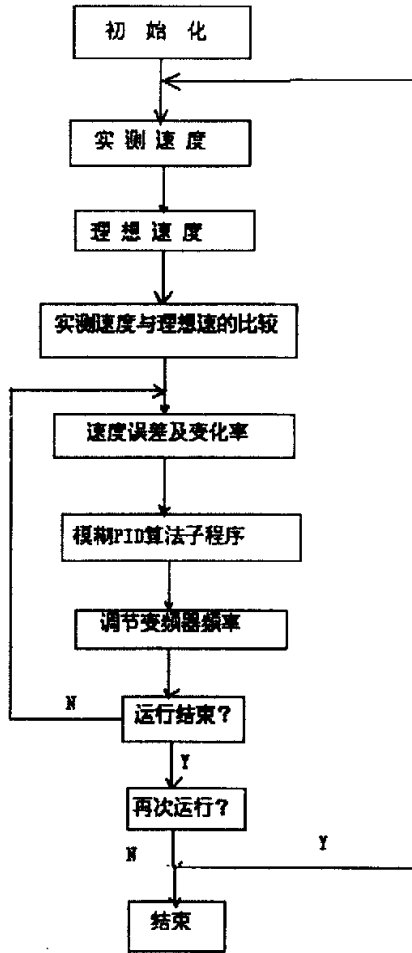


图 4.6 模糊 PID 控制系统程序流程图

#### 4.3.2 制动器模糊 PID 控制器的实现

模糊控制工具箱是进行模糊推理和模糊控制器仿真的工具包，它集成了 FIS 编辑器、隶属函数编辑器、模糊规则编辑器、规则浏览器和输出浏览器等可视化工具，使用户可以快速开发、设计模糊控制器。这些工具实质都是由开发者为 MATLAB 用户编写的 M 文件或 S 函数组成的，如 fuzzy.m 等等。这些以 .m 为后缀名的文件通常放在 Fuzzy 目录下，用户也可以在 MATLAB 编程环境中单独编程调

用这些文件或函数。本节利用 MATLAB 的模糊控制工具箱实现模糊控制器设计。

### (1) 定义输入输出变量

进入 MATLAB 编程环境后键入 fuzzy 即可打开 FIS 编辑器。本论文中的模糊控制器有两个输入、一个输出。使用 Edit 下拉菜单中的 Add input 命令添加输入变量，并且给输入、输出变量命名。

初步建立的模糊推理系统如图 4.7 所示。

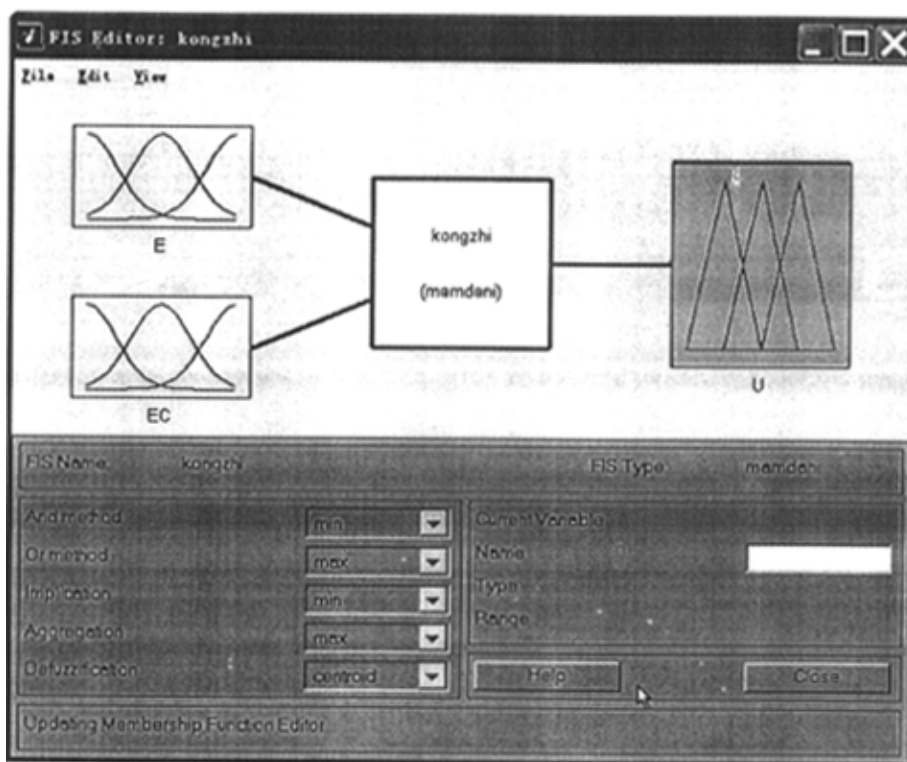


图 4.7 模糊控制工具箱编辑器主界面及初步的 FIS

### (2) 定义语言变量隶属函数

双击 FIS 编辑器中的输入变量或输出变量方框，就可以进入隶属函数编辑器主界面，对模糊变量的隶属函数进行编辑。选中要编辑变量的图标，确定当前变量量化等级的范围 (Range)；使用 Edit 下拉菜单中的 Add MFS 命令，打开确定变量隶属函数的窗口，如图 4.8 所示确定隶属函数的数量，即变量语言值模糊子集的大小（本论文中输入输出变量的隶属函数的数量为 7）。然后确定隶属函数的

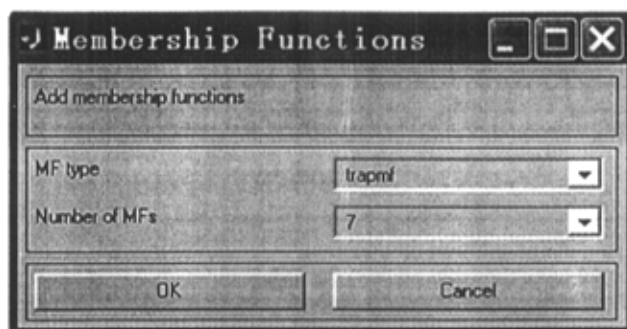


图 4.8 确定隶属函数窗口

类型，编辑器中提供了 8 种类型的隶属函数，本论文所选用隶属函数的类型为梯型和三角形。确定了变量的隶属函数以后可以得到如图 4.9 的隶属度函数编辑器。

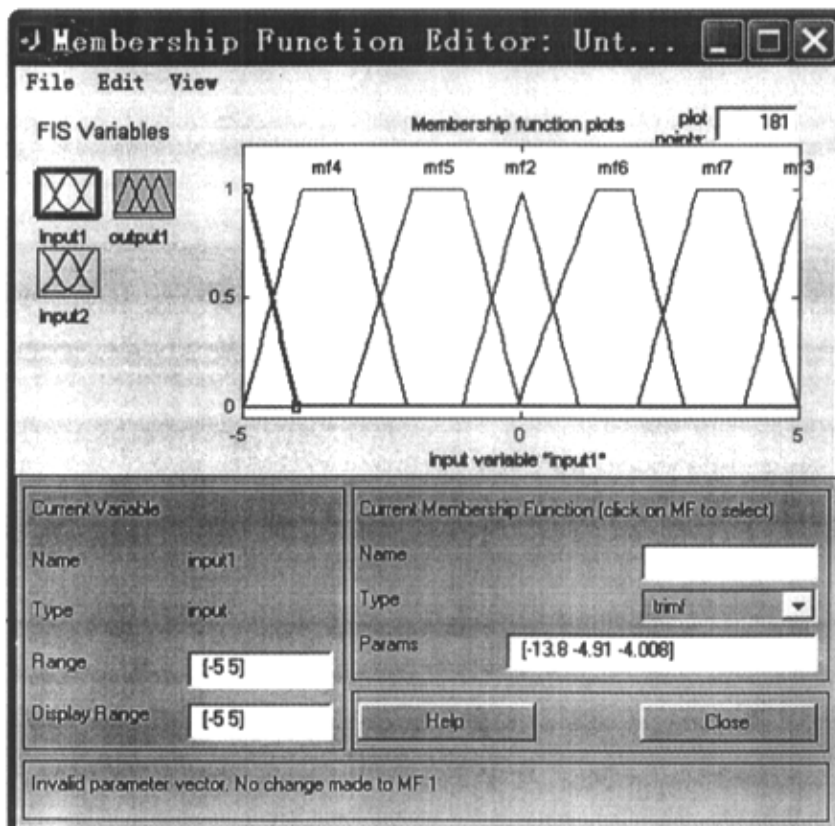


图 4.9 隶属函数编辑器

### (3) 定义模糊控制规则

双击 FIS 编辑器中流程图里的规则方框即可进入模糊规则编辑器主界面，其中模糊规则编辑框实际上是个文本编辑框。只要将前面得出的制动控制系统模糊控制规则表转换为 FIS 编辑器中的格式输入即可。完成后的规则编辑器如图 4.10 所示。

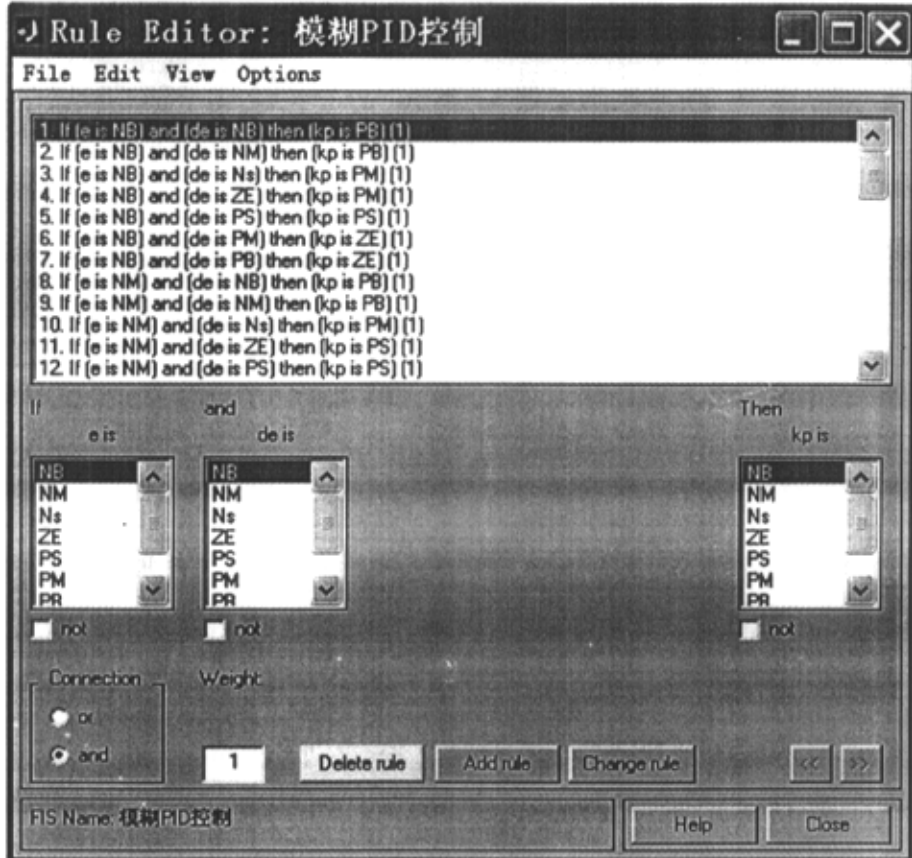


图 4.10 规则编辑器

因为两路输入均有 7 个语言值，所以共 49 条规则：

1. If (e is NB) and (de is NB) then (kp is PB) (1)
2. If (e is NB) and (de is NM) then (kp is PB) (1)
3. If (e is NB) and (de is Ns) then (kp is PM) (1)
4. If (e is NB) and (de is ZE) then (kp is PM) (1)
5. If (e is NB) and (de is PS) then (kp is PS) (1)
6. If (e is NB) and (de is PM) then (kp is ZE) (1)
7. If (e is NB) and (de is PB) then (kp is ZE) (1)

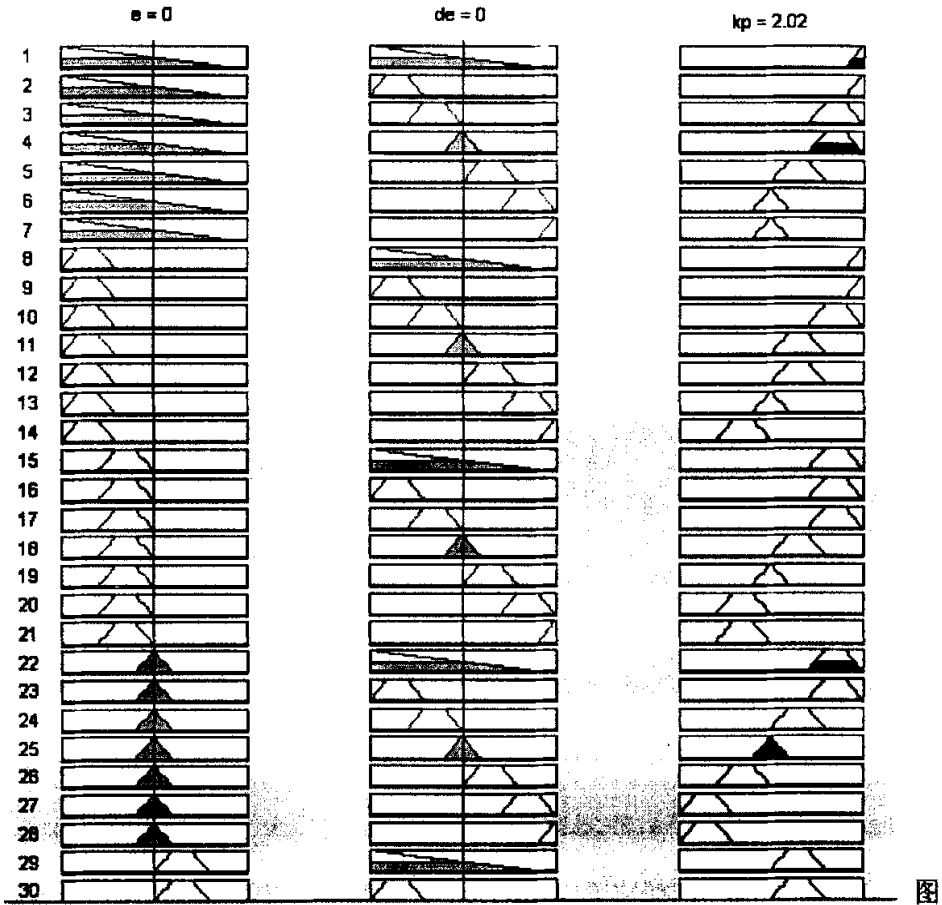


8. If (e is NM) and (de is NB) then (kp is PB) (1)
9. If (e is NM) and (de is NM) then (kp is PB) (1)
10. If (e is NM) and (de is Ns) then (kp is PM) (1)
11. If (e is NM) and (de is ZE) then (kp is PS) (1)
12. If (e is NM) and (de is PS) then (kp is PS) (1)
13. If (e is NM) and (de is PM) then (kp is ZE) (1)
14. If (e is NM) and (de is PB) then (kp is Ns) (1)
15. If (e is Ns) and (de is NB) then (kp is PM) (1)
16. If (e is Ns) and (de is NM) then (kp is PM) (1)
17. If (e is Ns) and (de is Ns) then (kp is PM) (1)
18. If (e is Ns) and (de is ZE) then (kp is PS) (1)
19. If (e is Ns) and (de is PS) then (kp is ZE) (1)
20. If (e is Ns) and (de is PM) then (kp is Ns) (1)
21. If (e is Ns) and (de is PB) then (kp is Ns) (1)
22. If (e is ZE) and (de is NB) then (kp is PM) (1)
23. If (e is ZE) and (de is NM) then (kp is PM) (1)
24. If (e is ZE) and (de is Ns) then (kp is PS) (1)
25. If (e is ZE) and (de is ZE) then (kp is ZE) (1)
26. If (e is ZE) and (de is PS) then (kp is Ns) (1)
27. If (e is ZE) and (de is PM) then (kp is NM) (1)
28. If (e is ZE) and (de is PB) then (kp is NM) (1)
29. If (e is PS) and (de is NB) then (kp is PS) (1)
30. If (e is PS) and (de is NM) then (kp is PS) (1)
31. If (e is PS) and (de is Ns) then (kp is ZE) (1)
32. If (e is PS) and (de is ZE) then (kp is Ns) (1)
33. If (e is PS) and (de is PS) then (kp is Ns) (1)
34. If (e is PS) and (de is PM) then (kp is NM) (1)
35. If (e is PS) and (de is PB) then (kp is NM) (1)
36. If (e is PM) and (de is NB) then (kp is PS) (1)

37. If (e is PM) and (de is NM) then (kp is ZE) (1)
38. If (e is PM) and (de is Ns) then (kp is Ns) (1)
39. If (e is PM) and (de is ZE) then (kp is NM) (1)
40. If (e is PM) and (de is PS) then (kp is NM) (1)
41. If (e is PM) and (de is PM) then (kp is NM) (1)
42. If (e is PM) and (de is PB) then (kp is NB) (1)
43. If (e is PB) and (de is NB) then (kp is ZE) (1)
44. If (e is PB) and (de is NM) then (kp is ZE) (1)
45. If (e is PB) and (de is Ns) then (kp is NM) (1)
46. If (e is PB) and (de is ZE) then (kp is NM) (1)
47. If (e is PB) and (de is PS) then (kp is NM) (1)
48. If (e is PB) and (de is PM) then (kp is NB) (1)
49. If (e is PB) and (de is PB) then (kp is NB) (1)

#### (4) 规则查看器

使用 FIS 编辑器等窗体 Edit 下拉菜单中的 View rules 命令, 可以打开规则查看器, 还可以针对模糊控制器的输入输出进行仿真检验。规则查看器用于显示各条模糊控制规则对应的输入量和输出量的隶属函数。通过指定输入量, 可以直观地显示所采用的控制规则, 以及通过模糊推理得到相应输出量的过程, 以便对模糊规则进行修改和优化。规则查看器如图 4.11 所示。



4.11 规则查看器

(5) 输出预览

当规则输入无误后,使用 FIS 编辑器等窗体 Edit 下拉菜单中的 View surface 命令,可以看到经模糊决策矩阵运算并解模糊化后得到的三维坐标图,如图 4.12 所示。

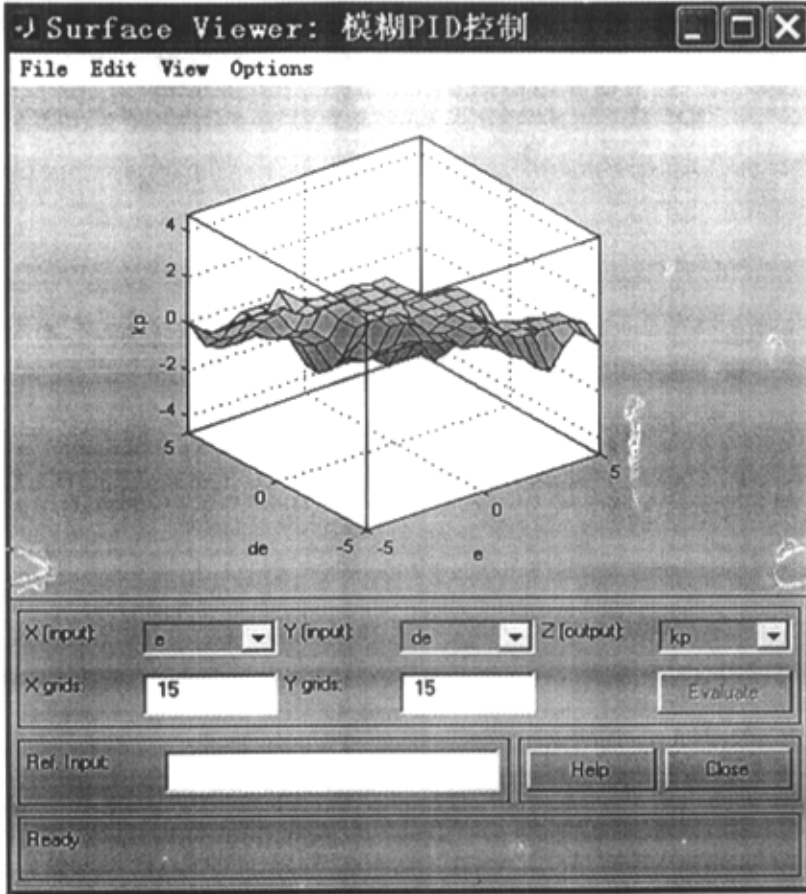


图 4.12 输出预览器

#### (6) 保存设计的模糊控制器

当模糊规则编辑完以后，将已设计好的模糊控制器模型存盘，文件名后缀为 .fis，可以随时调用该文件以修改模型。也可以使用 File 下拉菜单中的 save to workspace 命令，将计算结果以矩阵变量的形式存入内存缓冲区，从而可以在 Simulink 仿真平台上将该矩阵变量作为参数赋给一个封装的 fuzzy 模块，使设计好的模糊控制器作为模糊控制系统中的一个组成部分参与仿真过程。

## 4.4 制动器模糊控制仿真

### 4.4.1 仿真技术

所谓仿真(Simulation),就是模型实验,即通过对系统模型进行实验来研究一个存在的或设计中的系统。按照模型的建立方法,仿真可以分为三类:实物仿真、数学仿真和半实物仿真。

#### 1. 实物仿真

实物仿真又称为物理仿真,是以几何相似或物理相似为基础进行实验的仿真。实物仿真的典型实例就是对飞机或导弹进行风洞实验。这类方法往往具有模型建立复杂、花费多、周期长等缺点,在早期的仿真技术中应用较多。

#### 2. 数学仿真

数学仿真是以数学模型代替实际系统进行实验的仿真方法。广义上说,牛顿运动定律、欧几里德几何学都可以看成是对客观世界的仿真。这类仿真的关键在于数学模型的建立和求解,随着计算机技术的发展,数学仿真应用越来越广泛。

#### 3. 半实物仿真

半实物仿真是将物理模型、数学模型与实物相结合进行仿真的方法。很多系统往往不能准确地建立数学模型,这时就需要将实际系统模型放置在仿真系统进行仿真研究,这种方法可以获得较好的结果。

从20世纪40年代开始,人们将计算机引入到仿真技术中,随着科学技术、仿真理论及计算机的不断发展,仿真技术不断提高。仿真技术提高了科学研究水平,缩短了科学研究周期,将踢了科学研究成本及风险,促进了不同领域学科的融合。可以说仿真技术已成为科学研究中不可或缺的实用技术。

在大量计算机仿真软件中,MATLAB 是其中的优秀代表。MATLAB 现已被广泛应用于自动控制、信号处理、通信、数学、神经网络等许多不同学科的研究中。

### 4.4.2 MATLAB 模糊控制仿真

Simulink 是 MATLAB 下的一个软件包,是一个结合了框图界面和交互仿真能量的系统级设计和仿真工具。

在 MATLAB 的 Command Window 窗口中键入命令:

```
>>simulink
```

即可启动仿真模块库浏览器，如图 4.13 所示。

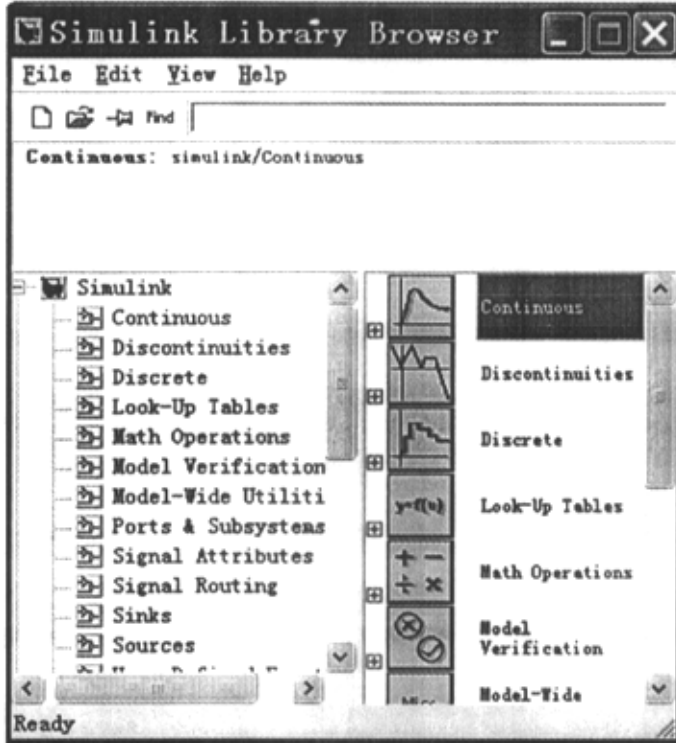


图 4.13 Simulink 模块库浏览器

Simulink 模块库浏览器主要用于浏览及选择模块，其模型搭建及仿真则在模型编辑窗口中完成。执行模块库浏览器的菜单“File”/“New”/“Model”命令，或采用快捷键[Ctrl+N]都可以打开模型编辑窗口。

仿真程序主要由五个部分组成：

- (1) 给定仿真初值；
- (2) 定义语言变量和隶属度函数；
- (3) 定义规则表矩阵；
- (4) 完成仿真计算；
- (5) 输出仿真图形。

本课题中将电力液压推杆制动器的电机等效为二阶系统，输入输出论域均选作[-5 5]，隶属度函数选用梯形和三角形两种来表示。



#### 4.4.2 仿真程序

具体仿真程序如下：

```

clear all;

close all;

%-----

%制动系统的传统 PID 控制与模糊 PID 控制的性能比较

%-----

%被控系统建模

num=20;

den=[10 5 1];

[a1,b1,c1,d]=tf2ss(num,den);

%系统参数

T=0.01;

h=T;

N=400;

%-----

%传统 PID 控制

%-----

e=0;

de=0;

ie=0;

kp=3;

ki=1.2;

kd=0.1;

x=[0;2];

for k=1:N

    %利用龙格-库塔法进行系统仿真

    u1=-(kp*e+ki*ie+kd*de);

    k0=a1*x+b1*u1;

    k1=a1*(x+h*k0/2)+b1*u1;

```

```

k2=a1*(x+h*k1/2)+b1*u1;
k3=a1*(x+h*k2)+b1*u1;
x=x+(k0+2*k1+2*k2+k3)*h/6;
y=c1*x+d*u1;
t(1,k)=k*T;
R(1,k)=(12-t(1,k))/3;
%计算误差、微分和积分
e1=e;
e=y-R(1,k);
de=(e-e1)/T;
ie=e*T+ie;
y1(1,k)=y;
end
%-----
%模糊 PID 控制
%-----
%定义输入、输出变量及隶属度函数
a=newfis('simple1');
%建立一个新的模糊系统
a=addvar(a,'input','e',[-5 5]);
%在 FIS 中添加输入变量
a=addmf(a,'input',1,'NB','trapmf',[-5,-5,-5,4]);
%在 FIS 中添加隶属度函数
a=addmf(a,'input',1,'NM','trapmf',[-5,-4,-3,-2]);
a=addmf(a,'input',1,'Ns','trapmf',[-3,-2,-1,0]);
a=addmf(a,'input',1,'ZE','trimf',[-1,0,1]);
a=addmf(a,'input',1,'PS','trapmf',[0,1,2,3]);
a=addmf(a,'input',1,'PM','trapmf',[2,3,4,5]);
a=addmf(a,'input',1,'PB','trapmf',[4,5,5,5]);
a=addvar(a,'input','de',[-5 5]);

```

```

a=addmf(a,'input',2,'NB','trapmf,[-5,-5,-5,4]);
a=addmf(a,'input',2,'NM','trapmf,[-5,-4,-3,-2]);
a=addmf(a,'input',2,'Ns','trapmf,[-3,-2,-1,0]);
a=addmf(a,'input',2,'ZE','trimf,[-1,0,1]);
a=addmf(a,'input',2,'PS','trapmf,[0,1,2,3]);
a=addmf(a,'input',2,'PM','trapmf,[2,3,4,5]);
a=addmf(a,'input',2,'PB','trapmf,[4,5,5,5]);
a=addvar(a,'output','kp',[-5 5]);
a=addmf(a,'output',1,'NB','trapmf,[-5,-5,-5,4]);
a=addmf(a,'output',1,'NM','trapmf,[-5,-4,-3,-2]);
a=addmf(a,'output',1,'Ns','trapmf,[-3,-2,-1,0]);
a=addmf(a,'output',1,'ZE','trimf,[-1,0,1]);
a=addmf(a,'output',1,'PS','trapmf,[0,1,2,3]);
a=addmf(a,'output',1,'PM','trapmf,[2,3,4,5]);
a=addmf(a,'output',1,'PB','trapmf,[4,5,5,5]);
%模糊规则矩阵
rulelist1=[1 1 7 1 1;
1 2 7 1 1;
1 3 6 1 1;
1 4 6 1 1;
1 5 5 1 1;
1 6 4 1 1;
1 7 4 1 1;
2 1 7 1 1;
2 2 7 1 1;
2 3 6 1 1;
2 4 5 1 1;
2 5 5 1 1;
2 6 4 1 1;
2 7 3 1 1;

```

31611;  
32611;  
33611;  
34511;  
35411;  
36311;  
37311;  
41611;  
42611;  
43511;  
44411;  
45311;  
46211;  
47211;  
51511;  
52511;  
53411;  
54311;  
55311;  
56211;  
57211;  
61511;  
62411;  
63311;  
64211;  
65211;  
66211;  
67111;  
71411;  
72411;

```

7 3 2 1 1;
7 4 2 1 1;
7 5 2 1 1;
7 6 1 1 1;
7 7 1 1 1];
a=addrule(a,rulelist1);
%在 FIS 中添加规则
b=newfis('simple2');
b=addvar(b,'input','e',[-5 5]);
b=addmf(b,'input',1,'NB','trapmf',[-5,-5,-5,4]);
b=addmf(b,'input',1,'NM','trapmf',[-5,-4,-3,-2]);
b=addmf(b,'input',1,'Ns','trapmf',[-3,-2,-1,0]);
b=addmf(b,'input',1,'ZE','trimf',[-1,0,1]);
b=addmf(b,'input',1,'PS','trapmf',[0,1,2,3]);
b=addmf(b,'input',1,'PM','trapmf',[2,3,4,5]);
b=addmf(b,'input',1,'PB','trapmf',[4,5,5,5]);
b=addvar(b,'input','de',[-5 5]);
b=addmf(b,'input',2,'NB','trapmf',[-5,-5,-5,4]);
b=addmf(b,'input',2,'NM','trapmf',[-5,-4,-3,-2]);
b=addmf(b,'input',2,'Ns','trapmf',[-3,-2,-1,0]);
b=addmf(b,'input',2,'ZE','trimf',[-1,0,1]);
b=addmf(b,'input',2,'PS','trapmf',[0,1,2,3]);
b=addmf(b,'input',2,'PM','trapmf',[2,3,4,5]);
b=addmf(b,'input',2,'PB','trapmf',[4,5,5,5]);
b=addvar(b,'output','ki',[-5 5]);
b=addmf(b,'output',1,'NB','trapmf',[-5,-5,-5,4]);
b=addmf(b,'output',1,'NM','trapmf',[-5,-4,-3,-2]);
b=addmf(b,'output',1,'Ns','trapmf',[-3,-2,-1,0]);
b=addmf(b,'output',1,'ZE','trimf',[-1,0,1]);
b=addmf(b,'output',1,'PS','trapmf',[0,1,2,3]);

```

```
b=addmf(b,'output',1,'PM','trapmf',[2,3,4,5]);
```

```
b=addmf(b,'output',1,'PB','trapmf',[4,5,5,5]);
```

```
rulelist2=[1 1 1 1 1;
```

```
1 2 1 1 1;
```

```
1 3 2 1 1;
```

```
1 4 2 1 1;
```

```
1 5 3 1 1;
```

```
1 6 4 1 1;
```

```
1 7 4 1 1;
```

```
2 1 1 1 1;
```

```
2 2 1 1 1;
```

```
2 3 2 1 1;
```

```
2 4 3 1 1;
```

```
2 5 3 1 1;
```

```
2 6 4 1 1;
```

```
2 7 4 1 1;
```

```
3 1 1 1 1;
```

```
3 2 2 1 1;
```

```
3 3 3 1 1;
```

```
3 4 3 1 1;
```

```
3 5 4 1 1;
```

```
3 6 5 1 1;
```

```
3 7 5 1 1;
```

```
4 1 2 1 1;
```

```
4 2 2 1 1;
```

```
4 3 3 1 1;
```

```
4 4 4 1 1;
```

```
4 5 5 1 1;
```

```
4 6 6 1 1;
```

```
4 7 6 1 1;
```

```
5 1 2 1 1;  
5 2 3 1 1;  
5 3 4 1 1;  
5 4 5 1 1;  
5 5 5 1 1;  
5 6 6 1 1;  
5 7 7 1 1;  
6 1 4 1 1;  
6 2 4 1 1;  
6 3 5 1 1;  
6 4 5 1 1;  
6 5 6 1 1;  
6 6 7 1 1;  
6 7 7 1 1;  
7 1 4 1 1;  
7 2 4 1 1;  
7 3 5 1 1;  
7 4 6 1 1;  
7 5 6 1 1;  
7 6 7 1 1;  
7 7 7 1 1];  
b=addrule(b,rulelist2);  
c=newfis('simple3');  
c=addvar(c,'input','e',[-5 5]);  
c=addmf(c,'input',1,'NB','trapmf',[-5,-5,-5,4]);  
c=addmf(c,'input',1,'NM','trapmf',[-5,-4,-3,-2]);  
c=addmf(c,'input',1,'Ns','trapmf',[-3,-2,-1,0]);  
c=addmf(c,'input',1,'ZE','trimf',[-1,0,1]);  
c=addmf(c,'input',1,'PS','trapmf',[0,1,2,3]);  
c=addmf(c,'input',1,'PM','trapmf',[2,3,4,5]);
```



```
c=addmf(c,'input',1,'PB','trapmf',[4,5,5,5]);
c=addvar(c,'input','de',[-5 5]);
c=addmf(c,'input',2,'NB','trapmf',[-5,-5,-5,4]);
c=addmf(c,'input',2,'NM','trapmf',[-5,-4,-3,-2]);
c=addmf(c,'input',2,'Ns','trapmf',[-3,-2,-1,0]);
c=addmf(c,'input',2,'ZE','trimf',[-1,0,1]);
c=addmf(c,'input',2,'PS','trapmf',[0,1,2,3]);
c=addmf(c,'input',2,'PM','trapmf',[2,3,4,5]);
c=addmf(c,'input',2,'PB','trapmf',[4,5,5,5]);
c=addvar(c,'output','kd',[-5 5]);
c=addmf(c,'output',1,'NB','trapmf',[-5,-5,-5,4]);
c=addmf(c,'output',1,'NM','trapmf',[-5,-4,-3,-2]);
c=addmf(c,'output',1,'Ns','trapmf',[-3,-2,-1,0]);
c=addmf(c,'output',1,'ZE','trimf',[-1,0,1]);
c=addmf(c,'output',1,'PS','trapmf',[0,1,2,3]);
c=addmf(c,'output',1,'PM','trapmf',[2,3,4,5]);
c=addmf(c,'output',1,'PB','trapmf',[4,5,5,5]);
rulelist3=[1 1 5 1 1;
1 2 3 1 1;
1 3 1 1 1;
1 4 1 1 1;
1 5 1 1 1;
1 6 2 1 1;
1 7 5 1 1;
2 1 5 1 1;
2 2 3 1 1;
2 3 1 1 1;
2 4 2 1 1;
2 5 2 1 1;
2 6 3 1 1;
```

27411;

31411;

32311;

33211;

34211;

35311;

36311;

37411;

41411;

42311;

43311;

44311;

45311;

46311;

47411;

51411;

52411;

53411;

54411;

55411;

56411;

57411;

61711

62511;

63511;

64511;

65511;

66511;

67711;

71711;

```
7 2 6 1 1;
7 3 6 1 1;
7 4 6 1 1;
7 5 5 1 1;
7 6 5 1 1;
7 7 7 1 1];
c=addrule(c,rulelist3);
%模糊 PID 控制系统仿真
e=0;
de=0;
ie=0;
x=[0;2];
ke=1/20;
kec=1/10;
kup=6;
kui=1.2;
kud=0.1;
for k=1:N
    e1=ke*e;
    del=kec*de;
    if e1>=5
        e1=5;
    elseif e1<=-5
        e1=-5;
    end
    if del>=5
        del=5;
    elseif del<=-5
        del=-5;
    end
end
```

```

%模糊推理，计算被控对象的控制输入
in=[e1 de1];
u1=evalfis(in,a);
u2=evalfis(in,b);
u3=evalfis(in,c);
kp=-u1/2+kup;
ki=-u2/1+kui;
kd=-u3/2+kud;
u=-(kp*e+ki*ie+kd*de);
%计算系统输出
k0=a1*x+b1*u;
k1=a1*(x+h*k0/2)+b1*u;
k2=a1*(x+h*k1/2)+b1*u;
k3=a1*(x+h*k2)+b1*u;
x=x+(k0+2*k1+2*k2+k3)*h/6;
y=c1*x+d*u;
y2(1,k)=y;
t(1,k)=k*T;
R(1,k)=(12-t(1,k))/3;
e1=e;
e=y-R(1,k);
de=(e-e1)/T;
ie=ie+T*e;
end
%绘制输出响应曲线
kk=[1:N]*T;
figure(1);
plot(kk,R,'k',kk,y1(1,:),'k',kk,y2(1,:),'k-');
title('输出响应曲线');

```

```
xlabel('时间');  
ylabel('速度');  
legend('理想曲线','PID 控制曲线','模糊控制曲线')
```

#### 4.4.3 输出响应曲线

运行仿真程序得到的输出曲线如图 4.14:

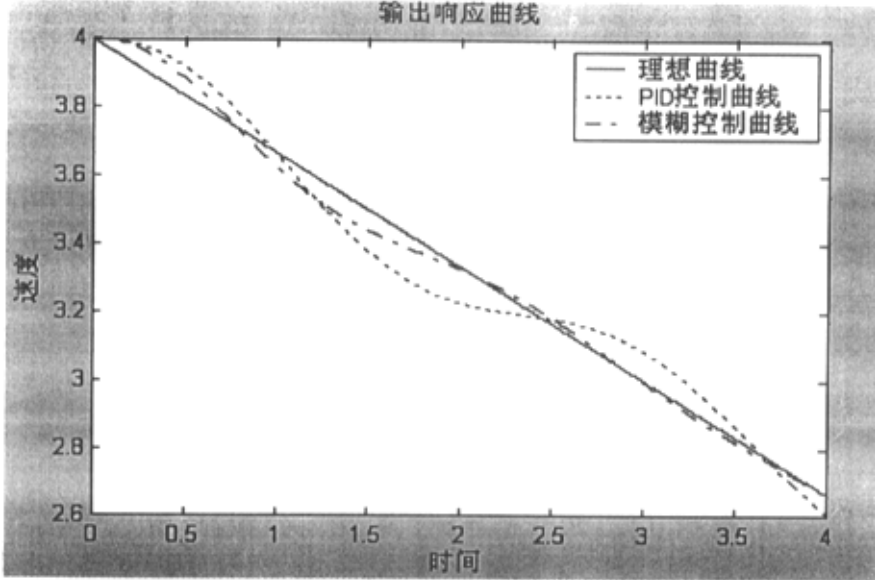


图 4.14 输出响应曲线

由上图可以看出,将模糊控制与PID控制相结合应用到电力液压推杆制动器的调速系统中,与传统的闭环PID控制相比,大大的提高了系统适应外部干扰和内部参数变化的鲁棒性,减小了超调,改善了动态特性。而与简单模糊控制相比,它又减小了稳态误差,提高了系统稳定性。

## 第五章 总结

本课题出于电力液压推杆制动器制动简单、不能适应复杂工况的要求，对制动器的控制方式进行研究，包括系统的组成及模糊控制器的设计。我们通过变频调速的制动方式，结合模糊控制的思想，实现了不同输入条件下的输出智能控制，并将传统的 PID 控制与模糊 PID 控制器相比较，进行动态仿真，得到了较满意的结果。主要来说，本课题完成了下列任务：

(1) 分析了当前的各种制动控制方案，针对电力液压推杆制动器确定了以模糊智能控制技术和变频调速技术相结合进行制动的控制方案；

(2) 以龙门起重机大车运行机构为研究对象，结合电力液压推杆制动器的特性建立智能制动控制器的开环系统，研究模糊控制算法、控制参数和决策准则。

(3) 在电力液压推杆制动器的基础上，将模糊控制与 PID 控制相结合，构建制动器的闭环控制系统，并与传统 PID 控制进行比较，同时进行仿真。

通过本文的介绍，可以发现模糊控制容易构造，相对简单，形象直观，适应性好，可以根据不同的被控对象修改参数，具有效率高、可靠性高、成本低、投资回收期短等优点。本文所作的工作拓宽了模糊控制的应用范围，也为控制策略的进一步提升奠定了基础，同时本文的工作对其他控制领域的研究也有一些启示。

虽然本课题对电力液压推杆制动器的研究取得了一些进展，但仍然存在一些问题：智能控制方法在不断发展，本课题只选用了模糊控制，而专家智能控制、神经网络控制及模糊神经网络控制有可能也具有不错的制动效果。今后的控制算法可能是多种智能控制技术的结合。

## 参考文献

- (01)杨长骅，起重机械，机械工业出版社，1982
- (02)王涛，摩擦制动器，华南理工大学出版社，1992.14~124
- (03)于岩，运输机械设计，中国矿业大学出版社，1998
- (04)张质文、虞和谦、王金诺、包起帆，起重机设计手册，中国铁道出版社，1998
- (05)董炜，脚踏常开式制动器在起重机上的应用，起重运输机械，1999（2）：15-16
- (06)马素平、寇子明，带式输送机液压能耗制动系统得智能化，煤矿机械，2000，（5）：35-37
- (07)刘武胜、路建湖，制动器的国内外现状及发展趋势，港口装卸，2000 年增刊：22-30
- (08)赵正彩，车轮盘式制动器的使用与分析，重型汽车，2000（05）：17-21
- (09)包寿红，离心机制动控制电路设计，电子与自动化，2000（06）：38-39
- (10)关源勇，电力液压推动器结构与性能比较，起重运输机械，2000（03）：29-31
- (11)戴建立、史志、王志信，下运带式输送机液力制动闭环控制的研究，煤矿机械，2001，（10）：22-23
- (12)周满山、于岩、包继华、吴思波，推杆制动器在小功率下运带式输送机上的应用  
起重运输机械，2001（12）：25-27
- (13)李晓明，电气制动的研究，山西电力，2001（05）：7-12
- (14)孙炳达，基于最优模式智能控制自适应交流调速系统的研究，电器传动，2001（06）：16-19
- (15)傅岳林，微机控制在提升机恒减速制动系统中的应用，矿山机械，2001（05）：42-44
- (16)寇子明、李军霞，液压调速制动系统的研究，湘潭矿业学院学报，2002（04）：27-30
- (17)徐小增、廖庆华、秦忆，智能控制方法的现状及发展，佛山科学技术学院学报（自然科学版），2002（04）：14-17



- (18)柳青, 异步电动机的软启动与软制动技术, 湖南工业职业技术学院学报, 2002 (02): 12-14
- (19)崔根伟, 一种新型下运胶带输送及液压制动系统, 机械, 2002 (29 卷 04 期):13-17
- (20)聂春华、陈胜根、付兰梅、曾志勇, 全自动免维护制动装置, 港口装卸, 2002 (06): 35-37
- (21)刘义伦、顾跃、梁明智, 基于 PLC 的变频调速恒压供水自动控制技术的应用, 广西机械, 2003 (02):37-39
- (22)赵明薇, 变频调速电梯系统的驱动和控制技术, 设备管理与维修, 2003 (05): 24-25
- (23)杜金城, 电气变频调速设计技术, 中国电力出版社, 1999
- (24)电机工程手册编辑委员会, 电机工程师手册, 机械工业出版社, 1987
- (25)丛爽, 神经网络、模糊系统及其在运动控制中的应用, 中国科学技术大学出版社, 2001
- (26)钟麟、王峰, MATLAB 仿真技术与应用教程, 国防工业出版社, 2004
- (27)冯冬青、谢宋和等, 模糊智能控制, 化学工业出版社, 1998
- (28)郑宏兴、姚纪欢、张成等, MATLAB 5.X 工具箱使用技巧与实例, 华中科技大学出版社, 2000
- (29)Lee,Ki-Chang;Jeon,Jeong-Woo;Hwang,Don-Ha;Kim,Yong-Joo Performance Evaluation of Antilock Brake Controller for Pneumatic Brake System Conference Record-IAS Annual Meeting(IEEE Industry Applications Society),v1,2003,p301-307
- (30)Lee,Tsu-Tian;Hsu,Chun-Fei;Lee,San Robust hybrid control for antilock braking systems Proceedings of the IEEE International Conference on Systems,Man and Cybernetics,v1,2003,p84-89
- (31)Podrigalo,M.A.;Volkov,V.P;Klimenko,V.I.;Efimchuk,V.M. Prediction of requirements to brake control of motor transport facilities Avtomobil'naya Promyshlennost,n,5,2003,p35-36
- (32)Svensson,Bo Brke Systems Designed for Marine Applictions Diesel and

Gas Turbine Worldwide,v35,n7,September,2003,p34

( 33 ) Massimo Losa Arrester Beds,Soil-Wheel Interaction and Braking Performance Article of Routers/Roads issue number 303 (July 1999)

( 34 ) Varuzan Kevorkijan Steel Surfaced Aluminum Brake Discs Dischi AL rivestiti in acciaio technologies MARZO 2003 ,pp.115-119

( 35 ) ZOU En;LIN Yi-qin;ZHANG Tai-shan Intelligence control and its Application Journal of Zhuzhou Institute of Technology sept.2003(Vol.17 No.5) ,pp.75-78

( 36 ) Marwitz,H.;Fischer,J.(1998) Braking Systems on Heavy Duty Trucks,Truck &Commercial Vehicle International 1998,pp.13-14

( 37 ) Kaneko,Tetsuya;Kageyama,Ichiro;Tsunshima,Hitoshi Braking stability of articulated vehicles on highway Vehicle System Dynamics ,n SUPPL,2003,p1-11

( 38 ) Dufrenoy,p.;Weichert,D. A thermomechanical model for the analysis of disc brake fracture mechanisms Journal of Thermal Stresses ,v26,n8,August,2003,p815-828

### 读研期间发表论文情况

- (1) 申伟, 张元培. 基于 MATLAB 的自适应神经网络模糊系统的应用. 制造业自动化, 2004 (增)