

## 摘要

基于 DSP (Digital Signal Processor) 的运动控制技术的突破, 为开放式数控系统的发展提供了新的机遇。新一代开放式数控系统将以基于 DSP 的运动控制器为核心, 融合 PLC 的功能, 与通用或工控机集成而成。本文针对教研室 PTC 系列活塞数控机床的发展要求探讨了采用 GT400-SV 运动控制器进行活塞加工控制的可行性。

介绍了课题的背景及国内外的动态。对电机进行理论建模, 并在此基础上对其进行仿真分析, 以对控制系统有较深入的认识。编写了控制系统各部分的测试程序, 对闭环控制系统采用了正弦信号控制与活塞数据控制两种实验方法, 为开发 Windows 控制程序提供了借鉴。

实验表明, 运动控制器所提供的控制方法不能满足活塞加工运动的要求。因此课题在最后提出对运动控制器和控制程序进行改进的方法, 并进行相应的实验, 实验结果表明该方法是切实可行的。课题为该系统的前期研究, 要将运动控制器应用于活塞加工中, 还需不懈的努力。

**关键词:** 运动控制器、活塞、直线电机、中断

## ABSTRACT

The breakthrough of the motion control technology provides an opportunity for the development of the opening CNC system. The new generation of opening CNC system takes the motion controller based on DSP for its core, including the function of PLC, and integrates with IPC. Aiming at the development of the numerical controlled piston machining system developed by our lab, this article discusses the possibility of the appliance of GT400-SV MC to piston machining control.

Introduces the background of the task and the domestic and foreign trends of development. A theory module is built on the motor, and simulation analyses are also executed. It made us understand the control system more clearly. Writes programs for testing all parts of the control system, carries through the experiments with sine and piston data on the control system, it will be helpful for developing the Windows control programs.

The experiments shows that the control methods provided by the MC do not satisfy the need of piston machining. Therefore, at the end of the task an improved method for the system is put forward, and corresponding experiments are carried out to verify it. The task is the prophase of the research, more work should be done to make the MC used in piston machining.

Keywords: motion controller, piston, linear motor, interrupt

# 独创性声明

本人声明所呈交的学位论文是我本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表和撰写过的研究成果，也不包含为获得国防科学技术大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文题目：基于运动控制器的活塞外圆加工技术

学位论文作者签名：于世江 日期：2002年11月14日

## 学位论文版权使用授权书

本人完全了解国防科学技术大学有关保留、使用学位论文的规定。本人授权国防科学技术大学可以保留并向国家有关部门或机构送交论文的复印件和电子文档，允许论文被查阅和借阅；可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

(保密学位论文在解密后适用本授权书。)

学位论文题目：基于运动控制器的活塞外圆加工技术

学位论文作者签名：于世江 日期：2002年11月14日

作者指导教师签名：冯正毅 日期：2002年11月14日

## 第一章 绪 论

### § 1.1 课题概述

#### 1.1.1 课题来源

本课题主要来源于我室研制的 PTC 活塞数控车削系统。该系统主要用来车削中凸变椭圆铝合金活塞外圆。它的最大特点是用控制计算机中的“软靠模”（即活塞数据文件）代替传统的凸轮、模板和立体靠模，因此具有很多突出的优点。本课题就是在原有系统的软硬件基础上研究以运动控制器为核心，基于 Windows 环境的控制系统，为研制车削加工系统做好技术上的准备。

本课题题目是“基于运动控制器的活塞外圆加工技术”，本课题是研究室自主开发项目。

#### 1.1.2 课题背景

众所周知，一些轴类机械零件，因其性能的要求，其横截面经常被设计成非圆的截面，常见的有活塞、凸轮等。其中活塞的外表又常为中凸变椭圆。这样的非圆截面或称异形截面用普通的数控车床是无法加工的。为此，教研室的前身之一的原八系 804 教研室，早在九十年代初便开始研制开发 PTC 活塞数控车削系统，并经过市场化的运作后，已成为我国活塞外圆车削加工中的主流设备，创出了产学研一体化的样板。该系统已经从刚开始的 PTC-1 升级到 PTC-4<sup>[1]</sup>。其主要的技术参数，如最大工件型线及最大椭圆度、最大工件型线误差、最大工件椭圆度误差等均达到了国内领先水平。

基于 DSP (Digital Signal Processor) 的运动控制技术的突破，为开放式 CNC 系统的发展提供了新的机遇。新一代开放式 CNC 系统将以基于 DSP 的运动控制器为核心，融合 PLC 的功能，与通用或工控 PC 集成而成<sup>[2]</sup>。另外，随着 Windows95/98、WindowsNT 操作系统的出现，32 位 Windows 应用程序漂亮的用户界面 (GUI)、多方的多媒体服务、几乎无限制的内存资源、多任务并发执行机制等优点是显而易见的，开发基于 Windows 的数控软件已经成为必然趋势。

为了适应市场的需求,使产品更具竞争力,本课题主要探讨 GT400-SV 运动控制器在活塞外圆车削系统中应用的可行性,并将控制系统的软件由 DOS 转移到 Windows 环境中。通过研究为未来研制非圆截面精密车削加工系统做好技术上的准备。

### 1.1.3 课题目的及意义

本课题的目的在于为特种数控的进一步发开开辟新的道路,在前人的基础上,开发以基于 DSP 的运动控制器为核心,与通用或工控 PC 集成,运行于 Windows 的新一代非圆截面车削系统。

意义就在于可以将 Windows 强大的图形用户接口,多任务能力以及强大的硬件和软件兼容能力与基于 DSP 的运动控制器无与伦比的伺服、插补和实时计算能力结合在一起,为用户提供具有高效低成本的现代化控制。能够始终把目光瞄准国内乃至世界在这一领域前沿,为我们的产品抢占市场提供有力的技术支持和保障。

## § 1.2 文献综述

### 1.2.1 基于 DSP 的运动控制器的优点

运动控制器一般采用高速多功能的或专用微处理器及信号处理器,如 DSP (Digital Signal Processor)。DSP 的高速化给伺服系统的控制提供了较大的灵活性,例如 ADSP2181 的指令周期达到 33ns,而且执行 ADSP2181 的任何一条指令都仅需要 33ns,如果将控制系统的采样频率定为 100kHz,按 33ns 的指令周期计算,在一个控制周期内循环程序的长度最多可以容纳 300 条指令,这就给复杂控制的实施提供了有利条件。

采用基于 DSP 的运动控制器加 PC 构成的新一代开放式数控系统的优点是多方面的。首先,系统集成商或使用者可以从自己选择的供货商那里购买硬件和软件加进自己的系统中,使系统更加耐用和灵活,而且成本低廉。用户再也不必依靠单一的供货商,不必支付高昂的费用而得到一个用途受到很大限制而且更新周期长的产品;其次,新一代开放式数控系统可以将 Windows 强大的图形用户接口,多任务能力以及强大的硬件和软件兼容能力与基于 DSP 的运动控制器无与伦比的伺服、插补和实时计算能力结合在一起,为用户提供具有高效低成本的现代化控制;第三,以基于 DSP 的运动控制器为核心的新一代开放式 CNC 不仅融合了传统 CNC 的功能,而且能提供许多传统 CNC 系统所不能提供的功能。总之,现代制造业正在追求一种理念,那就是设备的智能化,但要求设备的操作越来越简单,基

于 DSP 的运动控制技术正是适应了这一潮流。

### 1.2.2 国内外研究现状和发展趋势

基于 PC 的开放式 CNC 系统结合了 PC 的分析运算能力,大容量存储功能,图文显示优势及组网的灵活性,使得数控系统的使用具备了较为开放的模式。但由于 PC 计算能力的限制,阻遏了这类数控系统朝高速和高精度方向发展的步伐。

80 年代末基于 DSP(Digital Signal Processor)的运动控制技术的突破,为开放式 CNC 系统的发展提供了新的机遇。新一代开放式 CNC 系统将以基于 DSP 的运动控制器为核心,融合 PLC 的功能,与通用或工控 PC 集成而成<sup>[2]</sup>。

目前,世界上主要的运动控制产品供应商均为国外公司,如美国的 Delta Tau、Galil 公司和德国的 DSpace 公司等,他们的产品在中国也已经有了一定规模的市场。但是由于价格昂贵、体系庞大导致编程困难以及缺乏有力的技术支持等原因,这些产品要想进一步在中国推广使用却面临着较大的困难。近几年国内在运动控制器的技术上取得了较大的进展,从经济型 2 轴和 4 轴运动控制器、通用性 2 轴和 4 轴运动控制器到网络性运动控制器和精密型 4 轴和 8 轴运动控制器均能独立设计和制造。如深圳的固高科技公司及摩信公司等将为我国数控系统集成商提供高性能价格比的基于 DSP 的运动控制器产品,为他们开发具有自主知识产权的高速、高性能新一代开放式 CNC 系统打下坚实的基础

### 1.2.3 控制系统相关技术的发展

随着电子技术的突飞猛进的发展,高速多功能的或专用的微处理器大量的涌现,如 DSP 数字信号处理器近年来发展很快,ADSP2181 的指令周期达到了 33ns,而且执行 ADSP2181 的任何一条指令都仅需要 33ns,这是以前决不可想象的事情<sup>[3]</sup>。DSP 的高速化给伺服系统的控制提供了较大的灵活性,如将控制系统的采样频率定为 100KHz,按 33ns 的指令周期计算,在一个控制周期内循环程序的长度最多可容纳 300 条指令,这样就给复杂控制的实施提供了有利条件。由它和工控机所构成的二级控制系统,即上下位机的控制方法,它可以把工控机从复杂的计算中解脱出来,把大量的工作交给 DSP 去做,这样就使得 WINDOWS 环境下的实时控制成为可能。

硬件水平不断提高的同时,各种控制策略和控制理论也不断地发展应用。在直线交流伺服中所成功应用的控制策略大致如下:

### (1) 传统的控制策略

传统的控制策略如 PID 反馈控制、解耦控制等。其中 PID 控制算法蕴涵了动态控制过程中的过去、现在和将来的信息，而且其配置几乎为最优，具有较强的鲁棒性，是交流伺服电机驱动系统中最基本的控制形式，其应用广泛，并与其它新型控制思想结合，形成了许多有价值的控制策略。Smith 预估计器与控制器并联，可以使控制对象的时间滞后得到完全的补偿。在直线永磁交流伺服电动机系统中存在着多个电磁变量和机械变量，在这些变量之间存在较强的耦合作用，这种情况下，可以采用解耦控制算法加以解决，使各变量间的耦合减小到最低限度，以使各变量都能得到单独的控制。

### (2) 现代控制策略

在对象模型确定、不变化且为线性，以及操作条件、运行环境确定不变的条件下，采取传统控制策略是简单有效的。但在高精微进给的高性能场合，就必须考虑对象的结构与参数变化、各种非线性的影响、运行环境的改变以及环境干扰等时变和不确定因素，才能得到满意的控制效果。因此现代控制策略在直线伺服电动机的研究中引起了很大的重视。在这些现代控制策略中较为典型的如自适应控制，它主要有模型参考自适应控制和自校正控制两种类型。它们都根据对象的变化而对控制参数作出相应的调整。还有比如变结构控制、鲁棒控制、预见控制等，它们都有成功地应用实例。

### (3) 智能控制策略

对控制对象、环境与任务复杂的系统宜采用智能控制方法。模糊逻辑控制、神经网络和专家系统控制是当前三种比较典型的智能控制策略。其中模糊控制器专用芯片已商品化，因其实时性好、控制精度高，在伺服系统中已有应用。神经网络从理论上讲具有很强的信息综合能力，但目前缺乏相应的神经网络计算机的硬件支持，其在直线伺服中的应用有待于神经网络集成电路芯片生产的成熟。

## § 1.3 课题研究的主要内容

本课题在研究室原有的 MF 型直线电机闭环控制系统的基础上接入 Y0D71-2/4 型三相异步电动机来模拟主轴的运动，从而通过 GT400-SV 运动控制器来实现两轴联动的活塞加工过程。

本课题包括以下两方面的内容：

## 1 控制系统硬件的设计与调试

控制系统硬件主要由工控机、GT400—SV 运动控制器、光栅尺、功放、三相异步电动机、编码器等组成。在对硬件的调试过程中，还需要编写软件对每部分的功能进行测试，确保系统硬件正确的连接。

## 2 两轴联动控制系统软件

软件主要是在 C++builder5.0 下编写的实时控制软件及在 Matlab 编写的仿真程序。实时控制软件主要包括采用正弦信号进行控制和采用活塞数据进行控制两部分。

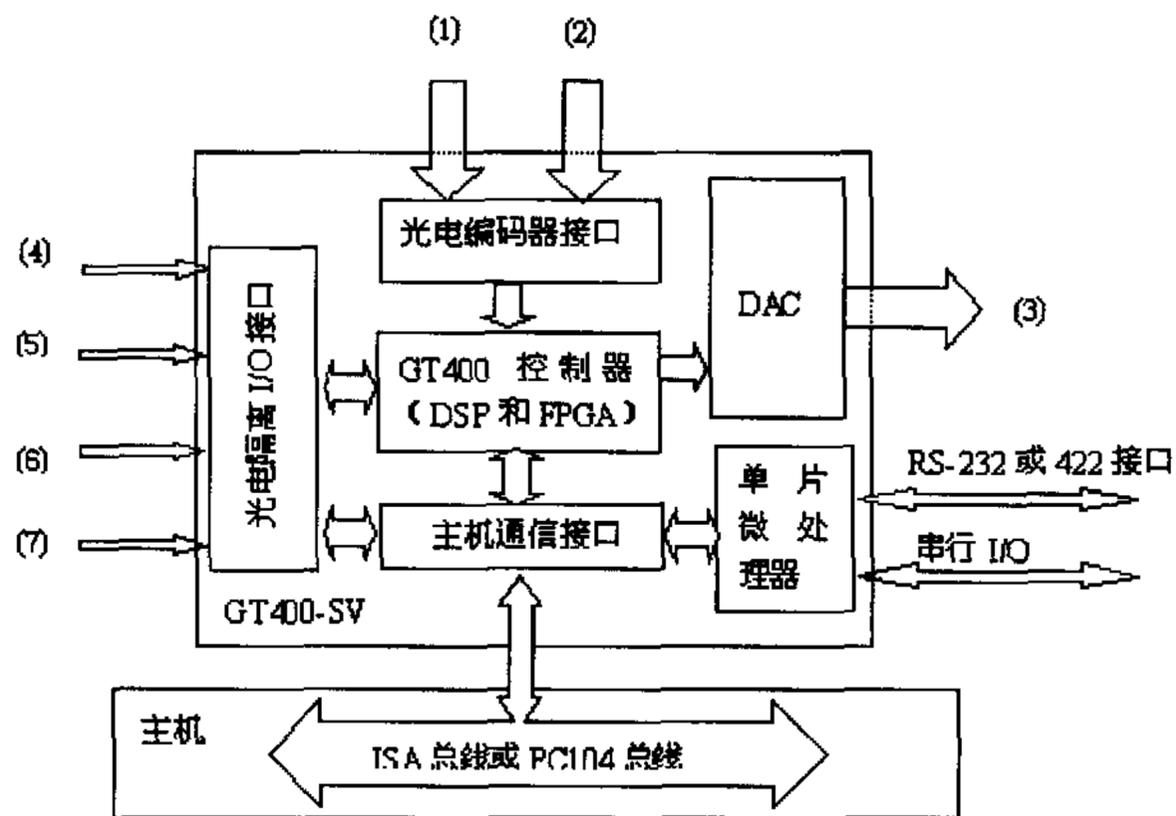
本课题用运动控制器代替原来的 D/A 及 I/O 板实现活塞加工过程的控制，分析其可行性，并尝试在 Windows 环境下编写控制软件，使得界面更加友好，操作和掌握起来都比较方便，真正做到所见及所得，但这些都无疑对实时性和对硬件的底层操作是一种挑战，但 Windows 环境也有 DOS 环境不可比拟的优点，如漂亮的图形用户界面(GUI)、几乎无限制的内存资源、多任务并发执行机制等等。所以这样的尝试和探索也是很有意义的。

## 第二章 控制系统的总体设计

### § 2.1 控制系统的原理及硬件结构

#### 2.2.1 控制系统结构

控制系统的核心为一块 GT400-SV 运动控制器，主轴编码器、光栅尺以及限位、报警等信号均接入控制器中。系统的原理如图 2.1 所示：



- (1) 光栅尺信号      (2) 主轴编码器信号  
 (3) 模拟量输出      (驱动直线电机)  
 (4) 正向限位信号   (5) 负向限位信号  
 (6) 原点信号        (7) 驱动器报警信号

图 2.1 系统原理图

GT400-SV 四轴运动控制器的核心是由美国 AD 公司的 ADSP21XX 系列数字信号处理器和 FPGA 组成。是一块以 IBM PC 或兼容机为主机的标准 ISA 总线的运动控制器。它能完成实时位置轨迹计算，闭环伺服控制，主机命令处理和控制器的 I/O 管理。

该运动控制器具有以下控制功能：

- 看门狗实时监测 DSP 的工作状态。
- 基于坐标系编程的连续轨迹控制，可实现空间直线、圆弧插补运动。
- 提供程序缓冲区，实现运动轨迹预处理，以获得高质量的运动控制，并降低主机通讯实时性的要求。
- 面向各控制轴实现点到点运动控制，具有可编程 S-曲线、梯形曲线、速度控制和电子齿轮运动控制方式。
- 使用 32 位(二进制)有效数字计算，实现高精度的轨迹控制。
- 伺服控制采用可编程数字 PID+速度前馈+加速度前馈滤波方式。
- 可编程设置伺服周期。四轴伺服（插补）周期为 162 微秒。
- 硬件捕获编码器 Index 信号和系统原点 Home 信号；保证控制系统具有较高的重复定位精度。
- 可编程设置位置限位、跟随误差极限、加速度极限、控制输出极限等参数，实现安全可靠的控制。

系统的硬件结构如图 2.2 所示。

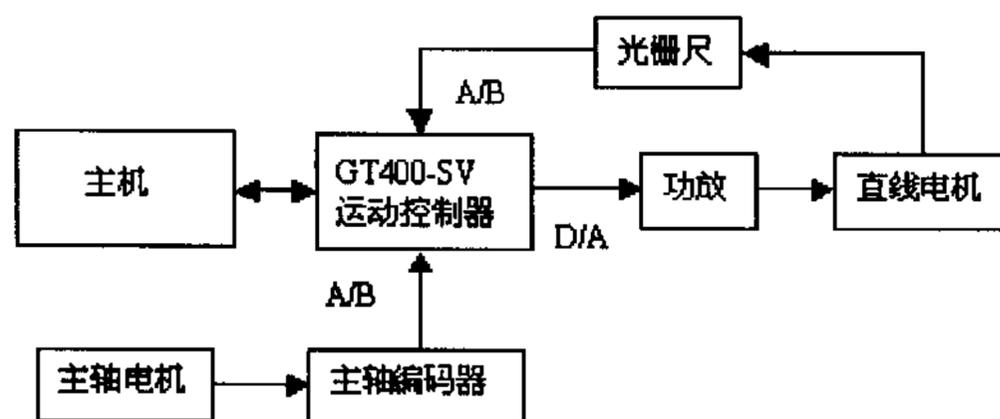


图 2.2 系统结构图

其中，GT400-SV 运动控制器的控制模式和参数由主机通过 I/O 接口设定，运动控制器根据主机发送的命令决定采用何种加减速控制曲线和运动模式。

设活塞截面型线采样点数  $N=72$ ，通过主轴编码器获得主轴的实际位置(每 5 度一次)，然后利用运动控制器的事件中断由主机根据活塞数据文件向运动控制器发送直线电机的目标位置，运动控制器在其内部的 PID 数字滤波器中将目标位置与光栅尺反馈的实际位置

进行比较得到位置误差，然后将位置误差值送到数模转换器（DAC）和运算放大器输出伺服电机的控制信号。如何获取运动控制器的事件中断通知主机发送活塞数据以及系统响应的实时性将是本课题的主要研究内容。

### 2.2.2 轴运动控制工作原理

图 2.3 为 GT400-SV 运动控制器的原理框图。图中所示的增量编码器的 A, B 相信号作为位置反馈信号，运动控制器通过四倍频、加减计数器得到实际位置。实际位置的信息存储在位置寄存器中，主机可以通过控制寄存器读取。运动控制器的目标位置由主机设定，通过内部加减速控制器得到控制器的当前运动位置、速度和加速度。然后将当前运动位置，速度和加速度送入数字伺服滤波器与控制器反馈得到的实际位置进行比较得到控制器的位置误差值，经过数字伺服滤波器后，送到数模转换器（DAC）和运算放大器输出伺服电机的控制信号。

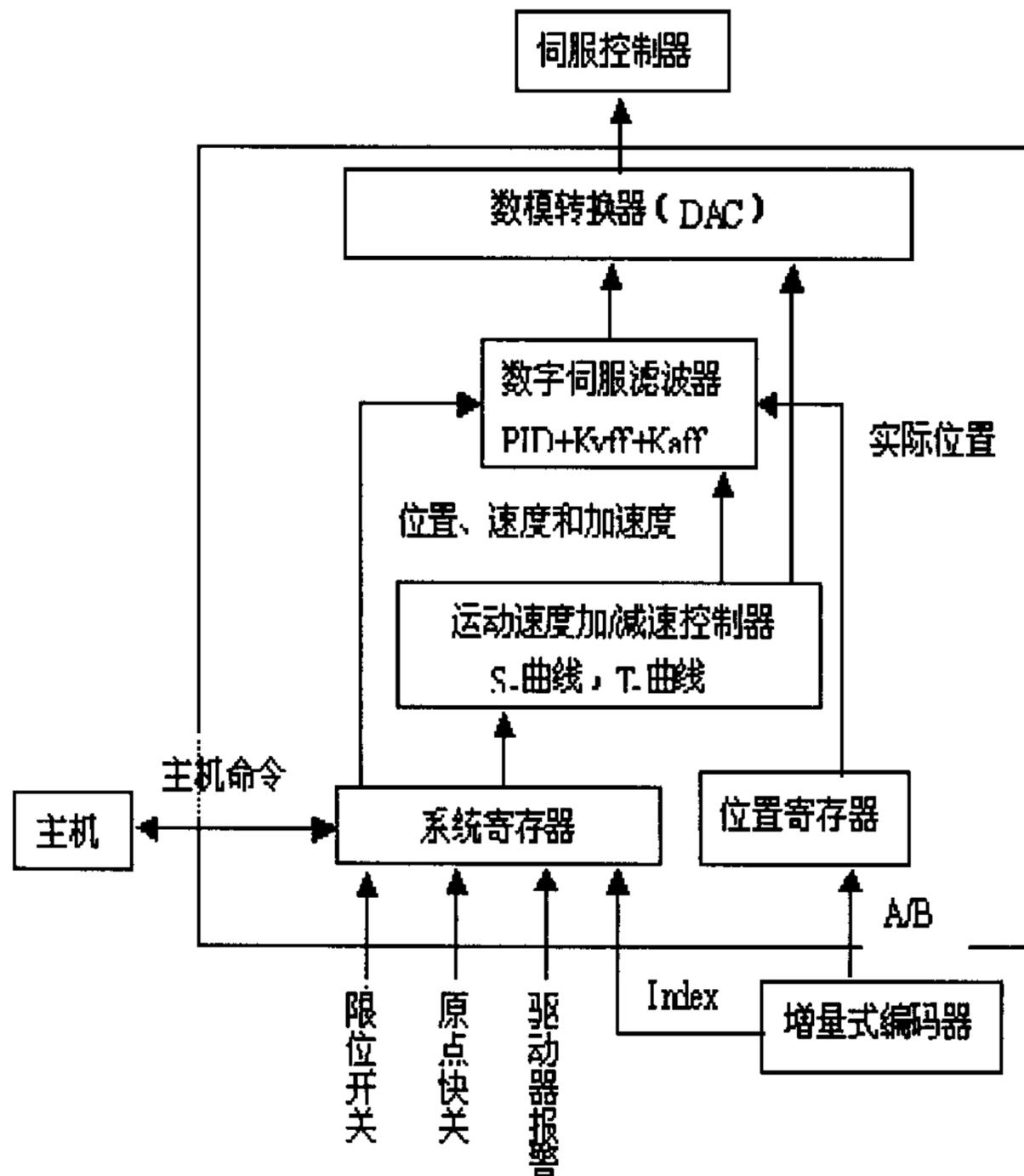
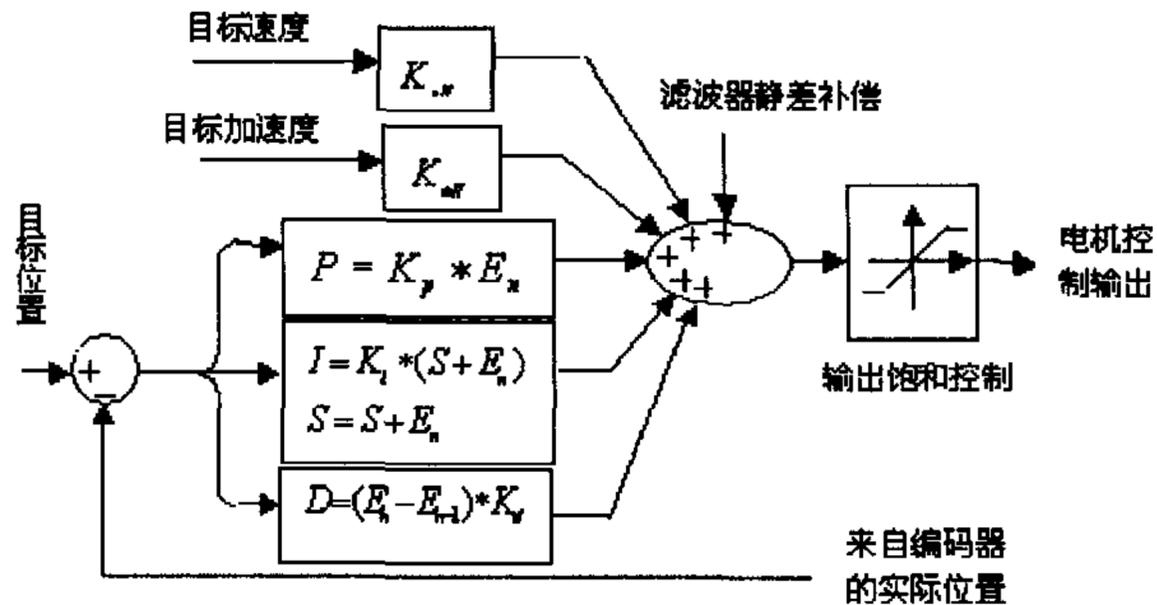


图 2.3 运动控制器原理框图

GT400-SV 运动控制器采用 PID 滤波器，外加速度和加速度前馈，即 PID+ $K_{VFF}$ + $K_{AFF}$  滤波器。通过调节各增益参数，该滤波器能对大多数系统实现精确而稳定的控制。伺服滤波器的参数可由主机设定。PID+ $K_{VFF}$ + $K_{AFF}$  滤波器的原理如图 2.4 所示。

图 2.4 PID+ $K_{VFF}$ + $K_{AFF}$  滤波器的原理图

数字伺服滤波器输出计算公式为：

$$E_n = (P_{target})_n - (P_{actual})_n$$

$$U_n = (E_n K_p + (E_n - E_{n-1}) K_d + \left( \sum_n E_n \right) K_i / 64 + V_{target} K_{vff} + ACC_{target} K_{aff}) / 4 + B \text{ 其中}$$

$U_n$  电机输出值

$E_n$  第  $n$  采样时刻的位置误差

$P_{target}$   $n$  采样时刻的目标位置

$P_{actual}$   $n$  采样时刻的实际位置

$\left( \sum_n E_n \right)$   $n$  采样时刻的误差积分值

$V_{target}$  期望速度单位：计数值/采样周期

$B$  电机偏移量

数字伺服滤波器的输出静差补偿主要用于补偿控制轴单方向外力的影响，如：机床垂直轴的重力。滤波器的输出静差补偿值可以通过主机命令设置。

## § 2.2 系统组成

系统为采用基于 DSP 的运动控制器加 PC 构成的新一代开放式数控系统，属于上、下位机控制的方式。

- 由 PC 构成上位机

该部分主要利用 Windows 强大的图形用户接口提供控制系统的人机界面，并完成系统的初始化、控制参数的设置、活塞数据处理、文件管理等功能。

- 以基于 DSP（下位机）的运动控制器为核心构成的闭环位置控制器

该部分主要包括以 ADSP2181 为核心的运动控制器、MF 型直线电机、放大器、光栅尺、编码器等。运动控制器通过主轴编码器获得主轴的实际位置（一般每 5 度一次），然后利用运动控制器的事件中断由主机根据活塞数据文件向运动控制器发送直线电机的目标位置，运动控制器在其内部的 PID 数字滤波器中将目标位置与光栅尺反馈的实际位置进行比较得到位置误差，然后将位置误差值送到数模转换器（DAC）和运算放大器输出伺服电机的控制信号。充分利用了运动控制器无与伦比的伺服、插补和实时计算能力。

控制系统具体的硬件组成：

1. MF 型直线电机

用于椭圆数控车床伺服刀架上的电机多为音圈式直线电机，这种电机“出力/质量比”较大，非常适合高频响的场合。按照结构形式，可以将该类电机分为两大类：MPK 型和 MF 型。本系统的直线电机为 MF 型，其原理如图所示：

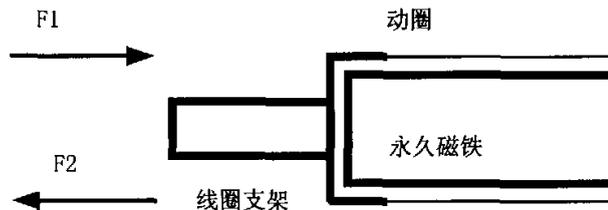


图 2.5 动圈式直线电机结构示意图

MF 模型直线电机采用永久磁铁产生作用磁场，动圈套在其上如图 2-1 所示，当在线圈(动圈)中通以电流时，线圈立即产生沿轴向的作用力。根据电流的方向不同，产生作用力的方向也不同，如图中 F1、F2 所示。

直线电机其结构形式如图 2.6 所示：

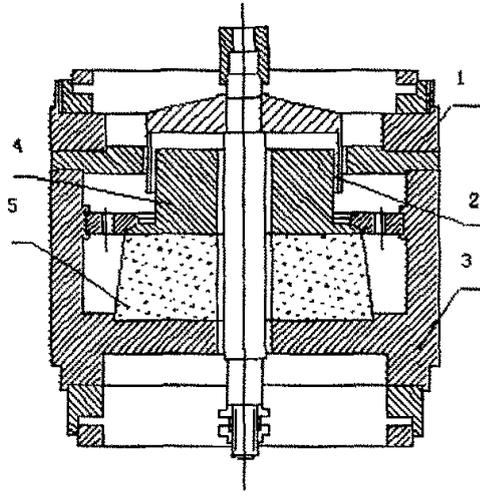


图 2.6 永磁式 MFK 型直流直线电机的装配结构

1、3、4：导磁体 DT6 2、线圈

电机的结构形式为直线导轨型。

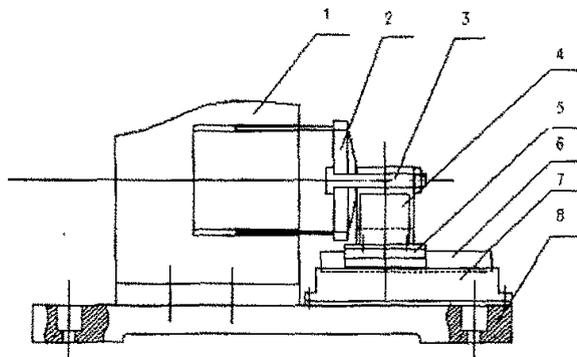


图 2.7 直线导轨式

1-电机底座 2-动圈 3-联结轴 4-联结块

5-光栅动尺 6-直线导轨 7-导轨底轨 8-底板

这种结构形式的优点为：

- 保证线圈的轴和磁座有较好的同轴度, 以使线圈能在环形磁槽内作往复运动而不“擦圈”。
- 能重复装拆, 每一次的重复定位精度都很高, 不会因为重复的装拆而带来调节上的困难。
- 方便润滑, 从而更好地保障其寿命。
- 便于光栅尺、行程开关、零位开关的安装、定位。
- 能保证一定的强度和刚度。

## 2. GT400-SV 运动控制器

运动控制器选用了固高公司的 GT400-SV 运动控制器, 该控制器的核心由美国 AD 公司的 ADSP2181 数字信号处理器和 FPGA 组成, 其硬件结构原理见图 2-6, 同时 GT400-SV 还是一种用户可编程运动控制器。提供给用户一块四轴运动控制器之外, 还提供一个用 C 语言编写的用户软件包。软件包以 C 语言源程序形式提供给用户。用户可以在自己的开发应用系统时, 调用该软件包所提供的功能函数。

GT400-SV 运动控制器的主要功能: 数字信号处理器和 FPGA 构成的 GT400-SV 内核完成运动控制器的实时轨迹规划、位置闭环伺服控制、主机命令处理和控制器 I/O 管理。基于篇幅方面的考虑, 对运动控制器就不作太多的介绍, 相关的情况可以参看用户手册

## 3. 位置检测元件—光栅尺

考虑到位置控制要求有很高的精确性、抗干扰性和较高的位移分辨率, 我们所选用的位置检测元件为英国雷尼绍公司(RENISHAW)的产品, 读数头型号是 RGH22, 分辨率是 0.5 $\mu$ m, 尺的型号是 50 线/毫米。

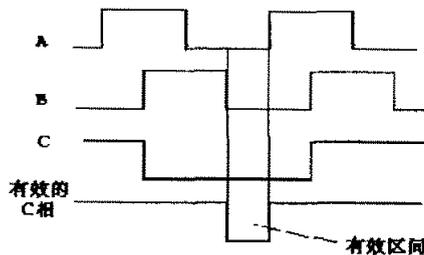


图 2.8 C 信号与 A, B 信号的关系

课题所选用光栅的输出信号是数字信号，提供两路位置输入信号和一路控制信号：

A 相信道    B 相信道    C 脉冲

A, B 相信号为方波，彼此相差为  $90^\circ$ 。正向运动时，A 相超前 B 相  $90^\circ$ ；负向运动时，B 相超前 A 相  $90^\circ$ 。一个伺服信道的完整相位周期将产生 4 个  $90^\circ$  相位差的跃变，GT400-SV 运动控制器对光栅的输入信号进行 4 倍频使运动控制分辨率提高 4 倍，从而使伺服运动更加平稳。

光栅尺输出的数字信号与 GT400-SV 的接口如图 2.9 所示：

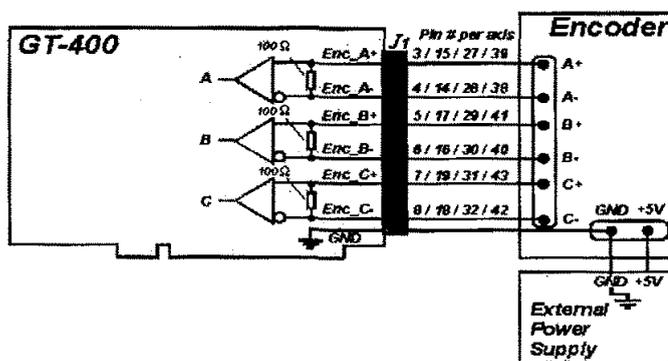


图 2.9 光栅与运动控制器接口电路示意图

选定好光栅的型号以后，还须要做的工作就是要对其对光栅所能承受的速度和加速度进行校核。

以一次椭圆为例：

$$S_1 = \frac{e}{4}(1 - \cos 2\theta)$$

$$\theta = \frac{2\pi n}{60}t$$

$$S_1 = \frac{e}{4} \frac{4\pi n}{60} \sin \frac{4\pi n}{60}t$$

$$S_1'' = \frac{e}{4} \left( \frac{4\pi n}{60} \right)^2 \cos \frac{4\pi n}{60} t$$

$$S_{1\max}' = \frac{e\pi n}{60}$$

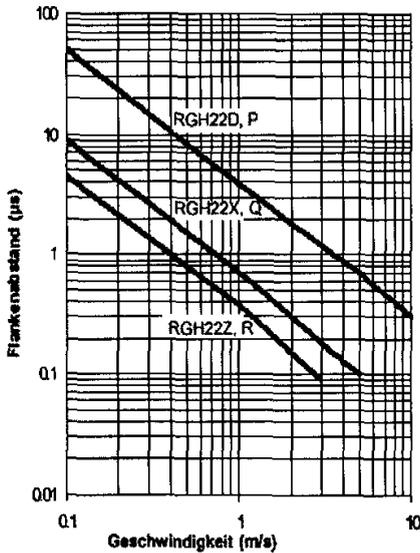
$$S_{2\max}'' = \frac{e}{4} \left( \frac{4\pi n}{60} \right)^2$$

1、n=1500rpm、e=0.8 时:

$$S_{1\max}' = 62.8\text{mm/s}$$

$$S_{2\max}'' = 2g$$

2、n=2000rpm、e=0.8 时:



$$S_{1\max}' = 83.8\text{mm/s}$$

$$S_{2\max}'' = 3.58g$$

3、n=3000rpm、e=0.8 时:

$$S_{1\max}' = 125.6\text{mm/s}$$

$$S_{2\max}'' = 8g$$

图 2.10

由图 2.10 所示, 该型号的光栅对于课题所要求的速度范围是可以达到的。通过光栅尺的用户手册, 允许最大的加速度是 10g, 因此加速度也可满足。

#### 4. 编码器

编码器采用长春东河光电技术有限公司的 PIF-XXX-C05D 型光电编码器。参数如表所示:

表 2-2 各类型号光栅尺速度范围

型式 记号	电源电压 DC(V)	消耗电流 (mA)	输出 方式	输出 电压 (V)		注 入 电 流 (mA)	最小负 载阻抗 ( $\Omega$ )	上升、 下 降 时 间 ( $\mu$ s)	响 应 频 率 (KHz)
				VH	VL				
05D	$5 \pm 0.25$	250	长线驱动	2.5	0.5	40	—	0.2	100

输出波形与信号位置精度:

90° 相位差两信号与零位信号从轴头看, 顺时针 (CW) 时的波形图:

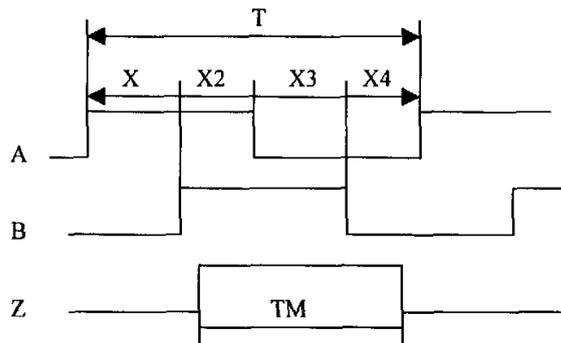


图 2.11 编码器输出波形

波形比:  $X1+X2=0.5T \pm 0.1T$

$X2+X3=0.5T \pm 0.1T$

相位差:  $Xn \geq 0.12T$  ( $n=1, 2, 3, 4$ )

信号位置精度: 准确度 =  $0.2T$

周期误差  $\pm 0.01T$

零位信号:  $TM=1T \pm 0.5T$

$T=360^\circ / N$  ( $N$  为每转输出脉冲数)

系统中使用的编码器为  $N=360$  和  $N=72$  两种。编码器与运动控制器的接口电路见图 2-8。

### 5. 变频器与三相异步电机

系统中使用了三菱 FR-E500 变频调速器。变频器与三相异步电机相连，带动编码器提供主轴位置信号。课题还探讨了 PC 与变频器的通信方法。

三相异步电机的参数如下：

型号：Y0D71-2/4                      额定功率：500/370 瓦  
 转速：2800/1400 转/分      电压：380 伏      电流：1.58/1.37

### 6. 功率放大器

采用的功放是航天部七零二所的型号为 7101 型功率放大器，它能将控制电压转化为电流以驱动直线电机。其特点是性能可靠，漂移小，频带宽（1KHZ），线性好，在工频范围内，可认为是一个线性的放大系统，这就可系统的简化提供了有利保证。功放允许控制输入电压正负 10 伏。功放与运动控制器的接口如图 2.12 所示：

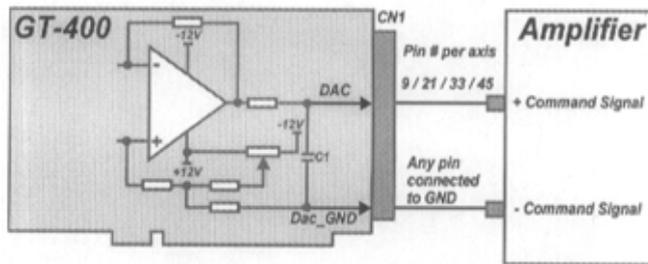


图 2.12 伺服电机驱动器连线

以上主要是介绍了一下直线电机测控系统中较为重要的部分，其余如工控机、行程开关及一些接口就不作一一介绍。

## § 2.3 本章小结

本章第一节主要介绍了系统的硬件结构及其原理。第二节对控制系统各部分诸如运动控制器、光栅尺、功放、编码器、变频器及其接口电路作了介绍。

### 第三章 控制系统建模仿真及实验

本章在前面完成硬件平台的构筑以后，开始对控制系统作理论上的建模，并为实际控制工作提供一些理论上的依据及一些可行性的分析，避免盲目性。

#### § 3.1 理论建模

##### 3.1.1 电机模型

为分析方便，图 3.1 和图 3.2 分别列出了 MF 型直线电机的电路简图和机械模型，从图 3.2 可以看出，除了 X 方向之外，其它的运动自由度均由相应的机构所限制。当线圈通电时，产生 X 方向的力 F，使线圈轴沿 X 方向产生直线运动。由于不存在弹簧的约束，线圈受力容易冲出，所以在实际的情况中，必须有限位措施。系统对运动体有粘性阻尼的作用，设阻尼系数为 C。

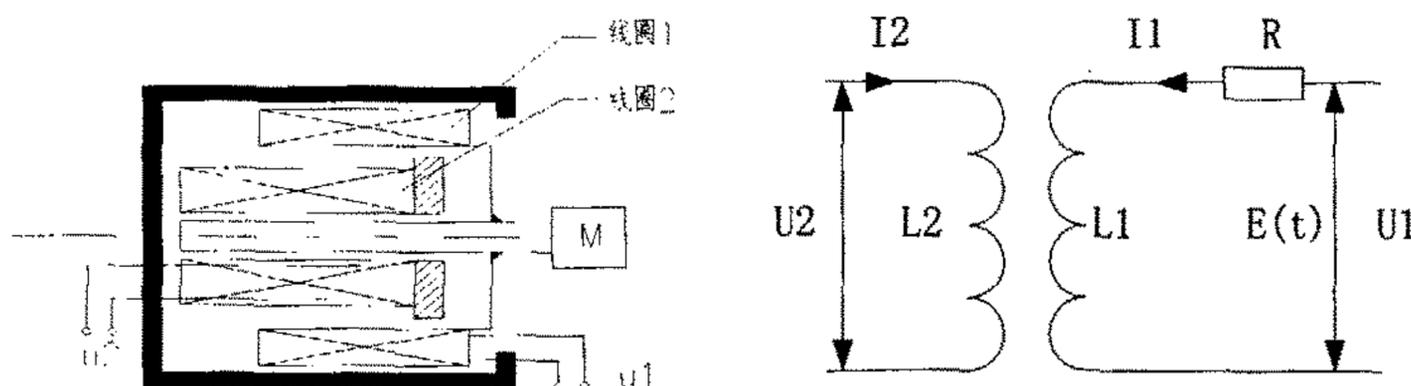


图 3.1 MF 型直线电机简化模型

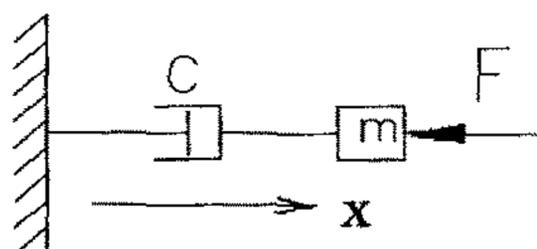


图 3.2 电机力模型

在电路模型中，L2、I2 表示直线电机的永磁体，线圈的电感为 L1，E 是控制信号， $E=E(t)$ ，R 为线圈的内阻与控制信号 E(t) 的内阻（即，功放的输出阻抗）之和。该机构的运行原理是：当线圈中通入电流 I1 时将产生磁场，该磁场与永磁场相互作用而产生作用力。由于电机线圈通过输出轴与外面的运动体是固联的，这个力就推动运动体在直线导轨上产生往复运动。可见该系统可以看作是由电系统、耦合磁场、机械系统所组成，系统框图如图 3.3 所示。

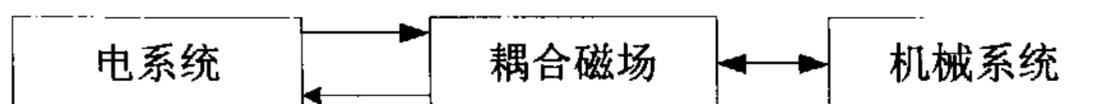


图 3.3 系统框图

本伺服系统的目的是使运动体的位移  $X$  跟踪所需的控制信号  $E(t)$  的变化, 以满足各种应用的目的。电机的运动方程可以由电系统的电压平衡方程及机械系统的力平衡方程来组成。

### 3.1.2 动力学方程

#### 1 电压平衡方程:

为便于分析, 永磁铁体可以用一等效线圈来代替, 因此才有图 3.1 中线圈  $L_1$ , 其等效的原则是使线圈的激励势等于永磁铁产生的加于其外部磁势, 这个等效线圈可称为永磁激磁线圈, 直线电机的可动线圈称作电激磁线圈。这样, 直线电机的运动就可以简化为两个线圈的运动。电压平衡方程如下:

$$\begin{cases} R_1 i_1 + \frac{d\phi_1}{dt} = u_1 \\ \frac{d\phi_2}{dt} = u_2 \end{cases} \quad \dots\dots \text{式3-1}$$

式中  $U_1$  和  $U_2$  分别为施加于线圈 1、线圈 2 的电压。 $R_1$  是线圈 1 的内阻,  $\phi$  表示磁通。

假设永磁线圈中的电流为  $i_2$ , 在线性情况下, 由线圈 1、线圈 2 组成的磁链可表示为

$$\begin{cases} \phi_1 = L_{11} i_1 + L_{12} i_2 \\ \phi_2 = L_{21} i_1 + L_{22} i_2 \end{cases} \quad \dots\dots \text{式3-2}$$

式中  $L_{11}$ 、 $L_{22}$  表示线圈 1、线圈 2 的自感,  $L_{12}$ 、 $L_{21}$  表示线圈 1 与线圈 2 之间的互感。由于永磁线圈 2 是一个等效于永磁铁的虚拟通电线圈, 因此该回路的电压平衡方程可以不考虑, 但须考虑由它产生的交链于线圈 2 的磁通。

将式 3-2 代入式 3-1, 在  $L_{11}$  为常数的情况下, 可得到如下方程:

$$R_1 i_1 + L_{11} \frac{di_1}{dt} + \frac{d\phi_{12}}{dt} = u_1 \quad \dots\dots \text{式3-3}$$

其中,  $\phi_{12} = L_{12} i_2$ , 上式可简化为

$$Ri + L \frac{di}{dt} + \frac{d\phi}{dt} = u \quad \dots\dots \text{式3-4}$$

2 力平衡方程:

力平衡在这里是指电机输出力与运动体之间的平衡。力平衡关系如下:

$$F_m = F_b + F_a + F_l + F_k \quad \dots\dots \text{式3-5}$$

式中:  $F_m$  直线电机产生的电磁力, 且有  $F_m = Bli$ ,  $l$  为磁场中线圈  $l$  的长度,  $B$  是电机的气隙磁密。

$F_b$  粘滞摩擦力。在此,  $F_b = (C + B_l^2/R) * X'$  ( $C$  为机械阻尼,  $B_l^2/R$  为电磁阻尼,  $X'$  为速度)

$F_k$  弹簧变形所产生力, 在此为零。

$F_a$  惯性力,  $F_a = m * d^2x/dt^2$

$F_l$ : 负载力, 在此因为电机暂无负载为零。

所以有:

$$F_m = m \frac{d^2x}{dt^2} + (C + \frac{B_l^2}{R}) \frac{dx}{dt} \quad \dots\dots \text{式3-6}$$

联系前面所推导出的公式以及力平衡公式, 将其联立可得:

$$\begin{cases} Ri + L \frac{di}{dt} + \frac{d\phi}{dt} = u \\ m \frac{d^2x}{dt^2} + (C + \frac{B_l^2}{R}) \frac{dx}{dt} = F_m \\ F_m = Bli \end{cases} \quad \dots\dots \text{式3-7}$$

在联立简化求解直线电机的传递函数中, 作以下几点说明:

(1) 在动子的行程内, 气隙场强  $B$  恒定, 当动位移为  $dx$  时线圈的磁通增量为

$$d\phi = Bldx$$

(2)  $L$  实际是很小的, 所以式中的  $L * di/dt$  项可以忽略不计。

(3) 两线圈相动运动, 必然会产生感生电动势, 从而使  $i$  发生变化, 在电磁阻尼这

一项中,  $B_1$  是电机的力常数, 根据设计, 它的值在 15-20 (N/A) 这样一个范围。因此, 电磁阻尼的大小取决于电阻  $R$  的大小。而  $R$  是由线圈和功放的输出阻抗两部分组成。线圈内阻是很小的, 约为 0.6 欧姆, 由此可知, 在直线电机已经给定的情况下, 电磁阻尼主要由功放的输出阻抗所决定。当阻抗很大时, 电磁阻尼很小, 可以忽略不计。然而如果功放的输出阻抗较小时, 则电磁阻尼是不容忽略的。

关于功放的阻抗匹配问题, 已经由其它同志解决好了, 所以在此就可以不考虑电磁阻尼问题, 使模型得以进一步简化。

综上三点可得电机传函:

$$G_p(s) = \frac{X(s)}{I(s)} = \frac{B_1}{ms^2 + cs} = \frac{B_1}{s(ms + c)} \quad \dots\dots \text{式3-8}$$

上式中, 动子的质量  $M=0.35\text{kg}$ , 及  $B_1=15\text{N/A}$ , 阻尼根据该类型的电机定为  $C=0.05$ , 这样分别将  $M$  值和  $B_1$  值代入, 初步认定该直线电机的传递函数为:

$$G_p(s) = \frac{15}{0.35s^2 + 0.05s} = \frac{42.86}{s^2 + 0.14s} = \frac{42.86}{s(s + 0.14)} \quad \dots\dots \text{式3-9}$$

## § 3.2 系统闭环控制实验

上一节通过对控制系统进行理论建模, 初步获得了直线电机的传递函数。下面对闭环系统进行控制实验, 从而与后面的 Matlab 仿真结果进行对比来验证理论模型的正确性。

### 3.2.1 闭环阶跃响应实验

在闭环阶跃响应实验中采用定时中断的方式来获得从阶跃信号输入直到稳定这一过程中直线电机的位置值。定时中断的中断周期设为 0.15 毫秒。阶跃信号输入值为位置值, 在控制程序中转换为脉冲数, 对应于光栅尺 A/B 相信号四倍频后的脉冲数。系统采用 GT400-SV 运动控制器的定时中断进行控制的原理见第四章。因为运动控制器的算法是“PID+速度前馈+加速度前馈”, 所以通过多次实验获得理想的阶跃实验结果, 从而确定 PID 参数值。图 3.4 为系统阶跃响应控制程序的流程图。

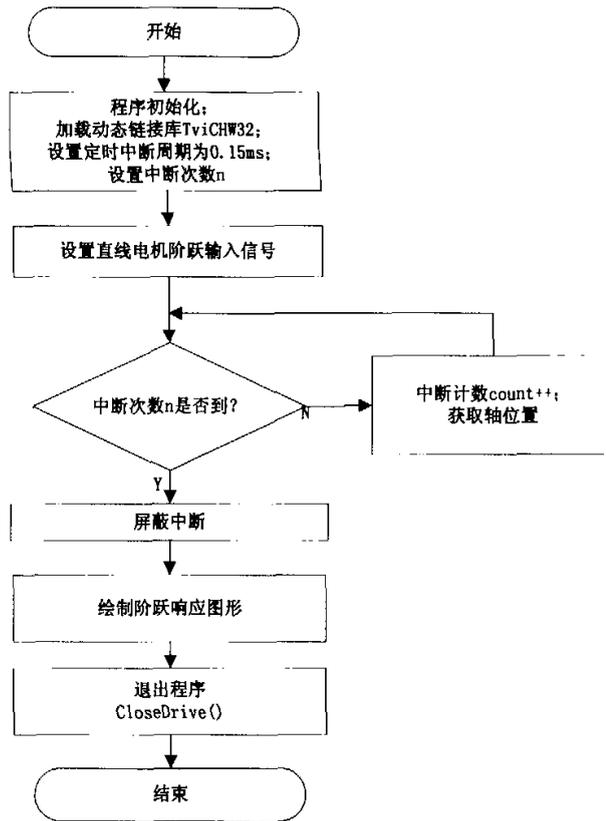


图 3.4 系统阶跃响应程序流程图

实验结果如图 3.5 所示，图中纵坐标为位置值（单位  $\mu\text{m}$ ），横坐标为定时中断计数值，每两个计数值间隔  $0.15\text{ms}$ 。从图中可以看出，电机能够以很快的速度达到期望值，并且超调量很小，震荡次数和调整时间都比较理想。此时的 PID 参数分别为 6, 0.5, 1。

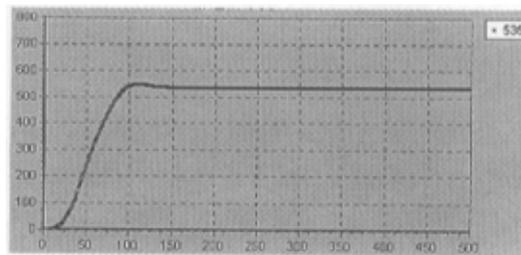


图 3.5 阶跃响应实验结果

### 3.2.2 闭环扫频实验

通过对系统进行闭环扫频，得到系统的幅频特性与相频特性，从而在一下节中与理论

建模并仿真得到的系统频率特性进行对比来验证模型。

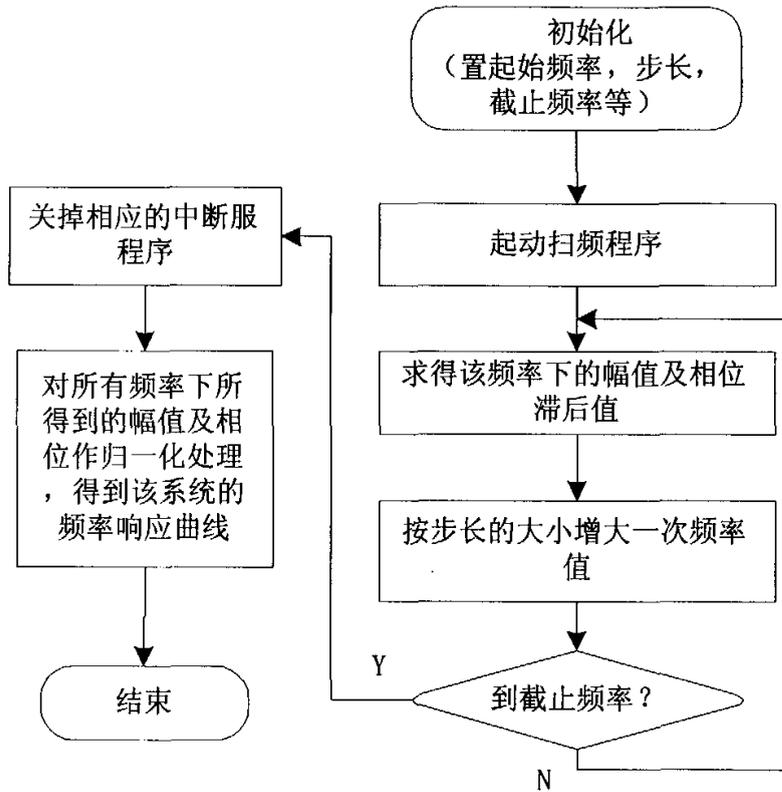


图 3.6 系统扫频程序流程图

由于电机本身的结构以及实时性的限制，扫频的截止频率仅为 50Hz，也就是仅于电机的低频至中低频一段，图 3.7 是 PID 参数分别为 6, 0.5, 1 时对系统的扫频结果。

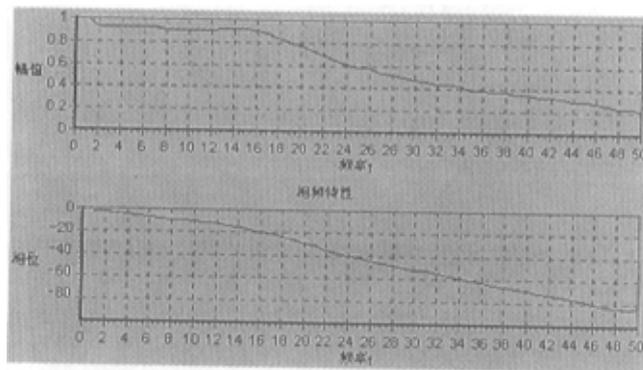


图 3.7 系统扫频结果

图 3.7 为根据扫频结果绘制的波特图。

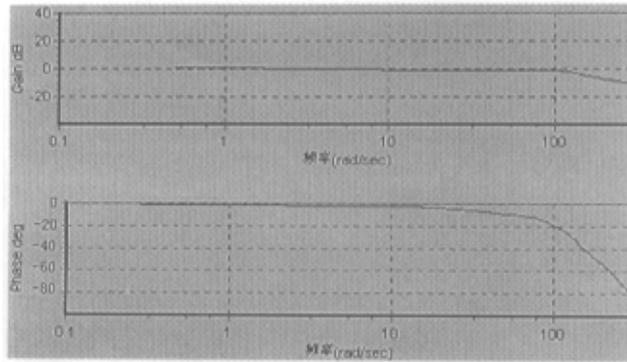


图 3.7 系统波特图

### § 3.3 MATLAB仿真

通过仿真，可以进一步加深对系统性能的认识，可以寻找一些有效控制该系统的控制方法。并通过与上面的实验结果进行比较来判断系统理论建模的正确性。

MATLAB 是一个高级的数学分析与运算软件，可以用作动态系统的建模与仿真。它是一个开放的环境，在这个环境下，人们开发了许多具有特殊用途的工具箱软件，如控制系统、信号处理、最优控制、鲁棒控制及模糊控制工具箱等。本章使用到了控制系统的仿真部分。

#### 3.3.1 控制器的选择

常用的控制器有很多种，如采用数字 PID、最小拍无纹波控制、模糊控制器等。针对该电机曾经用最小拍无纹波方法设计了一个控制器，通过仿真，达到了较好的效果，说明控制器的选择不是唯一的。因为在本课题中的运动控制器用到的算法是“PID+速度前馈+加速度前馈”，所以在此节讨论数字 PID 控制仿真方法的实现和结果分析。

#### 3.3.2 数字PID仿真

重写直电机的传递函数

$$G_p(s) = \frac{15}{0.35s^2 + 0.05s} = \frac{42.86}{s^2 + 0.14s} = \frac{42.86}{s(s + 0.14)}$$

显然该电机在开环状态下是不稳定的，需用 PID 方法去调整。根据上一节实验结果得到的 PID 参数为： $K_p=6$      $K_i=0.5$      $K_d=1$ 。

将所得参数在MATLAB提供的SIUMLINK工具箱中闭环仿真,分别到如图3.8和如图3.9所示的阶跃响应和闭环的波特图:

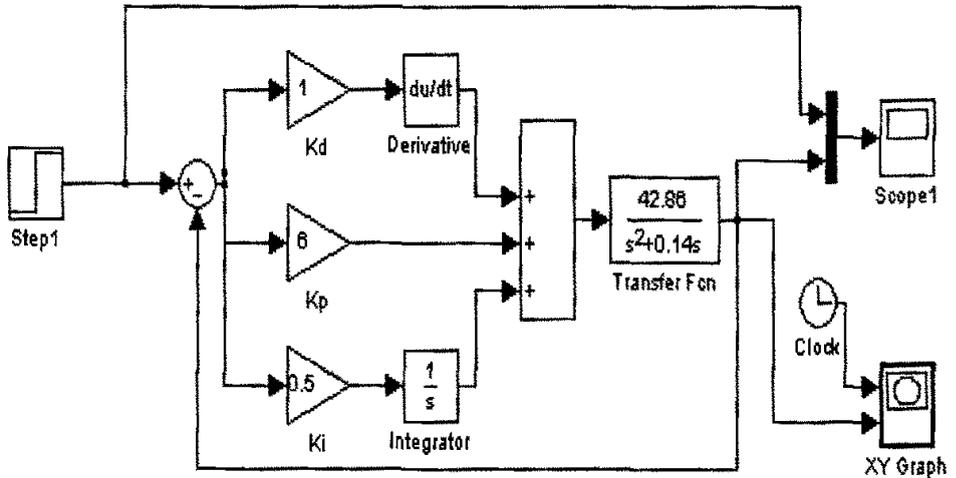
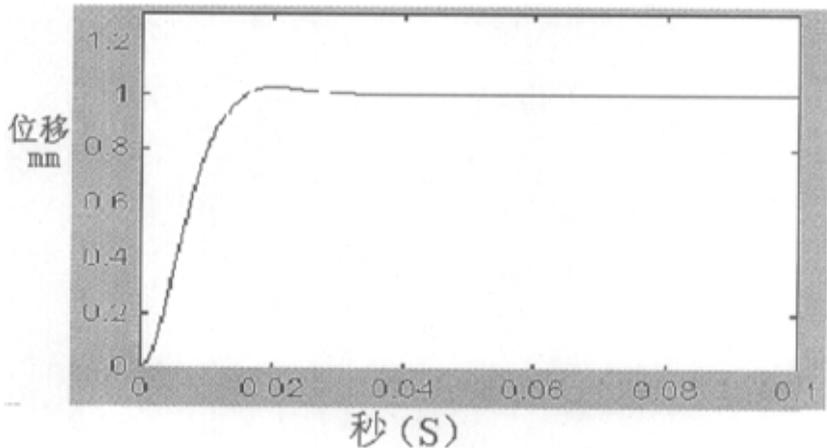


图 3.8 SIUMLINK 仿真环境



3.9 仿真的阶跃响应曲线

从仿真的阶跃响应曲线来看,超调量很小,调整时间和振荡次数也很短,与实验所得的结果很好的吻合。

通过该闭环波特图来看,截止频率应该在大约 300-500rad/s 之间,但相位滞后在 100rad/s 之后较为明显。与实验结果相比,在 100rad/s 时的相位滞后均为 20 度,模型在中低频一段是比较吻合的。当频率更高时,通过分析确定由于控制方法的实时性会造成较大的失真,具体的分析见§5.2。

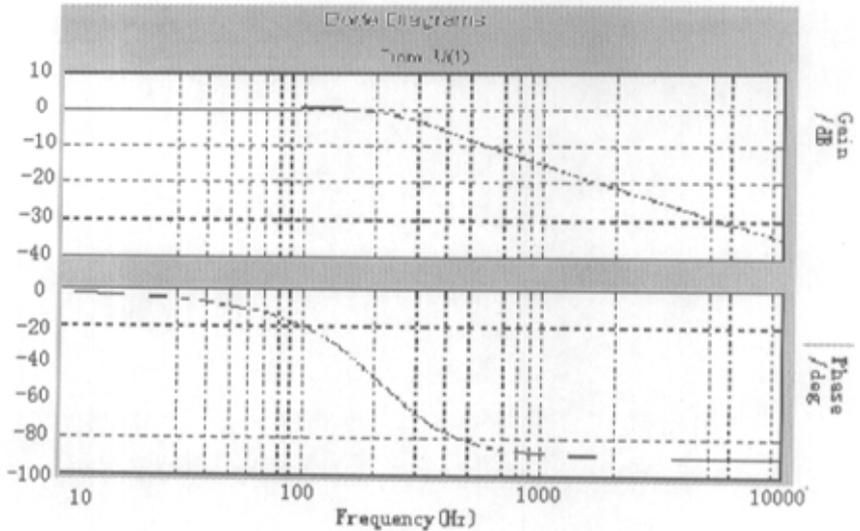


图 3-6 系统仿真的闭环波德图

### § 3.3 本章小结

本章的第一节在分析了 MF 型直线电机的具体结构的基础上，参照了前人在音圈式直线电机建模的成功经验，对控制系统作了一些简化。例如，在一定范围内认为功率放大器是线性的；电磁阻尼在一定的条件之下可以忽略不计等。这些假设都是有科学根据和实践证明的。另外对系统的机械阻尼  $C$  的估计也是参照了本教研室以及清华大学对同类直线电机的测试而得到的，况且机械阻尼本身就很小，在很多文献都忽略不计，例如在郑子文的硕士论文中就没考虑到阻尼。因此，我们完全有理由认为，因为 MF 型直线电机的结构本来就比 MFK 型直线电机的结构就简单，因此所存在的不确定性因素更少，理论分析所得到的模型将只会更加准确。当然，光有理论上的推导是不够的，必须有真实的实验结果相印证，这样才更加具有说服力，在第二节中，对该控制系统进行阶跃响应实验和闭环扫频，以往的直线电机的系统辨识常采用的是开环扫频，并且已经由本室研究生孟咸勇专门开发了扫频的软件，做得比较成熟。但是由于本课题所涉及的电机在开环状态不稳定，无法象 MFK 型直线电机那样直接得到结果，因此本课题专门做了闭环扫频软件，以期能够得到较好的结果。

第三节，在认为第一节所得到的模型是正确的情况下，并采用实验获得的 PID 参数通过用 MATALAB 软件进行仿真。将实验得到的阶跃响应曲线与仿真结果对比，两者较好的吻

合，并且通过对比分析实验与仿真得到闭环系统的波特图，进一步对系统更深一层有一个认识。

## 第四章 系统控制方法及实验

### § 4.1 活塞外圆表面加工运动形成原理

最常见的活塞外形是中凸变椭圆。图纸上常以椭圆度、形线和某些截面上的形状等数据来描述外形面。在传统的靠模法加工中，人们很少关心外圆面上某一点的具体坐标。在数控法加工过程中，则要建立活塞外圆面的数学模型。

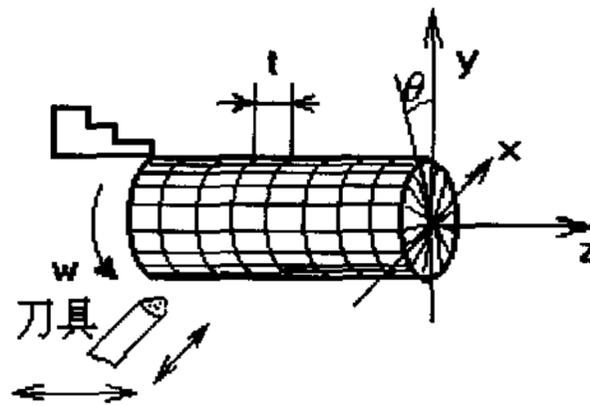
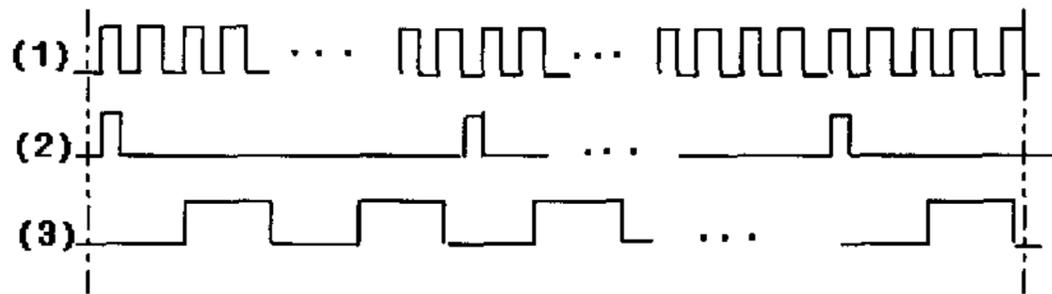


图 1

图 4.1 网格化的活塞外圆

首先，按照图纸要求将外圆曲面网格化，如图 4.1 所示。只要网格足够密 ( $t$  和  $\theta$  足够小)，则这些数字化了的点就能以足够的精度来代表活塞外圆曲面。在车削过程中，如果刀尖的轨迹能准确地通过这些网格结点，就能加工出足够理想的活塞。数控系统的任务就在于让刀具在  $X$  方向沿预先编好的轨迹  $S_x = (\theta_i, Z_j)$  运动。其中， $\theta_i$  表示第  $i$  个转角， $Z_j$  表示第  $j$  个截面。 $S_x = (\theta_i, Z_j)$  是根据图纸要求进行插补运算求出并存储在计算机中的数据。



(1) 主轴采用 (2) 主轴同步 (3) 纵向

图 4.2 主要控制信号

在本系统中,当交流电机(主轴)旋转时,编码器向计算机发送主轴采样信号和主轴同步信号(如图 4.2)。主轴每转一圈,发一个同步信号,对应着活塞的长轴位置。以同步信号为角向基准,每收到一个主轴采样信号,对应着刀尖相对于长轴的每一个角度 $\theta_i$ 。主轴采样信号触发运动控制器中断,在中断服务程序中发送轨迹 $S_i=(\theta_i, Z_i)$ 到运动控制器输出,该信号通过 D/A 转换,送到功放,驱动直线电机。当下一个主轴同步信号到来时触发中断使轴向位置 $\theta$ 清零,并发送纵向信号 $Z_i$ 驱动纵向步进电机前进一步。系统中没有对纵向进行控制,而只是对纵向信号进行计数。利用 GT400-SV 运动控制器实现上述运动过程的关键在于当工件转过 $\theta_i$ 角时如何产生中断并发送活塞数据 $S_i=(\theta_i, Z_i)$ 来控制直线电机的运动,并在每完成一个截面的加工后响应同步信号产生的中断对轴向位置清零。

## § 4.2 中断处理方法及测试程序

### 4.2.1 利用 TVicHW32 实现中断调用的方法

由于 Windows 操作系统是采用非抢占事件(消息)驱动为主要特征,强调设备无关性,对系统底层操作采取了屏蔽策略,因而无法立即响应外部事件中断,也就不能满足工业应用中实时事件处理和实时控制应用的要求。因此如何寻找一种方法来处理外部实时事件就显得尤为重要。

在这里使用 TVicHW32 控件来实现中断处理。该控件为 ActiveX 控件,除具备直接访问存储单元和端口的功能外,还提供了丰富的处理并口的属性和方法,以及处理硬件中断的属性、方法和事件,极大地拓展了应用程序对底层硬件的访问控制。TVicHW32 作为一个通用的设备驱动程序几乎可以被任何编程语言调用。通过它可以使用户的 Win32 应用程序在 Windows 95/98/ME/NT/2000 中直接访问和控制硬件而无需使用 Windows DDK(Driver Development Kit)。

在 C++Builder 中加载 TVicHW32 可以采用 VCL 控件和 DLL 两种方法。作为 VCL 控件时使用步骤如下:

下载 TVicHW32 压缩软件包并解压到一个目录中,如\HW。在 Windows 98 下,将文件 vichw00.vxd 复制到 Windows 98 的系统目录,例如, C:\WINDOWS\SYSTEM。在 Windows NT 下,将文件 vichw00.vxd 复制到 Windows NT 的 drivers 目录,例如, C:\WINNT\SYSTEM32\DRIVERS。该文件是控件访问硬件的驱动程序。

把 ocx 子目录下的 tvichw32.ocx 拷贝到\windows\system 子目录下，并在 DOS 命令行状态下键入以下命令进行注册：regsvr32 tvichw32.ocx。

在 C++Builder 中通过菜单 Component\Import ActiveX Control 将控件安装到控件面板中。安装后如图 4.3 所示，此时即可调用控件的函数对中断进行处理。

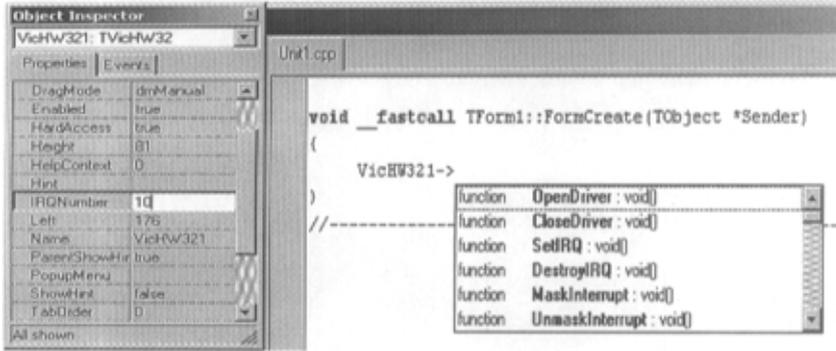


图 4.3 作为控件调用 TvicHW32

因为 VCL 控件仅仅是对 DLL 的包装，所以采用 DLL 方式有较好的性能。因此课题中采用动态链接库的方法。TvicHW32 在动态链接库中提供的与中断有关的函数主要有：

OpenTvicHW32()：加载驱动程序 (vichw32.vxd) 句柄；

CloseTvicHW32()：释放驱动程序句柄；

SetIRQ：为中断程序分配中断号；

MaskInterrupt()：屏蔽 SetIRQ 函数指定的中断号；

UnmaskInterrupt()：非屏蔽 SetIRQ 函数指定的中断号；

DestroyIRQ：释放中断占用的资源，屏蔽中断。

采用动态链接库调用 TvicHW32 实现中断响应的方法如下：

1. 在\*.h 文件中声明加载的函数

```
class TForm1 : public TForm
{
.....
public: //User declarations
```

---

```
HINSTANCE DllFile;
```

```
bool (__stdcall*OpenTVicHW32)(void);
```

```
void (__stdcall *CloseTVicHW32)(void);
```

```
.....
```

```
}
```

2. 在程序初始化时动态地加载 TVicHW32. DLL

```
void __fastcall TForm1::FormCreate(TObject *Sender)
```

```
{
```

```
    DllFile=LoadLibrary("Tvichw32.dll");
```

```
    if(DllFile)
```

```
    {
```

```
        OpenTVicHW32=(bool (__stdcall*)(void))GetProcAddress(DllFile, "OpenTVicHW32");
```

```
        CloseTVicHW32=(void (__stdcall*)(void))GetProcAddress(DllFile, "CloseTVicHW32");
```

```
.....
```

```
    }
```

```
}
```

3. 在相应的事件中设置中断号并使其处于非屏蔽状态

```
void __fastcall TForm1::StartBtnClick(TObject *Sender)
```

```
{
```

```
    .....; //对主轴和直线电机行初始化
```

```
    user.set_3();
```

```

user.set_intr_msk(msk);

//设置中断屏蔽字 msk, 使运动控制器可以响应指定的中断

OpenTVicHW32();

SetIRQ(10, IntrFunc); //设置中断号

UnMaskIRQ();

}

```

通过以上的设置即可在产生中断时调用中断服务程序进行处理。

#### 4.2.2 运动控制器中断

GT400-SV 运动控制器可以通过向主机发送中断请求方式, 使主机能及时对轴运动过程中出现的事件或过程作出处理。这种中断方式通常比主机来查询控制器的各种状态更为方便和有效。GT400-SV 运动控制器提供了两种中断, 一种是事件中断, 主要针对轴运动过程中出现的事件作出及时处理; 另一种是定时中断, 控制器以一定的周期向主机发出定时中断。由于两个中断共用主机的一根中断请求线, 因此主机在系统中只允许通过命令 `set_timer_int()` 和 `clr_timer_intr()` 来选择一种中断形式。

`set_timer_intr()` 命令设定控制器中断为定时中断, 定时中断的周期由控制器的伺服刷新周期和 `set_intr_time()` 命令的设定值共同确定。例如: 控制器的伺服刷新周期为 100 微秒, 主机用 `set_intr_time()` 命令设定的值为 10, 则定时中断的周期 =  $10 \times 100$  微秒, 即 1 毫秒。

`clr_timer_intr()` 命令将关闭控制器定时中断, 转向事件中断。控制器上电默认中断为事件中断。GT400-SV 运动控制器有表 4.1 中列举的几种事件能触发控制器中断请求。

表 4.1 轴事件中断标识定义

运动完成	运动完成时触发
驱动器报警	驱动器报警时触发
断点到达	断点到达时触发
捕获到当前	捕获到编码器的 index 脉冲或零位信

位置	号时触发
运动出错	轴运动误差超过设置的最大误差范围 时触发
负限位开关	负越位开关动作时触发
正限位开关	正越位开关动作时触发

对于每个轴和每个事件。主机可以通过 `set_intr_msk()` 命令确定其是否向主机申请中断。`Set_intr_msk()` 命令设定控制器的轴中断屏蔽寄存器来控制各种事件是否允许触发中断请求。中断控制屏蔽字如表 4.2 所示:

表 4.2 中断控制屏蔽字

Bit 位	说明
7-15	没有使用, 为 0
6	正向限位开关
5	负向限位开关
4	运动出错
3	Index/Home 捕获到
2	断点条件满足
1	轴驱动器报警
0	轴运动完成

当屏蔽字中某一位设置为“1”时, 允许该位标识的中断事件向主机产生中断申请, 如果设置为“0”时, 不允许该位标识的中断事件向主机产生中断申请。运动控制器产生中断申请后, 如果主机允许中断, 则 CPU 会响应这个中断, 主机可以通过调用 `get_intr()` 函数获取中断轴的状态寄存器的状态字。对于 ISA 总线 PC 机, 中断入口地址及中断号是相对应的。通过跳线设置, 本系统使用主机的 10 号中断。

在进行两轴联动的控制实验之前, 首先对运动控制器的定时中断及以上的事件中断进行了测试, 实验显示各种中断均能正常产生。具体程序参见附录。

## 4.2.3 对编码器输出信号的测试

由活塞外圆表面加工过程可知，采样信号和同步信号是否准确对系统有极大的影响。下面对编码器的采样和同步信号进行测试，以确保其准确无误。

GT400-SV 运动控制器采用增量编码器作为主轴的位置输入信号，并对输入信号进行四倍频使轴运动控制分辨率提高 4 倍，从而使轴伺服运动更加平稳。图 4.3 为测试程序的流程图：

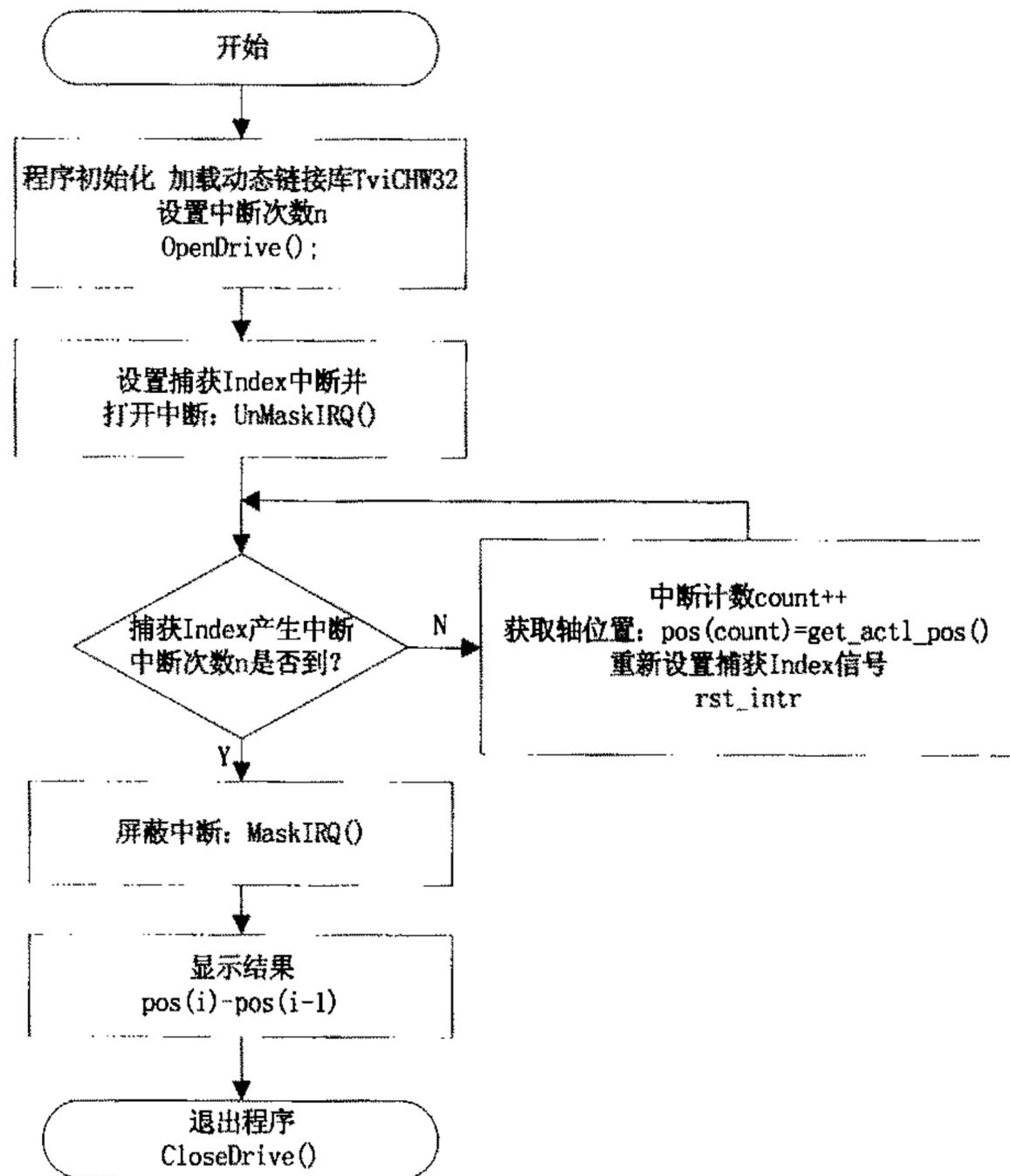


图 4.4 编码器测试程序流程图

测试时采用的编码器每转输出脉冲数为 360。因为运动控制器对编码器输入信号进行了四倍频，所以每两个同步信号之间控制器接收到的脉冲计数应为  $360 \times 4 = 1440$ ，即每四个

脉冲对应主轴转过 1 度。程序运行结果如图 4.4 所示, 可以看到主轴每转一圈输出脉冲数均为 1440, 从而保证了主轴同步和采样信号的准确。

### § 4.3 正弦信号的两轴联动控制实验

#### 4.3.1 控制器的运动轴状态

GT400-SV 运动控制器为每一个轴提供了一个状态寄存器, 用来保存各轴的运动状态。该状态寄存器是一个 16 位寄存器, 用 `get_status()` 命令可以查询当前寄存器的内容。状态寄存器的各标识位定义如表 4.3 所示。标识位 8-10 位和 12-13 位指示连续状态信息, 主机不能对它们进行复位, 而 0-7 位标识轴的不同事件状态, 它们能向主机提出中断申请。这些标识位由运动控制器置位, 并必须由主机复位, 否则它们将一直保持有效。

表 4.3 轴状态寄存器定义

位	含 义						
0	运动完成标识位。如果轴的运动完成, 该位置 1。该标识位只在 S 曲线, 梯形曲线及速度跟踪三种运动方式时有效。						
1	轴伺服驱动器报警标识位。如果轴伺服驱动器报警, 该位置 1。						
2	断电到达标识位。在设置断点条件下, 断点到达该位置 1。						
3	Index/Home 标识位。在设置位置捕获命令后, 控制器检测到 Index/Home 将该位置 1。						
4	运动出错标识位。如果位置误差超过允许范围, 控制器将该位置 1。只有轴不再处于运动错误状态时, 才能对它复位。						
5	正限位开关触发标识位。如果正限位开关被触发, 该位置 1。						
6	负限位开关触发标识位。如果负限位开关被触发, 该位置 1。						
7	命令出错标识位。如果出现命令错误, 控制器将该位置 1。						
8	电机 on/off 状态 (1 表示 on, 0 表示 off)。						
9	轴 on/off 状态 (1 表示 on, 0 表示 off)。						
10	运动状态标识位。它连续指示轴是否在运动。如果在运动, 为 1; 如果静止, 为 0。						
11	限位开关有效标识位。						
12, 13	当前轴号标识 (13bit=高位, 12bit=低位)。当前轴号的编码如下 <table style="margin-left: auto; margin-right: auto; border: none;"> <tr> <td style="padding: 0 10px;">Bit13</td> <td style="padding: 0 10px;">Bit12</td> <td style="padding: 0 10px;">轴</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> </tr> </table>	Bit13	Bit12	轴	0	0	1
Bit13	Bit12	轴					
0	0	1					

	0	1	2
	1	0	3
	1	1	4
14	设定 Home 信号捕获标识。		
15	设定 Index 信号捕获标识。		

在程序中，无论是对轴的初始化、设置中断，还是判断控制器触发何种中断都需要对轴的状态进行判断，因此轴的状态字是整个程序编写的一个主要依据。

### 4.3.2 程序实现及流程图

控制程序的核心是响应主轴的采样信号和同步信号中断并进行相应的处理（发送数据和轴位置清零）。

#### 1. 采样信号

对采样信号的中断采用轴断点事件中中断的方法。GT400-SV 运动控制器提供了两种断点，一种是参数断点，即：主机设定一个断点位置比较值，当轴运动到达该位置时触发断点；另一种是为事件断点，即：主机设定事件断点模式，当事件满足时触发断点。事件断点主要包括轴运动完成事件断点和原点（Home）开关触发事件断点。考虑到主轴每转过一个固定的角度时产成一个采样信号，所以只能将断点设置为参数断点。主机采用 `set_brk_cn(pos)` 命令设置轴断点目标位置（`pos` 为断点目标位置，也就是控制器接收到的脉冲数），`set_pos_brk()` 命令设定轴断点为正向位置断点。当运动轴实际位置大于或等于设定的断点位置时触发该断点并清除轴断点标识，同时将轴状态字的 `bit2` 置 1。

对于 360 点的编码器，主轴每转过一度对应运动控制器接收到 4 个脉冲（控制器对输入四倍频）。程序中设定主轴每转过 5 度产生一个采样中断，因此每次采样中断发生时将断点目标位置 `pos` 设为当前的断点目标位置加 20。

#### 2. 同步信号

GT400-SV 运动控制器允许使用增量编码器的 Index 信号触发中断并捕获轴的位置。Index 信号来自增量编码器的 C 相信号，当该信号和 A、B 相的四倍频信号均为低电平时产生 Index 信号，Index 信号触发轴的高速位置捕获寄存器获得当前轴的实际位置。该信号

即可作为主轴的同步信号。主机通过 `set_capt_index()` 命令设置捕获 Index 信号，并将轴状态字的 bit15 置 1。当运动控制器捕获到 Index 信号后，轴状态寄存器的 Index/Home 标识位 (bit3) 置 1，并清除状态寄存器的 Index 捕获或 Home 捕获设定标识位 (bit15)。主机必须重新设定 Index 捕获命令才允许下一次 Index 信号捕获发生。控制器所捕获的位置是 Index 触发脉冲到来时刻的轴当前实际位置，捕获位置精度为  $\pm 1$  个脉冲。

GT400-SV 运动控制器捕获位置寄存器采用硬件完成，因此轴的运动速度不影响控制器的捕获精度。

### 3. 程序流程图

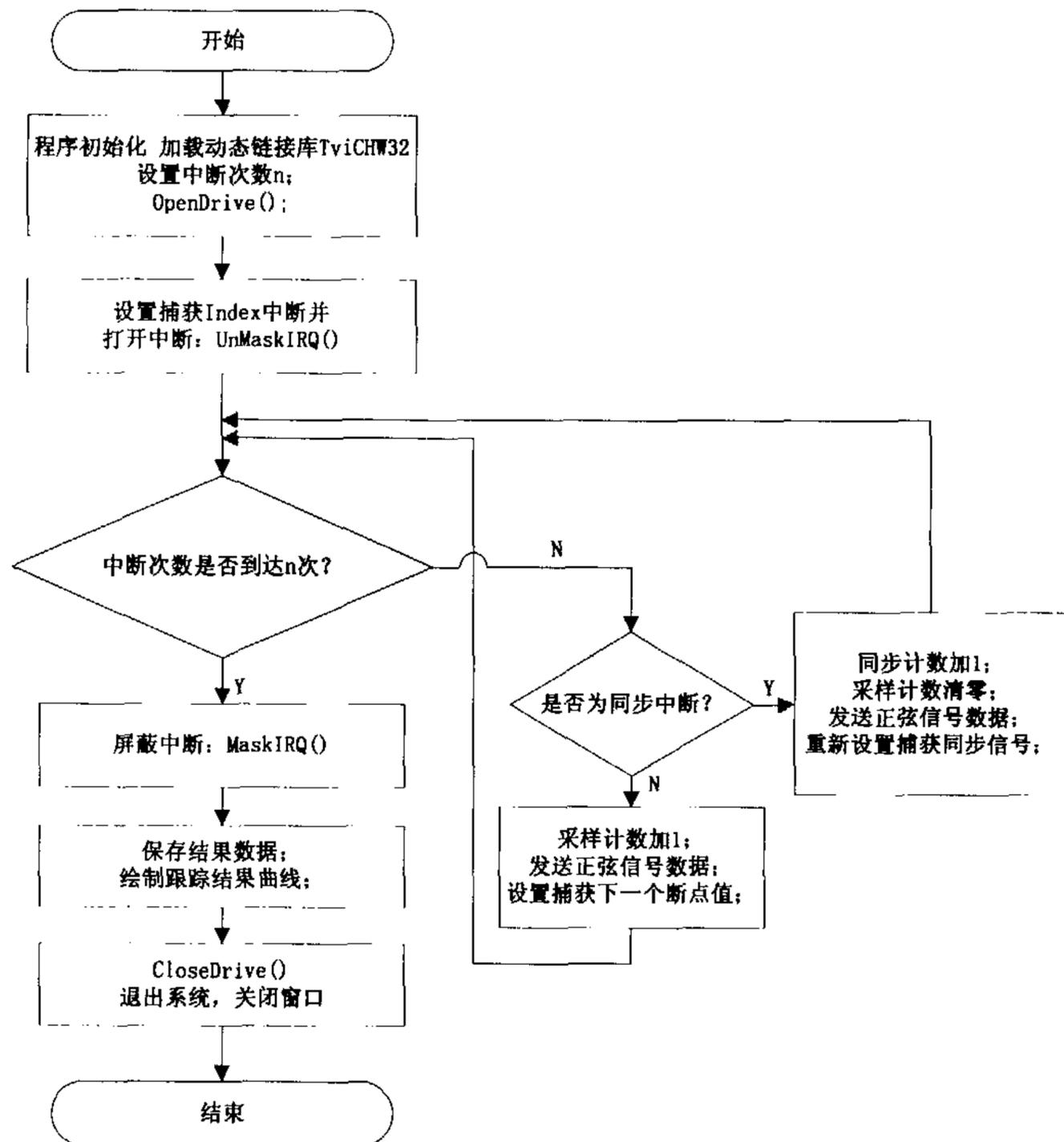


图 4.5 正弦信号控制程序流程图

### 4.3.3 轴运动出错问题的解决方法

GT400-SV 运动控制器为其伺服控制轴提供了轴运动出错监控功能。对于闭环伺服控制而言,在某些情况下,轴的实际位置可能和期望位置差别很大。这时通常意味着一些危险情况存在,例如电机或编码器故障,编码器 A, B 信号接反或断线,机械摩擦太大或机械故障造成运动轴阻塞等。为了能检测这种情况,增强系统的安全度和延长设备使用寿命,GT400-SV 运动控制器中设置了可编程修改的轴最大误差值。运动控制器在每一个伺服周期内都通过比较最大位置误差值与实际位置误差值来检测是否出现运动错误。如果轴的实际位置误差超过了控制器设置的最大位置误差值,控制器就认为轴出现了运动错误。发生运动错误时,控制器将轴状态寄存器中运动错误标识位(bit4)置 1,如果轴模式寄存器中运动出错停止标识位置 1,则轴自动停止。

虽然这一保护措施极大地保障了系统安全,但却带来了一个问题。因为运动控制器没有对主轴发送位置控制命令,而只是获取其实际位置,所以主轴的目标位置一直为 0,因此主轴的实际位置误差就等于主轴的实际位置,随着主轴的旋转,实际位置误差不断增大,当超过 32768(可设置的最大位置误差值,为默认值)时将产生运动出错事件,使程序无法正常运行。

控制程序中,通过在中断的服务程序中调用函数 `synch_prfl()` 使主轴的目标位置寄存器设置为实际位置寄存器的位置值,从而使轴的实际位置误差一直小于 32768。GT400-SV 运动控制器采用双缓冲机制把轴参数和相关的控制命令下载到运动控制器中。控制器在没有接收到主机的命令之前,这些下载的参数和运动命令均不生效。因为 `synch_prfl()` 是一个双缓冲命令,因此必须调用 `update()` 函数进行刷新。控制器接收到主机的刷新命令之后,将在下一个伺服控制周期将该命令复制到有效的寄存器中,使其生效。

### 4.3.4 不同频率下的信号跟踪结果

#### 1. 使用 360 点编码器

360 点的编码器每转一圈输出 360 个 A 相或 B 相脉冲,通过运动控制器的四倍频后,位置计数器的计数为 1440 脉冲/转。设采样间隔为 5 度,则断点间隔应设为 20。不同频率下的跟踪结果如下。图中横坐标为采样点数,纵坐标为幅值(脉冲数)。

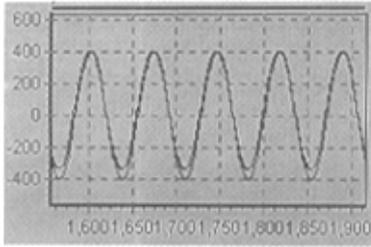


图 4.6 360 点编码器、5Hz

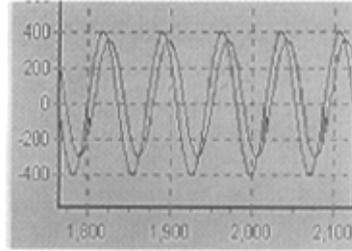


图 4.9 360 点编码器、20Hz

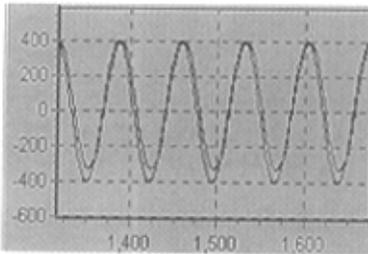


图 4.7 360 点编码器、10Hz

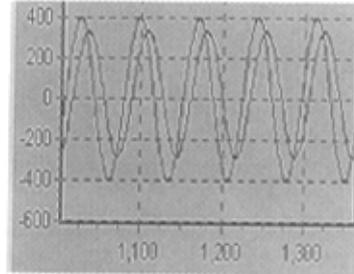


图 4.10 360 点编码器、25Hz

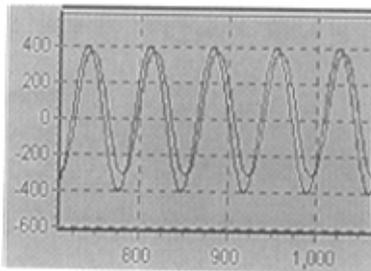


图 4.8 360 点编码器、15Hz

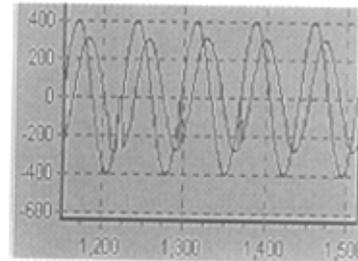


图 4.11 360 点编码器、30Hz

从图 4.6-图 4.11 可以看出实际的正弦信号与发送的正弦信号之间的幅值与相位是不断变化的,随着频率的增大,实际的正弦信号幅值逐渐降低,相位滞后也越来越大。当频率达到 20Hz 时,出现了丢点的现象。

## 2. 使用 72 点编码器

72 点的编码器每转一圈输出 72 个 A 相或 B 相脉冲,通过控制器的四倍频后,位置计数器的计数为 288 脉冲/转。设采样间隔为 5 度,则断点间隔应设为 4。不同频率下的跟踪结果如下。图中横坐标为采样点数,纵坐标为幅值(脉冲数)。

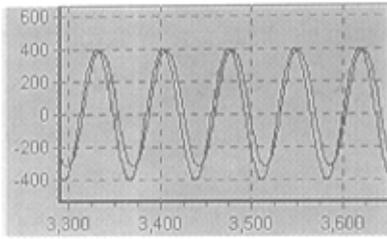


图 4.12 72 点编码器、10Hz

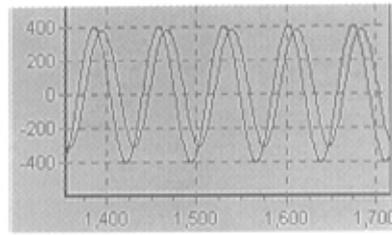


图 4.15 72 点编码器、25Hz

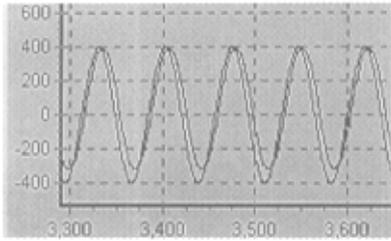


图 4.13 72 点编码器、15Hz

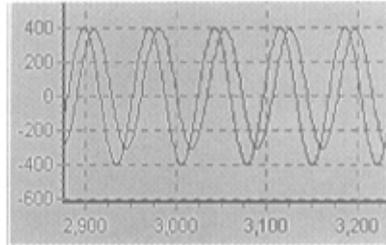


图 4.16 72 点编码器、30Hz

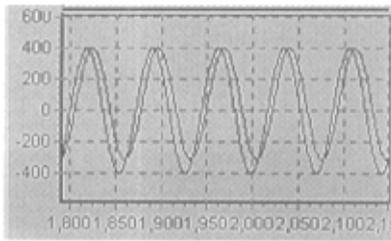


图 4.14 72 点编码器、20Hz

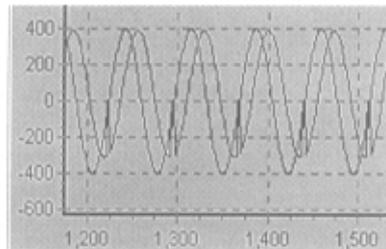


图 4.17 72 点编码器、35Hz

从图 4.12-图 4.17 可以看出实际的正弦信号与发送的正弦信号之间的幅值与相位是不断变化的,随着频率的增大,实际的正弦信号幅值逐渐降低,相位滞后也越来越大。当频率达到 35Hz 时,出现了丢点的现象。

### 4.3.5 相位校正

从前面的正弦信号跟踪实验可以到相位随着频率 $\omega$ 的不断提高而产生越来越大的滞后,在非圆截面零件加工过程中,这种滞后的相位差会导致工件表面形状在横截面上的分布产生偏移,严重影响加工质量,因此必须对微进给装置控制系统实行相移校正控制。可实现控制系统超前校正的途径有以下三类:一是模拟式超前校正;二是数字式超前校正;三是超前输出控制。

倘若微进给系统工作在某一固定频率处时,用模拟式超前校正网络可获得很好的相移校正效果,但要在较宽的通频带内都达到零相位差的校正仍存在着较大的困难。

在数字式超前校正控制结构中,由于采用计算机来实现数字相移校正,因此可根据控制对象的相频特性确定通频带内各频率点的校正参数,对于不同频率的输入信号,可使用对应的校正参数来实现相移校正,从而达到全通频带内实现零相位跟踪的控制效果。

超前输出控制的原理是提前输出目标控制量,由此来减少控制系统的相位误差.这一控制策略是通过多次微进给控制实验总结出来的,并且在微进给装置控制过程中能取得良好的相位补偿效果。设信号频率为 $\omega$ ,微进给系统输出相位滞后量为 $\phi$ , $T$ 为采样周期,则超前输出量为下式所描述,其中 $X_0(kT)$ 为预输出量,而 $X_r[(k+N)T]$ 为提前输出量。

$$\begin{cases} X_0(kT) = X_r[(k+N)T] \\ N = \frac{\phi}{\omega T} \end{cases}$$

对应不同的频率,只要得到输出的相位滞后量为 $\phi$ ,就可以求出提前输出量 $X_r[(k+N)T]$ 进行相移校正。但由前面的正弦信号跟踪结果可以看出,随着频率的不同,系统输出的相位滞后是变化的。为了能够在不同频率的输入信号时使用对应的校正参数来实现相移校正,可以将变频器的控制引入主机中。这样就可以通过频率的设置查表得到对应的校正参数实现校正,或者每次更改主轴频率时求得校正参数再进行加工。对变频器的通信方法见附录。

## § 4.4 活塞数据的两轴联动控制实验

### 4.4.1 活塞数据来源

前面的程序中实现了主轴采样和主轴同步中断,并用正弦信号进行了测试。下面的实验使用实际的活塞数据按照活塞加工运动过程控制直线电机的进给运动。

对于一个指定的活塞,用户首先输入原始型线数据、原始椭圆度数据和原始截面数据,然后通过型线插值、椭圆度插值、截面数据处理得到相应的数据,最后将以上的数据合成得到活塞加工数据。因为对活塞数据的处理不属于本课题的研究范围,因此实验中调用教研室相关课题处理后生成的活塞数据进行实验。

### 4.4.2 瑞软表格控件

列表控件是应用系统最常用的一种控件，一个高效、美观、功能强大的列表控件是创建一个优秀系统的关键。用表格来组织、管理数据，对数据进行访问是编程的常用方法，可以大大提高编程的效率。为了使用相关课题得到的活塞数据，这里采用了瑞软表格控件。

RsGrid 控件是由 ReadSoft 公司开发的 ActiveX 控件产品，RsGrid 控件不仅具备基本的列表控件功能，还提供了许多优秀的列表新特性：

- 漂亮高效的表格显示功能
- 强大的数据库功能
- 强大精致的报表功能
- 丰富的数据类型
- 灵活的数据操纵功能。

使用 RsGrid 控件可以对表格数据进行编辑，保存，打印等，极大的提高了编程效率，也使得软件界面美观漂亮。

#### 4.4.3 程序实现及流程图

##### 1. 程序结构及功能

程序结构如图 4.18 所示：

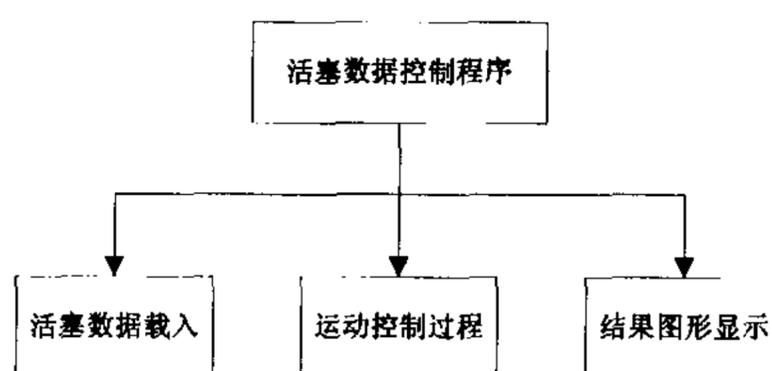


图 4.18 活塞数据控制程序结构

系统初始化时载入 Tวิช32 动态链接库，并对主轴和直线电机的参数进行初始化。各部分的主要功能为：

##### 活塞数据载入部分

活塞数据载入部分通过瑞软表格控件将存储在硬盘中的活塞数据载入到系统初始化部分为活塞数据分配的内存空间中，实验中载入 100 个截面的活塞数据。

## 加工控制过程部分

加工控制部分与正弦信号测试程序基本相同。在发送活塞数据控制直线电机运行的同时，读取每个采样点时刻直线电机的实际位置并存储到系统初始化部分为返回值分配的内存空间中。

## 结果图形显示部分

结果图形显示部分利用活塞数据和返回值绘制椭圆曲线，并可以对返回数据进行相位补偿。

## 2. 系统流程图

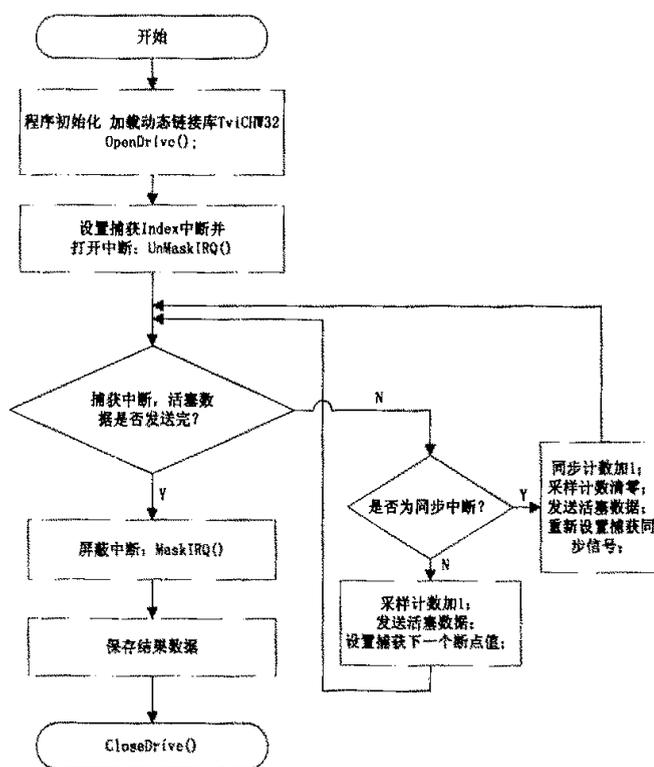


图 4.19 活塞数据控制程序流程图

## 4.4.4 活塞数据控制结果

实验中采用了 360 点的编码器，并针对主轴的不同运动情况得到不同的活塞数据跟踪结果。主要有以下三种情况：

## 1. 跟踪从主轴电机启动开始直到稳定在 30Hz

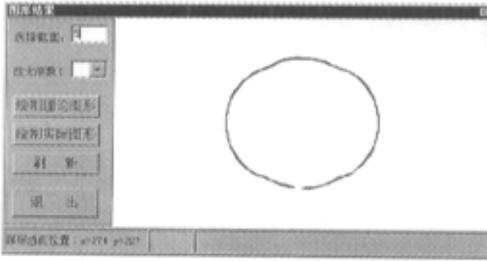


图 4.20

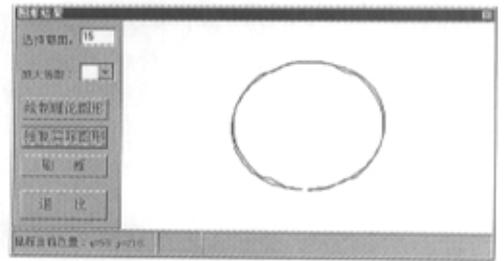


图 4.21

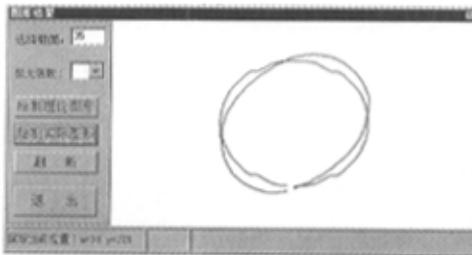


图 4.22

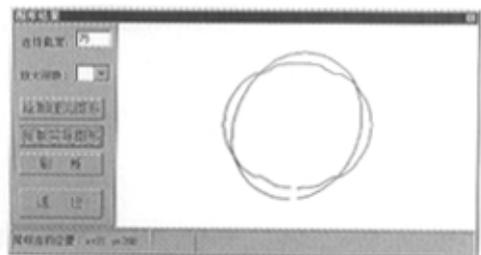


图 4.23



图 4.24

图 4.20-1.23 是从根据返回值画出的 100 个截面中具有代表性的四个截面。因为在实验中电机从启动到转速稳定在 30Hz 的过程中频率逐渐变大，所以随着截面数的增大（对应于频率增大）所画出的椭圆旋转角度变大（对应相位滞后变大）。

图 4.24 是由一系列根据返回值得出的截面所组成的，从图中可以直观地看到不同频率引起不同的相位滞后，从而导致不同截面的椭圆转过不同的角度。

## 2. 主轴电机稳定运行于 5Hz

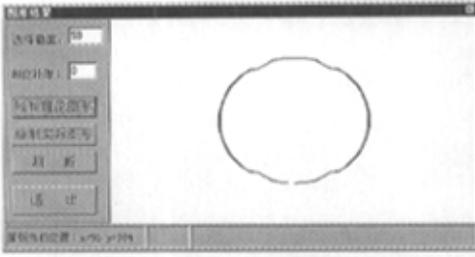


图 4.25

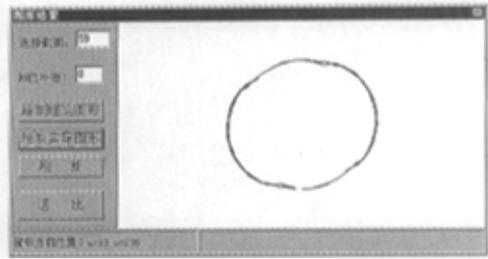


图 4.26

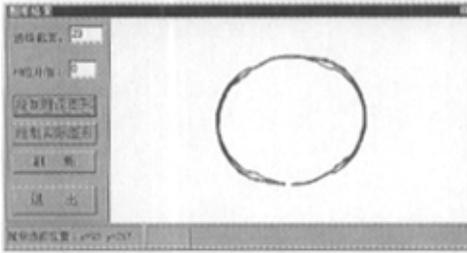


图 4.27

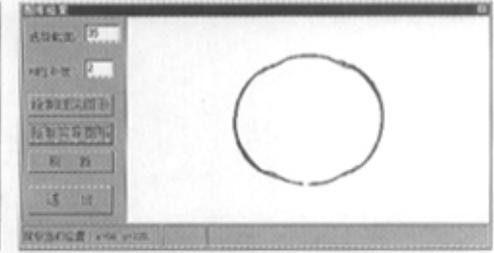


图 4.28

图 4.25 是由根据载入的活塞数据绘出的 5 个椭圆所组成，分别为截面 5、15、25、35、45。图 4.26 为根据返回数据绘出的与图 4.24 对应的 5 个截面椭圆。图 4.27 为图 4.25 和图 4.26 的合成。图 4.28 是通过图 4.27 中的实际椭圆进行相位补偿后得到的。可以得出结论，当程序运行在较低的频率时（5Hz），理论图形与实际图形有较好的吻合。

### 3. 主轴电机稳定运行于 30Hz

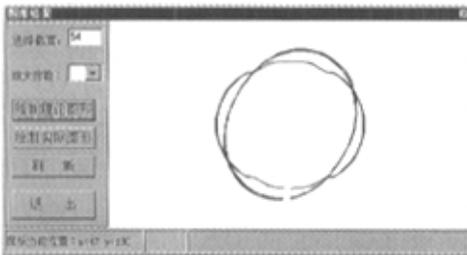


图 4.29

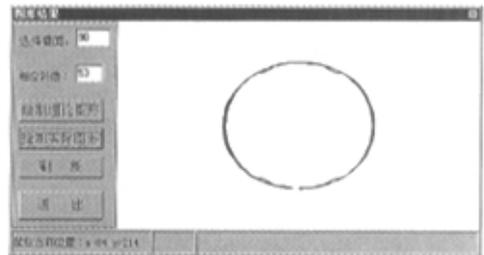


图 4.30

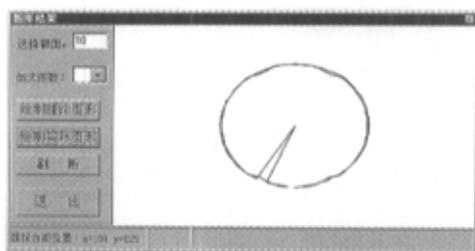


图 4.31

图 4.29 是由理论截面 5、15、25、35、45 与实际截面 5、15、25、35、45 所组成的。图 4.30 则为相位校正后的结果。

从图 4.29-4.30 可以看出，当程序运行在 30Hz 时，系统产生了较大的相位滞后。另外实际曲线与理论曲线相比有较大的失真，并且出现了丢点的现象，如图 4.31 所示。

#### § 4.5 本章小结

本章首先介绍了活塞加工运动形成的原理，分析得出如何响应主轴的采样和同步中断为控制程序的核心内容。然后给出了在 C++Builder5.0 中如何利用 TvicHW32 控件包对中断进行响应的方法，并根据需要编制了运动控制器时间中断、事件中断测试程序以及编码器测试程序。在两轴联动控制过程中，分别采用正弦信号和活塞数据进行实验。

## 第五章 实验结果分析

### § 5.1 ADSP2181性能分析

因为 GT400-SV 四轴运动控制器的核心是美国 AD 公司的 ADSP2181 数字信号处理器，所以首先对该芯片是否适用于活塞加工过程进行分析。采样周期  $T$  在计算机控制系统中是一个重要参数，从信号的保真度来考虑采样周期不宜太长，采样定理给出了下限频率的公式： $\omega_s \geq 2\omega_m$ ， $\omega_m$  是被采样信号的最高频率， $\omega_s (=2\pi/T)$  为采样的角频率。从控制性能来考虑，采样周期  $T$  应尽可能地短，也即  $\omega_s$  尽可能的高，但是采样频率越高，对处理器的性能要求也就越高。

ADSP2181 数字信号处理器的指令周期为 33ns。如果取采样频率  $f=100\text{KHz}$ ，则在一个采样周期内 DSP 程序指令的为 300 条。

下面计算对主轴运行于不同频率时采样周期内 DSP 程序指令数目：

设：

主轴转速： $n=3000\text{rpm}$ ；

活塞截面型线采样点数： $N=72$ ；

单位置指令周期内 PID 的调节次数： $M=20$ ；

单调节周期内 2181 的指令周期数为  $I$

采样周期  $T$  间隔内直线电机得最大位移量： $L=V_{\max}T$

取  $V_{\max}=83.8\text{mm/s}$  ( $n=2000\text{rpm}$ 、 $e=0.8$ 、 $c=0$ )

则：采样周期  $T$ 、采样频率  $f$  可计算如下

(1)  $M=20$  时

$$f = (n/60) \times M \times N$$

$$= 72000 (\text{Hz})$$

$$T = 1/f$$

$=13.889(\mu s)$

$I=416$ (条)

$L=1.16\mu m$  ( $n=2000rpm$ 、 $e=0.8$ 、 $c=0$ )

(2)  $M=30$  时,

则  $f=108(kHz)$ ,  $T=9.259(\mu s)$ ,  $I=278$ (条)

$L=0.775\mu m$  ( $n=2000rpm$ 、 $e=0.8$ 、 $c=0$ )

(3)  $M=40$  次时,

则  $f=144(kHz)$ ,  $T=6.944(\mu s)$ ,  $I=208$ (条)

$L=0.582\mu m$  ( $n=2000rpm$ 、 $e=0.8$ 、 $c=0$ )

由以上的计算可以看出, ADSP2181 的性能可以满足活塞加工的要求。控制系统的采样周期  $T$  具有较宽松的选择环境。但由于系统使用的是基于 ADSP2181 的 GT400-SV 运动控制器, 它只提供了时间中断和事件中断来响应主轴的采样或同步信号, 所以下面对其采样周期进行分析。

## § 5.2 运动控制器采样周期

### 5.2.1 采用时间中断控制

GT400-SV 运动控制器使用 `set_timer_intr()` 命令选择中断方式为定时中断, 定时中断的周期由控制器的伺服刷新周期和 `set_intr_time()` 命令的设定值共同确定。例如, 控制器的伺服刷新周期为 100 微秒, 主机设定 `set_intr_time(10)`, 则定时中断地周期 =  $10 \times 100$  微秒, 即 1 毫秒。而运动控制器的伺服刷新周期由命令 `set_smpl_time(i)` 确定, 伺服刷新周期 =  $i \times 25$  微秒。

下面的实验通过定时中断判断主轴是否到达采样位置, 如果是则发送正弦信号值。实验中使用 72 点编码器, 设定主轴转速为 30Hz, 并通过以下命令:

```
set_intr_time();
```

```
set_smpl_time(2);
```

```
set_intr_time(2);
```

设定运动控制器的定时中断周期  $T=25*2*2=100$  毫秒。图 5.1 为程序流程图，运行程序后通过示波器观察发现发送的正弦信号有丢点现象。

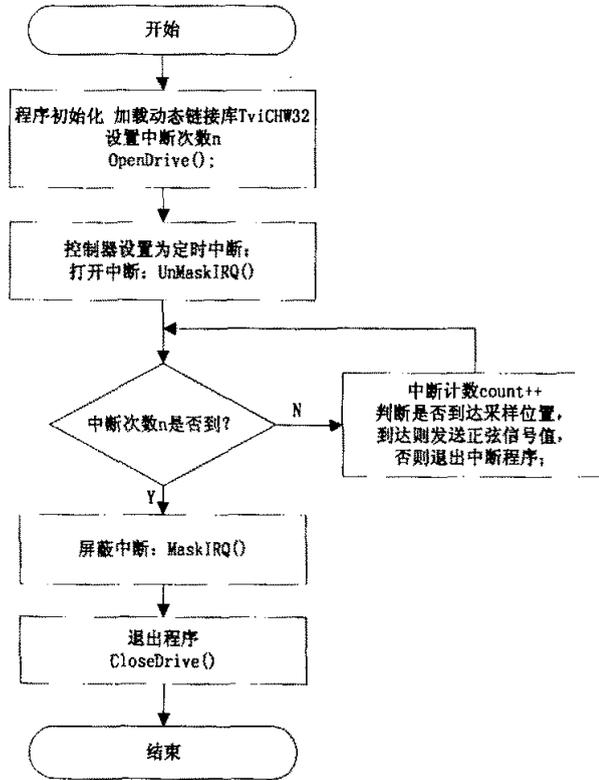


图 5.1 定时中断控制程序流程图

分析:

定时中断（采样信号）频率： $1/T=10000\text{Hz}$ ;

主轴位置信号（被采样信号）频率：主轴转速  $30\text{Hz}$ ，即每秒 30 转。系统采用 72 点的编码器，每转输出 72 个 A 相或 B 相脉冲，通过控制器四倍频后主轴每旋转一周运动控制器位置计数为 288。所以主轴位置信号的频率为  $30*288=8640\text{Hz}$ 。

可见采样信号的频率与被采样信号频率接近，所以程序执行过程中会发送丢点现象。

### 5.2.2 采用断点中断控制

因为实际加工过程是根据网格化的活塞位置发送活塞数据的，所以不能以定时的间隔来进行控制。这就要求必须利用运动控制器的事件中断进行控制。

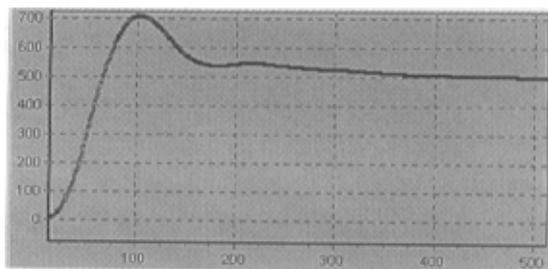


图 5.2 采用定时中断的阶跃响应

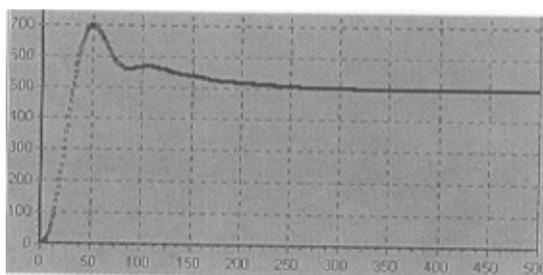


图 5.2 采用断点中断的阶跃响应

为了确定断点中断与时间中断的关系,通过实验得到如图 5.2 和图 5.3 所示的阶跃响应结果。图中横坐标均为中断次数,纵轴显示位置值,单位是脉冲数。两次实验在相同的条件下进行的, PID 参数均为  $k_p=10$ ,  $k_i=1$ ,  $k_d=1$ 。由图中可以看到采用定时中断时中断发生 100 次阶跃响应达到峰值,而采用断点中断时中断发生 50 次阶跃响应达到峰值。因此可以得出结论,断点中断响应速度约为定时中断的一半。

因此,采用运动控制器的时间中断或断点事件中中断来响应主轴的采样和同步信号是难以达到要求的。这一结果与第四章的实验吻合,在低频时能够取得比较理想的效果,频率较高时有明显失真甚至丢点现象。例如,在采用活塞数据的控制过程中,主轴运行于 30Hz。实验从启动主轴开始发送活塞数据,频率逐渐升高并达到 30Hz,其结果如图 5.3 所示,可以看到在频率较低时,理论椭圆与实际椭圆相吻合,随着频率逐渐升高实际椭圆失真越来越大。由于随着频率升高相位滞后越来越大,所以图中实际的椭圆随着主轴频率的升高转过的角度也越来越大。

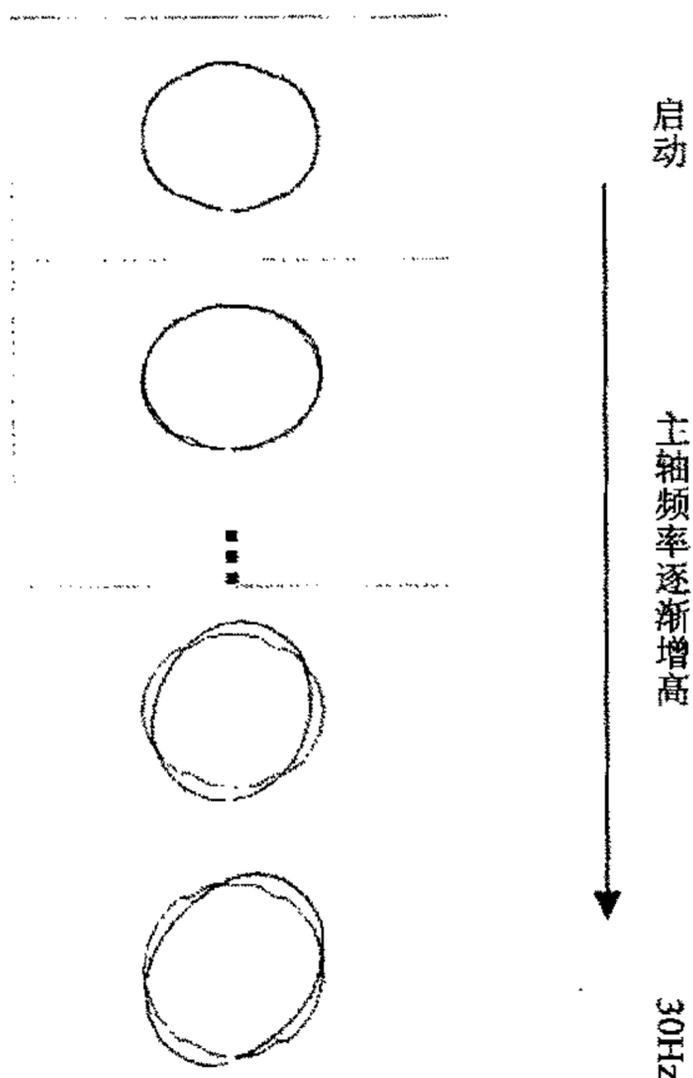


图 5.3 主轴频率对实验结果的影响

### § 5.3 本章小结

本章结合实验对采样频率进行了分析。通过分析可知，对于 GT400-SV 运动控制器仅利用其已有的时间中断及事件中断进行活塞加工的控制是难以达到要求的。因此，如何利用现有运动控制器充分发挥其数字信号处理芯片的高性能来满足活塞加工的要求将是需要进一步探讨的问题。

## 第六章 结论与展望

课题根据活塞加工运动的原理, 研究采用以 ADSP2181 数字信号处理器为核心的 GT400-SV 运动控制器控制活塞加工过程的可行性。在实验中分别采用正弦信号与实际活塞数据对系统进行了控制, 根据实验结果并通过分析得出以下结论:

1. 作为 GT400-SV 运动控制器的核心, ADSP2181 数字信号处理器具有较高的性能, 完全满足活塞加工运动的频率要求。

2. GT400-SV 运动控制器作为一种通用的控制器, 它提供了时间中断及事件中断两类中断处理方法, 但不论哪种中断方法能够响应的频率都达不到活塞加工的要求, 这就导致了实验中在主轴频率较低时能够取得较好的效果, 而当增大频率达到活塞加工要求时, 系统产生了很大的失真, 中断发生丢点现象。

因此, 为了充分发挥 ADSP2181 的性能, 从而将 GT400-SV 运动控制器应用于活塞加工系统中, 就必须改变运动控制器响应中断的方式。为了使轴的伺服运动更加平稳, GT400-SV 运动控制器对编码器输入信号进行 4 倍频使轴运动控制分辨率提高四倍。这对于直线电机的控制是完全必要的, 但是对于主轴的运动来说却增加了位置信号的频率, 例如使用 360 点的编码器则每转产生 1440 个位置信号, 增大了中断响应的难度, 而且也是不必要的 (一般活塞加工时, 主轴每转过 5 度进行一次采样, 相当于上述编码器的 20 个位置信号)。另外考虑到主机与运动控制器之间数据传输对系统性能的影响, 下面提出对运动控制器及控制程序进行改进的思路:

1. 提高运动控制器的伺服刷新周期, 充分利用 ADSP2181 数字信号处理器的性能。

2. 去掉对主轴编码器信号的四倍频, 直接利用编码器输出的 A/B 相信号作为中断请求信号, 也就是主轴的采样信号。

3. 将控制过程中每次主轴采样时由主机发送活塞数据到控制器进行控制改为程序初始化时一次将活塞数据从主机载入控制器的存储器中, 以使每次采样时控制器自动发送数据。若数据量较大也可将活塞数据分成若干部分, 在控制过程中不断载入控制器存储器。

在课题的最后阶段, 尝试着按照上述的思路对控制程序进行改进并进行实验。控制程

序的流程如图 6.1 所示。

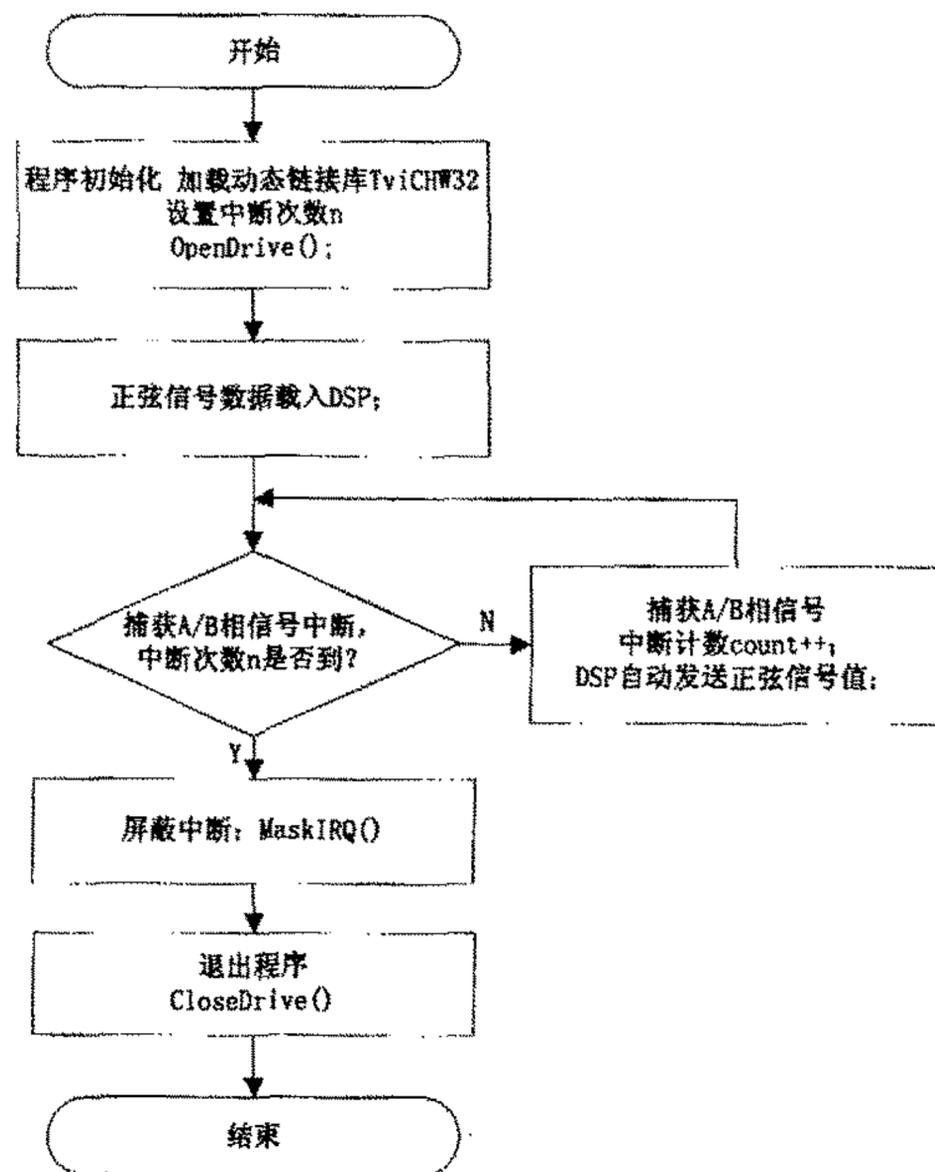


图 6.1 改进后控制程序流程

程序执行后,通过示波器观察得到比较理想的波形,相信按照以上的思路可以将改进后的运动控制器应用于活塞加工过程中。由于时间的关系本课题没有做进一步的实验和分析。

## 参 考 文 献

- [1]杨舜洲 陈志华 崔红娟《PTC-IV活塞车削控制系统》《发动机配件技术》，1997年第1期
- [2]黄金秋. 基于开放式结构的高性能数控系统的研制. 制造技术与机床. 1998(8)
- [3] GT400-SV 四轴伺服电机运动控制器用户手册 固高科技有限公司
- [4]王先逵、陈定积、吴丹《机床进给系统用直线电动机综述》《制造技术与机床》2001年第8期
- [5]ADSP-2111 User's Manual March 1990
- [6] Creppe Renato C, de Souza, Carlos R. 《Dynamic behavior of a linear induction motor》. 《Proceedings of the Mediterranean Electro-technical Conference》 IEEE picataway, NJ, USA, 98Ch36056, 1998, 2:1047-1051
- [7]蒙以正 著《MATLAB 5. X 应用与技巧》科学出版社, 1999年12月
- [8]沈美明、温冬婵 著《80X86 汇编语言程序设计》清华大学出版社 2001年9月
- [9]WilliamBuchanan. PC Interfacing, Communications and Windows Programming, Addison Wesley Longman Inc, 2001 :230-247.
- [10]三菱变频调速器 FR-E500 使用手册:108-118
- [11]张万里, 陈战林. C++Builder5. 0 高级开发技巧与范例. 北京:电子工业出版社, 2001.
- [12]陈立定. Windows98 下单台 PC 与多台变频器的串行通信控制. 制造业自动化, 2001(03)
- [13] 胡均安, 张道德, 刘革新, 杨光友, 周国柱. 用单片机实现对变频器的网络控制. 制造业自动化, 2000(11)
- [14]孟威勇《直线伺服系统的特性研究及输入数据的预处理》国防科技大学硕士学位论文, 1998年1月
- [15]RG2 Encoder System RGH22 READHEADS Installation Guide
- [16]施文康 余晓芬 著《检测技术》机械工业出版社, 2000年5月
- [17] 郭庆鼎, 王成元, 周美文, 孙廷玉著《直线交流伺服系统的精密控制技术》机械工业出版社, 2000。
- [18] 袁俊哲, 王先逵著《精密和超精加工技术》机械工业出版社, 1999年7月

- [19]孙增圻等 著《智能控制理论与技术》清华大学出版社, 广西科学技术出版社, 1997年4月
- [20]冯勇等 著《现代计算机控制系统》哈尔滨工业大学出版社, 1997年1月
- [21]何克忠 李伟 著《计算机控制系统》 清华大学出版社 1999年
- [22]邓中亮 王先逵《微进给装置的计算机数字控制系统的设计与实现》《自动化学报》990218
- [23] 则次俊郎, 进行波超音波モータを用いた小型マニピュレータの制御性能, 日本机械学会论文集(C)篇, 1995, 61(581):124-130
- [24] 范大鹏, 尹自强, 郑子文 《直线电机在精密加工的应用》 《制造技术与机床》 1997年第5期
- [25] 杨正新, 陈志华、杨舜洲、崔红娟、涂阳虎 《直接驱动的发展与未来》《中国机械工程》2001年第5期
- [26] 胡敏强 《超声波电动机的研究及其应用》《微特电机》2000第5期
- [27] 杨叔子, 杨克冲 著《机械工程控制基础》华中理工大学出版社 1984年4月
- [28] 李智慧 秦成 著《C++Builder4 从入门到精通》清华大学出版社, 1999年8月
- [29]张屹 杨波 戴路 著《WINDOWS9X、NT 及 WINDOWS2000 下对硬件的直接操作访问》《微计算机信息》, 2000年第5期
- [30]杨强 李堂秋 著《WIN9X 虚拟设备驱动程序编程指南》清华大学出版社, 1999年6月
- [31]郑子文《非圆异型零件精密磨削加工及控制技术》国防科学技术大学硕士学位论文 1998年月1月
- [32]DAREN HAZZAH 著, 孙喜明 译《WINDOWS VxD 与设备驱动程序权威指南》中国电力出版社, 2001年5月
- [33]陈志华《非圆截面零件车削控制研究》国防科学技术大学工学博士学位论文 2000年5月
- [34]仲志丹、张洛平、张青霞《PID 调节器参数自寻优在运动伺服控制中的应用》《洛阳工学院学报》2000年第21卷
- [35] H. Yonezaz, Y. Hirata, H. Sasai, Positioning Table with High Accuracy and High Speed, Annals of the CIRP, Vol. 39/1/1990

- [36]谭浩强 著《C 程序设计》清华大学出版社, 1993 年 6 月
- [37]薄建全《非圆表面数控车削系统的实验研究》清华大学硕士学位论文 1998 年 5 月[2]
- [38]魏克新 王云亮 陈志敏 著《MATLAB 语言与自动控制系统设计》机械工业出版社, 1997 年 8 月
- [39]张雄伟 曹铁勇 著《DSP 芯片的原理与开发应用》电子工业出版社, 2000 年 9 月
- [40]东北工学院机械设计、机械制图教研室 编《机械零件设计手册》冶金工业出版社, 1974 年 4 月

## 致 谢

本文是在导师杨正新副教授的悉心指导下完成的，导师严谨的科研作风和治学态度使我终身受益，在此谨致以深切的谢意！在课题的进一步研究中得到了本教研室杨舜洲教授、崔红娟副教授的关怀与帮助，解旭辉副教授无私地为课题提供了运动控制器，并且非常耐心细致解答板子上所遇到的疑问，在此也表示我最真诚的感谢。

另外还要感谢本室的刘双百老师、戴光华老师、刘新辉老师、杨良才老师、吴汉春老师、邢兴东、张翊成老师。感谢舍友郜业猛、杨帆，因为课题的相关性，从他们那里获得了很大的帮助。

在生活上还要感谢队干部的关心和帮助！

## 附录 变频器通信方法的研究

交流电机变频调速具有节能降耗、实现工业自动化和提高生产率的显著特性，广泛应用于以节能和无级调速为目的的调速控制。在多变频器系统中，运用数字通讯技术组成变频器网络，采用数字信号传输技术，可使网络的分散性与变频器本身的自治性达到完美统一，在提高生产效率的同时降低成本。针对这一需要，德国西门子、日本东芝、三菱等公司推出了带有 RS-485 通信接口的变频器，使用户能方便灵活地利用变频器的强大功能来设计各自的工业控制系统。

线程是 32 位操作系统的主要特点，它支持抢占式的多任务机制，是操作系统的基本调度单元。可以在控制系统软件中将各管理和控制模块置于独立的线程中，从而实现数控软件的并行处理操作。

下面对如何在 C++Builder5.0 中实现基于线程的 PC 对多台变频器的异步串行通信进行了研究。

### 1 系统结构

图 1 为系统的网络结构，PC 机的 RS-485 接口通过分配器与多台 FR-E500 变频器的 PU 接口相连接，最多可达 32 台。RS-485 作为一种多点、差分数据传输的电气规范现已成为业界应用最为广泛的标准通信接口之一。这种通信接口允许在简单的一对双绞线上进行多点、双向通信、它所具有的噪声抑制能力、数据传输速率、电缆长度及可靠性是其他标准无法比拟的。系统中每台变频器具有不同的站号，这样 PC 便能通过 RS-485 通信线对连接的所有变频器进行控制。

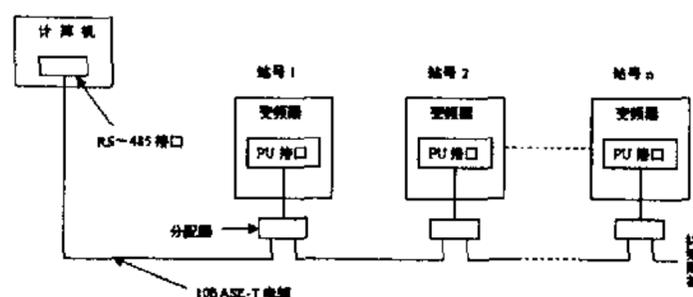


图 1 系统网络结构

### 2 FR-E500 系列变频器的串口通信协议

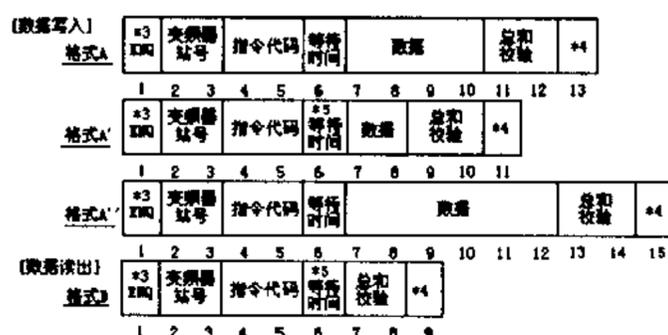
FR-E500 系列变频器采用异步控制协议，通讯方式为半双工。在 PC 和 FR-E500 系

列变频器之间进行通信前，通讯规格必须在变频器的初始化中设定，如果没有进行初始化设定或有一个错误的设定，数据将不能进行传输。这里 FR—E500 系列变频器的通讯规格设定如下：

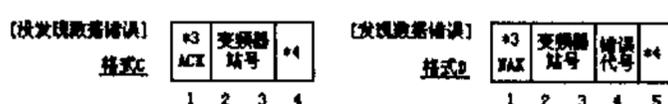
操作模式取值 1，即将变频器设定为通讯操作模式；通讯速率取值 192，即 19200bps；字节长 8 位，停止位长 1 位；奇偶校验 取值 0，无奇偶校验。事实上，通讯指令本身已采用总和校验方式；通讯重试次数取 10，如果错误连续发生 10 次，变频器将报警停止；通讯校验时间间隔取值为 9999，表示变频器与单片机无通讯状态的时间不受限制；等待时间设定取 1，表示变频处理来自计算机命令的时间为 10ms。

PC 与 FR—E500 系列变频器的通信使用十六进制数，数据在计算机和变频器之间自动使用 ASCII 码传输。数据格式类型主要有以下几种：

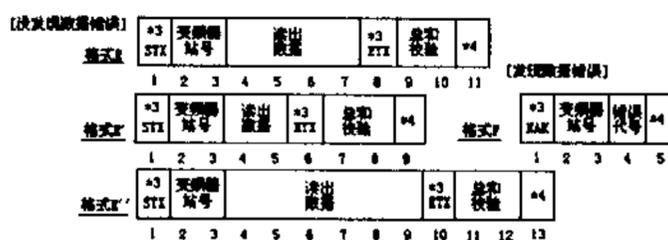
### 从计算机到变频器的通讯请求数据



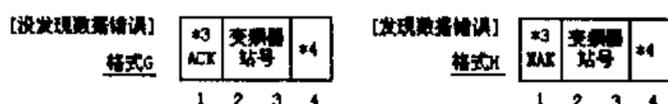
### (2) 写入数据时从变频器到计算机的送信数据



### (3) 读出数据时从变频器到计算机的应答数据



### (4) 读出数据时从计算机到变频器的送信数据



控制代码：

STX: Start Of Text (数据开始), H02;

EXT: End Of Text (数据结束), H03;

ENQ: Enquiry (通讯请求), H05;

ACK: Acknowledge (未发现数据错误), H06;

NAK: Negative Acknowledge (发现数据错误), H15。

数据格式中\*3 表示控制代码, \*4 表示 CR (回车, H0D) 或 LF (换行, H0A) 代码。具体的指令代码和数据见 FR-E500 变频器使用手册。

### 3 在 C++Builder5.0 中实现通信

在 Win32 操作系统中, 编写异步串行通信程序典型的方法有: 采用 API 函数编程的方法; 使用微软提供的 ActiveX 组件的方法; 使用第三方厂家提供的组件的方法。本文主要介绍采用 API 函数并基于线程的异步串行通信方法。比较起来, 该方法使用较多的底层设置, 是几种方法中最复杂的情况。但通过该方法可以了解串行通信中的内部细节问题, 有助于编写数据传输更加可靠, 更具灵活性的应用程序。

#### 3.1 建立自定义通信类 Tcommunication

建立自定义通信类的目的是将串口操作所必需的 API 函数封装在其中, 从而提供若干接口关系简单的成员函数供外部调用。该通信类的基类是 Tobject。其成员变量及函数定义如下:

```
public:
    HANDLE hCom;
    ...
    bool __fastcall ComInitialization(AnsiString DevicePort, int
    BaudRate, int ByteSize, int Parity, int StopBits);
    char * __fastcall ReceiveProcess();
    BOOL __fastcall WriteCommBlock(LPSTR lpBlock, DWORD nLength);
private:
    char Buffer[BLOCK_LENGTH+1];
```

```
DWORD BufferOffset;  
  
OVERLAPPED osRead, osWrite;  
  
DWORD __fastcall readCommBlock
```

(LPSTR Block);

接收缓冲区 Buffer 用于保存从变频器返回的数据，并由函数 ReceiveProcess() 进行相应处理。OsRead 和 osWrite 是用于重叠方式接收和发送操作的 OVERLAPPED 类型的数据结构。重叠操作是一种异步输入和输出操作，是操作系统内部多线程工作机制的具体应用。区别于非重叠操作方式，在重叠操作方式下，函数立即返回，输入或输出动作完成与否通过查询事先定义好的事件是否处于信号态作出判断。该事件保存在 OVERLAPPED 类型的数据结构中。

串行通信设备初始化函数 ComInitialization() 用于对指定的串口进行初始化设置，该函数被主线程调用。初始化过程主要包括：创建串行通信设备的文件句柄，指定为重叠操作方式。API 函数以创建文件的方式使用串行通信设备的句柄，这样从串口接受数据就是读文件，发送数据就是写文件；设置串口内部接收和发送缓冲区的大小；设置端口、波特率、数据位、停止位和奇偶校验位等参数。采用读取、修改和回写串口的 DCB 数据结构来实现；调用 CreateEvent() 函数为接收和发送操作的重叠结构建立信号事件；接收缓冲区 Buffer 计数器清零。

在完成了串行通信设备的初始化后，接收实际是在操作系统内部自动进行的，接收的字符自动放在串口内部输入缓冲区中。数据块接收函数 ReadCommBlock() 把读取串口内部的输入缓冲区变为读取便于操纵的缓冲区 lpBlock，并返回串口内部接收字符数。该函数声明为：DWORD \_\_fastcall ReadCommBlock(LPSTR lpBlock)。函数中首先定义一个 COMSTAT 类型的变量 ComStat，然后调用 ClearCommError() 函数将获取的串口状态信息数据存入 ComStat 中。根据串口内部接收字符数 Comstat.cbInQue 以重叠方式调用 ReadFile() 函数接收指定的字符。

调用 ReadFile() 函数隐含一个使 osRead 数据结构的 hEvent 事件复位的操作。ReadFile() 函数立即返回，指定长度的字符并未全部读入指定的缓冲区中，动作交于后台执行，此时返回值为 false，且 GetLastError() 函数的返回值为 ERROR\_HANDLE\_EOF。此时只有等到 osRead 的 hEvent 事件为信号态，方能说明 ReadFile() 函数的动作全部完成。

串行通信设备发送处理函数 `WriteCommBlock()` 完成指定长度数据块（发送到变频器的命令数据）的发送。该函数声明为：`WriteCommBlock(LPSTR lpBlock, DWORD nLength)`，`lpBlock` 参数为存储变频器命令数据的缓冲区，`nLength` 为所要发送的命令数据的长度，根据读出或写入操作而有所不同。在线程中调用该函数，每调用一次只发送一个字符。该函数以重叠操作方式调用的 `WriteFile()` 函数。

### 3.2 创建线程派生类 `TthreadComm`

线程用来监视串行通信设备的通信事件，在这里只涉及收到有效字符事件和发送缓冲区空事件，它们分别驱动接收和发送操作。

在线程派生类中定义 `AnsiString` 串 `recvBuffer` 存放从串口接收到的变频器返回的数据，通过处理后放到主窗口的 VCL 中显示。要访问主线程的 VCL 控件，线程必须通过调用 `Synchronize()` 函数才能实现。`Synchronize()` 是线程的同步方法，按照要求 `Synchronize()` 调用的函数应为无参数和返回值的函数，`RecvBuffer` 就是用来替代这个被调用函数的参数。`AnsiString` 串 `sendBuffer` 放置要发送的命令，发送是逐个字符进行的，减一计数器 `sendCount` 用来记录 `sendBuffer` 中将要发送的字符数。

在线程派生类的构造函数中以挂起方式启动线程，并在线程终止时自动释放内存。发送缓冲区计数器 `sendCount` 清零，最后在调用者以非挂起方式显式地启动线程。

```
__fastcall TThreadComm::TThreadComm(bool CreateSuspended) : TThread(true)
{
    //线程终止时自动释放内存
    FreeOnTerminate=true;

    //发送缓冲区的计数器清 0
    sendCount=0;

    if(CreateSuspended==false)

        Resume(); //启动线程
}
```

重载线程的 `Execute` 方法，用来监视收到有效字符和发送缓冲区空事件，根据不同的

事件来启动接收或发送操作。SetCommMask 函数用来设置通信事件掩码。通信事件除了收到有效字符和发送缓冲区空外还有 CTS 管脚电平改变、奇偶校验错、覆盖和帧格式错等事件。这里屏蔽其它的通信事件，只考虑收到有效字符和发送缓冲区空事件，其字符常量分别为 EV\_RXCHAR 和 EV\_TXEMPTY。设置了事件掩码后，等待通信事件函数 WaitCommEvent() 监视指定的通信事件。该函数在线程的 Execute 方法中循环调用，只要线程没有终止，它就一直监视指定的通信事件。

当发生收到有效字符事件时，通信派生类的 ReceiveProcess() 函数调用 ReadCommBlock() 函数读取字符并进行相应处理后存入 recvBuffer，然后利用线程同步函数 Synchronize() 访问主线程中的 VCL 控件。

当发生发送缓冲区空事件时（即可以发送的情况），调用 SendOneByte() 函数将发送缓冲区中当前位置的字符发送出去。SendOneByte() 函数为线程派生类的成员函数，它调用通信派生类的 WriteCommBlock() 函数发送缓冲区中当前位置的字符。

### 3.3 创建应用程序主窗体

在主窗体的 OnCreate 事件中获取初始化设置参数初始化指定的串口，创建串行通信对象实例，创建并启动线程，实现对通信事件的监视和操纵。

最后启动数据块发送函数，将要发送的命令一次性装入线程内部的发送缓冲区，同时设置其计数器值。发送命令的首字节以启动整个命令的发送。命令的发送过程是主线程发送命令的首字节并间隔一段时间后，发生发送缓冲区空事件。该事件被线程的 Execute 方法的通信事件监视函数捕获，从而触发发送下一个字符的动作。发送首字节之前，发送缓冲区是空的。但此时并不发生发送缓冲区空事件，只有在线程之外发送了命令的首字节，才引起发送缓冲区空事件。该事件驱动线程中的后续数据的发送，直至整个命令发送完毕。

最后判断接收到的数据（变频器返回的数据）格式并对其进行相应的处理，从而实现监视、报警等功能。

通过以上的方法可以灵活的编写 PC 与多台变频器的通信程序，实时检测各变频器的运行状态，以完成运行、监视、读取和设置变频器参数等功能，将在变频器面板上的操作完全转移到软件中进行。整个控制系统灵活方便，具有很大的实用性。

另外 ActiveX 控件 MSCOMM32 提供了访问串行通信设备简单的接口，通过操纵控件的

属性、方法和事件可以实现对串口的多种操作。该方法能够满足一般的通信要求，与本文介绍的方法相比较容易理解。