

摘要

在机器人学的研究领域中,如何有效地提高机器人控制系统的控制性能始终是学者十分关注的一个重要内容。为了满足当前机器人位置跟随精度和定位精度高的要求,论文研究并开发了基于 DSP 的机器人位置控制系统。论文研究主要涉及以下几方面:

首先,传统 PID 控制算法结构简单,调整方便,但是不易在线实时调整参数,对一些复杂过程不能进行有效的控制。论文提出了模糊 PD 控制算法结合单神经元自适应 PID 算法的双模控制,并且研究了模糊 CMAC 控制算法。模糊控制学习速度快,但控制精度不高;单神经元结构简单,具有良好的调整性和自适应性,但在输入量大范围变化时学习速度变慢,论文将两种算法的优势有机结合起来,取长补短,提高整个系统的快速性和精确性。CMAC 神经网络具有信息分类存储的能力,较其他网络更适用于复杂环境下的非线性实时控制。论文在 MATLAB 平台上进行了仿真,仿真结果表明这两种算法都可以满足机器人位置控制系统的要求。

其次,采用运算速度较高的 DSP 作为运动控制板的主芯片。硬件控制平台采用合众达公司的 SEED-DEC2812 模板。软件设计包括系统初始化程序、系统主程序、中断服务子程序和中断处理子程序及其各程序模块的设计,并且在 DSP 系统开发环境下进行系统仿真试验。

再次,选用了传输速度很快的 USB2.0 作为通讯电路。USB 的编程主要分为两个部分,一是主机方面的驱动程序的编写,一是有关 DSP 与 USB 的数据与命令的交换,主要内容为机器人关节控制命令的传递、机器人运行状态的监测。

最后,在 WINDOWS 操作系统下利用 VC++采用多文档模板结构设计了用户界面程序,实现对硬件的驱动,完成数据结果的显示与统计。软件与硬件有机地结合在一起,构成一个整体,共同实现功能。

关键词: DSP; 机器人; 交流伺服电机; 双模控制

ABSTRACT

In robot research area how effectively enhances the robot control system's control performance is always an important content which the scholars pay much attention to. To satisfy high requirements for position follow accurateness and location precision of robot system at present. This paper researches and develops robot position control system based on DSP. The work includes:

Firstly, the PID position control algorithm has simple structure and convenient regulation, but it can't change the parameter online in real time or it can't control some complex process effectively. this paper studies a dual-mode control strategy which combined fuzzy PD control with adaptive control based on single neuron and a kind of fuzzy CMAC network control algorithm. Fuzzy control learns with high-precision but low precision; single neuron has simple structure, good adjustability and self-adaptation, but the learning speed is low when the importer does large -scaled changes, This paper combines advantages of the two algorithm seamlessly, learn from other's strong points to offset one's weakness, to enhance the system's fastness and accuracy. For the ability of CMAC's classifying information storage, it is more fit for non-linear real-time control of complicated environment than other nets. It simulates on MATLAB console, the result shows that the two algorithms can satisfy the robot position control system demands.

Secondly, this paper adopts DSP which has higher speed of operation as main chip of motion control circuit. The hardware control platform is based on SEED-DEC2812 testing board. The software includes the initialization program, the main program, the interrupt service program and the interrupt processing program and etc. We present the emulated result in the end.

Thirdly, this paper adopts USB2.0 with higher transmission rate rather than general serial communication mode. The programmer includes two main parts: one is host-driven process design, the other is data and order exchange between DSP and USB, the main content is transfer of robot joint control order and monitor for robot running state.

Finally, with VC++'s multi-document template technology, this paper designs the interface programs under WINDOWS operating system to realize the driving of hardware, achieve the display and statistic. Software and hardware combines with each other seamlessly, realize function as a whole.

Keywords: DSP; Robot; AC Servo Motor; Dual-mode Control

独创性声明

本人声明所呈交的论文是我个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或撰写过的研究成果，也不包含为获得北京工业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了谢意。

签名： 吕雁 日期： 2007.6.12

关于论文使用授权的说明

本人完全了解北京工业大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

（保密的论文在解密后应遵守此规定）

签名： 吕雁 导师签名： 吕雁 日期： 07.06.12

第1章 绪论

1.1 机器人技术发展概况

早在三千多年前的西周时代，我国就出现了能歌善舞的木偶，称为“倡者”，这可能是世界上最早的“机器人”^[1]。

当代的工业机器人起源于数控机床和遥控操作器。1954年美国的 George Devol 将遥控操作器的连杆机构与数控铣床的伺服轴结合起来，研制出了世界上第一台通用工业机器人(Engelberger)，操作者可以对它进行轨迹控制，使它进行一系列的点位运动。此后，随着电子计算机、自动控制理论的发展、工业生产的需要以及空间技术的进步，机器人技术在一些发达国家迅速发展起来^[2]。

70年代，随着计算机技术、现代控制技术、传感技术、人工智能技术的发展，机器人得到了迅速发展。1974年 Cincinnati Milacron 公司开发成功多关节机器人；1979年，Unimation 公司又推出了 PUMA 机器人^[3]，它是一种多关节、全电动驱动、多 CPU 二级控制；采用 VAL 专用语言；可配视觉、触觉、力觉等传感器，在当时是一种技术先进的工业机器人。现在的工业机器人结构大体上是以此为基础的。这一时期的机器人属于“示教再现”(Teach-in/Playback)机器人^[4]。只具有记忆、存储能力，按相应程序重复作业，但对周围环境基本没有感知与反馈控制能力。

进入80年代，随着传感技术，包括视觉传感器、非视觉传感器(力觉、触觉、接近觉等)以及信息处理技术的发展，出现了第二代机器人，即有感觉的机器人。它能够获得作业环境和作业对象的部分有关信息，进行一定的实时处理，引导机器人进行作业。而机器人的控制器技术也已经由单一的硬件研究逐步过渡到软件研究、由具体控制器过渡到通用开放式体系结构、由单独控制过渡到多机协调控制。在这个阶段，应控制技术、一些先进的控制方法被广泛应用于机器人控制系统，如模型参考自适应最优控制技术等，大大改善了机器人控制系统的控制效果。

第三代机器人是目前正在研究的“智能机器人”。它不仅具有比第二代机器人更加完善的环境感知能力，而且还具有逻辑思维、判断和决策能力，可根据作业要求与环境信息自主地进行工作。控制器和网络化控制等要求。为了克服传统的控制方法不能解决机器人系统的变参数和非线性的问题，神经网络控制在机器人控制中获得了广泛的应用^[5]。现代机器人技术集中了机械、电气电子控制、计算机、通讯和传感等各方面的成果，如图 1-1 所示，是当代科技发展的前沿。

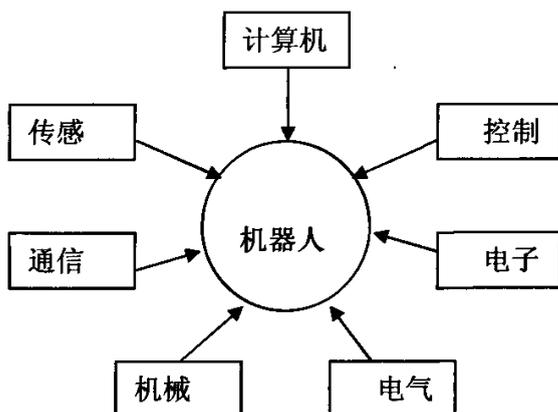


图 1-1 机器人的相关技术
Fig. 1-1 Relevant techniques of Robots

1.2 运动控制方法发展概况

目前，一个运动控制系统或电机控制系统的实现方法主要有以下几种^[6]：

(1) 以模拟电路硬接线方式建立的运动控制系统。早期的运动控制系统一般是采用运算放大器等分立元件以硬接线方式组成的模拟控制系统，其优点在于：通过对输入信号的实时处理，可实现系统的高速控制；由于硬接线方式可以实现无限的采样频率，因此控制器的精度较高而且具有较大的宽度。

然而，与数字系统相比，其缺点也是很明显的：器件老化和环境温度变化对构成系统的元器件的参数影响很大；构成模拟系统需的元器件较多，增加了系统的复杂性，也使系统的可靠性降低；由于系统采用硬接线，当系统设计完成以后，升级和功能修改几乎是不可能的；受系统规模的限制，很难实现运算量大、精度高、性能更先进的复杂控制算法。这些缺陷使它很难应用于一些功能要求比较高的场合。

(2) 以微控制器为核心的运动控制系统。利用微控制器所构成的系统与模拟电路相比具有以下优点：绝大多数控制逻辑由软件实现，电路变得简单；微控制器具有更强的逻辑功能，运算速度快、精度高、具有大容量的存储器，因此有能力实现较为复杂的算法；微控制器的控制方式主要由软件实现，修改控制规律时，仅需对软件进行修改，灵活性和适应性强；无零点漂移，控制精度高；可提供人机界面，实现多机联网工作。

然而，由于微控制器一般采用冯-诺依曼总线结构，处理速度和能力有限，软件编程难度较大，且一般芯片集成度较低，不具备运动控制系统的专用外设。

因此,基于微控制器构成的系统仍需较多的元器件,这增加了系统电路的复杂性,降低了可靠性,也难以满足运算量较大的实时信号处理的需要,难以实现控制算法。

(3) 在通用计算机上用软件实现运动控制策略。在通用计算机上,利用高级语言编制相关的控制软件,配合驱动电路板、信号交换接口,就可以构成一个运动控制系统。这种实现方法利用计算机的高速度、强大的运算能力和方便的编程环境,可以实现高性能、高精度、复杂的控制算法,而且软件的修改也很方便。但是,这种方式的一个缺点在于系统体积大,难以应用于工业现场,而且难以实现实时性要求较高的信号处理算法^[7]。

(4) 利用专用芯片实现的运动控制。为简化电机模拟控制系统电路,同时保持系统的快速响应能力,一些公司推出了专用电动机控制芯片,如 Ti 公司的 UCC3626, UCC2626 等。利用专用电机控制芯片构成的运动控制系统具有速度快、系统集成度高、使用元器件少、可靠性好等优点;同时,专用电动机芯片的价格便宜,进一步降低了系统成本,因而这种实现方法被广泛应用于成本敏感的场所。

然而,受专用电机控制芯片本身的限制,其缺点主要包括:软件算法固化在芯片内部,降低了系统灵活性,且不具备扩展性;受芯片制造工艺限制,实现算法较为简单,控制精度也较低;用户不能对芯片编程,不便对系统升级。

(5) 以可编程逻辑器件为核心的运动控制系统。由于 FPGA/CPLD 等可编程器件的发展,人们可以利用它们的系统开发软件或 VHDL 等开发语言,通过软件编程实现某种运动控制算法,然后将这些算法下载到相应的可编程逻辑器件中,从而以硬件方式实现最终的运动控制系统。

这种系统优点主要有:系统的主要功能在单片 FPGA/CPLD 中实现,减少了元器件个数,缩小了系统体积;具有良好的扩展性和可维护性,通过修改软件并重新下载到目标板上的相关器件中,就可以对系统升级;系统以硬件实现,响应速度快,可实现并行处理;开发工具齐全,通用性强。然而,这种系统实现方法的成本较高。控制算法越复杂,对可编程逻辑器件的集成度要求越高,芯片价格越昂贵。因此,考虑到系统成本,一般使用可编程逻辑器件实现较为简单的运动控制系统。

(6) 以可编程 DSP 控制器为核心构成运动控制系统。DSP 属于精简指令集计算机,增强型哈佛结构,具有独立的程序存储空间和数据存储空间,运算速度极快,为系统实时性提供了有力保障,也使复杂算法的实现成为可能,特别是面向电机控制应用的 DSP,它通过把一个高性能的 DSP 内核和常用外围设备集成为一个芯片的方法,将 DSP 的高速运算特性和优化的控制特性结合起来,成为运动控制系统核心芯片的最佳选择。基于 DSP 控制器的运动控制系统实际上是一个单片系统,因为整个电机控制所需的各种功能都可以用 DSP 控制器来实现,

因此可以大幅度减小系统体积,减少外部器件的个数,增加系统可靠性。另外,由于各种功能都通过软件实现,系统升级容易,扩展性和可维护性都很好。同时,DSP 控制器的高性能使最终系统既能满足要求比较低的系统,也可以满足对性能和精度要求较高的场合需要。

1.3 机器人的运动学和动力学建模

1.3.1 机器人的运动学建模

机器人运动学问题是研究机器人关节变量 $\mathbf{q} = [q_1, \dots, q_n]^T$ 与其末端夹持器位置和指向间的关系,因末端夹持器与杆 n 固连,故其位置和指向由系 n 唯一确定,亦即由奇次变换矩阵 0A_n 唯一确定,因此也可以说运动学问题是研究 \mathbf{q} 与 0A_n 间的关系^[8]。

设有坐标系 0, 将其绕系 0 原点的一轴转动一个角度或沿系 0 的一轴移动一定长度后得到一个新的坐标系 1, 这些转动或移动对应的变换矩阵称为奇次变换矩阵^[9]。连续相对运动的奇次变换矩阵可按相对运动的次序连续右乘各次运动的奇次变换矩阵得到。由连续相对运动奇次变换矩阵的求法可得到机器人的运动学方程

$${}^0A_n(\mathbf{q}) = {}^0A_1(q_1) {}^1A_2(q_2) \dots {}^{n-1}A_n(q_n) \quad (1-1)$$

对于图 1-2 所示的水平双连杆刚性机器人,杆长分别为 l_1, l_2 , 称原坐标系为系 0, 以连杆 1 的末端为原点, 连杆 1 为 x 轴的坐标系为系 1, 以连杆 2 的末端为原点, 连杆 2 为 x 轴的坐标系为系 2。

绕 x 轴转动 θ 对应的奇次变换矩阵为

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

沿 x 轴移动 d 对应的奇次变换矩阵为

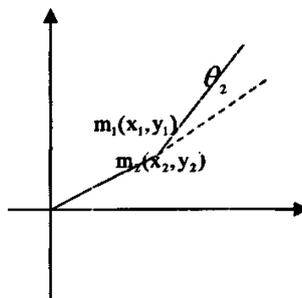


图 1-2 两连杆机械臂

Fig.1-2 Two link manipulators

$$T_x(d) = \begin{bmatrix} I & \begin{bmatrix} d \\ 0 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix}$$

沿 y 轴移动 d 对应的奇次变换矩阵为

$$T_y(d) = \begin{bmatrix} I & \begin{bmatrix} 0 \\ d \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix}$$

故得到图 1-4 所示的双连杆机器人的奇次变换矩阵分别为

$${}^0A_1 = T_x(x_1)T_y(y_1)R_z(\theta_1) = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & l_1\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & l_1\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1A_2 = T_x(x_2)T_y(y_2)R_z(\theta_2) = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & l_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & l_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} {}^0A_2 &= {}^0A_1 {}^1A_2 \\ &= \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & 0 & l_1\cos\theta_1 + l_2\cos(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & 0 & l_1\sin\theta_1 + l_2\sin(\theta_1 + \theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

(1-2)

1.3.2 机器人的动力学建模

常用的机器人动力学建模有利用欧拉方程建模和利用拉格朗日函数建模^[10]。^[11]。论文介绍基于拉格朗日方程的动力学建模方法。

拉格朗日函数 L 定义为系统的动能 T 和势能 V 之差，即 $L=T-V$ ，其中 T 和 V 可以用任何方便的坐标系来表示。系统的动力学方程式即拉格朗日方程如下：

$$F_i = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} \quad i=1,2,\dots,n \quad (1-3)$$

其中, q_i 表示动能和势能的坐标, \dot{q}_i 表示相应的速度, F_i 表示作用在第 i 个坐标上的力或者力矩, n 表示连杆数目。且这些力、力矩和坐标称为广义力、广义力矩和广义坐标。

对于图 1-2 所示的水平双连杆刚性机器人, 假设连杆的转矩为 F_1 和 F_2 , 质量为 m_1 和 m_2 , 且以连杆末端的点质量表示, 连杆的长度为 d_1 和 d_2 , 广义坐标为 θ_1 和 θ_2 , g 为重力加速度, h 为基座的高度。

(1) 首先计算连杆 1 的动能 T_1 和势能 V_1 :

$$\text{由图可知, } T_1 = \frac{1}{2} m_1 v_1^2, v_1 = d_1 \dot{\theta}_1, V_1 = m_1 gh$$

所以得到连杆 1 的动能 T_1 和势能 V_1

$$T_1 = \frac{1}{2} m_1 d_1^2 \dot{\theta}_1^2, V_1 = m_1 gh \quad (1-4)$$

(2) 再求连杆 2 的动能 T_2 和势能 V_2

$$T_2 = \frac{1}{2} m_2 v_2^2, V_2 = m_2 gh, v_2^2 = \dot{x}_2^2 + \dot{y}_2^2$$

$$x_2 = d_1 \cos \theta_1 + d_2 \cos(\theta_1 + \theta_2)$$

由图可知, $y_2 = d_1 \sin \theta_1 + d_2 \sin(\theta_1 + \theta_2)$

$$\dot{x}_2 = -d_1 \sin \theta_1 \dot{\theta}_1 - d_2 \sin(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2)$$

$$\dot{y}_2 = d_1 \cos \theta_1 \dot{\theta}_1 + d_2 \cos(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2)$$

所以可以得到连杆 2 的动能 T_2 和势能 V_2

$$T_2 = \frac{1}{2} m_2 d_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 d_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + m_2 d_1 d_2 \cos(\theta_2) \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2), V_2 = m_2 gh \quad (1-5)$$

由机器人的动能和势能, 可得到双连杆机器人的拉格朗日函数 L :

$$L = T - V = T_1 + T_2 - V_1 - V_2$$

$$= \frac{1}{2} (m_1 + m_2) d_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 d_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + m_2 d_1 d_2 \cos(\theta_2) \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) - m_1 gh - m_2 gh \quad (1-6)$$

对 L 求偏导数和导数:

$$\frac{\partial L}{\partial \theta_1} = 0;$$

$$\frac{\partial L}{\partial \theta_2} = -m_2 d_1 d_2 \sin(\theta_2) \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2);$$

$$\frac{\partial L}{\partial \dot{\theta}_1} = (m_1 + m_2) d_1^2 \dot{\theta}_1 + m_2 d_2^2 (\dot{\theta}_1 + \dot{\theta}_2) + m_2 d_1 d_2 \cos(\theta_2) (\dot{\theta}_1 + \dot{\theta}_2) + m_2 d_1 d_2 \cos(\theta_2) \dot{\theta}_1;$$

$$\frac{\partial L}{\partial \dot{\theta}_2} = m_2 d_2^2 (\dot{\theta}_1 + \dot{\theta}_2) + m_2 d_1 d_2 \cos(\theta_2) \dot{\theta}_1;$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} = (m_1 + m_2) d_1^2 \ddot{\theta}_1 + m_2 d_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 d_1 d_2 \cos(\theta_2) (\ddot{\theta}_1 + \ddot{\theta}_2)$$

$$- m_2 d_1 d_2 \sin(\theta_2) \dot{\theta}_2 (\dot{\theta}_1 + \dot{\theta}_2) + m_2 d_1 d_2 \cos(\theta_2) \ddot{\theta}_1 - m_2 d_1 d_2 \sin(\theta_2) \dot{\theta}_1 \dot{\theta}_2;$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_2} = m_2 d_2^2 (\ddot{\theta}_1 + \ddot{\theta}_2) + m_2 d_1 d_2 \cos(\theta_2) \ddot{\theta}_1 - m_2 d_1 d_2 \sin(\theta_2) \dot{\theta}_1 \dot{\theta}_2;$$

将上面相应各导数和偏导数代入式(1-1), 即可得到力矩 F_1 和 F_2 的动力学方程式:

$$F_1 = \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} - \frac{\partial L}{\partial \theta_1} = [(m_1 + m_2) d_1^2 + m_2 d_2^2 + m_2 d_1 d_2 \cos(\theta_2) + m_2 d_1 d_2 \cos(\theta_2)] \ddot{\theta}_1$$

$$+ [m_2 d_2^2 + m_2 d_1 d_2 \cos(\theta_2)] \ddot{\theta}_2 - m_2 d_1 d_2 \sin(\theta_2) \dot{\theta}_2^2 - 2m_2 d_1 d_2 \sin(\theta_2) \dot{\theta}_1 \dot{\theta}_2;$$

$$F_2 = \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_2} - \frac{\partial L}{\partial \theta_2}$$

$$= [m_2 d_2^2 + m_2 d_1 d_2 \cos(\theta_2)] \ddot{\theta}_1 + m_2 d_2^2 \ddot{\theta}_2 - 2m_2 d_1 d_2 \sin(\theta_2) \dot{\theta}_1 \dot{\theta}_2 - m_2 d_1 d_2 \sin(\theta_2) \dot{\theta}_1^2$$

将上式化为矩阵形式即可得到双连杆机器人的动力学方程:

$$\begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} C_{111} & C_{121} \\ C_{211} & C_{221} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1^2 \\ \dot{\theta}_2^2 \end{bmatrix} + \begin{bmatrix} C_{112} & C_{122} \\ C_{212} & C_{222} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \dot{\theta}_2 \\ \dot{\theta}_2 \dot{\theta}_1 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \quad (1-7)$$

$$\text{即} \quad F = H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) \quad (1-8)$$

式中 q ——关节的广义坐标。

1.4 课题来源、研究内容与研究意义

1.4.1 课题来源

国际标准化组织(ISO)对“机器人”一词的定义是:“机器人是一种自动的、位置可控的、具有编程能力的多功能操作机,这种操作机具有几个轴,能够借助可编程操作来处理各种材料、零件、工具和专用装置,以执行各种任务”。机器人技术以机械、电子及信息技术为基础,以自动控制理论为媒介,是一门综合性工程技术,从一般的工业生产,如装配、焊接,到特殊的应用领域,如医疗外科、太

空作业,机器人在现代化工业领域的各个方面得到了广泛的应用,随着工业生产的不断发展,对机器人的作业性能也提出了越来越高的要求^[12]。

本课题来源于北京工业大学机器人研究所的科学研制项目。北京工业大学机器人研究所是北京“211”工程实验室工程项目实践基地,同时也是为相关人员研究先进的机器人运动控制算法而设立的机器人技术研究机构。

机器人研究所已经研制出了双连杆机器人系统的动力学机构,并且组装了相应的双连杆机器人的电机驱动系统,论文在此基础上研制了一套双连杆实验室机器人控制器的软、硬件平台,通过它来实时准确的驱动机器人关节电机的运动,实现操作者对机器人系统的有效控制。

1.4.2 研究内容

论文的主要研究内容包括:

- (1) 在现有机器人系统硬件的基础上,通过引入高性能的 DSP 控制器,组建了一套完整的双连杆机器人控制系统。
- (2) 在 MATLAB 平台上对两种控制算法进行仿真,选择最佳算法。
- (3) 主要采用 C 语言实现机器人系统的 DSP 控制器的控制功能。
- (4) 基于 VC++6.0 平台设计双连杆机器人控制器的用户界面程序。
- (5) 分别在机器人控制器的 DSP 侧与 PC 机侧编写程序,实现两者之间的 USB 通信功能。
- (6) 针对双连杆机器人系统所使用的关节驱动电机,利用多种控制策略,进行实验、调试。

1.4.3 研究意义

长期以来,机器人的控制问题一直是控制领域众多学者的研究课题,现代控制理论所提供的几乎所有方法都在其控制系统上作过尝试。机器人控制的研究不仅具有普遍的工程意义,还具有深入的理论意义。

本课题通过研制双连杆机器人系统,可以深入地研究机器人运动学算法、动力学分析、模糊控制、神经网络等先进控制算法,并且我们选用了 DSP 控制器来实现智能机器人运动控制系统。由于 DSP 具有极高的速度,这可以大大缩短系统的采样周期,提高系统的调节频率,从而提高系统的动态性能;另外, DSP 具有丰富的硬件资源,集成了闪存、高速 A/D 转换器,可以大大简化控制系统的硬件结构,大大提高系统的可靠性。

目前各种机器人的生产和研制在发达国家方兴未艾,而在我国无论开发应用

还是学术研究方面都落伍于世界先进水平,因此开展对机器人技术的研究已成为我国实现现代化的当务之急。我国的机器人研究和开发始于70年代,并在80年代掀起了发展机器人的高峰,许多高等院校和研究部门正在从事该领域的研究工作,并取得了一定的成效。

1.5 本章小结

在本章,首先对机器人技术和运动控制方法的发展情况进行了简要的介绍,接着分析了机器人的建模方法,包括机器人的运动学建模和动力学建模,最后列出了课题的来源、研究内容和意义。

第 2 章 基于 DSP 的机器人控制器的硬件设计

2.1 机器人位置控制系统总体结构

机器人控制器是机器人系统的核心部分, 机器人性能的优劣、功能的强弱与控制器密切相关。论文采用基于三级 CPU 控制的控制机模型, 即上位计算机为一级 CPU, 它担当系统管理以及运动学计算、轨迹规划等; DSP 为二级 CPU, 它主要控制交流伺服电机的位置环, 完成全部关节的运动控制算法实现; 交流伺服驱动器为第三级 CPU, 实现了交流伺服电机的速度环和电流环的数字控制。整个控制系统可以分为下位控制子系统和上位监控子系统。如图 2-1 所示。

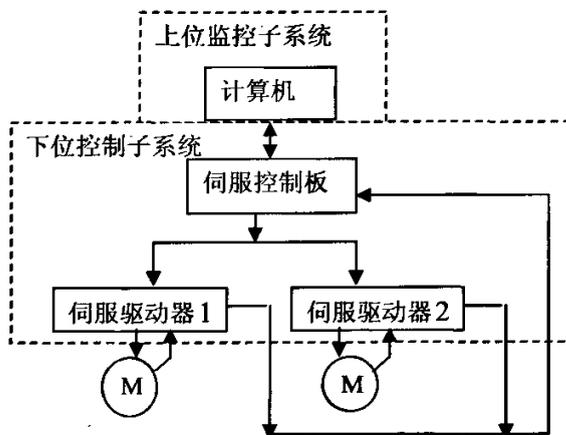


图 2-1 控制系统总体框图

Fig.2-1 The overall diagram of control system

(1) 下位控制子系统: 由伺服控制板和伺服驱动器组成。主要实现机器人各关节(电机)直接、精确的位置数字控制。该子系统以基于 DSP 的伺服控制板为控制中心, 实现机器人各关节运动的轨迹精细插补、协调运动。伺服控制板能实时接收上位监控子系统下传的控制参数, 反馈各关节的实时的位置参数给上位控制子系统, 进行伺服驱动器位置环的控制。伺服驱动器主要实施对各关节电机的速度环、电流环的控制, 驱动电机运转。

(2) 上位监控子系统: 由监控计算机组成。主要对机器人系统进行实时的监视与控制。该子系统具有友好的人机界面, 依据用户给定的末端执行器的位姿信息, 进行运动学运算、轨迹插补等操作, 计算出对应于每个关节电机的控制参数, 并下传至下位控制子系统。此外, 该子系统能实时接收下位子系统反馈的各关节位置信息, 有效地对系统进行实时操作。

控制系统采用典型的两级子系统控制方式,可以有效地将系统的控制功能分离,并使系统结构更具模块化,大大增强系统的可靠性,降低项目研发的风险。上、下位子系统之间可以通过通用 USB 通信,并采用中断事件触发数据接收,能实时地实现稳定可靠的高速数据交换。此外,整个系统几乎都是全数字控制,所以系统的控制精度和可靠性很高,操作和维护也很方便。

2.2 机器人关节驱动装置

2.2.1 交流伺服电动机

机器人关节的驱动电机又称执行电机,它跟其它电机的主要不同点在于,它要接受控制信号的控制:有了控制信号,立即转动;控制信号消失,立即停止^[13]。对它的选用必须以能适应频繁的位置变化和速度指令变化要求,并具有能以大的加速度进行频繁起停、停止等功能为主要考虑对象。

目前在机器人的运动控制中较常使用的电机有步进电机、直流伺服电机和交流伺服电机。

步进电机的点位控制性能好,没有累积误差,可直接使用数字信号进行控制,而且没有电刷,维护方便,寿命长,但其过载能力弱,能量转换效率较低,而且容易失步,因此步进电机不适合于构成高性能的电机驱动系统。

直流伺服电机具有控制特性好、效率高等特点,而且启动转矩大,转子的转动惯量小,能够获得大的加速度,所以很久以来直流伺服电机在高性能伺服控制系统中一直占据着重要的地位。但是由于直流伺服电机具有碳刷和机械换向器,致使电机的维护工作量大,寿命短,而且干扰严重,一旦损坏,就很难修复。

交流伺服电动机是一种能够实现高性能伺服控制的驱动电机。它没有换向器与电刷结构,却拥有与直流伺服电机同样出色的控制性能^[14, 15]。

交流伺服电机通常具有以下一些特点:

- (1) 良好的速度控制特性,在整个速度区可实现平滑控制,几乎无振荡。
- (2) 高速控制。
- (3) 高精度的位置控制(取决于采用的编码器的分辨率)。
- (4) 额定运行区域内,实现恒力矩。
- (5) 90%的高效率,发热低。
- (6) 惯量低。
- (7) 低噪声,没有电刷的磨损,免维护。
- (8) 不产生磨损颗粒,没有火花,适用于无尘、易爆环境。

鉴于上述的分析,本课题决定采用 Panasonic 公司的小惯量型三相交流伺服

电动机实现对机器人系统各个关节的驱动控制,第一关节的电机参数如表 2-1 所示。



图 2-2 松下交流伺服电机 MINAS-A 系列

Fig.2-2 Panasonic AC Servo motor MINAS-A series

交流伺服电机的主要特征参数如表 2-1 所示:

表 2-1 交流伺服电机的主要特征参数

Table 2-1 Mostly character parameter of AC servo motor

	电机
关节 1	型号: MSMA042A1A** 小惯量、有键槽、无制动器 输出功率: 400W 输入电压: 3 相 220V 输入电流: 2.0A 额定速度: 3000r/min 带增量式编码器, 2500p/r 分辨率: 10000p/r

2.2.2 电机的驱动与控制

我们采用与电机相配套的交流伺服驱动器来控制电机的运转。附录 1 中给出了驱动器内部结构方框图。驱动器采取典型的三环控制结构: 内环为电流环, 中环为速度环, 外环为位置环。电流环主要用于电机运行时, 检测电机的相电流变化, 通过反馈环节调节电流变化, 防止电流突变, 从而改善系统电参数性能。电流环采用的是 PI 控制, 但它的参数不能改变, 随驱动器的型号而定。在速度环结构中, 电机通过编码连接器将其运动转角位移反馈给驱动器, 驱动器则通过内

部微分电路, 调节及修正速度指令反馈量。该速度环采用的是 PI 调节器, 其参数可由用户自己设定。位置环是整个下位机控制的核心。电机经编码连接器将其运转信号反馈给驱动器控制主电路, 通过驱动器内置的处理器进行差分、放大、参数调整等数据处理, 改善位置闭环控制性能。该位置环采用的是 P 调节器, 其参数是可变的, 用户可以根据需要设计自己的参数。此外, 驱动器内还有错误检出和保护电路。

附录 2 中给出了驱动器控制方框图。驱动器有三种控制方式: 位置控制方式、速度控制方式和转矩控制方式。本控制系统采用的为速度控制方式, 可通过设计 Pr02 参数将伺服驱动器调整为速度控制方式, 位置环控制在 DSP 控制器上做。速度控制信号连接图如图 2-3 所示。

在速度控制模式中, 位置环控制器通过 SPR, GND 引脚给伺服驱动器输入一个 $-10V \sim +10V$ 的模拟电压, 就可以控制电机实现从反向最大转速到正向最大转速之间的速度变化。电机的转速与输入电压之间的关系是线性的, 通过 pr50 (速度指令输入增益) 来确定。论文设定 pr50=300, 如输入 10v 时转速为 3000r/min, 达到额定转速。

速度环的参数通过交流伺服驱动器的常规自动增益参数优化调整获得。调整后的控制参数为: pr11 (第一速度环增益) 为 220, pr12 (第一速度环积分时间常数) 为 50。

2.3 机器人关节 DSP 运动控制器

数字信号处理 (Digital Signal Process) 是一门涉及许多学科而又广泛应用于许多领域的新兴学科, 于 20 世纪 60 年代开始围绕着数字信号处理的理论、实现和应用等几个方面发展起来的。70 年代, 由于大规模集成电路 (LSI) 的发展, 使得用硬件实现数字滤波器成为可能, 并开始研究将微处理器应用于数字信号处理。到了 80 年代, 随着超大规模集成电路 ((VLSI) 的应用和微处理器技术的发展, 单片通用数字信号处理器和各种单片专用数字信号处理器纷纷涌现出来。

近年来, 随着微电子学、数字信号处理技术、计算机技术等学科的迅猛发展, DSP 的应用逐渐成熟, 并在电机控制、通信、图像处理等领域显示出其独有的优势。

DSP 芯片, 也称数字信号处理器, 是一种特别适合于进行数字信号处理运算的微处理器, 其主要应用是实时快速地实现各种数字信号处理算法 (如采集、变换、滤波、估值、增强、压缩、识别等处理), 以得到符合人们需要的信号形式。若没有特殊注明, 在本论文中 DSP 均指数字信号处理器。

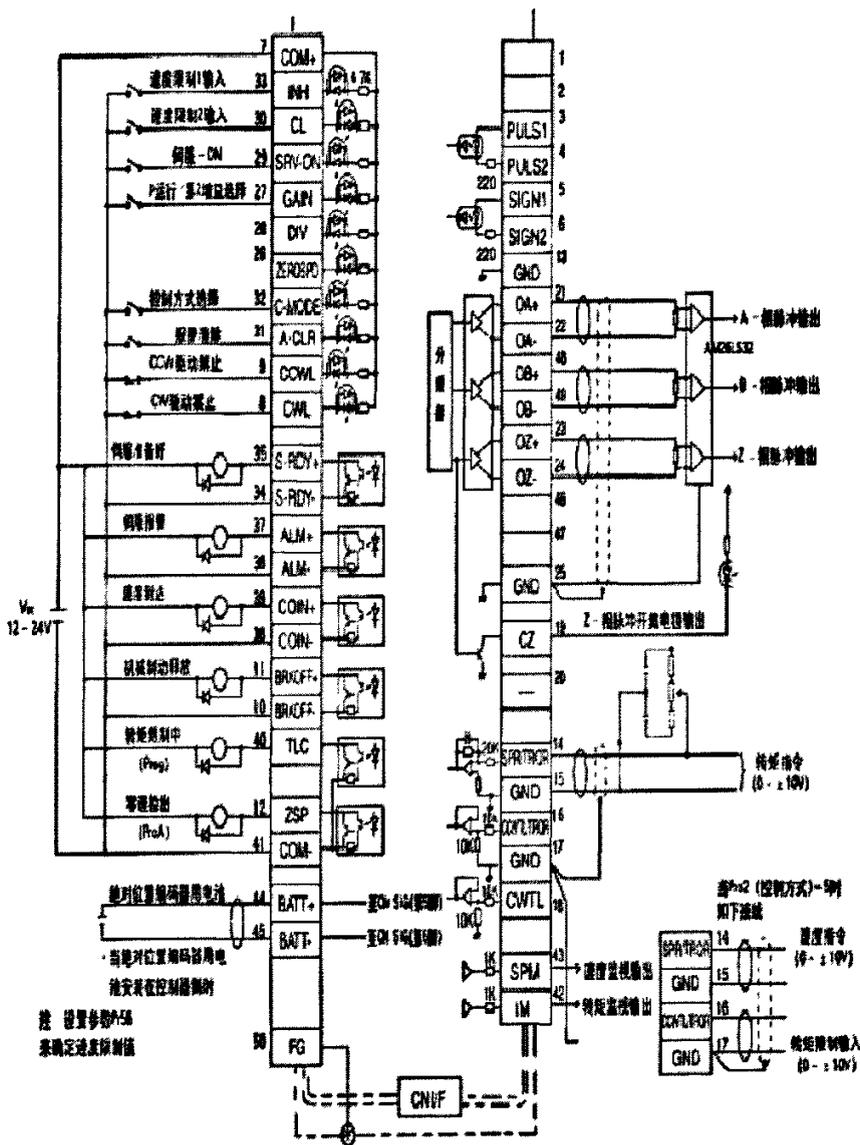


图 2-3 伺服电机速度控制信号连接图

Fig.2-3 Juncture diagram of servo motor speed control systems

目前，DSP 技术发展得很快，美国德州仪器 (Texas Instruments, 简称 TI) 公司、模拟器件 (Analog Devices, 简称 AD) 公司、MOTOROLA 公司等几家公司占有较大的市场份额。其中，尤其是 TI 公司的 TMS320 系列芯片应用最为广泛，包括 2000 系列、5000 系列、6000 系列，分别应用于电机控制、通信、图像处理等领域。

机器人关节的伺服运动控制器需要实时接受机器人操作者的控制指令，计算并输出关节电机的控制信号，监控机器人系统的状态，因此其处理器必须具有较强的运算能力、较短的运行周期以及高度可靠的性能。论文研究使用的是 TI 2000 系列最新产品 TMS320F2812，附录 3 中给出了 TMS320F2812 的外观图片，此产

品是当今世界最先进、功能最强大的 32 位定点 DSP 芯片。它既有较高的运算精度 (32 位), 又有强大的事件处理能力 (达到 150MIPS), 特别适用于数据处理繁重的测控场合^[16, 17, 18]。TMS320F2812 功能结构框图如图 2-4 所示。

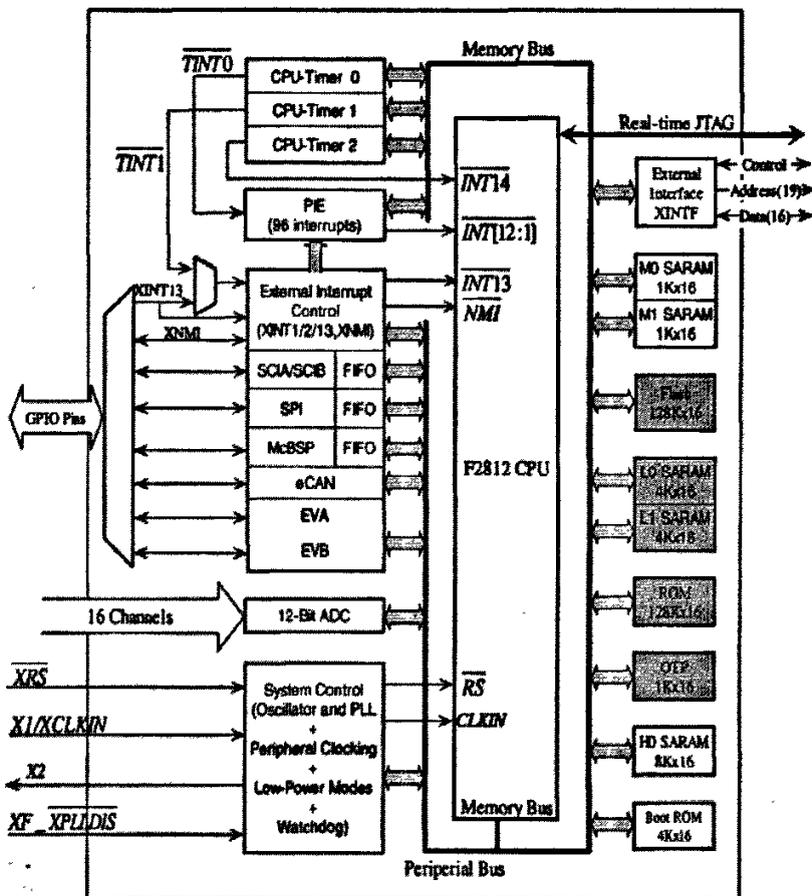


图 2-4 TMS320F2812 功能结构框图

Fig.2-4 TMS320F2812 function construction diagram

其主要特点有:

(1) 高性能静态 CMOS 技术

- 工作频率 150MHz(时钟周期 6.67ns);
- 低功耗设计;
- Flash 编程电压 3.3V。

(2) 高性能的 32 位中央处理器

- 16 位 X 16 位和 32 位 X 32 位的乘累加操作;
- 16 位 X 16 位的双乘法累加器;
- 哈佛总线结构;
- 快速中断响应和处理能力;
- 代码高效(支持 C/C++和汇编语言);

- 与 TMS320F24X/F240X 处理器的源代码兼容。
- (3) 片上存储器
- 最长达 128K X 16 位 (F2812) 的 Flash 存储器;
 - 1K X 16 位的 OTP 型只读存储器;
 - L0 和 L1: 两块 4K X 16 位的单周期访问 RAM(SARAM);
 - H0: 一块 8K X 16 位的单周期访问 RAM(SARAM);
 - M0 和 M1: 两块 1K X 16 位的单周期访问 RAM(SARAM)。
- (4) 引导 (BOOT) ROM
- 带有软件启动模式;
 - 数学运算表。
- (5) 外部存储器接口 (F2812)
- 最多 1MB 的寻址空间;
 - 可编程等待周期;
 - 可编程读/写选择时序;
 - 三个独立的片选信号。
- (6) 时钟与系统控制
- 支持动态改变锁相环的倍频系数;
 - 片上振荡器;
 - 看门狗定时器模块。
- (7) 外部中断扩展 (PIE) 模块可支持 45 个外设中断
- (8) 3 个 32 位的 CPU 定时器
- (9) 电机控制外围设备, 两个与 F240xA 兼容的事件管理器模块, 每一个管理器模块包括:
- 两个 16 位的通用目的定时器;
 - 8 通道 16 位的 PWM;
 - 外部可屏蔽功率或驱动保护中断;
 - 三个完全比较单元;
 - 三个捕捉单元, 捕捉外部事件;
 - 正交脉冲编码电路;
 - 同步模数转换单元。
- (10) 串口外围设备
- 串行外围接口 (SPI);
 - 两个串行通信接口 (SCI), 标准的 UART;
 - 增强的 eCAN 2.0B 接口模块;
 - 多通道缓冲串行接口 (McBSP) 。
- (11) 12 位的模数转换模块

- 2X8 通道的输入多路选择器;
 - 两个采样保持器;
 - 流水线最快转换周期为 60ns, 单通道最快转换周期为 200ns。
- (12) 高达 56 个独立的可编程、多用途通用输入/输出 (GPIO) 引脚

2.4 各个功能模块分析

2.4.1 中央处理器

中央处理器是指我们选用的 TMS320F2812 型 DSP 芯片。我们用它来实现高速下位机控制算法、与上位机通讯和中断处理及异常事件的处理等操作。

2.4.2 编码器电路

检测元件是闭环伺服控制系统中必不可少的组成部分。在实际应用中,采用脉冲编码器作为角位移传感器,把机械转角变成电脉冲。编码器分为光电式、接触式和电磁感应式。数控机床上只使用体积小、精度高、工作稳定、可靠性高的光电式脉冲编码器。因为编码器产生的信号,是作为伺服闭环系统位置环的检测反馈信号,直接影响被控制轴的运动性能、控制系统的控制精度和智能水平,所以,不仅对编码器的性能要求高,同时也对处理编码器信号的电路要求也高。

光电脉冲编码器分为增量式脉冲编码器和绝对式脉冲编码器。增量式旋转编码器每转一周产生一系列脉冲,脉冲的多少表示角位移的大小;绝对式旋转编码器是通过读取编码盘上的图案来表示轴的位置,直接得到角度坐标的绝对值。论文采用的是增量式脉冲编码器^[19]。

在电机旋转时,光电编码器会利用其光码盘产生 A、B 两路正交的高速脉冲信号;而且电机每旋转一圈,编码器还会产生一个对应于电机零位的脉冲信号(称之为 Z 脉冲)。每个驱动器分别接收对应的光电编码器反馈的 A 相、B 相脉冲和 Z 脉冲,通过内部处理使之变成单路的位置脉冲和方向信号,并经过双向的计数处理,最后得到 DSP 所能接受的电机位置增量信息。

2.4.3 数/模转换输出电路

SEED-DEC2812 开发板上使用了一片 DAC7724 做模拟量输出, DAC7724 有 4 路 D/A 输出,输出的模拟信号范围为 $\pm 10V$ ^[20]。

DAC7724 为 4 通道、12 位分辨率、±10V 信号量程、建立时间 10us、±15V 供电的 D/A 转换器。它内部有 2 级锁存，先将数据锁存到缓冲器中，然后再锁存到输出寄存器，实现多通道同时输出。其原理框图如下图所示：

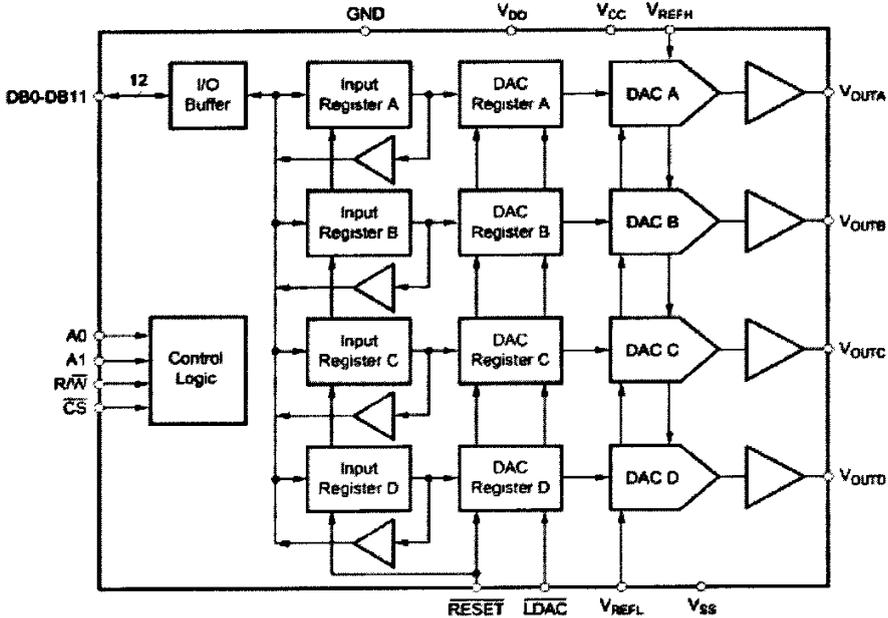


图 2-5 DAC7724 原理框图

Fig.2-5 DAC7724 schematic diagram

DAC7724 采用直接二进制编码，理想的输入数字量与输出模拟量之间的关系如下式所示：

$$V_{OUT} = V_{REFL} + (V_{REFH} - V_{REFL}) \times N \div 4096 \quad (2-1)$$

其中 N——数字输入；

V_{OUT} ——模拟输出；

V_{REFL} ——低参考电压，可以为-10V 或 0V，缺省为-10V；

V_{REFH} ——高参考电压，可以为+10V 或+5V，缺省为+10V

复位时，输出锁存器中的数值为 800H，所以当配置为±10V 输出时，模拟输出的复位值为 0V。

2.4.4 USB 通信接口电路

在 SEED-DEC2812 开发板上集成了 Cypress 公司的 CY7C68001 芯片^[21]。CY7C68001 上集成了 USB2.0 收发器(物理层)、USB2.0 串行接口引擎 SIE 链路层，实现底层通信协议)。CY7C68001 则作为 F2812 的外设，USB 的应用层协议由 F2812 编程实现。CY7C68001 采用并行异步存储器接口与 F2812 相连接，主

机可以唤醒 F2812，也可以配置 USB。

CY7C68001 芯片具有如下特点：

- 符合 USB2.0 标准，最高速度可达 480Mbps；
- 支持控制节点 0，用于处理 USB 传输的申请；
- 内部有 4K 字节的 FIFO 资源；
- 具有内部的锁相环；
- 具有同步与异步的 FIFO 接口。

CY7C68001 有两个外部接口：

命令接口：用来访问 CY7C68001 寄存器、Endpoint0 缓冲器、及描述表。

FIFO 数据接口：用来访问 4 个 1K 字节的 FIFO 中的数据。

这两个外部接口均可以通过同步或是异步的方式进行访问。在 SEED-DEC2812 模板中均采用异步的方式进行访问的。

2.5 本章小结

本章主要介绍了基于 DSP 的机器人位置控制系统的硬件结构。首先分析了整个系统的总体结构，将系统分为上位机子系统和下位机子系统两部分，接着详细介绍了下位机子系统的 DSP 控制器和交流伺服驱动器，并对各个功能模块进行了分析。

第 3 章 机器人位置控制器的控制算法研究

3.1 位置伺服系统的数学模型

系统所用松下伺服电机为永磁式同步交流伺服电机 (PMSM) [22, 23], 图 3-1(a) 为电机内部物理模型简图。取图中永磁体基波励磁磁场轴线(磁极轴线)为 d 轴(直轴), 而 q 轴(交轴)顺着旋转方向超前 d 轴 90° 电角度。 d_q 轴系随同转子以电角速度(电角频率) ω_r 一道旋转, 它的空间坐标以 d 轴与参考轴 as 间的电角度 θ_r 来确定。则可将图 3-1(a) 表示为图 3-1(b) 形式。

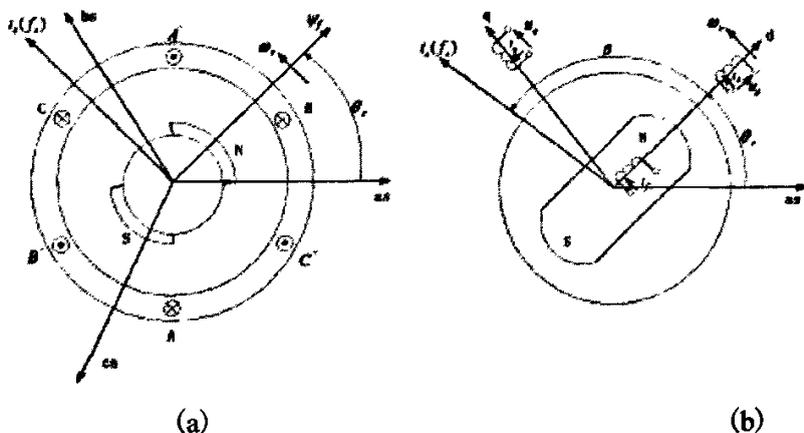


图 3-1 电机内部物理模型简图

Fig.3-1 The inner Physical model diagram of Motor

为了建立电动机的数学模型, 首先假设:

- (1) 忽略电动机铁心的永磁饱和;
- (2) 不计电动机的涡流和磁滞损耗;
- (3) 电动机的电流为对称的三相正弦电流。

由此可以得到如下的电压、磁链、电磁转矩和机械运动方程(式中各量为瞬态值)。

在转子参考坐标下的电压方程为

$$u_d = \frac{d\psi_d}{dt} - \psi_q \omega_r + R_s i_d \quad (3-1)$$

$$u_q = \frac{d\psi_q}{dt} - \psi_d \omega_r + R_s i_q \quad (3-2)$$

$$0 = \frac{d\psi_{2d}}{dt} + R_{2d} i_{2d} \quad (3-3)$$

$$0 = \frac{d\psi_{2q}}{dt} + R_{2q}i_{2q} \quad (3-4)$$

磁链方程为

$$\psi_d = L_d i_d + L_{md} i_{2d} + L_{md} i_f \quad (3-5)$$

$$\psi_q = L_q i_q + L_{mq} i_{2q} \quad (3-6)$$

$$\psi_{2d} = L_{2d} i_{2d} + L_{md} i_d + L_{md} i_f \quad (3-7)$$

$$\psi_{2q} = L_{2q} i_{2q} + L_{mq} i_q \quad (3-8)$$

以上各式中,

u_d 、 u_q ——d、q 轴电压;

i_d 、 i_q ——d、q 轴电流;

L_d 、 L_q ——d、q 轴电感;

R_s ——定子绕组每相电阻;

ω_r ——转子旋转的电角速度;

ψ_f ——永磁体基波励磁磁场过定子绕组的磁链。

结合上面各方程, 得到电机电压方程

$$u_d = L_d p i_d + p \psi_f - \omega_r L_q i_q + R_s i_d \quad (3-9)$$

$$u_q = L_q p i_q + \omega_r L_d i_d + \omega_r \psi_f + R_s i_q$$

其中 p ——微分算子。

电磁转矩为

$$T_{em} = p_n [\psi_f i_q + (L_d - L_q) i_d i_q] \quad (3-10)$$

机械动力学方程为

$$T_{em} = T_l + B\omega_r + Jp\omega_r \quad (3-11)$$

式中 T_l ——负载转矩

B ——运动阻尼系数

J ——转子惯量

在电机运行过程中, 若不计温度变化对永磁体供磁的影响, 则可认为 ψ_f 是恒定的, 可以得到:

$$\begin{aligned} p i_d &= (u_d - R_s i_d - \omega_r L_q i_q) / L_d \\ p i_q &= (u_q - R_s i_q - \omega_r \psi_f - \omega_r L_d i_d) / L_q \\ p \omega_r &= \{p_n [\psi_f i_q + (L_d - L_q) i_d i_q] - T_l - B\omega_r\} / J \\ p \theta &= \omega_r \end{aligned} \quad (3-12)$$

其中 p ——微分算子。

位置环和速度环控制器最终是通过 i_q 控制来达到目的的。在设计位置环, 速

度环和电流环时，可以用伺服电机简化的等效数学模型^[24, 25]。即使用 $i_d = 0$ 时的数学模型。此时伺服电机的状态方程为：

$$\begin{aligned} p i_q &= (u_q - R_s i_q - \omega_r \psi_f) / L_q \\ p \omega_r &= (p_n \psi_f i_q - T_l - B \omega_r) / J \\ p \theta &= \omega_r \end{aligned} \quad (3-13)$$

其中 p ——微分算子。

由上述 3 个公式可得到伺服电机的结构框图：

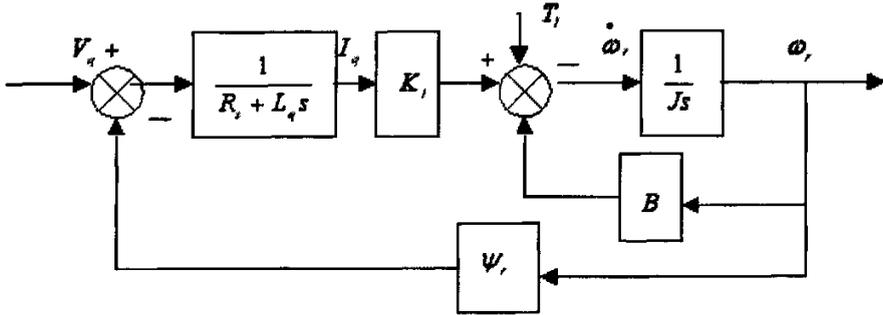


图 3-2 伺服电机的结构框图

Fig.3-2 structure diagram of servo motor

具有位置反馈、速度反馈和电流反馈的三闭环伺服系统的原理图如下图：

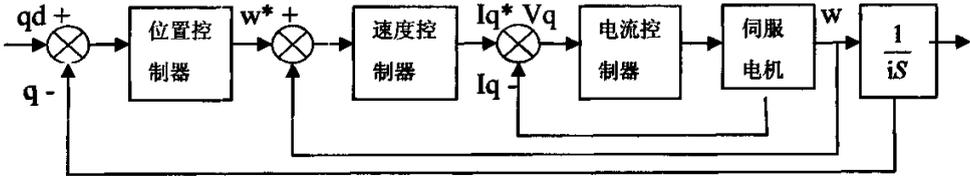


图 3-3 三闭环伺服系统原理图

Fig.3-3 theory diagram of three-closed loop servo system

利用伺服电机简化等效数学模型组成的系统结构如下图所示：

其中，电流环和速度环是内环，位置环是外环。采用位置控制方式时，需要对内环的数学模型进行简化。理论分析表明，当内环频率较低时，可以用简单的一节惯性环节代替内环。在本系统中，位置环的截止频率远低于内环截止频率，所以在分析设计的时候，可将内环按一阶近似处理，即 $\frac{K_s}{T_s s + 1}$ 。其中 K_s 为速度环增益； T_s 为速度环一阶近似等效时间常数。

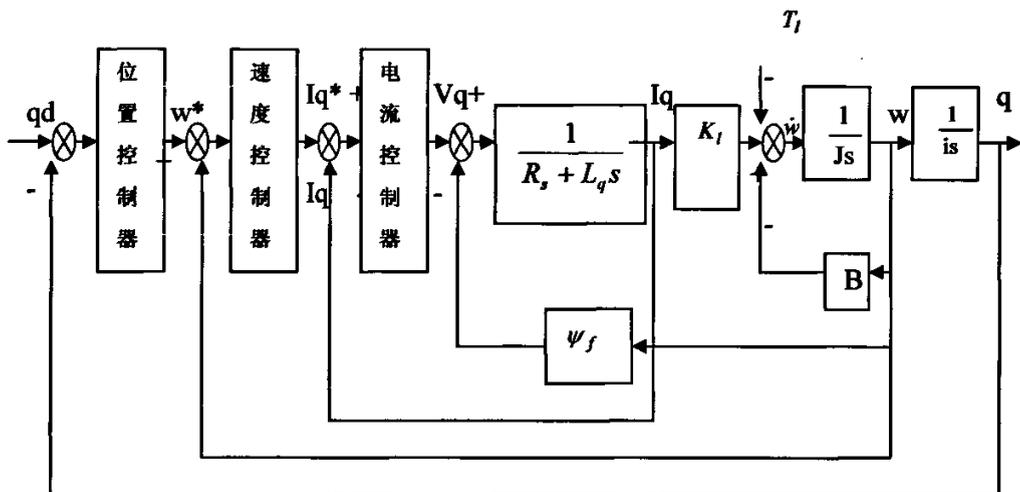


图 3-4 永磁同步电机位置伺服系统结构图

Fig.3-4 structure diagram of permanent-magnet synchronous motor position servo system
因此得到位置环的结构框图如图所示：

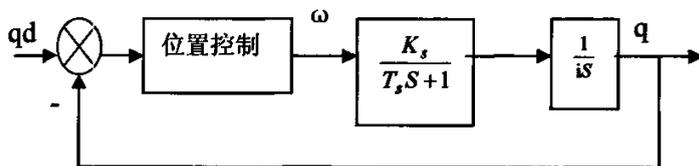


图 3-5 位置环的结构框图

Fig.3-5 Structure diagram of position loop

图中， i 为减速比。

通过驱动器控制面板上的参数设定，得到本系统的被控对象的第一关节数学模型可以近似用传递函数 $\frac{220}{3.5s^2 + 70s}$ (减速器比例为 70:1) 来表示，第二关节数学模型可以近似用传递函数 $\frac{220}{0.05s^2 + s}$ (无减速器) 来表示。

3.2 模糊 PD 结合单神经元自适应 PID 的双模控制算法

目前市场上成型的运动控制器的控制算法较多采用 PID 算法。机器人关节各自由度的转动惯量和摩擦系数都会随机器人姿态的改变而发生较大范围变化，PID 调节的鲁棒性不适应机器人协调控制精度和步行速度的提高^[26]。模糊控制有较强的鲁棒性，有学者将其用于运动控制。为保证控制的实时性，大多先将连续的控制量离散化，再离线计算出控制器输出量的查询表，在线控制时，通过查表的方式得出控制量，但这样会引入量化误差。当模糊控制器的输入输出量在查询表零档附近时，系统会出现稳态误差或稳态抖动，给控制精度带来不良影

响^[27]。单神经元因其结构简单，具有良好的调整性和自适应性，已成功应用于交流伺服系统。但单神经元控制在输入量大范围变化时调整速度变慢，是其有待解决的问题^[28]。

论文采用模糊 PD 控制算法结合单神经元自适应 PID 算法的双模控制(见图 3-6)。

图中

θ_d ——位置给定值；

θ ——位置输出值；

$e = \theta_d - \theta$ ；

U_f ——模糊 PD 控制器的输出值；

U_n ——单神经元自适应 PID 控制器的输出值；

U ——位置环控制器的输出值；

$e_U = U_f - U_n$

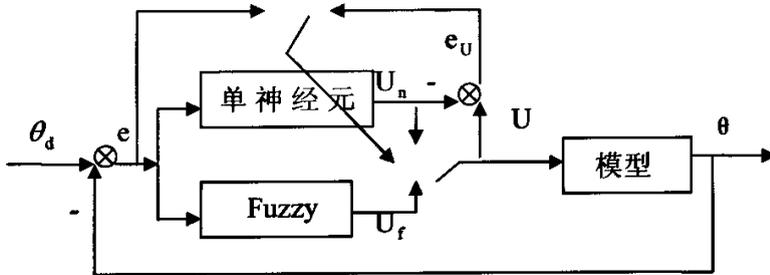


图 3-6 双模控制算法结构图

Fig.3-6 structure diagram of dual-mode control strategy

如上图所示，在位置伺服控制过程中，当位置偏差量较大时，模糊 PD 控制器起作用，即 $U = U_f$ ；单神经元自适应 PID 控制器的连接权沿 $E_U = \frac{1}{2}e_U^2$ 的负梯度方向进行修正，从而使 U_n 跟踪 U_f 。这时系统利用模糊 PD 控制器加快响应速度，克服单神经元 PID 控制调节速度慢的缺陷^[29]。当偏差量 E_q 减小到一定程度时，单神经元自适应 PID 控制器起作用，即 $U = U_n$ ，而其连接权也改为沿 $E_q = \frac{1}{2}e_q^2$ 的负梯度方向进行修正，最终实现 $q = q_d$ ，从而克服模糊 PD 控制存在稳态抖动和稳态误差的缺陷。当被控对象的参数发生变化时也可通过单神经元的自学习自调整获得良好的控制品质。在系统从模糊 PD 控制器切换到单神经元自适应 PID 控制器之前，由于单神经元自适应 PID 控制器的连接权沿 $E_U = \frac{1}{2}e_U^2$ 的负梯度方向进行修正，从而使 U_n 跟踪 U_f ，所以当切换控制器时，可实现无冲击切换。

3.2.1 模糊PD控制

将位置误差 e 和误差变化率 e_c 作为模糊控制器的输入，经过模糊化、模糊推理和解模糊过程，输出控制电压值 U_f 。输入和输出在 $[-6, 6]$ 之间分别划分为 7 个模糊子集即 NB(负大)、NM(负中)、NS(负小)、ZE(零)、PS(正小)、PM(正中)、PB(正大)； e 和 e_c 的论域经过标度变换后都离散量化为：-6、-5、-4、-3、-2、-1、0、1、2、3、4、5、6。输入的隶属函数采用高斯函数；输出的隶属函数也采用高斯函数；模糊规则基于专家的经验和控制工程知识，如表 3-1 所示；为提高在线控制的实时性，离线计算出控制查询表(见表 3-2)。模糊推理采用经典的极大—极小合成运算方法；解模糊方法采用重心法^[30,31]。

表 3-1 模糊规则
Table3-1 fuzzy rules

U_f e \ e_c	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NB	NM	Z	Z
NM	NB	NB	NB	NB	NM	Z	Z
NS	NM	NM	NM	NM	Z	PS	PS
Z	NM	NM	NS	Z	PS	PM	PM
PS	NS	NS	Z	PM	PM	PM	PM
PM	Z	Z	PM	PB	PB	PB	PB
PB	Z	Z	PM	PB	PB	PB	PB

模糊 PD 控制器的量化因子分别取 $k_e=10$ ， $k_{e_c}=0.07$ ；位置环的输出控制值即为驱动器的速度指令值，范围为 $0 \sim \pm 10v$ ，故比例因子设定为 $\frac{10}{6}$ ，将计算出的控制量送入系统的数学模型进行控制。

表 3-2 离线控制查询表
Table3-2 Query table of off-line control

U \ e	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-6	-5.24	-5.12	-5.00	-4.98	-4.78	-4.60	-4.47	-4.06	-3.16	-1.94	-0.69	-0.07	0.00
-5	-5.15	-5.05	-4.98	-4.72	-4.59	-4.06	-4.05	-3.98	-2.93	-1.90	-0.79	-0.02	0.04
-4	-4.78	-4.60	-4.58	-4.50	-4.77	-4.06	-3.16	-2.93	-2.62	-1.60	-0.35	0.34	0.45
-3	-4.17	-4.12	-4.10	-4.06	-4.05	-3.98	-2.93	-1.89	-1.57	-0.83	0.32	0.96	1.01
-2	-3.95	-3.89	-3.61	-3.47	-3.16	-2.93	-2.62	-1.58	-0.00	0.58	1.06	1.83	1.93
-1	-3.89	-3.82	-3.47	-2.84	-2.54	-1.90	-1.58	-0.83	0.58	1.90	2.54	2.84	2.91
0	-3.61	-3.47	-3.16	-2.54	-1.06	-0.58	-0.00	0.58	1.06	2.54	3.16	3.47	3.61
1	-2.91	-2.83	-2.55	-1.90	-0.58	0.83	1.58	1.90	2.55	2.86	3.47	3.82	3.88
2	-1.93	-1.83	-1.06	-0.58	-0.00	1.58	2.62	2.93	3.16	3.47	3.61	3.89	3.95
3	-0.97	-0.95	-0.32	0.83	1.58	2.16	3.22	3.98	4.06	4.08	4.12	4.14	4.17
4	-0.00	0.02	0.35	1.60	2.62	3.22	3.46	4.40	4.59	4.70	4.72	4.75	4.78
5	0.36	0.98	1.24	1.92	2.63	3.38	3.98	4.20	4.65	4.93	5.05	5.09	5.15
6	0.33	1.01	1.93	2.60	3.16	4.06	4.77	5.04	5.09	5.12	5.16	5.18	5.24

3.2.2 单神经元自适应 PID 控制器

PID 的增量式控制表达式为

$$u(k) = u(k-1) + K_p[e(k) - e(k-1)] + K_i e(k) + K_d[e(k) - 2e(k-1) + e(k-2)] \quad (3-14)$$

以上式为依据，构造单神经元控制器模型如图 3-7 所示，

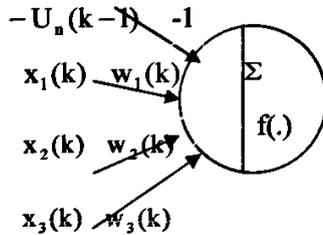


图 3-7 单神经元控制器模型
Fig.3-7 Model of single neuron controller

图中

$U_n(k)$ ——单神经元控制器的第 k 次采样时的输出值；

$-U_n(k-1)$ ——单神经元控制器的第 $k-1$ 次采样时的输出值取负，作为神经元的阈值；

x_i ($i=1,2,3$)——单神经元控制器的输入；

w_i ($i=1,2,3$)——单神经元控制器的连接权；

$f(\cdot)$ ——单神经元控制器的输出变换函数;

$$U_n(k) = f\{U_n(k-1) + x_1 w_1 + x_2 w_2 + x_3 w_3\} \quad (3-15)$$

$f(\cdot)$ 的函数表达式为

$$f(s) = \begin{cases} U_{\max} & U \geq U_{\max} \\ U & U_{\min} < U < U_{\max} \\ U_{\min} & U \leq U_{\min} \end{cases}$$

对照式(3-14)和式(3-15),可知图3-7所示的单神经元控制器能够实现PID控制算法。

输出的控制量 U_n 为伺服系统速度环的输入值, U_n 的最大、最小值分别为 $10v$ 、 $-10v$, 将其送入驱动器的速度指令引脚, 最终驱动电机运转。

该单神经元控制器的自适应功能是通过连接权 w_i 的自学习修正来实现的^[32]。

^{33]}。当模糊PD控制器起作用时, w_i 沿 $E_U = \frac{1}{2}e_U^2$ 的负梯度方向进行修正,

$$w_i(k+1) = w_i(k) - \eta \frac{\partial E_U(k)}{\partial w_i(k)} = w_i(k) - \eta \frac{\partial E_U(k)}{\partial U_n(k)} \frac{\partial U_n(k)}{\partial w_i(k)} \quad (3-16)$$

式中 $\eta > 0$ ——学习率。

通过学习, 使 U_n 跟踪 U_f , 当切换控制器时, 可实现无冲击切换。

当单神经元自适应PID控制器起作用时, w_i 沿 $E_q = \frac{1}{2}e_q^2$ 的负梯度方向进行修正:

$$w_i(k+1) = w_i(k) - \eta \frac{\partial E_q(k)}{\partial w_i(k)} = w_i(k) - \eta \frac{\partial E_q(k)}{\partial q(k)} \frac{\partial q(k)}{\partial U_n(k)} \frac{\partial U_n(k)}{\partial w_i(k)} \quad (3-17)$$

式中 $\eta > 0$ ——学习率。

3.2.3 仿真研究

论文以关节1为模型进行控制算法的仿真。单神经元自适应PID控制器的初始权值为 $w = [1, 0, 0]$, 学习速率分别为 $0.5, 0.5, 0.25$ 。系统采样周期为 $T=0.2ms$ 。

图3-8给出了单独使用模糊控制器和单神经元控制器, 以及使用双模控制算法的阶跃响应仿真结果。从图中可以看出, 当只采用模糊PD控制时, 系统响应速度较快, 但系统存在静态误差; 当只采用单神经元自适应PID控制时, 系统无静态误差, 但调整时间较长; 当采用双模控制时, 系统结合了以上两种控制器的优点, 同时满足了系统的快速性和准确性, 是一种较为理想的控制算法。

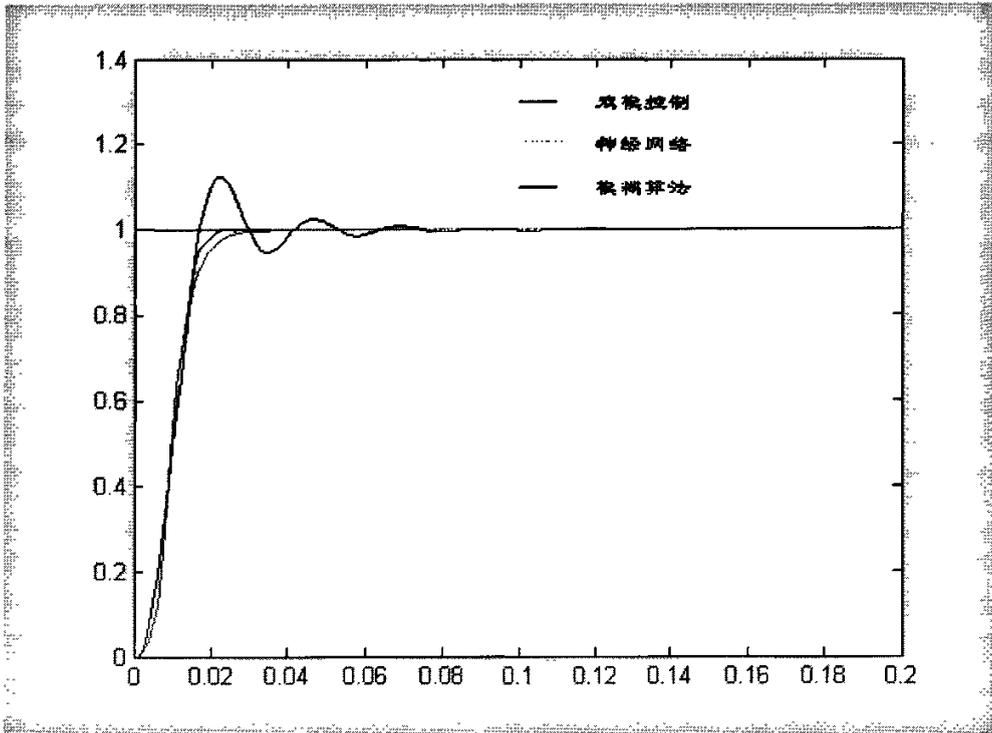


图 3-8 阶跃响应仿真结果
Fig. 3-8 Simulation results of step response

3.3 模糊 CMAC 控制算法

3.3.1 CMAC 技术

小脑模型关节控制器 (CMAC) 是由 Albus 最初于 1975 年基于神经生理学提出的, 它是一种基于局部逼近的简单快速的神经网络, 类似于 Perceptron 的相连记忆方法, 能够学习任意多维非线性映射, 迄今已广泛用于许多区域。特别是 Miller 等的突破性应用研究, 已使 CMAC 受到越来越多的关注^[34, 35, 36]。

与 BP 网络之类的全局逼近算法不同, CMAC 具有许多优点: 它具有局部逼近能力, 每次修正的权值极少, 学习速度快, 适合于在线学习; 具有一定的泛化能力, 相近输入给出相近输出, 不同输入给出不同输出; 具有连续 (模拟) 输入输出泛化能力; 具有寻址编程方式, 在利用串行计算机仿真时, 它使回响速度加快。

在 CMAC 技术里, 每个状态变量被量化并且问题空间被划分为离散状态。量化输入构成的向量制定了一个离散状态并且被用于产生地址来激活联想单元中存储的联想强度从而恢复这个状态的信息。图 3-9 描述了 CMAC 的空间划分和量化机制。这个简单的例子有两个状态变量 (v_1 和 v_2), 每个变量在论域上被

划分为两个离散的区域叫做“块”。块的宽度影响 CMAC 的概括能力。为了能用较简洁的矩阵形式来描述 CMAC 的机制和特性，在这里我们将块的数量限定为 2。譬如，v1 被划分为 A、B；v2 被划分为 a、b。区域 Aa、Ab、Ba、Bb 被称作“超立方体”。通过将每个变量以相同的方式平移一小段间隔(称为“元素”，如图 3-9 中的 1、2、3、4)，可以获得不同的超立方体。例如，v1 通过平移后的区域 C、D 和 v2 的 c、d 组成一层超立方体 Cc、Cd、Dc、Dd。我们规定，相同的划分方式组成一层超立方体，例如 v1 的第 p 种划分方式与 v2 的第 p 种划分方式对应组成第 p 层超立方体。用上述方法分解，我们可以得到一共有 Ne 层(在本图中为 3 层)的超立方体，其中每一层对应一种划分方式。从图中可以看出，两变量的每一种状态组合在每一层都被一个超立方体覆盖，对应所有的划分方式，一共被 Ne 个不同的超立方体覆盖。也就是说两变量的每一种状态将激活 Ne 个不同的超立方体，而 Ne 的大小由每个输入论域上块的数量和块中包含的元素数决定。CMAC 为每个超立方体分配一个物理的存储单元(联想单元)，每个联想单元中存储着相应的超立方体对于输出的影响强度(联想强度)，这样，两变量的每一种状态将激活 Ne 个不同的联想单元，被激活的 Ne 个联想单元又以不同的联想强度影响输出。

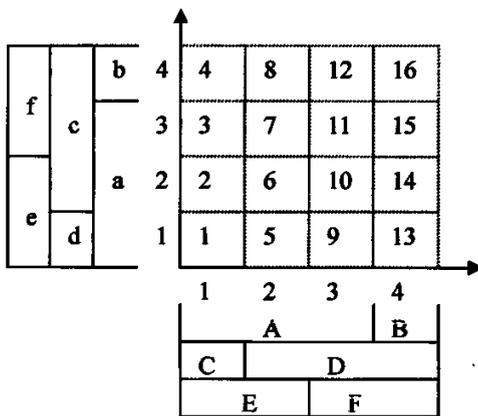


图 3-9 CMAC 的空间划分和量化机制

Fig.3-9 spatial partition and quantifying mechanism of CMAC

用 CMAC 技术，第 s 个状态对应的输出数据 y_s 可以表示为

$$y_s = C_s^T W = [c_{s,1}, c_{s,2}, \dots, c_{s,N_h}] \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_{N_h} \end{bmatrix} \quad (3-18)$$

式中， N_h 为总的联想单元的数量；W 代表联想强度的向量； C_s 是存储单元激活向量，该向量包含 N_e 个 1。

但是随着输入维数和分辨率的增加，CMAC 技术所需存储容量将呈几何级数增加。由于实现时受存储空间与实时性的限制，CMAC 的泛化能力较之全局

逼近的 BP 网络差距较大。而且 CMAC 在对输入空间进行划分时，将输入空间简单的划分为若干个“块”，输入状态与这些“块”之间是简单的“属于”与“不属于”的关系，是无法进行在线调整的，不能很好地应用于要求精度高且不确定性严重的场合。针对 CMAC 的这些缺点，在下一节将模糊理论引入 CMAC，提出了一种模糊小脑模型关节控制器 FCMAC。

3.3.2 模糊 CMAC 控制算法

模糊 CMAC 控制算法将模糊理论引入 CMAC，在对输入空间进行划分时和激活联想强度时都进行模糊化处理，使得 FCMAC 能够很好地反映人脑认知的模糊性和连续性，而且由于在模糊化处理时采用了连续可微分的高斯隶属函数，FCMAC 对输入空间的划分方式及输入状态激活联想强度的活性是可以在线调整的，具有较强的自学习能力，能够有效提高系统的控制精度。而且由于 FCMAC 的这种在线调整能力，因此无须再对“块”进行多种方式的划分，超立方体的数量将大大减少，相应的联想单元数量也将大大减少，这将有效地节省存储空间^[37, 38]。如上节中，双变量 CMAC 的块的数量为 2，对应的划分方式有 3 种，对应 $3 \times 4 = 12$ 个联想单元；而在双变量 FCMAC 中，如果块的数量是 2，由于无须多种划分方式，超立方体的数量是 $2 \times 2 = 4$ 个，联想单元的数量也是 4 个，大大地节省了存储空间。

FCMAC 的工作机理可以这样描述：通过对输入的模糊量化，得出输入向量激活联想强度的“活性”，进而激活联想强度以恢复系统的信息^[39, 40]。下面以双输入单输出的 FCMAC 为例说明 FCMAC 的结构(见图 3-10)。

控制器的第一层是将引入输入。

$$o_i^{(1)} = x_i, \quad (i=1, 2) \quad (3-19)$$

x_i 为第 i 个神经元的输入。

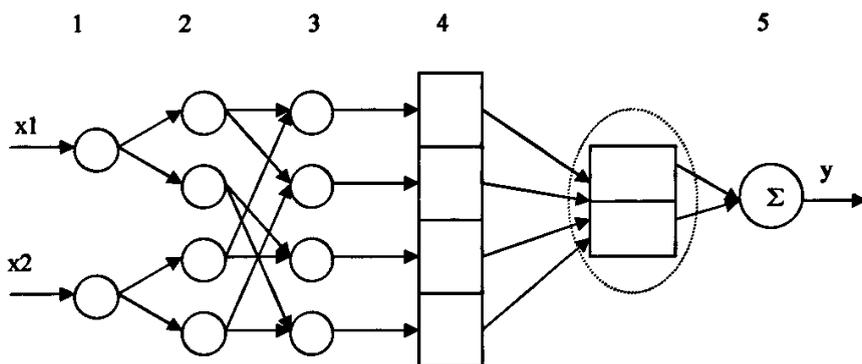


图 3-10 FCMAC 结构图
Fig.3-10 Structure diagram of FCMAC

第二层进行模糊量化，模糊量化的过程是在每个输入 x_i 的论域上定义 n 个块。在论文中，我们定义 $n=3$ 。输入 x_i 对应第 j 个“块”的隶属关系是高斯函数关系

$$o_{i,j}^{(2)} = \exp\left(-\left(\frac{x_i - a_{i,j}}{b_{i,j}}\right)^2\right) \quad (i=1, 2, j=1, 2, 3) \quad (3-20)$$

其中 $a_{i,j}$ 表示函数的中心值， $b_{i,j}$ 表示函数的宽度。

通过式(3-20)的定义，FCMAC 中输入状态变量与“块”之间的关系被模糊化了，它们之间的隶属关系不再是 CMAC 中简单的“属于”、“不属于”的关系，而是用连续的隶属度来表征。

第三层用于得到输入对联想单元的激活强度。在论文中通过模糊“与”操作来求得相应的激活强度 $o_{i,j}^{(3)}$ 。

$$o_{i,j}^{(3)} = \min\{o_{1,j}^{(2)}, o_{2,j}^{(2)}\} \quad (j=1,2,3) \quad (3-21)$$

第四层以第三层求出的激活强度激活联想单元中的联想强度。各联想单元的输出

$$o_{i,j}^{(4)} = o_{i,j}^{(3)} * w_{i,j} \quad (i=1,2;j=1,2,3) \quad (3-22)$$

式中 $w_{i,j}$ 为各联想单元中存储的联想强度。

第五层对联想单元的输出进行求和以恢复系统的信息，得到系统的输出。

$$y = \sum_{i=1}^2 \sum_{j=1}^3 o_{i,j}^{(4)} \quad (3-23)$$

如果系统的输入量过多或定义的“块”的数量过多，导致联想单元所需的存储空间过大，可以在第四层和第五层之间加入一层 hash 映射，如图 3-10 中的虚线框中部分，在 hash 单元中存放联想强度，在联想单元中存放散列地址编码，论文没有涉及到这种情况，没有加入 hash 映射。

论文采用位置误差 e 和误差变化率 e_e 作为 FCMAC 控制器的输入，通过经过量化、联想强度的激活、和输出求和过程，输出控制电压值 U 。 U 的最大、最小值分别为 $10v$ 、 $-10v$ ，将输出电压转化为模拟量，送入驱动器的速度指令输入引脚，进而驱动电机运转^[41]。

FCMAC 采用有导师的学习算法，分为离线学习和在线学习两个阶段。在离线学习阶段，采用模糊控制表中的样本数据进行训练，训练的值包括 w_y 以及高斯函数的参数 a_{ij} 、 b_{ij} ，在在线学习阶段将离线训练好的网络放入系统运行，根据定义的目标函数对权值进行微调，在线阶段只调整 w_y ，保证系统的实时性。

定义 FCMAC 的目标误差函数为 $E(k) = \frac{1}{2} [q_d(k) - q(k)]^2$ ， w_y 采用 BP 算法进行权

值修正:

$$w_{ij}(k+1) = w_{ij}(k) - \eta \frac{\partial E(k)}{\partial w_{ij}(k)} = w_{ij}(k) - \eta \frac{\partial E(k)}{\partial q(k)} \frac{\partial q(k)}{\partial U(k)} \frac{\partial U(k)}{\partial w_{ij}(k)} \quad (3-24)$$

式中 $\eta > 0$, ——学习率。

3.3.3 仿真结果及分析

论文以关节 1 为模型进行控制算法的仿真。系统采样周期为 $T=0.2\text{ms}$ 。FCMAC 中定义在“块”上的高斯基模糊隶属度函数的参数初值分别为 $a_{i1} = -0.33$, $a_{i2} = 0.33$, $b_{i1} = 0.3$, $b_{i2} = 0.3$ ($i=1,2$)。联想强度的初值为 $[-1,1]$ 之间的随机值。

图 3-11 给出了 FCMAC 控制器的阶跃响应仿真结果。可以看出, 控制器较好地完成了对阶跃曲线的跟踪, 满足了系统对实时性和精确性的要求。

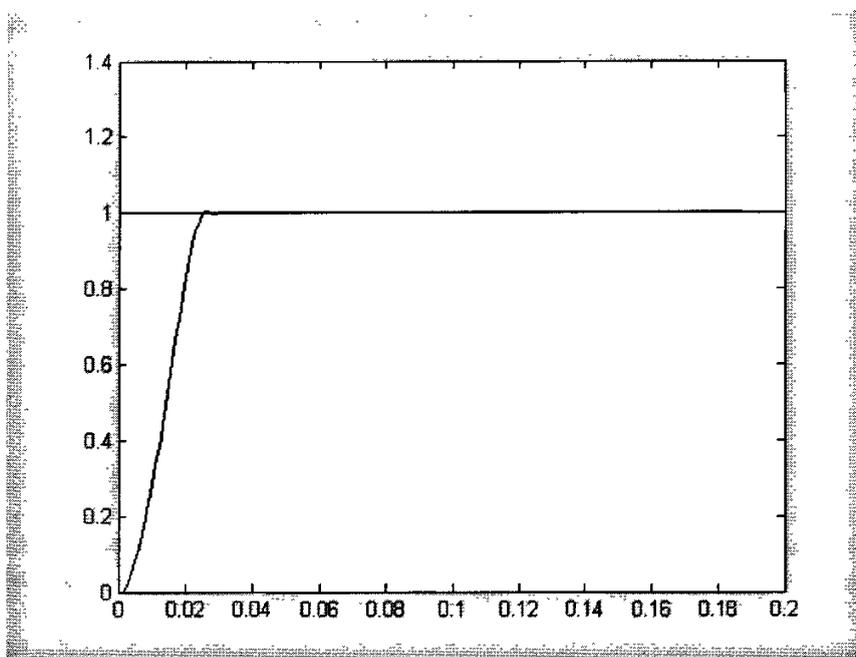


图 3-11 FCMAC 的阶跃响应仿真结果
Fig.3-11 Simulation results of FCMAC step response

3.4 本章小结

本章主要分析了交流伺服驱动系统的数学模型及两种智能控制算法。交流伺服驱动系统的数学模型可简化为一节惯性环节, 课题的仿真都是基于此简化的数

学模型进行的。课题采用的控制算法为基于模糊 PD 控制和单神经元自适应 PID 控制的双模控制算法，以及模糊 CMAC 控制算法，本章对这两种算法给出了详细的介绍。

第 4 章 机器人位置控制器的软件设计

机器人位置控制系统的软件设计包括三大部分^[42]：(1) 上位机的软件设计，它主要实现机器人各关节运动的轨迹规划，机器人系统的监控程序；(2) 机器人运动控制器的 DSP 控制程序的设计，它主要包括机器人控制主程序、中断服务子程序、电机控制子程序等；(3) USB 通信程序的设计。

机器人运动控制器的总体控制模型采用典型的两级控制方式，操作者通过上位计算机控制机器人系统进行轨迹规划，每隔 20ms 触发一次中断，将控制指令传输给 DSP 控制器。由于 DSP 控制器的运算速度高于上位机的运算速率，为了提高控制系统的精度，当 DSP 控制器接收到上位机指令后，对其进行简单的 100 等分，得到每次轨迹精细插补时的关节运动增量，调用控制子程序实现对关节电机的驱动，也就是说 DSP 控制器的控制周期为 0.2ms。精细插补完成后，将此时的机器人关节实际转角传递给上位机，与期望的关节转角进行比较，完成轨迹曲线的绘制。然后 DSP 再次查询 USB 的写中断，当上位机传输命令后再次进入精细插补，如此循环，直至结束。

4.1 DSP 控制器的软件设计总计结构

DSP 控制器的软件开发环境是 TI 公司开发的数字信号处理器系统集成开发环境 CC2000。按照功能的不同，DSP 控制器主要划分为初始化模块、中断服务程序模块、USB 通讯模块。DSP 控制器的主程序流程图见图 4-1。

4.1.1 初始化模块

系统初始化子程序主要完成以下功能：

(1) 系统寄存器初始化

包括对时钟倍频、看门狗定时器的寄存器、系统标志和状态寄存器、中断屏蔽寄存器和中断标志寄存器等设定，使 DSP 系统工作在预定的工作模式下。

(2) 定时器参数初始化

设置 DSP 内部定时器(T1, T2)的计数模式和相关计数寄存器。

(3) 全局变量的命名及初始化

论文中全局变量的命名都是实际变量所代表意义的字母的缩写。该命名方式使得在调试过程中修改和应用变量能够知道该变量的意义。

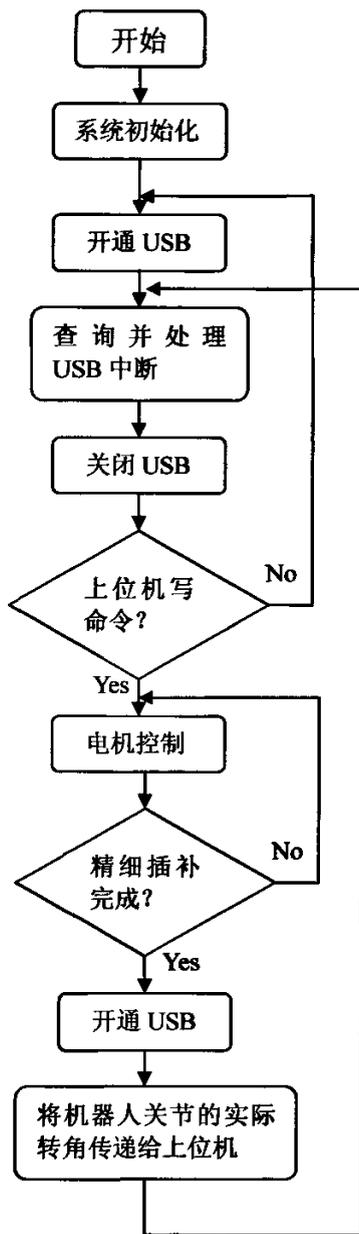


图 4-1 DSP 控制器的主程序流程图

Fig.4-1 Main program flow chart of DSP controller

(4) 中断的初始化

论文中用到了两种中断，一个是事件处理器周期中断，一个是为使用 CY7C68001 而在 DSP 中开设的外部中断。论文采用 EVA 中的定时器一、二同时定时中断，定时周期为 0.2ms。

4.1.2 中断服务程序模块

该模块中主要实现数据采集、智能控制算法、数/模转换输出的功能。这三部分的执行时间必须小于采样周期时间,否则将会出现在本次处理还没有完成的时候,下次的中断已经到了的情况,很快就会出现堆栈不够使用,系统也会因此崩溃^[43]。

(1) 数据采集模块

在每次采样时间到来的时候都需要得到该时刻的电机的位置值,论文使用的是松下的交流伺服电机,位置信号可以从电机尾部的光电编码器得到。这里电机驱动器中输出的信号是差分的正交编码脉冲,经过内部处理到达 DSP 的 QEP 电路。论文采用的是 DSP 中 EVA 模块的 QEP 2,采用通用定时器 2 来计数。EVA 模块的 QEP 电路的方向逻辑决定了两个序列中的哪一个是先导序列,从而确定通用定时器的计数方向,也就确定了电机的旋转方向。每次事件管理器周期中断时,读出通用定时器的数值,转化为电机在该时间段的实际位移。

(2) 控制算法模块

论文在第二章对神经网络的控制算法已有详述,这里就不多加介绍了。论文在实验时采用的是模糊 PD 结合单神经元自适应 PID 的双模控制算法。为了增加控制算法的精确性,将输入从本次期望脉冲增量值改为期望脉冲增量值加上历次误差积累值,将系统误差总值最终减小到零。

(3) 数/模转换输出模块

将控制模块计算出的控制量的值经过数/模转换,转化为相应的伺服驱动器的-10v~+10v 模拟输入电压,驱动电机转动。

论文采用的数/模转换芯片是 DAC7724。它内部有 2 级锁存,先将数据锁存到缓冲器中,然后再锁存到输出寄存器。数/模转换时将 D/A 转换的数值写到对应的输入数据寄存器中,将任意值写入 D/A 转换启动寄存器中,启动该片中所有 4 路 D/A 转换。表现为如下 2 条 C 语言指令:

```
*DA_PORTx=Vin[x];  
da_run();
```

第一条语句将需要转换的二进制数送入对应通道的 1 级缓存中,其中 x 代表需要处理的通道;第二条语句将第 1 级缓存的数字输入打入相应的输出寄存器中,并将数字输入转换为模拟输出。

4.1.3 USB 的通讯模块

USB 的编程主要分为两个部分,一是 HOST 方面的驱动程序的编写,一是有关 DSP 与 USB 的数据与命令的交换。HOST 方面的驱动程序已由厂家提供,

本课题主要完成的是 USB 与 DSP 通讯的应用程序。

F2812 是通过其 Zone 0 来完成 CY7C68001 的数据交换的。在这部分程序中主要完成以下功能：

完成对命令接口的读与写；

完成对 FIFO 接口的读与写；

完成 F2812 对 USB 的自举；

完成 F2812 与 USB 的控制命令的传输；

完成 F2812 与 USB 的各种方式的数据传输。

各部分的详细实现过程见下节。

4.2 USB 通讯系统的设计

通用串行总线 USB 目前已发展到 USB2.0。由于 USB 具有以下几方面的特点，所以普及迅速，逐渐确定了在计算机接口中的主导地位。

(1) 易用性。USB 支持热插拔和即插即用。

(2) 可扩展性。用户可以通过连接 USB 集线器到一个已有的端口来增加 USB 端口数量，集线器上还可以连更多的集线器。理论上可以连接 127 个外设到 USB 总线上。

(3) 快速性。USB1.1 版规定了两种传输速率：低速传输和全速传输。低速传输的速率是 1.5Mbps，全速是 12Mbps。这不但远远高于传统的串口传输速率，也比并口传输快了好多倍。而 USB2.0 允许的最高传输速率甚至高达 480Mbps。

(4) 可靠性。USB 的可靠性来自硬件设计和 USB 数据传输协议两方面的保证。USB 驱动器、接收器和电缆的硬件规范，消除了大多数可能引起数据错误的噪声；USB 协议使用了差错校验和数据重传机制，可以最大程度保证数据传输的准确性。

(5) 内置电源。USB 总线内置电源线，可以给外设提供 5V 和最多 500mA 的电源供应，满足大部分低功耗外设的电源要求。

由于 USB 具有这些突出的优点，不但一些传统外设开始提供 USB 接口，而且大量新型外设也把 USB 接口作为首选甚至惟一的接口，如 MP3 播放器、移动硬盘等。

由于 USB 的高可靠性和足够快的通信速率，USB 开始应用于工业级的实时通信和控制，例如机器人系统中示教盒与控制器的通信。论文讲述的用 USB 端口实现机器人上位机与下位机的通信也属于这些方面的应用。

4.2.1 CY7C68001 的命令接口读写模块

F2812 对 CY7C68001 寄存器的写操作由 3 个步骤组成:

- (1) 将写命令字写入命令口;
- (2) 将数据高 4 位命令字写入命令口;
- (3) 将数据低 4 位命令字写入命令口;

F2812 对 CY7C68001 寄存器的读操作由 2 个步骤组成:

- (1) 将读命令字写入命令口;
- (2) 读命令口。

以上操作均以 8 位数据宽度进行。

(1) 写命令口函数

为完成对命令口的操作,用函数将命令字传送给命令接口,在对寄存器读写操作时进行调用。

函数原型如下:

```
BOOL SX2_comwritebyte(unsigned int value)
```

当结果为 0 时,说明芯片自检尚未完成;当结果为非 0 时,说明芯片自检完成,可以执行操作。如果等待芯片自检过程中超时,则本次操作失败。

(2) 写 SX2 寄存器函数

由上面可知,写操作分 3 部分完成。写函数原型如下:

```
BOOL Write-SX2reg(unsigned char addr, unsigned int value)
```

参数 `addr` 为寄存器地址,首先将有效地址提取出来,然后调用命令字写操作函数 `SX2_comwritebyte`,将地址写入 SX2。写入地址后,先将高 4 位的数据提取出来,写入寄存器;再将低 4 位数据提取出来并写入,完成后返回 TRUE,如果写入地址错误或操作超时则返回 FLASE。

(3) 读 SX2 寄存器函数

读操作与写操作相同,首先提取地址,将要读取的寄存器地址写入 SX2,然后读取寄存器中的数据。与写操作不同,读操作需要改写 SX2 中断使能寄存器 (INTENABLE)中的 FLAGS 位,产生一个中断。该位在 SX2 任一输出端点的 F1F0 状态从空变为非空时置 1,同时产生中断。使用一个 bool 型全局变量 `readFlag` 来完成对 FLAGS 位的置位或复位,当需要产生读中断时,将该位置 1,中断得到响应后,该位自动清 0,并开始执行中断。读操作和写操作相同,当读取成功后,返回值为 TRUE;超时则返回值为 FALSE。函数原型如下:

```
BOOL Read_SX2reg(unsigned char addr, unsigned int *value)
```

4.2.2 CY7C68001 的 FIFO 接口读写模块

FIFO 的写函数允许用户一次性将多字节数据写入 FIFO, 这需要用户指定数据写入的端点, 数据以及数据长度。把多字节数据组织成单字节数据的数组, 然后将数组的首地址传递给端口, 则数据将被顺序输入。数据写入后, 返回 TRUE。

函数原型如下:

```
BOOL SX2_FifoWrite(int channel, unsigned int *pdata, unsigned length)
```

FIFO 读函数和写函数基本相同, 差别在于读函数为单字节操作, 不需要输入数据的长度。程序中, 通常要读取多字节的数据, 而读函数每次从 FIFO 只能一个字节, 因此如果想从 FIFO 中将数据完整读出, 则需要设置循环, 按照 FIFO 长度读取数据。读取数据之前首先需要查询 SX2 的相应端点 FIFO 的是否为空。如果非空, 则可以进行操作。函数原型如下:

```
BOOL SX2_FifoReadSingle(channel)
```

4.2.3 芯片自举模块

USB设备在正常工作以前, 第一件要做的事就是枚举。枚举就是让HOST认得这个设备, 并且为该设备准备其所请求的资源, 建立好主机和设备之间的信息沟通机制。

USB枚举是一系列有序而又连贯的过程, 哪怕只有一丝请求或者自报家门时出错, 甚至于在规定的时间内答不上来, 设备就会被停机。枚举自然就不会成功。

USB 自举的具体步骤为:

(1) 主机检测到有设备插上, 总线复位

当设备与主机连接时, 主机就会检测到一个连接条件, 主机开始进行总线复位。

(2) 主机读取设备描述符

主机使用默认地址(地址 0)读取设备描述符, 在这里主机使用地址 0 对设备的设备描述符进行读取。当主机接收到设备描述符后, 就认为真正有设备连接了, 即发送 1 个 0 字节的数据包到设备作为状态应答。

(3) 地址分配

在地址分配阶段里, 主机分配给设备一个地址。在以后的通信里设备就只对这个地址的信息做出应答。

(4) 主机从新的地址获取设备描述符

分配好了设备地址后,主机就从新的地址获取设备描述符。最后主机发送 0 字节的数据包作为状态应答。

(5) 主机读取配置描述符

主机读取完设备描述符后就读取设备的配置描述符。

(6) 读取描述符集合

主机除了读取设备描述符和配置描述符外,还要读取接口描述符和端口描述符。在这里主机使用再次读取配置的方法来读取配置描述符、接口描述符和端点描述符的集合。

(7) 设置配置

主机读取完描述符后,就需要对设备进行配置,使得设备从地址状态进入配置状态。

(8) 读取配置状态

主机设置完配置以后,设备即可使用。主机有时会对设备的配置状态进行读取。

(9) 读取接口状态

在配置完成后,主机会对接口的状态进行读取,这和读取配置一样也是可选的。

(10) 当主机能正确地收到这些数据之后,枚举成功,就可以进行数据传输。

4.2.4 控制命令和数据的传输模块

CY7C68001 的端点主要包括控制端点 0 和传输端点 2、4、6、8,端点 0 主要用于对控制命令的传输,它是双向传输端点;端点 2、4、6、8 主要用于数据传输,其中 2、4 端点的传输方向从主机到设备,6、8 端点的传输方向是从设备到主机。

根据 USB 协议,USB 设备支持 4 种传输方式:控制传输、批量传输、中断传输、等时传输。其中控制传输发送与设备的能力和配置相关的请求和数据;批量传输结构简单,对传输时间不严格的数据传输来说比较适合,可以发送大量数据而不会阻塞总线;中断传输对于在一个规定时间内传输一个中等数量的数据是很适合的;等时传输是流、实时传输,它对数据必须以一个恒定的速率或在规定的时间内传输是适合的^[44,45]。

批量传输能保证数据传输无差错,且论文传输的数据量较小,不会出现时间延误的现象,故选定批量传输作为数据的传递模式。控制传输有错误校验,用于组态设备和给设备发送命令,因此主机软件发送给 DSP 的指令采用控制传输来完成。

4.2.5 USB 通讯模块的执行过程

USB 通讯模块主要有以下几个部分组成^[46,47]:

(1) 初始化

板卡上电后,主程序执行初始化,包括系统控制寄存器、PIE 控制寄存器、PIE 矢量表、片内外设、GPIO 和外部中断的初始化。

在之后的初始化过程中,将 CY7C68001 芯片的端点 0 字节计数器寄存器清零。如果想使能其他端点,则清空对应的 F1F0,将 FIFO 状态寄存器清零。同时将芯片寄存器中的相应的使能位置位。

接下来定义中断发生的条件。这通过设定一系列的中断屏蔽位,每个中断屏蔽位代表一个中断发生的条件。

HOST 将检测到一个新的设备已接入,将进入自举阶段。

(2) 自举阶段

在这个阶段,进入循环,等待 SETUP 信令。当一个 SETUP 信令收到时,根据前面的设定,设备将产生一个中断,设备转入中断处理程序(ISR)并处理这个接收到的 SETUP 信令。在接收到第一个 SETUP 信令后,USB 的 HOST 将发送一连串的 IN、OUT 和 SETUP 信令包来枚举设备,设备也要求进行正确的响应。

(3) 循环阶段

在成功的完成枚举过程后,设备进入主循环,负责接受主机发送过来的数据或新的指令。在这个阶段之后,根据应用的设置,它有可能进入读取数据、SETUP 中断例程。此外,CY7C68001 芯片还支持其他的各种与 USB 有关的中断。

其流程图如图 4-2 所示。

4.3 上位机的软件设计

论文实现双连杆机器人的运动控制,在界面上分别定义机器人各关节的转角及运动时间,并实现关节运动曲线的监测。

4.3.1 机器人运动的轨迹规划

论文只控制机器人关节转角“点到点”(PTP)运动的功能。只要求控制机器人关节的起点转角和终点转角,控制时只要求快速、准确地实现关节的运动,而对两点之间的轨迹不做任何规定^[48]。

为了让规划的轨迹在起点和终点处的速度和加速度都能够平滑连续,我们采用多项式插值函数对机器人的各关节进行 PTP 运动的轨迹规划^[49]。对机器人的

某个关节构造一个三次多项式如下：

$$q(t) = a_0 + a_1 * t + a_2 * t^2 + a_3 * t^3 \quad (4-1)$$

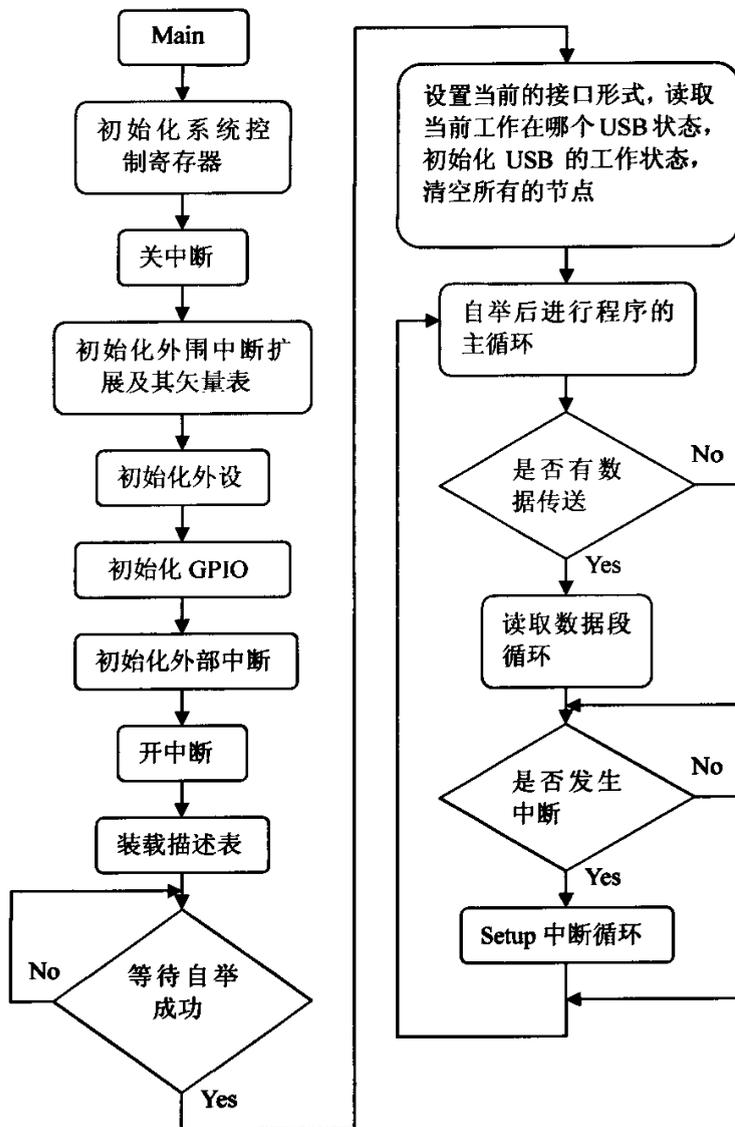


图 4-2 USB 程序开发流程图
Fig.4-2 Development flow chart of USB

设机器人关节的轨迹运行时间为 T ，运行转角为 q_T ，多项式应满足 $q(0)=0$ ； $q(T)=q_T$ ； $\dot{q}(0)=0$ ； $\dot{q}(T)=0$ ，依此可求出多项式各系数，得到机器人的关节轨迹规划。若上位机控制系统的插补周期为 T_s ，则插补步数 $N=(int)(T/T_s)+1$ 。该机器人关节在第 i 个插补段内需要转过的角度增量值 $\Delta q(i) = q((i+1) * T_s) - q(i * T_s)$ ($i=0,1, \dots, N-1$)，而第 N 次插补段内需要转过的角度增量值

$\Delta q(N) = q(T) - q((N-1) * Ts)$ 。通过比例关系将角度增量值转换为相应的脉冲增量值 Δp ，这便是上位机控制系统在每个插补周期 T_s 内要下传给下位控制系统的脉冲控制量。

根据上面的分析，可对机器人的关节轨迹规划的多项式插补进行软件编程，其程序流程如图 4-3 所示（以单关节为例）。

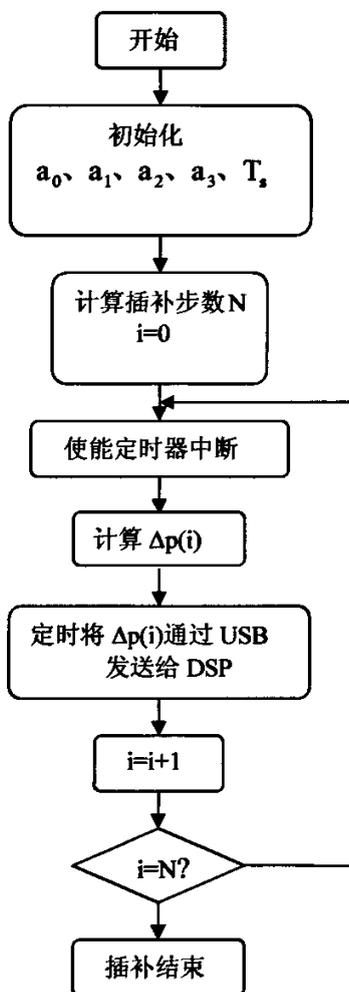


图 4-3 轨迹插补流程图
Fig.4-3 Flow chart of locus interpolation

4.3.2 客户端接口界面的设计

4.3.2.1 主机应用程序设计 USB 驱动程序的安装需要两个文件，一个为 usbfordec.inf 文件，一个为 usbfordec.sys 文件，均有厂商提供，论文不讨论驱动程序的设计。

主机应用程序通过驱动程序提供的接口函数来完成 USB 数据的读写操作。本系统中实现以下几个接口函数^[50, 51];

此外还有获得字符表描述符、配置描述符、管道描述符等函数,不再详细介绍。

表 4-1 USB 接口函数表
Table4-1 Interface function table of USB

接口函数	作用
BOOLEAN OpenDriver()	打开驱动程序, 通过调用 CreateFile ()函数, 返回一旬柄;并获得 USB 的接口信息。
void CloseDriver()	关掉驱动程序, 释放 USB 接口信息空间。
PVOID Sx2GetDeviceDesc()	通过调用 DeviceIoControl()向驱动程序发命令来获取 CY7C68001 的描述表。
BOOLEAN Sx2SendVendorReq(PVOIDmyRequest, char *buffer, int bufferSize, int *recnBytes)	通过调用 DeviceIoControl()向驱动程序发命令来向 CY7C68001 发送用户命令或访问 endpoint0。
BOOLEAN Sx2BulkdataTrans(PVOID bulkControl, char *buffer, int bufferSize, int *recnBytes)	通过调用 DeviceIoControl()向驱动程序发命令来访问 CY7C68001 的四个 F1F0 (endpoints)。

4.3.2.2 主机界面程序的设计 在采用以上函数的基础上, 应用程序就可以完成对 USB 设备数据读写的控制。课题采用 VC 十编程工具和 MFC 编程框架, 开发了基于文档和对话框的用户应用程序, 其界面如图 4-4 所示, 该程序可以完成以下功能:

(1) USB 接口状态的读取, 其中包括 USB 工作标准、读取设备描述表、读取注册表、读取 USB 管道接口信息等。

(2) 机器人运动参数的设定, 如各关节的期望转角、转动时间。

(3) 机器人关节位置的轨迹曲线绘制, 关节误差的显示。



图 4-4 实验界面
Fig4-4 experimental interface

4.4 本章小结

本章主要介绍了基于 DSP 的机器人位置控制系统的软件设计方案。将系统设计分为上位机设计、DSP 控制器设计和 USB 通讯程序设计三部分，并详细介绍了各部分的软件流程，进行了模块分析。

第 5 章 实验结果

5.1 实验研究及分析

在本章中，给出了本次实验的结果，包括关节轨迹、误差显示和主机读取的部分设备描述符。实验结果证明，本课题所用的模糊 PD 结合单神经元自适应 PID 的双模控制算法是正确的，可以满足实验的需要。论文所用的双连杆机器人的外观图片见附录 4。

图 5-1 展示了在 1s 内实现两关节转角均为 1 弧度的关节轨迹，图 5-2 给出了对应的误差显示。关节 1 转向为负，关节 2 转向为正，因为实验条件的限制，论文的第一关节为实际的机器人，第二关节实现的是仿真操作。实际操作因为受到各种干扰，导致误差比仿真操作要大，仿真操作的最大误差为 0.0044 弧度，实际操作的最大误差为 0.0036 弧度。因为实验所用的弧度值很小，所以在显示时为实际转角的 60 倍。

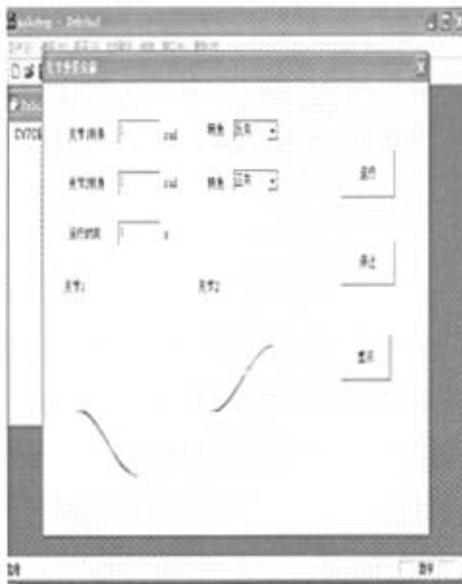


图 5-1 关节轨迹曲线
Fig5-1 the joint trajectory curve

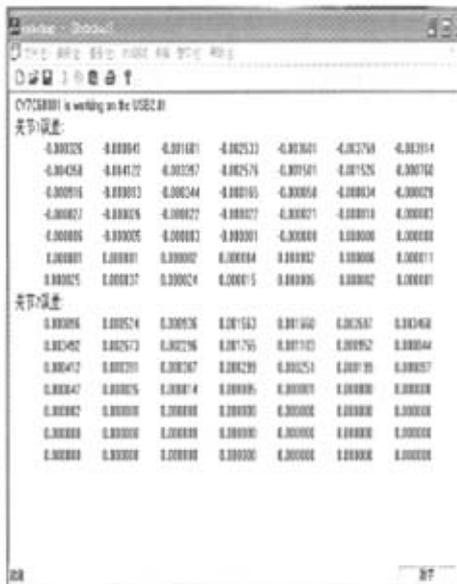


图 5-2 关节误差显示
Fig5-2 the joint trajectory show

图 5-3 给出了主机读取的部分设备描述符。

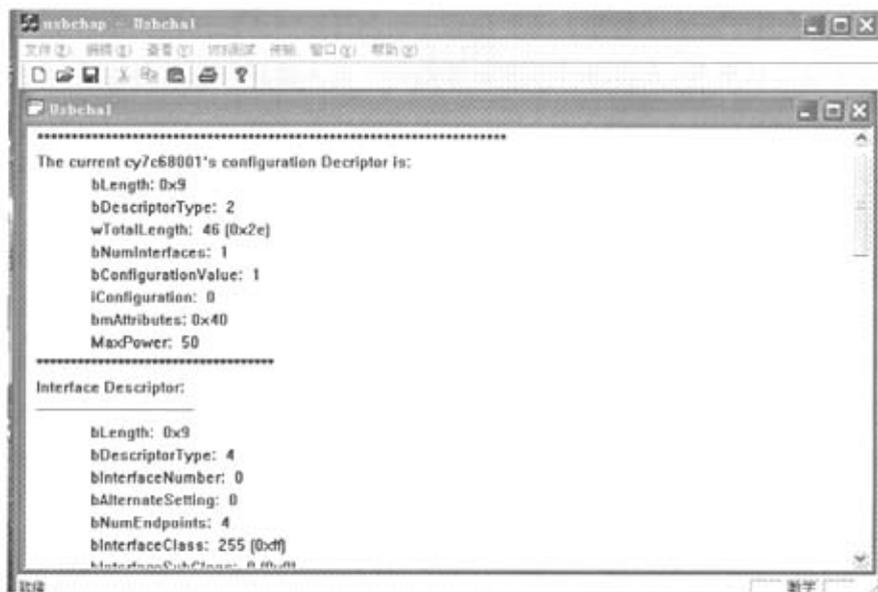


图 5-3 设备描述符
Fig5-3 Configuration Description

通过上述实验结果，可以证明论文的实验取得基本成功，正传或反转的关节最大误差都控制在 0.5% 范围内，完成了双连杆机器人控制系统的设计及调试。

5.2 本章小结

本章主要分析了基于实际对象的运动控制结果。实验结果表明，课题采用的双模控制算法很好地满足了控制系统的要求，实验取得基本成功。

结论

论文以北京工业大学大学机器人研究所的双连杆机器人作为对象,对机器人控制的有关理论问题进行了深入研究,设计了基于 DSP 控制器的机器人控制系统结构,在理论研究的基础上成功安装了双连杆机器人控制系统,为以后更深层次教学实验与学术研究提供了一个理想的实验环境。

论文完成的主要研究工作:

(1) 参考 Panasonic 交流伺服电机驱动器使用说明书,完成了双关节机器人操作系统的安装,通过驱动器控制面板控制机器人关节的转动参数,尽可能使系统运行平稳,准确。

(2) 采用多级 CPU 的控制结构设计机器人控制器

论文提出采用三级 CPU 控制的控制体系结构:第一级为 PC,第二级为 DSP 控制器,第三级为交流伺服驱动器。DSP 具有精度高、速度快的特点,可以实现对机器人系统的实时控制。

(3) 采用智能算法控制机器人关节的转动

论文提出了模糊 PD 控制算法结合单神经元自适应 PID 算法的双模控制,并且研究了模糊 CMAC 控制算法。仿真表明这两种控制策略都能很好的满足控制系统的要求,提高系统的快速性和准确性。

(4) 采用 USB 总线实现数据通信

在机器人控制系统中引入 USB 通信协议,这样能保证用户通过上位机与机器人系统进行信息交互的方便性、实时性和可靠性。

(5) 设计主机界面程序

在 VC++ 平台上设计了机器人关节运动的界面程序,可以实现 USB 接口的测试、机器人关节运动参数的输入、机器人关节运动状态的监测。

论文的主要特点及创新:

(1) 所采用的 TMS320F2812 是一种最新的数字信号处理器,主频达 150MHz,使机器人控制系统的采样周期、控制周期都大大缩短。

(2) 使用 USB 总线实现 DSP 伺服运动控制器与上位监控机的数据通信,这种通信方式既具有较高的传输速率和可靠性,能够满足对机器人进行实时控制的性能要求,又具有使用方便、支持热插拔和即插即用、便于系统扩展的特性。

(3) 机器人系统的上位控制指令周期是下位伺服控制周期的 100 倍。这样就既可以充分发挥下位 DSP 控制器的高速运算能力,提高伺服控制的精度和实时性能,又可以减轻上位计算机的运算负荷,降低对上位机系统计算性能的要求。

由于时间和诸多条件的限制,还有一些研究工作有待进一步的完善。在后续

的研究工作中，对机器人控制器的改进和研究将主要集中在以下几个方面：

(1) 可以将双连杆机器人进一步改装成多连杆机器人，将刚性机器人转换为柔性机器人，将一次运动转变为多次连续运动，加大机器人的运动范围，提高机器人系统的适用性。

(2) 论文的速度环和电流环控制是在现成的交流电机驱动器产品中实现的，在后续的工作中可以考虑使用 DSP 处理器来实现机器人关节驱动电机的全数字直接转矩伺服驱动控制。

(3) 提高机器人控制算法的鲁棒性和解耦性，进一步提高系统的稳定性和精确性。

参考文献

- 1 张福学. 机器人技术及其应用. 电子工业出版社, 1995:1~5
- 2 毛春利. 基于 DSP 和 FPGA 的四关节机器人控制器的研制. 中南大学硕士学位论文. 2005:1~5
- 3 蔡自兴. 机器人技术及其应用. 中南工业大学出版社, 1988:10~16
- 4 管家春. 足球机器人系统运动学动力学分析. 西华大学硕士学位论文. 2006:8~15
- 5 房海蓉等. 基于神经网络的机器人智能控制. 机器人技术与应用. 2002,4:28~31
- 6 陈全福. 智能移动机器人平台控制系统设计. 哈尔滨工程大学硕士学位论文. 2006:15~22
- 7 Luh. J. Y. S. Scheduling of Parallel Computation for a Computer Controlled Mechanical Manipulator. IEEE Trans. Syst. Man & Cyber. 1982, SMC-12(2)
- 8 霍伟. 机器人动力学与控制. 高等教育出版社, 2005:23~31
- 9 杨东超, 刘莉, 汪劲松, 陈晟. 利用奇次变换矩阵进行运动学正问题分析. 机械设计与研究. 2004,18(2):20~22
- 10 Wen Chen. Dynamic modeling of multi-link flexible robotic manipulators. Computers and Structures. 2001,79:183~195
- 11 王文琰. 柔性机械臂的建模与智能控制. 兰州理工大学硕士学位论文. 2005:7~11
- 12 雷勇, 马伟富, 涂国强. 基于 DSP 的微动机器人位置控制的实现. 电子技术应用. 2005,1:62~64
- 13 郑德兵. DSP 在开放式数控系统中的应用研究. 大连理工大学硕士学位论文. 2005:19~23
- 14 B. K. Bose. Motion Control Technology-Present and Future[J]. IEEE. Trans. Industry Applications. 1985,21:1337~1342
- 15 李长兵. 基于 DSP 控制技术的交流伺服驱动器研究. 广东工业大学工学硕士学位论文. 2005:1~5
- 16 Texas Instrument. TMS320F28x DSP System Control and Interrupt Reference Guide[M], Texas Instrument, Jul.2003
- 17 Smith S W. The Scientist and Engineer's Guide to Digital Signal Processing Second Edition[M]. California Technical Publishing. 1999
- 18 林正. 基于 TMS320F2812A 的永磁同步电机伺服系统设计. 浙江工业大学硕士学位论文. 2005:27~33
- 19 Texas Instrucments. TMS320F28x DSP Event Manager (EV) Reference Guide. Texas Instrucments, USA, 2003.
- 20 Texas Instrucments. DAC7724/DAC7725 Data Sheet. Texas Instrucments, USA, 2003.
- 21 Texas Instrucments. CY7C68001 Data Sheet. Texas Instrucments, USA, 2003.
- 22 苏奎峰. TMS320F2812 原理与开发. 电子工业出版社. 2005:355~369
- 23 Shigeo Morimoto. Wide-Speed Operation of Interior Permanent Magnet Synchronous Motors with High-Performance Current Regulator. IEEE transaction on industry applications. 2000,30(4):920~926
- 24 吴刚. 车辆驾驶机械手的研制与伺服运动控制研究. 吉林大学硕士学位论文. 2005:28~39
- 25 Wang Zhiliang; Xie Lun, Li Chongjian. Application of intelligent control theory in AC servo system. Journal of university of Science and Technology, Beijing, 1997
- 26 Liu,GP, Daley.S. Optimal-tuning PID control for Industrial system. Control Engineering

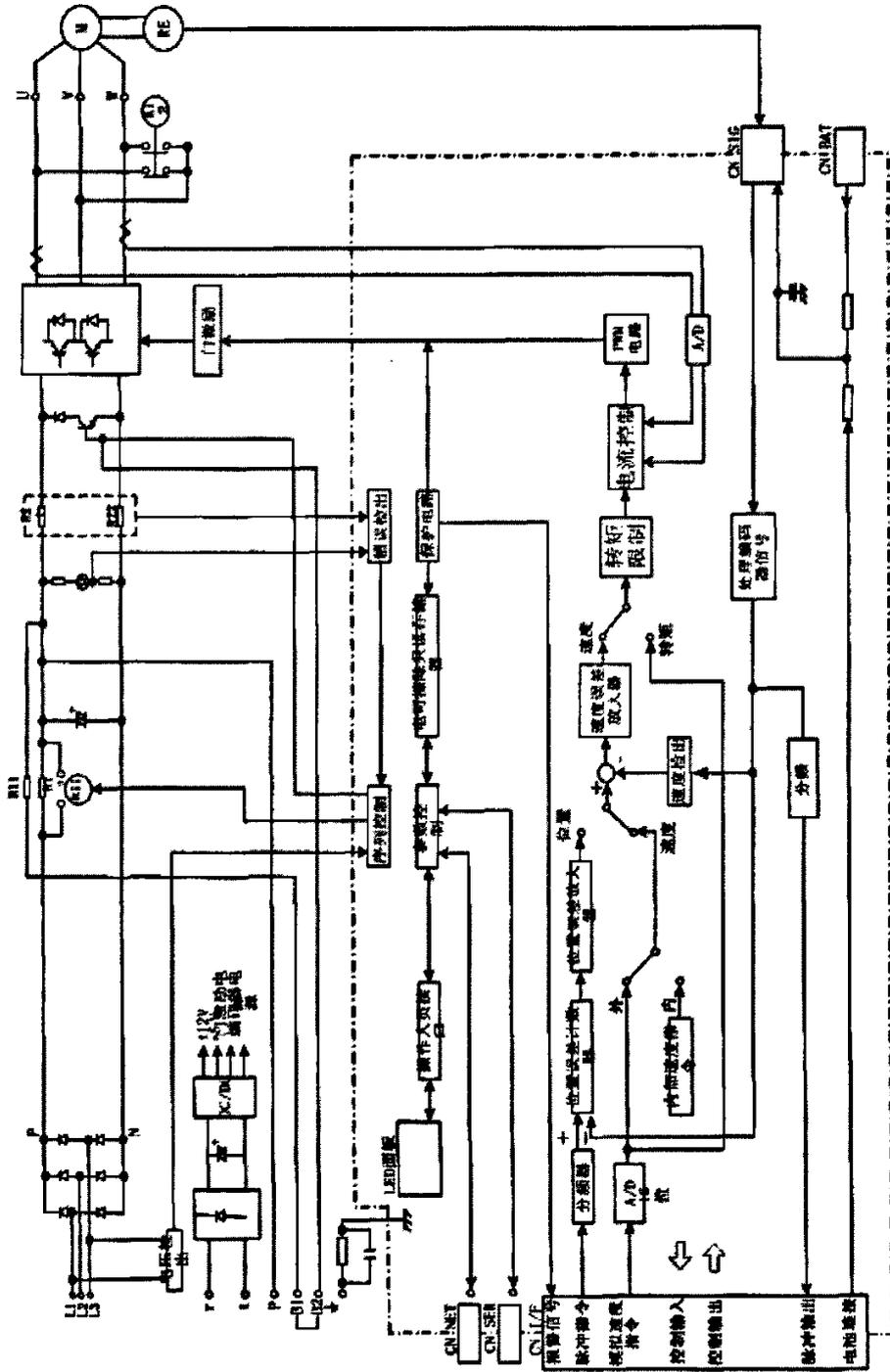
- Practice , 2001
- 27 S. Victor, K. Rafael. PD Control with Feedforward Compensation for Robot Manipulators: Analysis and Experimentation. *Robotica*. 2001,19:11~19
 - 28 J. M. G Cooper and J. J. Vidal. Genetic Design of Fuzzy Controller: the Cart and Jointed-pole Problem. *IEEE-Fuzz*. 1999:1332~1339
 - 29 Tang K.Z, Huang S.N, Tan K.K, Lee T.H. Combined PID and adaptive nonlinear control for servo mechanical systems *Macaronis*. 2004,14(6)
 - 30 陈贺. 仿人机器人控制系统的研究与开发. 河北工业大学硕士学位论文. 2005:34~46
 - 31 侯媛彬. 智能控制技术. 2. 北京工业大学出版社, 2001:177~200
 - 32 Yi Jikai, Wang Lin. The Application of Fuzzy Neural Networks to the Temperature Control System of Oil-Burning Tunnel Kiln. *Proceeding of the International Conference on Intelligent Processing*. 1997:512~516
 - 33 李海涛. 基于 DSP 的交流伺服位置控制系统的研究. 山东大学硕士学位论文. 2005:15~19
 - 34 G. Pristchow,. A Comparison of Linear Conventional Electromechanical Drives. *Annals of the CIRP*. 1998,47(2)
 - 35 Lin Chun-Shin, Chiang Ching-Tsan. Learning Convergence of CMAC Technique. *IEEE Transactions on Neural Networks*. 1997,8(6):1281~1292
 - 36 廖永忠. 基于 DSP 运动控制卡及控制算法的设计与研究. 湖南师范大学硕士学位论文. 2006:48~58
 - 37 Parks P C. A Comparison of Five Algorithms for the Training of CMAC Memories for Learning Control Systems. *Automatic*. 1992,28(5):1027~1035.
 - 38 王海涛. 几类不确定机器人模糊控制策略的研究. 燕山大学工学硕士学位论文. 2003:23~27
 - 39 Albus J S. Data storage in the Cerebella Model Articulation Controller(CMAC). *ASME. J. Dynam. Syst. Meas. Control*. 1975,97(3):228~233
 - 40 Cai,X.Z, Shin. K.G. Direct Control and Coordination Using Neural Networks. *IEEE Trans. On System, Man. And Cybernetics*. 1993,23(3):686~697
 - 41 蒋文萍. 基于模糊 CMAC 的 PMSM 位置伺服系统的分析和研究. 天津大学硕士学位论文. 2004:24~31
 - 42 周志红. 四关节伺服控制系统的研制. 中南大学硕士学位论文. 2004:11~17
 - 43 Texas Instrucments. TMS320F28x Control and Interrupts Reference Guide. Texas Instrucments, USA. 2003.
 - 44 李涵. USB 接口驱动的研究与设计. 山东科技大学硕士学位论文. 2005:5~14
 - 45 ZAO zhike. Research on application of USB interface in SCM communication. *Electric Drive Automation*. 2003,25(4):44~46
 - 46 廖良斌. 基于 DSP 和 USB 的图像采集系统的研究. 武汉理工大学硕士学位论文. 2005:37~45
 - 47 付云峰. 基于 DSP 的虹膜识别系统研究. 哈尔滨工程大学工学硕士论文. 2005:52~66
 - 48 Schachter S, Sahni S. Fast algorithm for polygon decor position. *IEEE Transactions on Computer-Aided Design*. 1991,7(4):473~483
 - 49 Eman, K F Soo-Hun Lee, J C Cesarone. "Trajectories" *International Encyclopedia of Robotics:Applications and Automation [J]*. New York, 1988:1796~1810
 - 50 Walter Olney. *Programming Windows Driver Model*. Microsoft Corporation Press. 1999:1~30
 - 51 张辉. 基于 DSP 的 USB 通道语音数据采集系统的实现方法. 广东工业大学工学硕士学位

论文. 2005:47~60

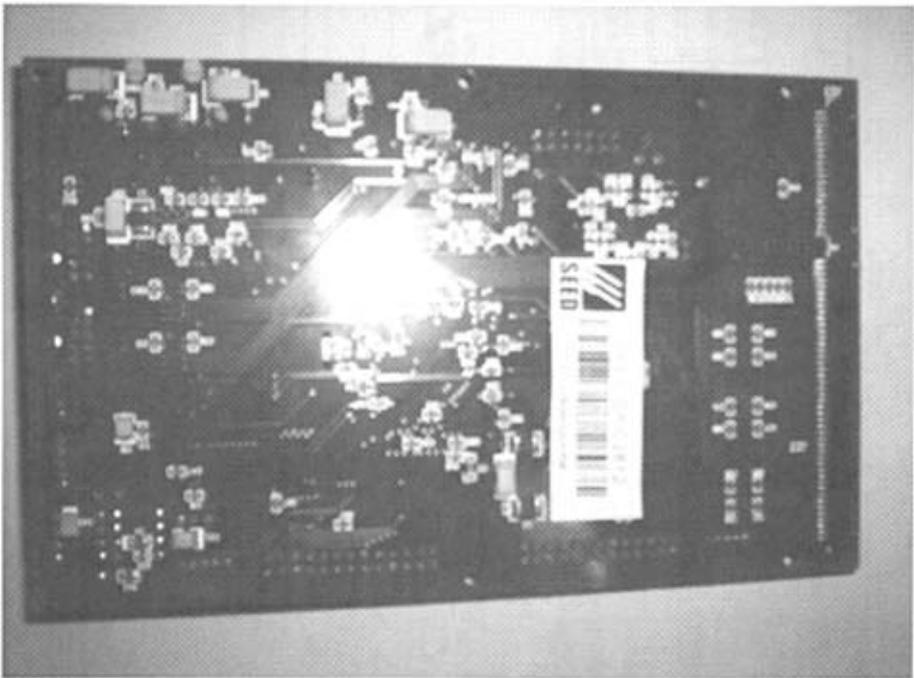
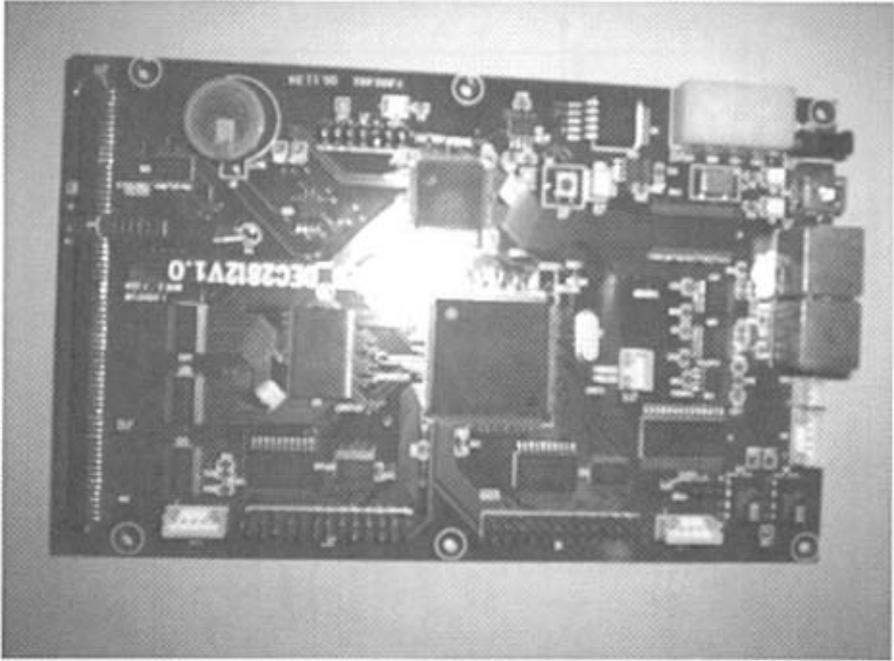
攻读硕士学位期间发表的学术论文

- 1 吕雁, 陈文楷, 唐润宏. 基于 F2812 DSP 的机器人位置控制系统研究. 仪器仪表用户杂志. 2007, (2):14~16 (已录用)
- 2 吕雁, 陈文楷, 唐润宏. 基于模糊 CMAC 的三连杆机械臂的最优控制器. 现代电子技术. 2007 (已录用)

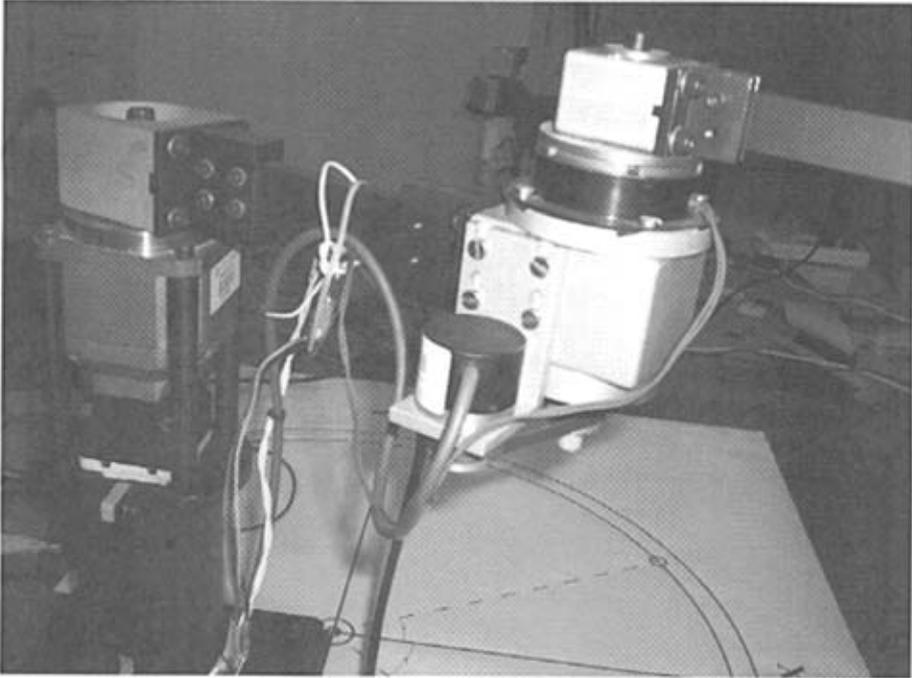
附录 1 驱动器结构图



附录 3 TMS320F2812 外观图



附录4 双连杆机器人



致谢

首先，我要衷心地向我的导师陈文楷老师表示感谢！陈老师以其严谨的治学态度和深厚的学术造诣，在我的选题和课题研究过程中给予了悉心的指导和帮助，使我的课题能够得以顺利完成。

同时，我也要感谢在本课题研究中曾给与我指导和帮助的高风老师、王广生老师、于乃功老师、崔刚老师、张会清老师，王老师和高老师对我的论文进行了认真审阅，并提出了许多宝贵的意见和建议。感谢电子技术研究室的崔刚老师。

论文的完成也离不开实验室的同学们的帮助和指点，他们都曾在学习和生活上给予我许多关怀和帮助。

另外，我的父母和家人支持并期待我完成学业，并为此承担了很多的生活压力。在论文顺利完成之际，我愿与他们分享其中的快乐，并感谢亲人们对我的殷切期望和无私无悔的支持。

谢谢所有关心和帮助过我的师长、同学、朋友和亲人们！谢谢你们！

最后，对评审论文的专家、学者表示衷心的感谢！