

基于 Nios II 的直流电机控制系统设计

桂春胜,陈万米,刘军江,王赛赛,范海艇
(上海大学上海市电站自动化技术重点实验室,上海 200072)

摘要:直流电机的控制一般采用单片机或 DSP 作为核心控制器,集成度较低并且稳定性较差。文中介绍了一种基于 Nios II 软核处理器的直流电机控制系统。利用 SOPC 平台设计并封装了 PWM 模块,在 FPGA 内嵌入 Nios II 软核作为微处理器,借助光电编码器实现电机测速反馈。同时,采用数字 PID 算法实现对直流电机的 PWM 闭环调速控制,并最后对数字 PID 算法进行改进。实验表明,改进后控制系统的动态性能得到了很大的改善,而且整个系统的设计简单,集成度高,系统稳定,同时也在实际的应用中得到了验证。

关键词:Nios II 软核;脉冲宽度调制;现场可编程门阵列;数字 PID

中图分类号:TP273

文献标识码:A

文章编号:1673-629X(2011)08-0157-04

Designing of DC Motor Control System Based on Nios II

GUI Chun-sheng, CHEN Wan-mi, LIU Jun-jiang, WANG Sai-sai, FAN Hai-ting
(Shanghai Key Laboratory of Power Station Automation Technology,
Shanghai University, Shanghai 200072, China)

Abstract: DC motor control generally uses the microcontroller or DSP as the core controller, which is less integrated and less stable. A DC motor control system based on soft core Nios II is proposed in this paper. The PWM module is designed and packaged on SOPC; The soft core Nios II is embed into the FPGA so it can work as a microprocessor; Photoelectric encoders are used to measure motor speeds. At the same time, the digital PID algorithm is used to achieve the PWM closed-loop DC motor speed control, which is improved finally. Experiments show that improved control system dynamic performance has been greatly improved, and the whole system design is simple, highly integrated, system stability, but also in practical application has been verified.

Key words: Nios II soft core; PWM; FPGA; digital PID

0 引言

直流电机是一种常用的机电能量转换元件,目前在直流电机的控制系统中,国内外大多采用单片机或 DSP 作为微控制器^[1],但是单片机或 DSP 控制电机时需要很多外围电路,占用较多端口资源,对系统的稳定也有很大的影响。

文中利用 Altera 公司的型号为 EP2C20 的 FPGA^[2] (现场可编程门阵列)芯片作为系统控制器,并内嵌 32 位 Nios II 嵌入式软核处理器^[3],用硬件描述语言设计 PWM (脉冲宽度调制)模块,然后在 SOPC (可编程片上系统)平台上将其封装成 IP 核,最后采用数字 PID 算法对直流电机进行闭环控制,能够很好解决用单片机或 DSP 作微控制器带来的缺陷。

1 PWM 模块设计

SOPC Builder 平台提供了一个组件编辑器,使用者可以创建和编辑一个自己的 SOPC Builder 组件。定制的组件可以像其他组件一样共享并重用。一个典型的组件一般由下面三部分组成:

(1) 硬件文件:描述组元逻辑的硬件描述文件;

(2) 软件文件:一个 C 语言头文件,定义了组件寄存器映像,及软件控制组件所需要的驱动程序;

(3) 组件描述文件(class. ptf):定义了组件的结构,为编辑器提供将组件集成到系统所需的信息,此文件基于所提供的软件、硬件以及编辑器 GUI 中指名的参数而自动创建。

1.1 PWM 部分的硬件逻辑设计

PWM 的硬件逻辑有三个部分组成,下面分别介绍。

(1) 任务逻辑。PWM 任务逻辑由一个输入时钟 (CLK)、一个输出信号 (PWM_OUTPUT)、一个使能寄存器、一个 32 位计数器和一个 32 位的比较电路组成,

收稿日期:2011-01-08;修回日期:2011-04-16

基金项目:机器人技术与系统国家重点实验室开放基金(SKLS-2009-MS-10)

作者简介:桂春胜(1984-),男,硕士研究生,研究方向为智能机器人与嵌入式系统方面的研究。

结构如图 1 所示。CLK 用来作为 32 位计数器的时钟信号,用来建立输出信号的周期。比较电路比较计数器当前值和占空比值,用来决定 PWM_OUTPUT 信号的输出。如果计数器当前值小于等于占空比值,PWM_OUTPUT 信号驱动逻辑 0;反之,驱动逻辑 1。PWM_OUTPUT 为该接口元件对外的输出接口信号^[4]。

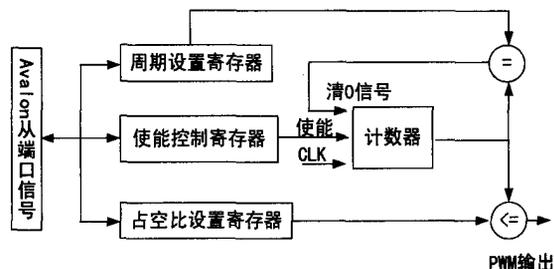


图 1 PWM 任务逻辑图

(2) 寄存器文件。寄存器文件提供对使能位、计数数值和占空比值的访问,设计将寄存器映像到 Avalon 从接口地址空间唯一的偏移地址^[5]。对寄存器可以进行读写访问,也就是说软件能读回前面写入的值,这也是任选的功能,目的是用硬件资源换取软件设计的方便性。也可以将寄存器设为只写,这可以节省逻辑资源,但不能用软件来读寄存器^[6]。

(3) Avalon 从接口。Avalon 从接口实现 PWM 组件与 Avalon 总线的通信,并且本身是可以进行读写操作的,对于读和写,没有任何建立和保持时间限制,不需要延迟,有零等待状态,使用本地的地址对齐原则。

用 Verilog 硬件描述语言来实现的硬件部分可由三个文件组成,分别为 pwm_register_file.v、pwm_task_logic.v 和 pwm_avalon_interface.v,其功能分别是寄存器文件、硬件的任务逻辑和 Avalon 总线设计接口。最后将文件封装成 IP 核,加到 SOPC 组件中。

1.2 PWM 组件的软件实现

PWM 组件的软件可以使用 Altera 提供的 API 接口函数,接口函数是在文件 altera_avalon_pwm_routines.h 中定义,在该文件中还提供了返回代码的定义。而在 altera_avalon_pwm_routines.c 文件中提供了 API 接口函数的具体实现源代码(该源代码主要是一些寄存器控制和运算)^[7]。在 PWM 外设的整个设计中,硬件设计是最底层,其上是寄存器的映射,再上是向用户提供的 API(应用程序接口),使用时直接调用 API 函数完成应用程序的设计。API 函数及其功能分别为:

- (1) 初始化 PWM 硬件函数: altera_avalon_pwm_init();
- (2) 使能 PWM 输出函数: altera_avalon_pwm_ena-

ble();

(3) 关闭 PWM 输出函数: altera_avalon_pwm_disable();

(4) 改变 PWM 占空比函数: altera_avalon_pwm_change_duty_cycle()。

1.3 PWM 死区保护

文中电机的驱动电路用的是 H 桥全桥电路,用 IR2110 作为控制芯片^[8],PWM 信号被用来控制全桥电路中四个 MOSFET 的开关。全桥驱动电路中,上桥和下桥是由电源直接串联两个 MOSFET 接地的,如果上下桥同时打开,则会出现短路(直通)现象。MOSFET 完全开启和关断均存在一定死区时间,且完全开启与关断的时间延时是不一样的,这样便会造成上桥尚未关断时下桥已经打开的状况,导致上下桥直通,会影响系统的稳定性。在 PWM 信号输出之前加入适当延时,可防止直通现象,起到 MOSFET 死区保护的作用。

2 电机测速

2.1 电机测速原理

要实现对直流电机地精确实时的控制,就需要知道电机速度反馈信息。一般电机自己带的增量式旋转光电编码器能够产生正交编码脉冲信号,利用产生的这两路脉冲信号来检测电机的实时转速。

在码盘上均匀地刻制一定数量的光栅,当电机旋转的时候,码盘随之一起转动,通过光栅的作用,次序不断地开放或封闭光通路,在接收装置的输出端便可以得到频率与转速成正比的方波脉冲序列,从而可以计算转速。为了获取转速的方向,增加了一对发光与接收装置,使两队发光与接收装置错开光栅节距的 1/4,则两组脉冲序列 A 和 B 的相位差 90 度。正转时 A 相超前 B 相;反转时 B 相超前 A 相^[9]。

常用的旋转编码器的数字测速方法有三种,M 法、T 法和 M/T 法。M 法适用于电机高速旋转的时候;T 法适用于电机低速旋转的时候;M/T 法(见图 2)将两者结合起来,在电机高速和低速旋转的时候都比较适用。

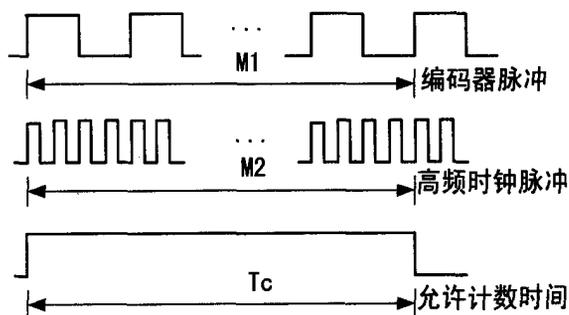


图 2 M/T 测速法

文中采用 M/T 法测速,它将前两种结合起来,在高速和低速时误差都比较小。在 T_c 时间内编码器输出的脉冲个数为 M_1 ,同一时间间隔里面高频时钟的脉冲个数为 M_2 ,高频时钟的频率为 f ,电机转一周输出的脉冲个数为 Z (Z 为倍频系数乘以码盘光栅分辨率),则准确的测速时间为 $T_i = M_2/f$,电机转速 n 以 r/min 为单位,则:

$$n = \frac{60M_1}{ZT_i} = \frac{60M_1f}{ZM_2} \quad (1)$$

2.2 测速的分辨率分析

分辨率用来衡量系统对电机转速变化的分辨能力,一般用改变一个计数字所对应的转速变化量来表示。高速时,测速的分辨率为:

$$Q = \frac{60(M_1 + 1)}{ZT_c} - \frac{60M_1}{ZT_c} = \frac{60}{ZT_c} \quad (2)$$

由上式可见,测速分辨率与实际转速大小无关,提高分辨率唯一的办法就是增大 T_c 或码盘光栅分辨率 Z 。

低速时,测速的分辨率为:

$$Q = \frac{60f}{Z(M_2 - 1)} - \frac{60f}{ZM_2} = \frac{60f}{ZM_2(M_2 - 1)} = \frac{Zn^2}{60f - Zn} \quad (3)$$

上式说明,当电机低速时,其分辨率与转速的值相关,当转速越低, Q 值越小,分辨率越高。

3 数字 PID 控制器设计

3.1 数字 PID 算法

PID 控制是较早发展起来的控制策略之一,因其算法简明,参数物理意义明确,理论分析体系完整且对于大多数过程都具有良好的控制效果和鲁棒性,所以至今仍被广泛应用并得到改进^[10]。数字 PID 控制算法分为位置式数字 PID 算法和增量式数字 PID 算法,文中采用的是增量式数字 PID 算法^[11]。与位置式 PID 控制算法相比,增量式 PID 控制算法不易产生较大的累积偏差,误动作时影响小,不会产生积分失控现象,而且易于实现无扰切换。

已知理想的模拟 PID 控制器传递函数为

$$D(s) = \frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i}s + T_d s \right) \quad (4)$$

式(4)中, $U(s)$ 为控制器输出信号的拉氏变换式, $E(s)$ 为其输入信号的拉氏变换式, K_p 为比例系数, T_i 为积分时间常数, T_d 为微分时间常数。

将上式离散化,整理后可以得到:

$$\Delta u(k) = u(k) - u(k-1) = K_p [e(k) - e(k-1)] + K_i e(k) + K_d [e(k) - 2e(k-1) + e(k-2)] \quad (5)$$

K_p, K_i, K_d 分别为比例系数、积分系数、微分系数; k

为采样序号, $k = 1, 2, 3 \dots$; T 为采样周期; $u(k)$ 为 k 次采样时刻控制输出; $e(k)$ 为 k 次采样时刻的偏差值。

可将式(5)进一步可以写成为

$$\Delta u(k) = Ae(k) - Be(k-1) - Ce(k-2) \quad (6)$$

$$\text{式中 } A = K_p \left(1 + \frac{T}{T_i} + \frac{T_d}{T} \right)$$

$$B = K_p \left(1 + \frac{2T_d}{T} \right)$$

$$C = \frac{K_p T_d}{T}$$

则数字 PID 控制的最终输出为:

$$u(k) = u(k-1) + \Delta u(k) \quad (7)$$

3.2 数字 PID 算法软件实现

PID 控制系统结构图见图 3。

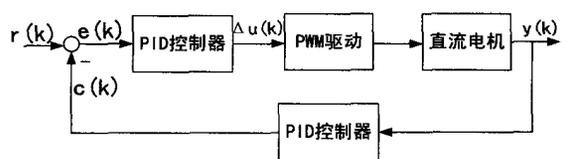


图 3 PID 控制系统结构图

Nios II 软核处理器设定定时器,每 2ms 采集一次直流电机编码器反馈的数据,得到电机的实际转速 $c(k)$ 。同时,Nios II 软核处理器读取控制系统给定的转速 $r(k)$,以及记录上次给定的转速 $r(k-1)$ 和 $r(k-2)$,通过计算可以得到 $e(k)$ 、 $e(k-1)$ 和 $e(k-2)$,再计算 PID 控制算法,输出 $\Delta u(k)$,即可得到实际控制值。算法程序流程图如图 4 所示。在 PID 算法源代码的编写时,充分注意到了算法实现的灵活性和适应性,使算法能很轻易得到修正,根据实验和经验在线整定

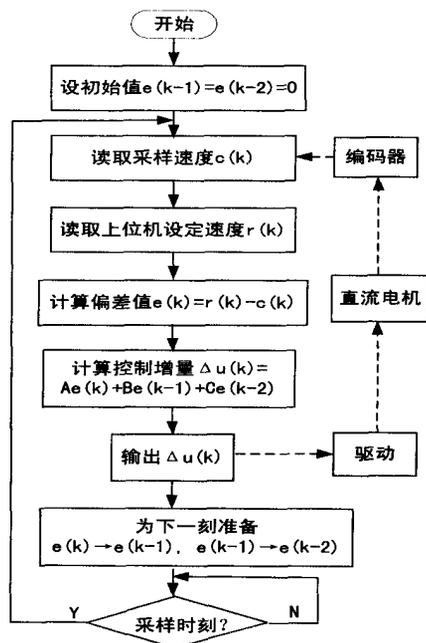


图 4 PID 算法程序框图

参数。

3.3 数字 PID 控制算法改进

(1) 抗积分饱和 PID 控制算法。

由于直流电机的启动、急停或速度突变等会引起控制偏差的突增,使 PID 输出急剧变化,甚至远远超过电机的最大转速,但由于控制量的限制,会导致偏差的增加,使积分项在运算中有较大的积累,这样要经过一段时间才能脱离饱和区,此时系统的输出已经明显地出现较大超调。随后,反方向的偏差又很大,又在反方向上重复上述现象,从而使系统产生振荡,使得电机则总是处在某种极限转速的状态。

解决上述的积分饱和的方法之一就是 PID 控制器输出的控制量加以限幅,设上限为 U_{max} , 下线为 U_{min} , 分别为直流电机限定的最大和最小速度。按 PID 算法计算出控制量 $u(k)$ 后,执行如下操作:

当 $u(k) > U_{max}$ 时,取 $u(k) = U_{max}$, 并取消积分作用;

当 $u(k) < U_{min}$ 时,取 $u(k) = U_{min}$, 并取消积分作用;

当 $U_{min} \leq u(k) \leq U_{max}$ 时,执行积分运算。

(2) 不完全微分 PID 控制算法。

微分环节改善了整个系统的动态性能,但对于干扰特别的敏感。微分项在电机的运动控制系统中,输出仅在第一个周期起激励作用,对于时间常数较大的系统,其调节作用很小,不能达到超前控制误差的目的^[12]。驱动电机有一定的时间延迟,如果不能及时达到要求的速度,可能会使输出失真。为克服上述的缺点,在 PID 算法中加一个一阶惯性环节,形成一个低通滤波器($G(s) = 1/[1 + T_f(s)]$),如图 5 所示,构成不完全微分 PID 控制。

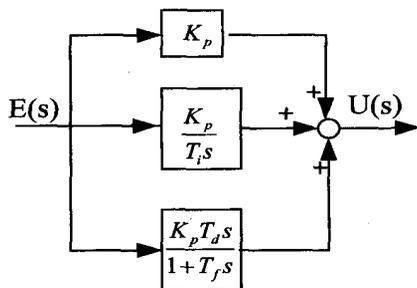


图 5 不完全微分 PID 控制算法结构图

图 5 所示的不完全微分 PID 结构,传递函数为:

$$U(s) = (K_p + \frac{K_p}{T_i s} + \frac{K_p T_d s}{1 + T_f s}) E(s) \quad (8)$$

将上式离散化最后可整理为:

$$u(k) = \frac{T_f}{T_f + T} u(k-1) + \frac{K_p T_d}{T_f + T} [e(k) - e(k-1)] \quad (9)$$

当单位阶跃输入时,完全微分的输出在第一个采样周期内的脉冲高度下降,不能实现连续的控制,容易引起系统的振荡。加入不完全微分环节后,输出能够有规律的衰减,系统输出缓慢变化,控制效果比较理想。

(3) 算法改进后的实验比较。

通过观察电机在带负载情况下的转速/时间图可以直观比较出 PID 算法改进后的效果。如图 6 所示,所标注的为经过算法改进后驱动电机的输出响应曲线。可以看出,无论在快速性,超调的控制还是稳态误差都比先前有很好的改善。

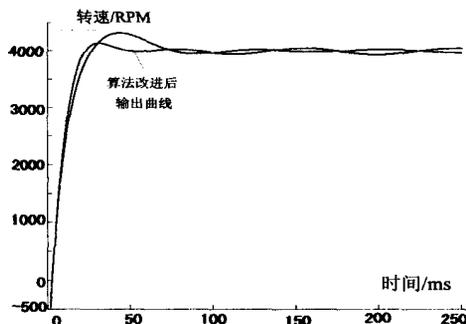


图 6 算法改进前后的电机响应特性差异图

4 结束语

文中设计了一种利用 FPGA 内嵌 Nios II 软核的方式来实现对直流电机的调速,这种新型的控制方式充分利用了 FPGA 的优势,使系统的集成度更高更稳定。采用了数字 PID 算法并对算法进行了改进,使得直流电机的运行效果更好。实际上,本系统已经用在了上海大学自强队的足球机器人上,其性能在参加国内外的比赛中得到了验证。

参考文献:

- [1] 姚文刚,余国强,孟小锁.基于多 DSP 架构的电机控制系统[J].计算机技术与发展,2006,16(6):48-50.
- [2] 王本有,苏守宝,汪德如.一种基于 FPGA 的 CPU 设计[J].计算机技术与发展,2008,18(6):221-224.
- [3] Altera Corporation. Nios II Software Developer's Handbook[EB/OL]. 2004. <http://www.altera.com>.
- [4] 冯志华,高社生.基于 NiosII 软核处理器的 PWM 控制器设计[J].微处理机,2008,2(1):54-56.
- [5] 李英兰.Nios II 嵌入式软核 SOPC 设计原理及应用[M].北京:北京航空航天大学出版社,2006.
- [6] Altera Corporation. Developing Peripherals for SOPC Builder [DB/OL]. 2004. <http://www.altera.com>.
- [7] 杨小明.Nios 系统中 Avalon 从外设(PWM)的设计和研究[D].昆明:昆明理工大学,2007.
- [8] 陈万米,张冰,朱明,等.智能足球机器人系统[M].北

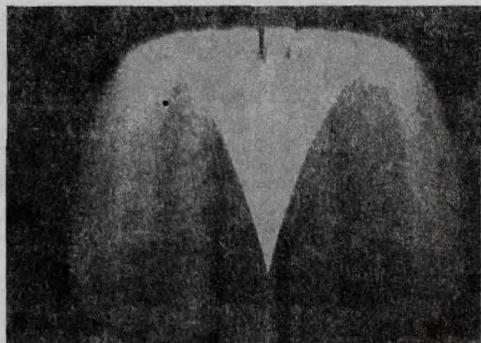


图 3 文中模拟喷泉的显示效果

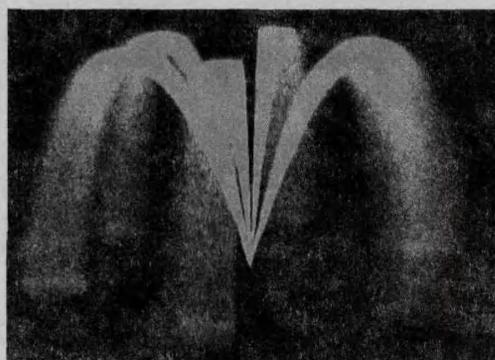


图 4 喷泉股数为六股的显示效果

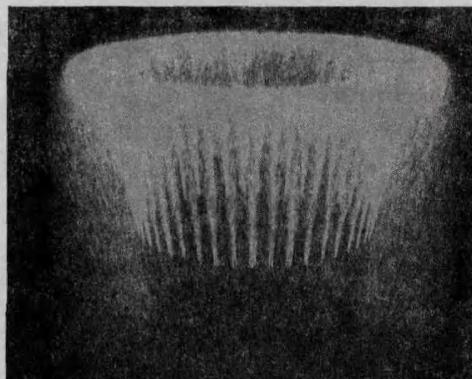


图 5 初始位置不在原点的显示效果

4 结束语

文中将纹理映射技术与粒子系统理论相结合,通过改变喷射角度、喷泉股数和矢量乘积等影响喷泉的真实感因素,从而对喷泉进行实时模拟。实验结果表明,该方法在实时性和逼真性方面取得了较好的效果。不过,文中只是从实时性、逼真性、简单性程度实现的一种模拟,由于不规则物体的表面很难把握,所以在喷

泉水珠粒子和地面接触处很难进行模拟,同时水粒子的真实感光照很难反应。通过改变喷泉股数、喷泉喷射角度以及喷泉的初始位置等因素可以观察到不同的喷泉的效果。在今后的工作中,将水粒子的真实感光照模型作为研究的重点,以及喷泉水滴落回水面与障碍物碰撞的效果,同时加入风力、阻力等因素,实现更加逼真的模拟效果。

参考文献:

- [1] Reeves W T, Lucasfilm. Particle Systems—A Technique for Modeling a Class of Fuzzy Objects[J]. ACM Transactions on Graphics, 1983, 2(2): 213-217.
- [2] Reeves W T. Particle System: A Technique for Modeling a Class of Flussy Object [C]//SIGGRAPH' 83. [s. l.]: [s. n.], 1986: 359-376.
- [3] 马骏,朱衡君. 基于动态纹理和粒子系统的喷泉模拟[J]. 北方交通大学学报, 2005, 2(1): 90-94.
- [4] 和平鸽工作室. OpenGL 高级编程与可视化系统开发[M]. 北京: 中国水利水电出版社, 2003.
- [5] 杨春雨. 基于粒子系统的喷泉动画模拟[D]. 长春: 吉林大学, 2008.
- [6] Witkin A. Physically Based Modeling Particle System Dynamics[EB/OL]. 2001. http://74.125.155.132/scholar?q=cache:sFtVaEvBw&J:scholar.google.com/+An+Introduction+to+Physically+Based+Modeling:Particle+System+Dynamics&hl=zh-CN&as_sdt=0.
- [7] 蒋恒恒,汤宝平,章国稳. 基于 OpenGL 与粒子系统实现三维喷泉模拟[J]. 计算机测量与控制, 2009, 17(9): 1717-1723.
- [8] 魏开平,朱晓华,沈显君,等. 基于纹理映射和粒子系统的三维喷泉实时模拟[J]. 计算机工程与设计, 2007, 28(11): 2586-2588.
- [9] 卫丽芬,李仰军. 基于粒子系统的喷泉模拟实现[J]. 电子测试, 2010, 2(2): 23-26.
- [10] 肖何,何明耕,白忠建. OpenGL 中基于粒子系统的喷泉模拟实现[J]. 计算机仿真, 2007, 24(12): 201-204.
- [11] 丁纪云,陈利平,李思昆. 基于 OpenGL 的烟花动态模拟方法的研究与实现[J]. 计算机工程, 2002, 28(4): 240-241.
- [12] 侯阳,迪克. 三维图形动画编程实例[M]. 北京: 海洋出版社, 1993.

(上接第 160 页)

京: 清华大学出版社, 2009.

- [9] 陈伯时. 电力拖动自动控制系统[M]. 第 3 版. 北京: 机械工业出版社, 2007.
- [10] 张赵良. 基于 NIOSII 软核的移动机器人控制器的研究与实现[D]. 兰州: 兰州理工大学, 2009.

- [11] Zhang Wanfeng. DSP implementation of predictive control strategy for power factor correction[J]. Apecc, 2004(2): 67-73.
- [12] 陶永华. 新型 PID 控制及其应用[M]. 第 2 版. 北京: 机械工业出版社, 2002.