

[自控·检测]

DOI:10.3969/j.issn.1005-2895.2012.03.017

# 基于 AT91SAM9261 的步进电机 S 曲线加减速控制研究与实现

简毅, 刘宝

(重庆大学机械工程学院, 重庆 400030)

**摘要:**步进电机按照 S 型曲线加减速时不存在柔性冲击, 适合于高速运动。采用了嵌入式 ARM 片上系统 AT91SAM9261 作为控制平台, 研究了其脉冲产生机制, 对 S 曲线加减速的算法及其曲线方程作了分析推导并研究了将曲线转化为脉冲的离散实现过程; 然后列举了实际运动中可能出现的加减速情况并针对各情况分析了实时运动控制过程。最后通过脉冲测试证明, 所提出的加减速算法及运动控制方法可行有效, 能够适应各种不同运动参数, 提高了步进电机的效率和稳定性。图 7 表 1 参 11

**关键词:**步进电机; 运动控制; S 曲线加减速; 嵌入式; AT91SAM9261 系统

中图分类号: TP273.5; TM383.6

文献标志码: A

文章编号: 1005-2895(2012)03-0062-05

## Research and Implementation of S Curve Acceleration and Deceleration Control for Step Motor Based on AT91SAM9261

JIAN Yi, LIU Bao

(School of Mechanical Engineering, Chongqing University, Chongqing 400030, China)

**Abstract:** When the step motor accelerates or decelerates to S curve, there is no soft impact, which is good for high-speed motion. The embedded ARM soc AT91SAM9261 was adopted as the control platform to study the pulse generating mechanism. The algorithm and equation of S curve was analyzed and derived. Moreover, the paper also described the process of discretization which converts curve to pulse. Next, the possible cases of acceleration and deceleration status in actual motion, which were determined by real time motion control, were listed. Finally, the pulse testing indicates that the algorithm and motion control methods of acceleration and deceleration proposed are feasible and can be applied to motions with various parameters as well as improve the efficiency and stability of step motor. [Ch, 7 fig. 1 tab. 11 ref.]

**Key words:** step motor; motion control; S curve acceleration and deceleration; embedded; AT91SAM9261 system

### 0 引言

步进电机是将电脉冲信号转变为角位移或线位移的开环控制元件, 具有精度高、运行可靠、无累计误差等优点, 因而被广泛应用于数字控制系统中<sup>[1]</sup>。步进电机启动时脉冲频率不能高于空载启动频率, 否则电机会发生丢步或堵转。因此要达到较高的运行频率电机必须有一个加速过程。常见的加减速算法有直线加减速、指数曲线加减速等, 因其均具有柔性冲击, 故不适合高速运动<sup>[2]</sup>。针对于此, 文中提出了一种基于嵌

入式 ARM 片上系统 AT91SAM9261 的 S 曲线加减速算法并研究了其实现和运动控制过程, 从而解决了步进电机加减速过程中的柔性冲击问题, 增强了电机高速运动时的稳定性和可靠性。

### 1 AT91SAM9261 脉冲发生原理

#### 1.1 AT91SAM9261 简介

AT91SAM9261 是 Atmel 公司生产的一款以 ARM926EJ-S ARM/Thumb 处理器为核心的高性能 ARM 片上系统。主时钟频率 190 MHz 时性能高达

收稿日期: 2011-10-31; 修回日期: 2011-11-10

作者简介: 简毅 (1968), 男, 重庆人, 重庆大学机械工程学院副教授, 硕士生导师, 主要研究方向为机电一体化技术、智能控制与计算机协同监控。通信作者: 刘宝; E-mail: lbecu2012@126.com

210 MIPS。片内集成 32 kB 的 ROM 和 160 kB 的 RAM 可以实现单周期最大总线速度存取<sup>[3]</sup>。片内集成高级中断控制器,3 个并行 I/O 控制器,3 通道 16 位定时器/计数器,可以检测频率或产生波形<sup>[4]</sup>;丰富的片上资源大大简化了系统外围设计并降低了开发成本。

## 1.2 脉冲频率控制

AT91SAM9261 内部集成的定时器/计数器 (TC) 通道可以产生指定频率的脉冲波形。每个通道内集成  $R_A$ 、 $R_B$  和  $R_C$  3 个比较寄存器和 1 个计数器。在波形操作模式下,当计数器值与  $R_A$  或  $R_C$  寄存器值相等时 TIOA 引脚电平发生改变(与  $R_C$  相等时计数器同时被清零)并引发中断,以此产生脉冲波形,如图 1 所示。

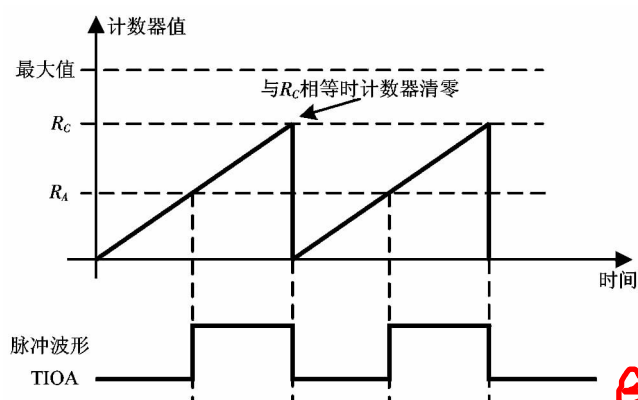


图 1 脉冲波形控制

Figure 1 Pulse waveform control

因此,通过编程  $R_A$  和  $R_C$  寄存器值就可以控制 TIOA 引脚产生指定宽度和频率的脉冲信号, $R_A$  用于控制脉冲宽度, $R_C$  控制脉冲频率,通常  $R_A$  值取  $R_C$  的一半。 $R_C$  值计算如下

$$R_C = f_{\text{时钟}} / f_{\text{目标}} \quad (1)$$

其中,  $f_{\text{时钟}}$  为计数器时钟频率,可通过编程改变;  $f_{\text{目标}}$  为目标脉冲频率。

## 2 S 曲线算法及实现

### 2.1 S 曲线算法分析

S 曲线加减速的特点是存在加加速度,故加速度没有突变,在速度变换过程中不存在柔性冲击<sup>[5-6]</sup>,适用于高速运动。完整的 S 曲线加减速过程如图 2 所示,共分 5 个阶段。

以拐点时刻  $t_1$  为分界点,在  $0 - t_1$  时间段内,速度由起始速度  $v_0$  加速至拐点速度  $v_1$ ,加加速度  $a'$  为常数  $c$ ,加速度  $a$  由 0 逐渐增加;  $t_1 - t_2$  时间段内,速度由  $v_1$  加速至运行速度  $v_t$ ,  $a'$  为负值,  $a$  逐渐减小至 0;  $t_2 - t_3$  为匀速段,保持速度  $v_t$  不变;  $t_3 - t_4$  段、 $t_4 - t_5$  为减速段,与加速段类似,只是加加速度  $a'$  为负值。

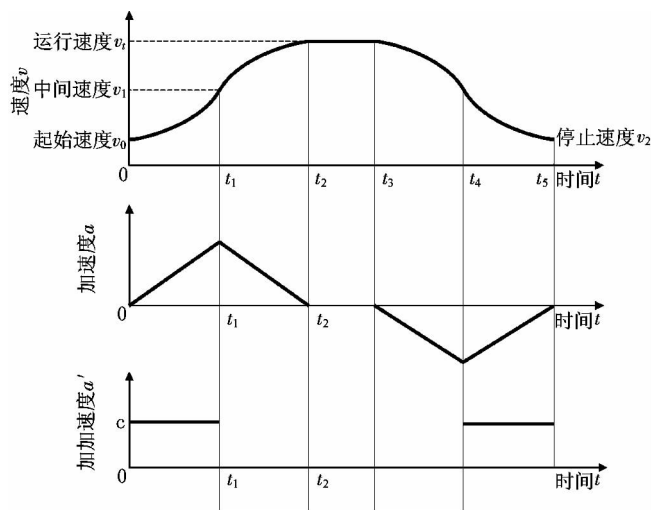


图 2 S 曲线解析

Figure 2 S Curve analyzing

加速段 S 曲线方程 ( $v - t$  函数) 推导如下:

1)  $0 - t_1$  时间段内,加加速度为常数  $c$  ( $c > 0$ ),故加加速度为

$$a = \int_0^t a' dt = \int_0^t c dt = ct \quad \frac{da}{dt} = c$$

对上式进行积分得 S 曲线方程 ( $v - t$  函数) 为

$$v = v_0 + \int_0^t a dt = v_0 + \int_0^t ct dt = v_0 + \frac{1}{2} ct^2 \quad (2)$$

2)  $t_1 - t_2$  时间段内,加加速度为  $-c$ ,对加加速度进行积分得加速度为

$$a = \int_0^t a' dt = \int_0^{t_1} c dt + \int_{t_1}^t (-c) dt = 2ct_1 - ct \quad (3)$$

(3) 式中拐点时刻  $t_1$  可由拐点速度  $v_1$  确定,由 (2) 式可得

$$t_1 = \sqrt{2(v_1 - v_0)/c} \quad (4)$$

将 (4) 式代入 (3) 式并积分得 S 曲线方程 ( $v - t$  函数) 为

$$\begin{aligned} v &= v_1 + \int_0^t a dt = v_1 + \int_0^t (2ct_1 - ct) dt \\ &= 2ct \sqrt{2(v_1 - v_0)/c} - \frac{1}{2} ct^2 - 2v_1 + 3v_0 \end{aligned} \quad (5)$$

公式 (2) 和 (5) 分别为  $0 - t_1$ 、 $t_1 - t_2$  时间段的 S 曲线方程,式中  $c$  值为常数,决定加减速过程的快慢,其最佳值在工程实践中通过反复实验得到,  $v_1$  一般取  $v_0$  和  $v_t$  的中间值。减速段曲线算法与加速段一致,可看作加速段的逆过程。

### 2.2 加减速曲线实现

将连续的加减速曲线转化为电脉冲信号控制步进电机的基本方法是将曲线离散化为各个频段<sup>[7]</sup>,在内

存中生成频段表,然后查表依次加载各个频段,使控制系统实际产生的脉冲曲线与理论加减速曲线相拟合。由于频率与速度是倍数关系,两者只相差一个脉冲当量,故曲线算法对频率同样适用。加速阶段曲线离散如图3所示。

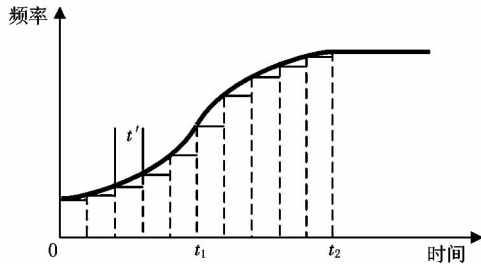


图3 曲线离散化

Figure 3 Curve discretization

离散实现的具体方法为将整个加速时间等分为若干时间段,每个时间段的长度固定为 $t'$ ,一般取 $t' = 2/f_{\text{start}}$ ,其中 $f_{\text{start}}$ 为电机的起始频率。然后将时间 $t$ 从 $t_0$ 开始以 $t'$ 为跨度依次递增并计算每个 $t$ 时刻处的曲线理论值,此值即为每个频段的频率值。由于拐点时刻 $t_1$ 前后曲线方程不同故需要在递增计算过程中判断拐点时刻并即时切换算法。具体流程如图4所示(设系统需求的运行频率为 $f_{\text{mov}}$ ,电机启动频率为 $f_{\text{start}}$ )。

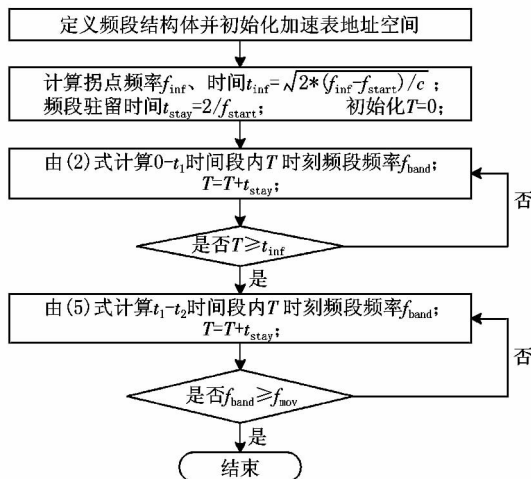


图4 程序流程图

Figure 4 Program flowchart

上述过程用C语言实现的核心代码如下:

```
typedef struct //加减速频段结构体
{
    long BandF; //频段频率
    long BandP; //频段脉冲
```

```
    long TotalP; //累计脉冲
} BAND;

#define ACC_ADDR_BEGIN 0x20600000 //定义加减速频段表在内存中的开始地址

const int c = 1000; //加加速度为固定值,由反复实验确定

void BandTabCreate(long StaF, long MovF) //频段表建立函数 参数为起始频率 StaF, 目标频率 MovF
{
    BAND * BandAddr = (BAND *) ACC_ADDR_BEGIN; //各频段结构地址

    long InfF = (StaF + MovF)/2; //拐点频率
    double InfT = sqrt(2 * (InfF - StaF)/c); //拐点时间

    double St = 2.0/StaF; //频段时间跨度
    double T = 0; //频段开始时刻
    while(T < InfT) //计算 0 - t1 时间段内 T 时刻频段频率
    {
        BandAddr -> BandF = StaF + c * T * T/2; // (2)式曲线方程实现
        T = T + St; //下一频段时刻
        BandAddr = BandAddr + sizeof(BAND); //下一频段内存地址
    }

    while(BandAddr -> BandF < MovF) //计算 t1 - t2 时间段内 T 时刻频段频率
    {
        BandAddr -> BandF = 2 * c * T * sqrt(2 * (InfF - StaF)/c) - c * T * T/2 - 2 * InfF + 3 * StaF; // (5)式曲线方程实现
        T = T + St; //下一频段时刻
        BandAddr = BandAddr + sizeof(BAND); //下一频段内存地址
    }
}
```

以上仅对频率的实现做了分析,每个频段还包含本频段脉冲数及累计的脉冲总数用于实时控制,对 $v-t$ 函数进行积分即可得到位移时间曲线,然后对其进行离散操作即可,方法与频率类似。

由(2)式可以看出,当加加速度值固定时, $0 \sim t_1$ 时间段内的各频段频率值固定,只是频段个数与运行

频率有关,因此可以在系统初始化时以最大频率为运行频率提前建立好  $0 \sim t_1$  时间段的频段表,运动开始前只需查表即可,节省了时间。而  $t_1 \sim t_2$  时间段的频段表则必须在每次运动开始前动态建立。

减速阶段可以看作加速阶段的逆过程,加速阶段的曲线方程同样适用于减速阶段的实现,且 2 者离散实现过程基本一致<sup>[8]</sup>,不同点在于减速阶段起始频率换作停止频率,并且频段表的加载顺序与加速阶段相反。

频段表建立完毕后须将其加载至 AT91SAM9261 的计数器/定时器模块,具体过程为将每个频段(包括匀速段)的频率根据(1)式转化为  $R_c$  寄存器值并装载至寄存器,并在脉冲中断子程序中实时记录脉冲数,当其达到本频段脉冲数时则切换到下一频段,直至加载完所有频段,完成指定运动。

### 3 运动实时控制

#### 3.1 运动情况分析

在实际运动控制过程中,由于运行速度和运动距离的限制,实际运动曲线可能会与图 2 的不同<sup>[9]</sup>,实际运动的主要情况如图 5 所示。

图 5(a)和(b)的启动频率和停止频率均小于运行频率,当实际运动距离允许电机达到运行频率时按照(a)运动,否则为(b)。在图 5(a)中电机按照 S 曲线正常加速到运行频率,然后以运行频率匀速运动一段时间,最后减速至停止频率,而图 5(b)中电机未加速到运行频率便开始减速,以保证电机的正常运行,不发生失步,此时控制系统需要调整最高运行频率。当运行距离极小(微距)时,直接以启动频率(或停止频率)匀速完成运动<sup>[10]</sup>。

在图 5(c)和(d)中,运行频率介于启动频率和停止频率之间,电机直接以运行频率停止或启动,若实际运动距离太小则同样需要实时调整运行频率值。

若启动频率和停止频率均大于运行频率,则直接以运行频率启动并停止,只有匀速运动阶段,图中省略。

#### 3.2 实时运动控制

为了实现以上各种运动情况,需要对步进电机进行实时运动控制。假设电机启动频率为  $f_{start}$ ,停止频率为  $f_{end}$ ,目标运行频率为  $f_{mov}$ ,需要发送的脉冲总数为  $p_{all}$ ,从  $f_{start}$  加速至  $f_{mov}$  的累计脉冲数为  $p_{acc}$ ,从  $f_{mov}$  减速至  $f_{end}$  的累计脉冲数为  $p_{dec}$ ,则运动实时控制流程如图 6 所示。

控制系统首先判断  $f_{mov}$  与  $f_{start}$  和  $f_{end}$  之间的大小关

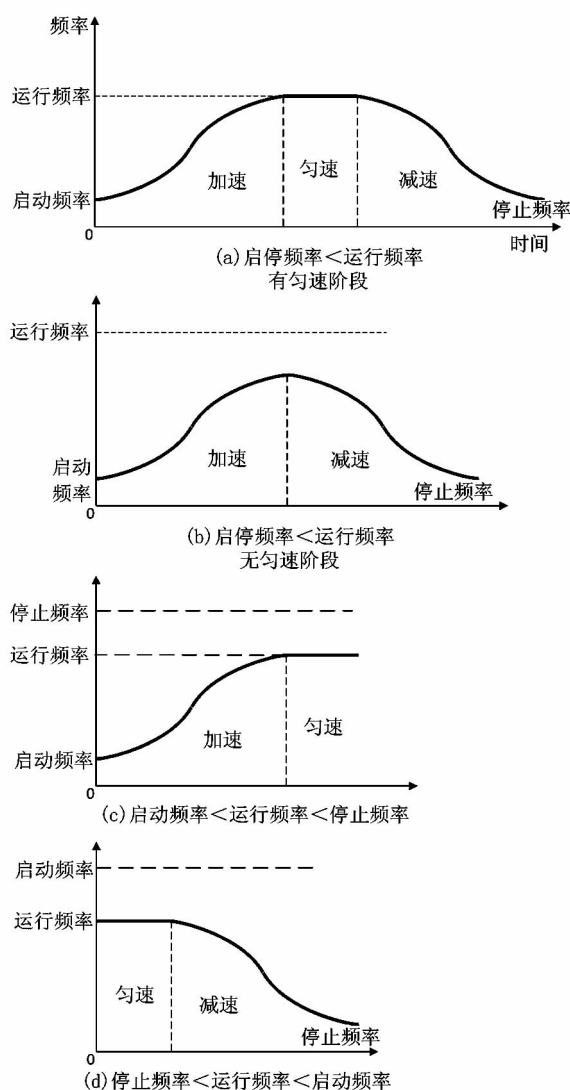


图 5 实际总体运动曲线

Figure 5 Curves of actual movement

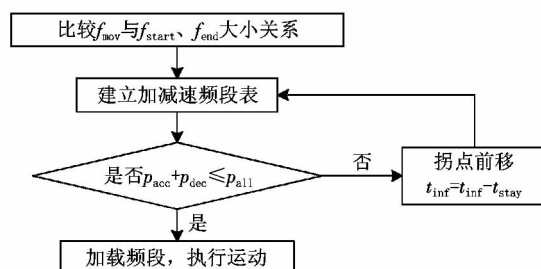


图 6 实时控制流程

Figure 6 Realtime control flowchart

系。若  $f_{start}$  和  $f_{end}$  均小于  $f_{mov}$ , 则分别以  $f_{start}$  和  $f_{end}$  为起始频率,  $f_{mov}$  为目标频率建立加减速频段表, 然后判断加减速累计脉冲之和是否超过运动指定的总脉冲数, 若没有超过, 说明在指定距离内可以达到指定运行频率,



则将各频段频率对应寄存器值依次加载至 AT91SAM9261 的  $R_c$  寄存器, 执行运动, 与图 5(a) 对应; 若超过总脉冲数, 说明在指定距离内电机不能达到运行频率, 则将拐点频段 ( $0 \sim t_1$  时间段内最后一个频段) 的前一频段作为新的拐点频段, 并将此频段频率的两倍作为新的运行频率重新建立频段表, 再次判断累计脉冲数, 重复此操作, 直到调整至加减速累计脉冲之和不再超过总脉冲数为止, 然后加载频段, 完成运动, 与图 5(b) 对应; 当  $f_{mov}$  在  $f_{start}$  和  $f_{end}$  之间时, 只需建立加速段或减速段, 控制过程类似, 分别对应图 5(c) 和 (d)。

#### 4 曲线验证

为了验证步进电机的实际运行频率变化是否符合 S 曲线, 采用光电式脉冲编码器配合 AT91SAM9261 的捕捉模式 TC 模块作为测试系统。编码器通过键与步进电机相连用作位置检测装置, 其发出的检测脉冲作为输入信号传送至 AT91SAM9261 芯片的 TIOB 引脚。在捕捉模式下, 当 TIOB 引脚检测到脉冲上升沿时 TC 计数器值会被装载至 TIOB 寄存器 (计数器同时被清零) 中并触发中断, 通过在中断服务程序中读取 TIOB 寄存器值即可获取脉冲间隔。

设步进电机控制系统启动频率为 1 000 Hz, 停止频率为 1 200 Hz, 运行频率为 6 000 Hz, 运动距离为 5 mm, 测试系统 TC 模块时钟频率  $f_{Clk}$  为 160 MHz, 从电机启动至停止过程中每隔 0.2 s 采样读取 TIOB 寄存器值  $N$  如表 1 所示 (选取加速部分)。

表 1 脉冲间隔记录

Table 1 Pulse interval record

时刻/s	TIOB 寄存器值 $N$	时刻/s	TIOB 寄存器值 $N$	时刻/s	TIOB 寄存器值 $N$
0.2	1 600	1.6	808	3.0	373
0.4	1 568	1.8	701	3.2	355
0.6	1 481	2.0	610	3.4	341
0.8	1 355	2.2	533	3.6	331
1.0	1 212	2.4	473	3.8	325
1.2	1 066	2.6	430	4.0	321
1.4	930	2.8	398	4.2	320

寄存器值除以时钟频率即为实际时间间隔, 将时间间隔取倒数即得到各时刻处频率值

$$f = f_{\text{时钟}} / N$$

根据上式将各时刻寄存器记录值转化为频率值并将其绘成曲线, 如图 7 所示。

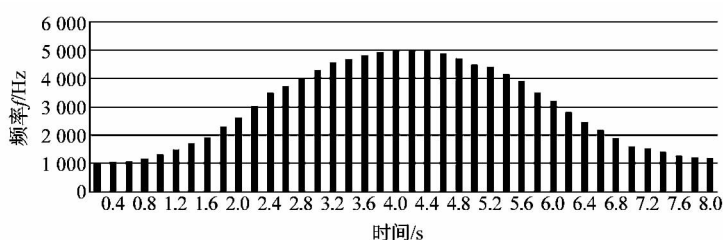


图 7 脉冲测试结果

Figure 7 Pulse test result

由图 7 可以看出, 步进电机加速过程与减速过程均符合 S 曲线规律, 编码器发出的脉冲频率间切换平滑, 电机运行稳定<sup>[11]</sup>, 而且控制系统根据实际运动距离要求, 将电机只加速到 5 000 Hz 即开始减速, 与图 5(b) 中运动情况相吻合, 缩短了加减速时间, 提高了效率, 达到了预期的控制效果。

#### 5 结语

加减速控制是步进电机运动控制中的重要环节, 直接影响系统的效率和稳定性。文中研究了基于嵌入式片上系统 AT91SAM9261 的 S 型加减速曲线的算法、实现及其实时运动控制方法, 并通过脉冲测试进行了验证。结果表明, 该算法及其实现过程能够使步进电机快速平滑地按照 S 型曲线完成加减速过程, 无柔性冲击, 而且通过实时运动控制能够使步进电机在各种运动条件下高效完成运动, 稳定可靠, 在运动控制中具有较好的实用意义, 特别适合于高速运动的场合。

#### 参考文献 (References):

- [1] 张强, 吴红星. 基于单片机的电动机控制技术[M]. 北京: 中国电力出版社, 2008.
- [2] 李晓菲, 胡鸿. 步进电机加减速控制规律[J]. 机电产品开发与创新, 2006(1): 122-123.
- [3] 黄智伟, 邓月明, 王彦. ARM9 嵌入式系统设计基础教程[M]. 北京: 北京航空航天大学出版社, 2008.
- [4] Atmel Corporation. AT91ARM/Thumb-based Microcontrollers AT91SAM9261 Preliminary [EB/OL]. 2007-11-26 [2011-10-20]. <http://www.atmel.com>.
- [5] 陆鑫森, 赵国军, 邢海潇, 等. 电梯运行速度曲线的控制[J]. 轻工机械, 2010, 28(6): 70-72.
- [6] 欧海平, 周砚江, 张华叶, 等. 基于 SOPC 的步进电机加减速 PWM 控制器 IP 核设计[J]. 机电工程, 2011, 28(6): 708-711.
- [7] 张振华. 数控系统加减速算法及定位技术研究[D]. 大连: 大连理工大学学报, 2007.
- [8] 张新荣, 徐保国. 基于正弦细分驱动技术的步进电机控制系统设计[J]. 电测与仪表, 2010(8): 25-26.
- [9] 李晓辉, 郭义杰, 冷洪滨. S 曲线加减速控制新方法的研究[J]. 组合机床与自动化加工技术, 2007(10): 50-53.
- [10] 陈友东, 王田苗, 魏洪兴, 等. 数控系统的直线和 S 形加减速研究[J]. 中国机械工程, 2006, 17(8): 1600-1603.
- [11] 许良元, 桂贵生, 彭丹丹. 加速度连续可变的加减速控制规律研究[J]. 组合机床与自动化加工技术, 2005(3): 12-13.