

# 基于 STM32 的步进电机 S 曲线加减速算法的优化

仪慧玲, 张仁杰

(上海理工大学光电信息与计算机工程学院, 上海 200093)

**摘要:** 步进电机加速启动或者减速停止的时候, 容易发生失步或者过冲现象。S 曲线加减速算法比较平稳, 因此广泛应用于运动控制系统。本文基于 STM32 控制系统通过 Modbus 通信协议给驱动器发送内部自发脉冲命令, 在 S 曲线的基础上对算法进行优化, 不仅保证电机运行过程的平稳, 而且可以提高电机的脉冲响应频率。通过对脉冲产生的基准时钟进行不同的设置, 实现步进电机低速和高速情况下对脉冲频率精度的需求。

**关键词:** 步进电机; S 曲线; STM32; Modbus; 脉冲响应频率

**中图分类号:** TP391      **文献标识码:** A

## The optimization of S curve algorithm based on STM32

YI Hui-ling, ZHANG Ren-jie

(School of Optical - Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

**Abstract:** Stepper motor is likely to out of step or overshoot when it accelerates to start or slowing down to stop. The S curve arithmetic is widely used in motion control systems because it is stationary. On the basis of the S curve, this paper optimizes the algorithm based on STM32 control systems and Modbus communication protocol. It's not only to guarantee the machine running smoothly, but also can improve the impulse response frequency of the motor. By setting the reference clock it can achieve the demand of the accuracy of the impulse frequency during the low speed or high speed.

**Key words:** stepper motors; scurve; STM32; Modbus; impulse response frequency

## 0 引言

步进电机是一个将脉冲信号转换成角位移的执行机构<sup>[1-2]</sup>, 其工作原理是每接收一个脉冲, 就按设定的方向转过一个固定的角度, 可以通过控制脉冲的数量来控制电机的角位移量, 通过控制脉冲的频率来控制电机的速度及加速度。理论上脉冲频率越高, 电机的运转速度越快。但在实际应用中, 脉冲频率太高时, 步进电机的内部反电动势会增加, 会导致步进电机发生失步或者堵转现象。所以要考虑最合适的脉冲输出方案, 才能实现步进电机的精确定位。

在步进电机控制系统中, 有两种可以选择的工作模式, 外加脉冲模式和内部自发脉冲模式<sup>[3-4]</sup>。其中内部自发脉冲模式下可以设置点动频率、启动频率、运行频率、升、降频时间、脉冲数等参数。通过

RS485 总线并使用 Modbus 协议接收上位机的运动控制命令以及设定的参数, 使 CPU 按照设定的参数输出相应的升、降频脉冲用于控制步进电机运动。Modbus 协议功能完善、协议开放、使用简单, 已经成为一种通用的工业标准<sup>[5]</sup>。为了实现精确的定位, 必须保证脉冲信号严格按照要求输出。本文中运动控制选用 STM32 系列的芯片, STM32 的时钟频率可以达到 72MHz, 并且定时器资源比较丰富, 通过对高、低频脉冲信号输出的不同情况的分析, 对 S 曲线进行优化, 提出了一种可行的方法确保了电机在低速启动以及高速运转时的稳定性和可靠性<sup>[6-7]</sup>。

收稿日期: 2014-03-14

作者简介: 仪慧玲(1989-), 女, 在读硕士研究生, 研究方向为步进电机控制系统。

1 自发脉冲设计方案

步进电机是由数字信号控制运动的,转速随着输入脉冲的频率变化而变化。电机工作在自发脉冲模式下时,脉冲是由 CPU 控制输出给驱动器,驱动电机运动<sup>[8]</sup>。电机一个完整的运动过程,要经过从低速启动、加速到目标速度的加速过程,以目标速度匀速运动、到减速的过程才能使电机稳定的运动。内部自发脉冲工作模式下,要解决的问题就是电机的加减速过程,即以何种曲线加减速的问题。

曲线加减速在伺服电机系统中被广泛使用,在步进电机系统中也同样适用。顶部圆滑的曲线,可以确保在最终趋于恒定速度时,不会出现速度过冲的现象。

步进电机加速过程是由基础频率和跳跃频率(步进电机在基础频率上逐渐提高的频率)组成加速曲线。通过计算式来获得加速度的数据,可以为 CPU 的速度控制提供理论依据。通常情况下 CPU 的速度控制,不是实时地用公式计算,而是把事先算好的数据填入 CPU 内存中。速度控制是利用 CPU 的定时器来实现的,也就是用规定的时间来触发一个脉冲。这个规定的时间也就是定时器的时间常数。计算式得到的数据为频率值,其倒数就是时间值。然后按照 CPU 定时器的工作频率进行换算就可以得到速度控制所需的时间常数。

步进电机工作在自发脉冲时的工作原理如图 1 所示。

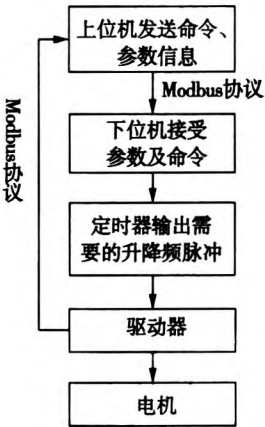


图 1 自发脉冲工作原理

上位机通过 Modbus 协议给下位机发送命令, CPU 收到自发脉冲命令之后,关闭外部脉冲接收端,打开内部自发脉冲所需的定时器。通过设定该定时器的定时时间来控制输出脉冲的频率,脉冲频率按照 S 曲线加减速方式进行输出,从而使电机可以平稳的运行。

2 S 曲线加减速算法的理论实现

加减速曲线的种类有很多,如图 2 所示为典型的 S 曲线模型,其中 PPS 表示每秒脉冲数,STEP 表示步长。为了获得更好的加减速效果,本系统选用的是 logistic 回归函数,logistic 函数的基本模型为:

$$y(x) = \frac{1}{1 + e^{-x}}$$

曲线特点是开始增长缓慢,但是在以后的某一范围内迅速增长,达到一定程度后,增长又缓慢下来最终趋于一个稳定值。正好适合电机的加速轨迹,既可以符合加速度曲线形态,又可以快速进入目标速度。本系统中使用的 logistic 回归函数模型如下:

$$y = (k - c) \times \frac{1}{1 + e^{-a(x - b)}} + c$$

其中,  $k$  表示增益,  $a$  是倾斜率,  $b$  是偏移量,  $c$  是起始增益。

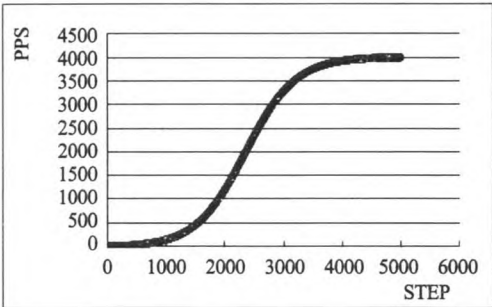


图 2 经典 S 曲线模型

倾斜率  $a$  的变化对曲线的影响如图 3 所示。

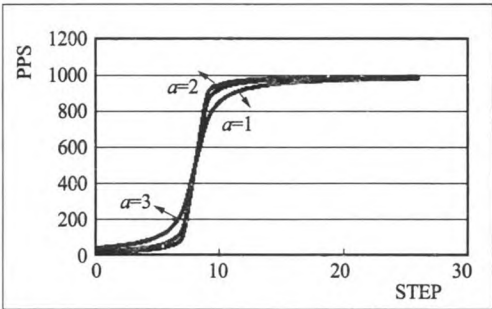


图 3 logistic 函数模型中倾斜率对曲线的影响

由上图可以看出,倾斜率越大,曲线就越陡,也就是上升速度越快,本系统中取  $a = 1$ ,由曲线走势可以看出  $a = 1$  的时候,曲线较平滑,并且可以快速达到目标速度。

图 4 表示了偏移量  $b$  对曲线的影响情况,从上图可以看出,偏移量只是单纯地左右偏移,而曲线形态不发生变化,本系统中取  $b = 5$ ,可以实现快速达到目标速度的要求。

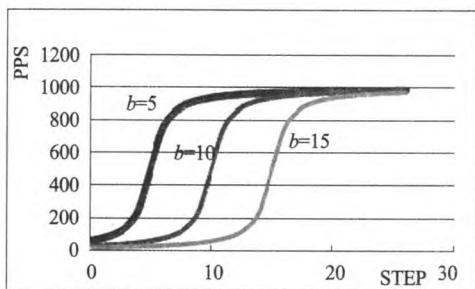


图4 logistic 函数模型中偏移量对曲线的影响

增益  $k$  设置为 1000, 因此系统选定的曲线模型为:

$$y = (1000 - 200) \times \frac{1}{1 + e^{-(x-5)}} + 200$$

本系统中 S 曲线模型数据表中存放有 5000 个数据, 即模型中的升频步长为 5000 步。模型表中的数据越多, 相对精度越高, 但是占用 CPU 的内存容量也越大。自发脉冲选用 5000 点来模拟升降频过程已经可以满足要求。

当前频率的计算过程:

$$\text{平均频率} = \frac{(\text{起始频率} + \text{最高频率})}{2}$$

$$\text{实际升频步长} = \text{平均频率} \times \text{升频时间}$$

$$\text{当前步长对应模型表中的位置} = \text{实际的当前步}$$

$$\text{长位置} \times \frac{\text{模型表总步长}}{\text{实际升频步长}}$$

所以可以求出当前的频率为:

$$\text{当前频率} = \frac{\text{起始频率} + \text{模型表频率} \times (\text{最高频率} - \text{起始频率})}{\text{模型表最高频率}}$$

模型表中 X 轴代表步长, Y 轴即为相对应的频率。程序中变量的最大字长定义为 unsigned int 型, 即为四字节字长, 所以变量的最大值为  $2^{32}$ 。本系统所要实现的最高频率为 1MHz, 假设最大升频时间为 999ms, 则最大的升频步长约为:

$$\frac{100000 + 0}{2} \times 999 \times 10^{-3} = 499500$$

为保证 4 字节变量不出现溢出, 模型表纵轴的最大取值为  $2^{32}/1000000 = 4294$ , 所以模型表的最高频率可以取 4000<sup>[9]</sup>。

由前述当前频率的计算表达式可以求出当前频率值, 频率的倒数即为输出脉冲的周期值。本系统中定时器的模式设置为 PWM 模式输出<sup>[10]</sup>, 根据该数值只要设定好定时器的定时时间即可实现相应频率脉冲的输出。

### 3 S 曲线加减速算法的具体实现办法

STM32 时钟最高能达到 72MHz, 若采用 72MHz 的时钟基准, 则定时器计数一次的时间为  $1/72\mu\text{s}$ 。而 STM32 是 16 位的定时器, 最大计数值为 65535, 所以若定时器的时间基准选为 72MHz, 则脉冲的最大计数周期为  $65535/72 = 910\mu\text{s}$ , 此时频率约为 1.1kHz。即最小的频率只能达到 1.1kHz, 往往步进电机的启动过程的起始频率会比较低, 1.1kHz 并不能满足要求。可以实现的方法有两种, 一是对定时器的时间基准进行分频处理; 二是使用定时器的级联, 但是会增加 CPU 资源的消耗。

对定时器时间基准进行分频时, 对于分频系数的选择也要慎重考虑, 既要考虑到定时精度的问题, 又要考虑升频过程中最大、最小频率对总的定时时间的要求<sup>[11]</sup>。为了增加定时时间, 即实现低频脉冲的输出, 分频系数越高越好, 当分频系数为 72, 即定时器时钟为 1MHz 时, 定时器计一个数的时间为  $1\mu\text{s}$ , 最大的定时时间为  $65535\mu\text{s}$ , 此时对应的频率为 15.3Hz, 理论上可以满足电机起始频率的要求<sup>[11]</sup>。

如图 5 所示为最高频率设置为 300kHz, 即对应的定时时间为  $3.3\mu\text{s}$ , 定时器基准时钟频率设置为 1MHz 下的定时器脉冲输出波形。CPU 计一个值的时间为  $1\mu\text{s}$ , 所以定时器计数的理论值为 3.3, 但是定时器的计数值设置只能为整型变量, 所以计数值 3.3 会自动转换成整型变量 3, 即脉冲的实际输出频率为 330kHz, 与目标频率相差 30kHz, 误差较大。分析可知分频系数为 72 且定时时间不为整数时, 都会存在误差。

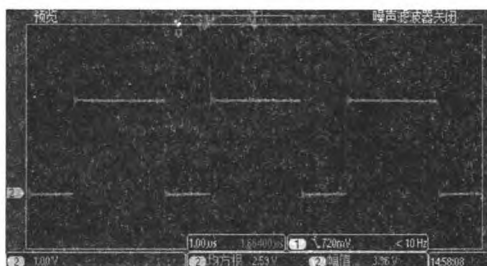


图5 定时器时钟为 1MHz、最高频率 300kHz 脉冲输出

设分频系数为  $div$ , 当前频率为  $f$ , 当前的理论计数值为  $CNT$ , 定时器的基准时间为  $TB$ 。则:

$$TB = \frac{div + 1}{72} \times 10^{-6} \text{s}$$

$$CNT = \frac{72}{(div + 1)f} \times 10^6$$

可求出其误差为:

$$\Delta f = 1/([CNT] \times TB) - 1/(CNT \times TB)$$

分频系数  $div$  越小, 对应的时间基准  $TB$  越小,

误差也就越小。

当目标频率为 400kHz 的时候,理论的定时时间应为  $2.5\mu\text{s}$ 。图 6 为目标频率为 400kHz,定时时钟 6 分频的时候的脉冲输出,此时示波器观测到的波形周期已经接近理想值  $2.5\mu\text{s}$ 。定时器的时钟为 12MHz,所以计数基准时间为  $0.083\mu\text{s}$ ,对应的定时器计数值的理论值为,CPU 处理的时候会把计数值设置为 30,所以定时误差为,实际的输出脉冲频率为,所以频率误差为 1.6kHz。

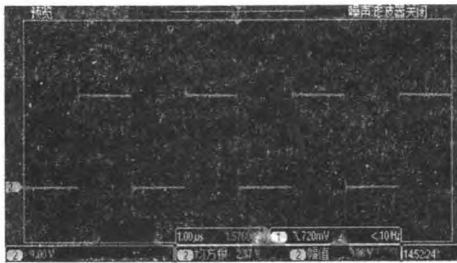


图 6 定时时钟为 12MHz、最高频率 400kHz 脉冲输出

当分频系数较小的时候,精度较高,但无法满足低频的需求;当分频系数较大的时候,高频的时候对应的精度达不到。所以电机的升降频过程中,定时器的设置要综合考虑高低频不同的需求。

最终本系统的解决方案是在低频的时候定时器的分频系数设置为较大分频值,高频的时候使分频系数降低。目前采取的是低频时候分频系数为 72,即低频可以达到 15Hz;当频率达到 200kHz 的时候改变分频系数,设置分频系数为 1,即定时器的基准时钟为 72MHz,计一个数的时间为  $1/72\mu\text{s}$ ,经过试验验证无论是低频还是高频完全可以满足

精度的要求。

4 结束语

在步进电机的加减速过程中,S 曲线是常用的一种较为精确的算法,启动和停止时的平稳性都比较好,可以根据不同的电机性能调整速度的变化率。本文在常用的 S 曲线基础上提出了一种更加可靠的实现方法,在低频和高频段都能可靠准确地实现相应频率脉冲的输出。目前该算法已经成功运用于步进电机控制系统中。

参考文献:

[1] 丛爽,李泽湘.实用运动控制计数[M].北京:电子工业出版社,2006.

[2] 朱海.几种常见步进电机控制方法庶谈[J].科技经济市场,2013(12):114-115.

[3] 王运.基于 DSP、CPLD 的运动控制器设计、实现及运动控制算法研究[D].成都:电子科技大学,2005.

[4] 鲍吉龙,叶平.工业监控系统的网络化发展[J].微计算机信息,2006,6(1):66-68.

[5] 唐义思.步进电动机运行电路设计中应注意的问题[J].科技创新导报,2013(27):57.

[6] 蒙博宇. STM32 自学笔记[M].北京:北京航空航天大学出版社,2012.

[7] 紫文才,何邦贵,紫振钦.基于 FPGA 的步进电机优化控制[J].现代电子技术,2013,36(23):142-144.

[8] 王鸿钰.步进电机控制技术入门[M].上海:同济大学出版社,1990.

[9] 史新福.32 位微型计算机原理接口技术及其应用[M].2 版.西北工业大学出版社,2001:243-257.

[10] 刘宝廷.步进电机及其驱动控制系统[M].哈尔滨:哈尔滨工业大学出版社,1997:57-82.

[11] 袁涛.单片机 C 高级语言程序设计及其应用[M].北京:北京航空航天大学出版社,2001:203-255.

责任编辑:薛慧心

\*\*\*\*\*  
(上接第 177 页)

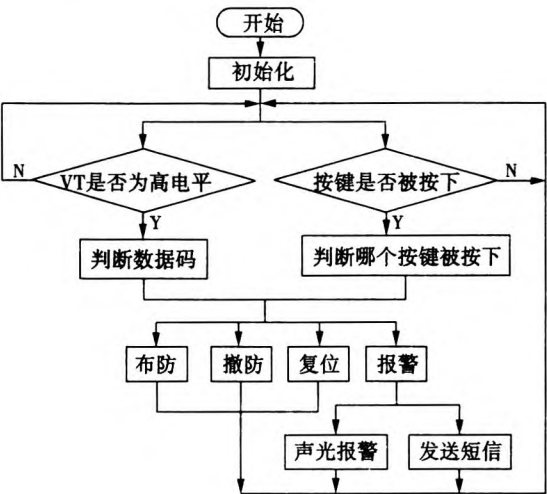


图 6 主机程序流程图

4 结束语

本文设计了一种基于单片机的家用防盗报警系统,具备门窗探测和室内探测功能。系统采用本地报警和短信报警结合,最大程度提高报警及时性,减少误报。探测器与主机之间采用无线通讯,避免布线的麻烦。遥控器与按键配合使用,人机交互性能良好。且系统整体成本较低,适合家庭等小型用户使用。

参考文献:

[1] 阎石.数字电子技术基础[M].北京:高等教育出版社,2006.

[2] 楼然苗,李光飞. MCS-51 系列单片机设计实例[M].北京:北京航空航天大学出版社,2006.

[3] 姚彦茹.热释电红外传感器在防盗系统中的应用[J].信息技术,2007(5):87-88.

[4] 刘举庆,刘莲秋.遥控编解码电路的新应用[J].信息技术,2011(11):46-49.

[5] 陌尘. 编码解码芯片 PT2262/PT2272 工作原理[J]. 电子制作,2004(8):44-46.

责任编辑:张荣香