

基于 ARM Cortex-M3 的运动控制系统与算法设计

朱 阁, 郑力新, 林晓海

(华侨大学 信息科学与工程学院, 福建 厦门 361021)

摘 要: 提出一种基于 ARM Cortex-M3 为核心的运动控制器的新运动控制方案, 并给出关键算法。控制系统采用“ARM 运动控制器+PC 机”的结构。PC 机实现界面功能以及部分预处理功能, 运动控制器则完成关键的算法与处理。在数控冲孔机控制上进行的应用表明, 该系统具有较高的可行性。

关键词: ARM Cortex-M3; 运动控制系统; 算法

中图分类号: TP273

文献标识码: A

文章编号: 1674-7720(2013)01-0077-03

The motion control system and algorithm design based on ARM Cortex-M3

Zhu Ge, Zheng Lixin, Lin Xiaohai

(College of Information Science and Engineering, Huaqiao University, Xiamen 361021, China)

Abstract: A new motion control program is put forward based on the motion controller core of ARM Cortex-M3, into which the key algorithms are interpolated. The control system uses the structure of the ARM motion controller and PC machines. PC can realize the interface function and a part of the pre-processing functions, and the key algorithms and processing are completed by the motion controller. The application on CNC punching machine control shows that the system has the high feasibility.

Key words: ARM Cortex -M3; movement control system; algorithm

在现代工业控制领域中, 基于专用运动控制芯片或者 DSP 控制器的运动控制系统应用较为广泛^[1], 但成本较高。新兴的 ARM Cortex 系列具有低功耗、低成本、高性能和开发环境优秀的特点, 不仅在功能上能够满足工控领域对速度及功能的要求, 更具有很好的成本优势。本文采用 ARM Cortex-M3 处理器为运动控制器设计开放式运动控制系统^[2-3], 并在此硬件平台上研究运动控制策略与关键算法, 实现运动控制处理功能。

1 控制系统

1.1 控制系统硬件结构

系统采用“运动控制器+PC 机”的开放式结构, 如图 1 所示。系统以 ARM Cortex-M3 微处理器作为核心处理芯片的运动控制器, 与 PC 机构成的多处理器结构, 实现多轴步进电机的控制。

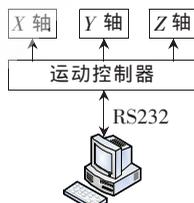


图 1 系统结构

运动控制器主要由 ARM Cortex-M3 微处理器、存储芯片、接口芯片以及外围接口等构成, 如图 2 所示。处理器芯片选用 ST 公司的 STM32F103VCT6, 其具有高性能、

低成本、低功耗的特点, 时钟频率最高能够达到 72 MHz, 并拥有丰富的外围设备和功能强大的定时器^[4]。

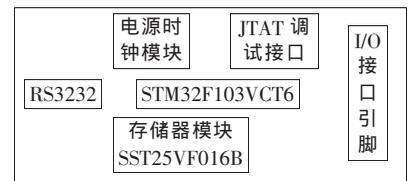


图 2 运动控制器结构图

ARM Cortex-M3 处理器主要实现数据处理、系统控制及中断处理等。数据处理: 对 PC 机下载的加工数据进行进一步处理, 主要是如何将微小线段的端点坐标、速度等信息通过特定算法转变为驱动步进电机驱动器的脉冲信号和方向信号。系统控制: 按照要求完成模块功能的分配与处理。中断处理: 完成定时器中断、串口通信中断、急停中断、位置控制等。

1.2 控制系统软件

运动控制系统软件程序主要有两大部分, 一是用于人机交互的界面程序, 包括系统初始化模块、参数设置模块、模板选择模块以及显示模块等; 二是为实现运动控制输出的算法处理程序, 主要包括速度前瞻性处理算法、坐标变换与主轴判断、脉冲分配以及两轴联动算法。

2 主要算法设计

运动控制系统需要通过运动控制算法来获得控制量的输出,以达到对机械运动部件的控制^[5]。本系统通过微小线段来拟合要加工的曲线,在PC机上进行前端预处理得到加工图形的坐标位置、端点数、端点速度等数据,并在运动控制器中进一步处理^[6-7]。系统的关键算法设计有:两轴联动的实现、速度前瞻性处理算法以及脉冲分配规划。

2.1 两轴联动

首先,根据X、Y轴的脉冲数来确定主从轴,脉冲数多的为主轴。为了实现联动,则必须在同一时间内将X、Y轴的脉冲数目发送完。

两轴联动主要是靠3个定时器级联来实现,如图3所示。其中,定时器2作为总线时钟的分频器,实现一个动态的“主时钟”;定时器3和定时器4分别作为“主时钟”的计数器,对定时器2产生的“主时钟”进行计数,以产生符合系统所需脉宽的脉冲。

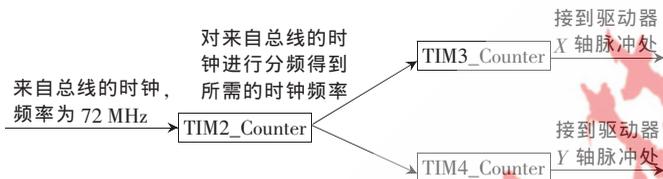


图3 定时器结构

在算法中利用上文介绍的定时器结构,在合理设置各个定时器的计数值的情况下,只要保证主轴的脉冲根据升降频曲线表的脉冲要求发送完需要发送的脉冲数,就能使从轴的脉冲同时发送完成。具体实现方法如下:

设定定时器3控制X轴,定时器4控制Y轴。要加工一条直线的时候,X轴要发XPlus个脉冲,Y轴要发送YPlus个脉冲。如若主轴为Y轴,并且该时刻主轴的半脉宽为 $N \mu\text{s}$,在此令 $\text{TIM3_Counter}=\text{YPlus}-1$, $\text{TIM4_Counter}=\text{XPlus}-1$,则定时器2输出一个脉冲为 $N/\text{XPlus} \mu\text{s}$,可以根据定时器2的输入输出关系:

$$\frac{N}{\text{XPlus}} = \frac{1}{72} (\text{TIM2_Counter} + 1)$$

求出TIM2_Counter的值: $\text{TIM2_Counter} = \frac{72 \times N}{\text{XPlus}} - 1$ 。

反推验证:在 $\text{TIM2_Counter} = \frac{72 \times N}{\text{XPlus}} - 1$ 时,X轴(从轴)的脉宽为:

$$\frac{1}{72} (\text{TIM2_Counter} + 1) \times (\text{TIM3_Counter} + 1) = \frac{N \times \text{YPlus}}{\text{XPlus}}$$

可得:

$$\frac{Y \text{ 轴频率}}{X \text{ 轴频率}} = \frac{\frac{1}{N}}{\frac{1}{\text{XPlus}}} = \frac{\text{YPlus}}{\text{XPlus}}$$

综上所述,为使Y轴发出YPlus个脉冲时,从轴会

配合产生XPlus个脉冲,实现两轴联动,只需对定时器进行如下配置:

$$\text{TIM3_Counter} = \text{YPlus} - 1;$$

$$\text{TIM4_Counter} = \text{XPlus} - 1;$$

$$\text{TIM2_Counter} = \frac{72 \times N}{\text{XPlus}} - 1。$$

2.2 速度前瞻性处理^[1,7]

系统要加工的圆弧是由大量的微小线段来逼近的,为了保证这些微小线段在高速运行时能够连续执行,不出现停顿现象,系统在上位机的内存中开辟一个空间,预先将加工的弧线数据读出,存储到该空间中,再根据设计好的步进电机升降频曲线对速度进行预先规划处理。预先规划包括:相邻直线段拐角速度优化、减速点预测以及连续微小线段始末速度确定。

在高速运动时,如果两个微小线段的拐角处速度差很大,则会使得拐角处产生很大的加速度,有可能导致过冲甚至堵转现象。因此在加工过程中,为使得连续微小线段的曲线加工流畅,必须保证微小线段间的速度过渡是连续的,不能发生突变。为保证轨迹精度并避免产生突变,两个运动轴的速度变化率不能超过该轴电机的驱动能力。在该约束条件下需要对减速点进行提前预测,从而保证每条小线段在运动到线段终点的时候能够达到预定所能达到的最高速度,并且不会过冲。

结合线段主轴的脉冲数目和升降频曲线表,推算出到达该小线段末端能达到的最高速度和最低速度及该线段终点的速度范围。再在这个范围从最高速度开始检查满足速度不突变条件的速度作为该线段的末点速度,并将这个速度或者这个速度在升降频曲线表中的相邻速度作为下一条小线段的起始速度。通过该方式可以规划出构成整条曲线的微小线段的端点速度。而后再利用规划好始末点速度来进行主轴的加减速脉冲分配,确定减速点的位置就可以避免电机运动过程中出现的过冲。

2.3 脉冲分配规划

以X轴为主轴为例,根据主轴的起点速度(line1.begin)、终点速度(line1.end)、主轴脉冲数(line1.Xlength)以及设定好的升降频曲线来规划加速、匀速、减速的脉冲数^[6]。具体算法如下:

如果起点速度和终点速度均为最大速度时,则加速和减速脉冲均为零,匀速脉冲数为整个主轴脉冲: $\text{line1.speed_yunsu} = \text{line1.Xlength}$ 。

如果起点速度等于最大速度,终点速度小于最大速度时,则:

$$\text{line1.speed_up} = 0;$$

$\text{line1.speed_down} = \text{Max_down2_VA}(\text{line1.end});$ (最高速度到终点速度需要的脉冲数)

$\text{line1.speed_yunsu} = \text{line1.Xlength} - \text{Max_down2_VA}(\text{line1.end});$

如果起点速度小于最大速度,终点速度等于最大速度时,则:

```
line1.speed_up=VA_up2_Max(line1.begin);(从起始速度到最高速度需要的脉冲数)
```

```
line1.speed_down=0;
```

```
line1.speed_yunsu =line1.Xlength -VA_up2_Max(line1.begin);
```

如果两端的的速度均小于最大速度时:

当主轴脉冲数大于等于起点加速到终点速度需要的脉冲数加上从最大速度降到终点速度脉冲数,则运动脉冲的分配如下:

```
line1.speed_up=VA_up2_Max(line1.begin);
```

```
line1.speed_down =Max_down2_VA (line1.end);
```

```
line1.speed_yunsu=line1.Xlength-VA_up2_Max(line1.begin)-Max_down2_VA(line1.end);
```

当主轴脉冲数小于起点加速到终点的速度需要的脉冲数加上从最高速度降到结束点速度脉冲数;若开始的速度小于末端速度则只有加速脉冲(等于主轴脉冲);若开始的速度大于末端速度则只有减速脉冲;若开始的速度等于于末端速度则只有匀速脉冲。

以 ARM Cortex-M3 运动控制器替代传统运动控制卡的运动控制系统,降低了控制系统的成本,同时开阔了 ARM Cortex-M3 在运动控制领域的应用。本文提供的方法已经成功应用于数控冲孔机的运动控制,实现了数控冲孔机的预定功能,充分证明了 ARM Cortex-M3

实现优秀运动控制器的可行性。

参考文献

- [1] 郑魁敬,高建设.运动控制技术及工程实践[M].北京:中国电力出版社,2009.
- [2] 黄阳明,郑力新.基于 LPC2294 和 MCX314AS 的经济型数控系统[J].微型机与应用,2010,29(8):83-85.
- [3] 金振林,黄大贵.基于 ARM 的高精高速运动控制[J].制造技术与机床,2007(5):42-44.
- [4] 彭刚,秦志强.基于 ARM Cortex-M3 的 STM32 系列嵌入式微控制器应用实践[M].北京:电子工业出版社,2011.
- [5] 庞牧野.嵌入式数控系统连续轨迹控制算法的研究与实现[D].成都:电子科技大学,2011.
- [6] 郑灼,李兴根.单片机控制的步进电机升降频规律与实现[J].微电机,1999(4).
- [7] 刘青山,高霖.基于运动控制卡的 PC 数控进给速度前瞻控制[J].机械科学与技术,2009(9).

(收稿日期:2012-06-06)

作者简介:

朱阁,女,1986年生,硕士研究生,主要研究方向:电气工程。

郑力新,男,1967年生,博士,主要研究方向:机器视觉与智能控制。

林晓海,男,1987年生,硕士研究生,主要研究方向:电工理论与新技术。