A-PDF Split DEMO : Purchase from www.A-PDF.com to remove the watermark

DOI:10.3969/j.issn.1001-4551.2013.05.009

基于Matlab/Simulink开发面向DSP的铣刀开槽插补算法*

黄少青,郑仲谦,李 迪,王世勇
(华南理工大学 机械与汽车工程学院,广东 广州 510640)

摘要:针对在DSP中直接编写和调试插补算法代码以及验证其正确性的种种困难,提出了采用代码混编的方式开发面向DSP的铣刀开槽插补算法。利用Matlab/Simulink工具对插补算法进行了建模,在算法设计阶段对插补运动的轨迹和速度规划进行了仿真;实现预想的设计效果后,利用Matlab的RTW工具自动生成了面向DSP的实时C代码,并将实时代码移植到DSP中,实现了从Matlab代码到DSP代码的无缝过渡。PCB微型铣刀加工螺旋槽过程中需要通过两轴联动实现插补,在一轴旋转的同时另一轴实现导程方向的进给可以插补出螺旋轨迹。通过将该插补算法进行简单的变换利用便可以应用于该PCB微型铣刀的螺旋槽加工过程中,最后对加工出来的铣刀进行了检测,铣刀的螺旋槽的形状和尺寸精度能够满足加工要求。实验结果表明,利用代码混编的方式可以解决直接在DSP中编写插补算法的困难,具有较高的实际应用价值。

关键词:印刷电路板;插补算法;DSP;Matlab;Simulink;建模与仿真;自动代码生成
中图法分类号:TP273;TH39;TH164 **文献标识码**:A **文章编号**:1001-4551(2013)05-0552-05

Cutter slotted interpolation algorithm for DSP based on Matlab / Simulink

HUANG Shao-qing, ZHENG Zhong-qian, LI Di, WANG Shi-yong
(School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510640, China)

Abstract: Aiming at the difficulties of editing, debugging the interpolation code directly in DSP and confirm correctness of the code, a method to develop the cutter slotted interpolation algorithm for DSP by using code mixed programming was proposed. Firstly, the interpolation algorithm was modeled by using Matlab/Simulink tool, The planning of locus and velocity of the interpolation motion was simulated while the algorithm was designed. After the desired effect was achieved, the real-time code was generated and transplanted to DSP automatically through RTW, so the seamless transition from Matlab code to DSP code was realized. The process of grinding the spiral groove of PCB miniature milling cutters needs two linked shafts to implement interpolation, and the spiral trajectory was interpolated when one axis rotates and the other feeds along the lead direction at the same time. The algorithm was utilized in the process of grinding the spiral groove by a simple transformation. After the milling cutter was produced, its shape and size were tested and their precision can meet the processing requirements. The experimental results show that the method can solve the difficulties of editing and debugging the interpolation code directly in DSP. It has a high practical value.

Key words: printed circuit board(PCB); interpolation algorithm; digital signal processing(DSP); Matlab; Simulink; modeling and emulation; automatic code generation

收稿日期:2012-12-05
基金项目:国家高技术研究发展计划("863"计划)资助项目(2011AA04A104,2012AA040909);广东省教育部产学研结合资助项目(2011A090200054)
作者简介:黄少青(1987-),男,广东揭阳人,主要从事高性能嵌入式控制系统方面的研究. E-mail:Darren_SCUT@163.com

0　引　言

目前，国内外对用于PCB钻孔的微型铣刀的需求量剧增，自动化磨槽设备应运而生。本研究提到的磨槽机是采用DSP+FPGA的控制结构并嵌入软PLC内核，利用PLC实现自动化生产。加工螺旋槽实际上只需要两个轴联动便可以插补出螺旋轨迹，插补算法在DSP调试开发中，一般不能使用程序调试的断点调试工具，这是因为断点调试只能用于修正程序的逻辑错误，不能获得连续时间上的信息，而插补算法设计一般需要连续时间上的信息用于分析。所以嵌入式系统应用程序的设计和调试异常困难，许多与应用平台无关的算法模块的设计调试需要等到硬件平台和系统平台搭建完成后才能开始设计或验证，而且后续的设计调试仍然十分繁琐，开发效率极低[1]。

针对该问题，本研究提出采用Matlab/Simulink工具对算法进行建模仿真，验证无误后利用RTW工具自动生成C实时代码移植到DSP中，通过这种方法可以大大减少开发者的工作量，提高效率，缩短开发周期。

1　DSP插补代码开发流程

基于Matlab/Simulink平台的DSP插补代码开发流程图如图1所示。

首先，本研究根据设计思路在Simulink搭建好模型并进行仿真验证，直至仿真结果符合设计要求，然后利用Matlab提供的工具Real-Time Workshop(RTW)编译，自动生成面向TI编译器的代码，并进一步完成代码的编译、链接和下载，最后在搭建好的机床上生产加工，观察产品的形状及检查各项指标验证该方法的可行性。从整个过程来看，开发者只需在Simulink图形化建模和仿真环境下搭建简单的模型，利用已有的工具实现全自动的代码生成，无需编写一行代码，便可以得到准确并高度优化的DSP代码。

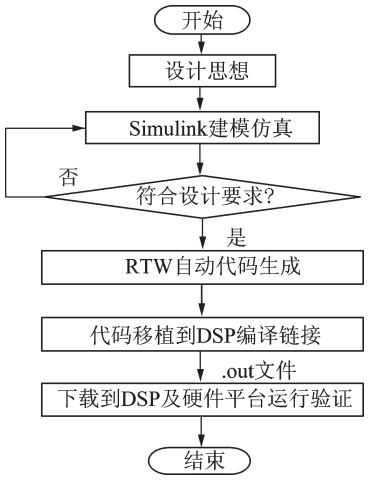

图1　DSP代码开发流程

2　插补算法Simulink建模与仿真验证

Simulink动态建模与仿真系统具有丰富而功能强大的器件库，提供了实时方便的仿真手段。Simulink是Matlab提供的实现动态系统建模与仿真的一个软件包。它让用户把注意力从编程转向模型的构造，为

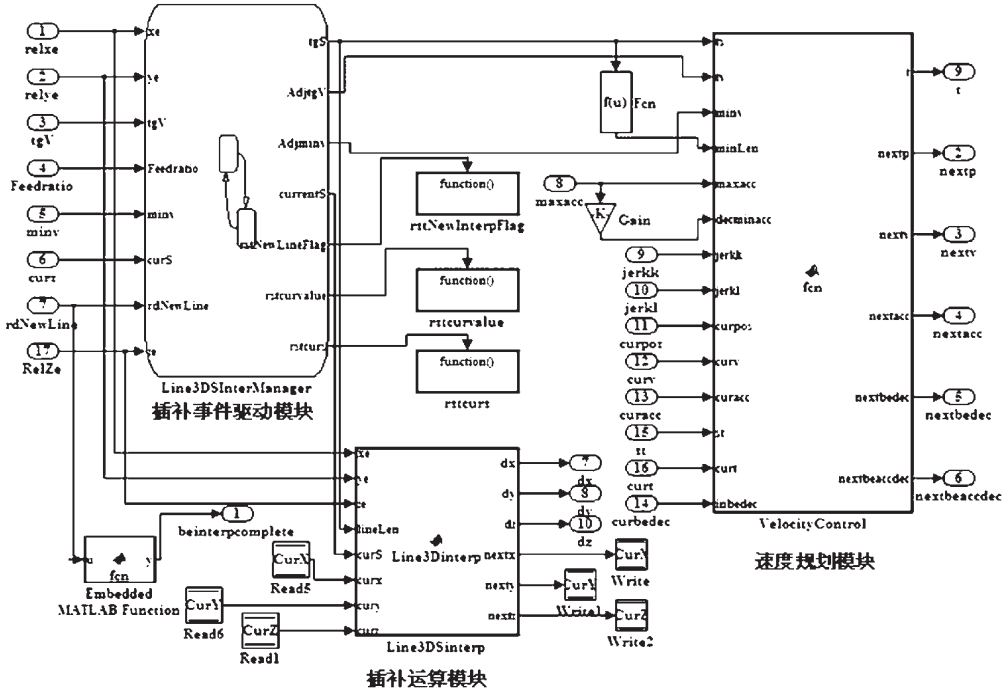

图2　运动控制直线插补模块

用户省去了许多重复的代码编写工作,不必一步步从最底层开始编起[2]。交互式的开发环境为开发人员提供了更快捷、直接明了的方式,而且用户可以立即看到系统的仿真结果[3]。

为了使代码更具通用性,本研究编写的是三轴插补的算法,当需要两轴插补时,只需令其中一轴的插补距离为零即可。

2.1 系统建模

运动控制直线插补的功能模块图如图2所示。该模块主要分为三大功能模块:插补事件驱动模块、速度规划模块和插补运算模块[4]。

插补事件驱动模块用来仿真插补运动时的周期调用功能是根据Stateflow有限状态机实现的。Stateflow生成的监控逻辑可以直接嵌入到Simulink模型下,两者之间能够实现无缝连接。所谓有限状态机是指系统中存在可数的状态,在某些事件发生时,系统从一个状态转换成另一个状态,故有限状态机又称为事件驱动的系统。在有限状态机的描述中,研究者可以设计出由一种状态转换至另一种状态的条件,并将每对可转换的状态均设计出状态迁移的事件,从而构造出状态迁移图[5]。

本研究中插补事件驱动功能模块图如图3所示。

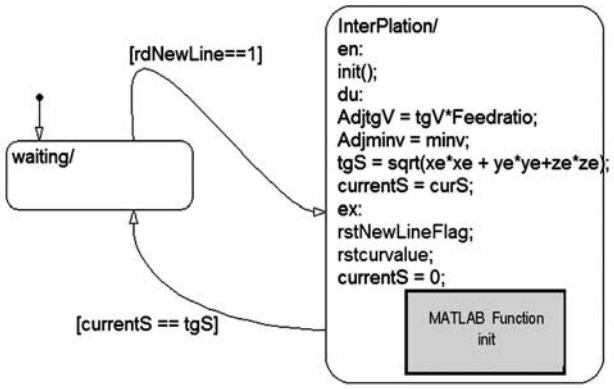

图3 直线插补事件驱动模块

刚进入驱动模块时事件处于wait状态等待,直至外部输入信号rdNewLine的值变为1时实现状态的跳转,进入到插补状态InterPlation。第一次进入到插补状态的时候本研究调用init函数,之后每次进入都会执行du:段下的代码,直到插补结束条件即当前位置等于目标位置成立时退出插补状态,重新回到wait状态等待。

速度规划模块和插补模块使用Embedded MATLAB Function模块,用户采用M代码形式编写自己的算法代码,对输入参数进行一系列相关运算处理后再输出,整个过程需要写代码的工作就是在这个环节中完成的。通过这种形式的连接,研究者自己编写的M代码就能和Simulink中模型图联系起来,最后利用RTW工具将M代码转换为C实时代码[6]。本研究采用先进行速度规划再进行插补运算的形式编写插补运动模块。速度规划模块实现S形加减速运动控制[7],这里不详述。插补运算模块根据规划模块得出的下一个插补位置计算各个方向轴周期输出的脉冲增量,完成一次插补。整个模型的详细过程如下:

(1)以当前点为起始点,输入插补目标位置点坐标(X_e, Y_e,Z_e)、起始速度、加速度、目标位置和当前位置等参数后进入插补事件驱动模块。

(2)进入事件驱动模块后判断是否进行状态迁移并进行计算,输出目标位置tgS以及设置现在的位置。插补等待状态激活后跳转到插补计算状态,输出相关信号。

(3)进入速度规划模块,速度规划模块根据给定参数计算出速度、加速度等参数,并计算得到下一个插补位置。

(4)进入插补运算模块,本研究根据以上各步骤得到的结果计算出各个方向轴该周期输出的脉冲增量dx、dy和dz,完成一次插补计算过程,存储新的位置作为当前所在位置。

(5)由速度规划模块和插补运算模块输出的参数作为输入信号再次输入到事件驱动模块中,判断是否到达目标位置;如果还没到达则继续进行第(2~4)步,否则,从插补状态跳转到等待状态。

2.2 功能仿真

本研究利用Matlab/Simulink进行3D直线插补仿真,验证算法运算过程的正确性。调试过程中研究者需要在模型中添加显示模块和数据存储模块,在算法运行过程中对运动参数的变化进行记录,将需要观察的变量利用To Workspace导入到Matlab工作空间中,最后在Matlab的Command窗口利用plot3和plot函数将需要观察分析的数据进行绘图,通过查看图形评估算法,直到算法满足应用要求。

进行3D直线插补仿真。本研究以脉冲量为长度单位,插补周期时间T,设定插补最大速度为300 pulse/T,起始速度为2 pulse/T,最大加速度为10 pulse/T^2,加加速度和减加速度分别设为0.1 pulse/T^2,直线插补的目标位置为(40 000,50 000,60 000),仿真结果如图4所示。由图4可见,运动轨迹的确为直线,且停止位置确实是所要求的目标位置。

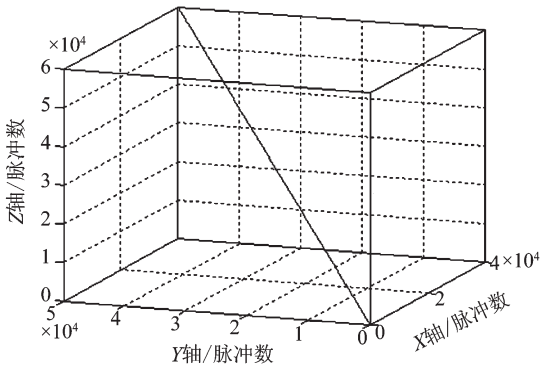

图4　三轴直线插补仿真图

另外，本研究对3D插补中每个坐标轴在每个周期内发送的脉冲即各个轴的速度进行记录，并导入到Matlab工作空间中，利用plot函数进行绘图，各个轴的速度变化如图5所示。

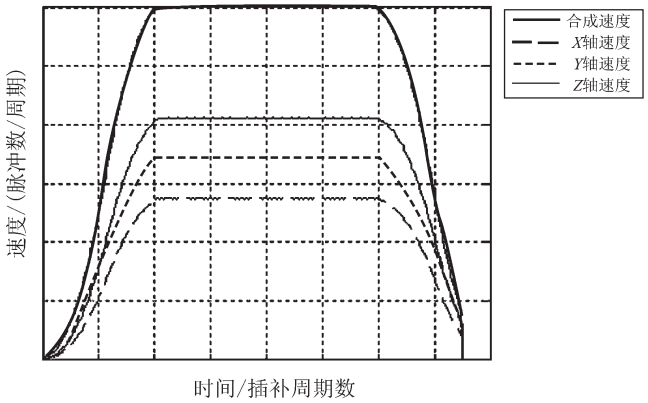

图5　三轴插补各轴的速度曲线

由图5可见各个轴的速度关系，每个轴在运动过程中都有明显的加、减速过程，并且形状为S形，即实现了S形速度规划的目标，它们的合成速度即为设定的插补速度。需要注意的是，如果研究者希望得到比较理想的S形速度曲线，则需要对插补速度、加减速、加加速度以及减加速度等参数进行合理的设置并不断地加以尝试，在实际加工过程中也要考虑这些参数设置的合适与否，以免机床产生振动，影响加工精度。本研究中需要用到的是2D直线插补，2D直线插补是3D直线插补的一个特殊情况，只需将一个目标轴位置设定为零即可实现2D直线插补。

至此完成直线插补的仿真实验，由仿真结果可知该直线插补算法是满足使用要求的。

3　利用RTW生成实时C代码

RTW是和Matlab、Simulink一起使用的一个工具，它可以直接从Simulink模型生成代码并且自动建立可以在不同环境下运行的程序，这些环境包括实时系统和单机仿真。RTW能够应用的场合十分广泛。本研究利用RTW快速原型工具自动生成准确并高度优化的DSP可执行代码，大大缩短了系统的开发周期[8]。模型建好后，研究者打开“Configuration Parameters”属性页进行代码自动生成过程中一些参数的设置。

本研究所生成的代码的目标系统是型号为C6000系列的DSP板，选择正确的硬件平台有利于代码的优化及对运行环境的匹配，因此在Hardware Implementation进行硬件的选择，本研究中选择只生成源代码，所设置的参数值如表1所示，build之后便可以在当前目录下看到所生成的源代码了[9]。

表1　参数设置值

选项	修改项	修改值
Solver	Stop time	inf
	Type	Fixed-step
	Solver	discrete
Hardware	Device vendor	Texas Instrument
	Device Type	C6500
Real-Time Workshop	System target file	ert.tlc
	Make command	make_rtw

4　代码移植到DSP中

在编译后可以看到众多的H文件和C文件，研究者将文件中的函数移植到DSP中，对模型中的输入信号进行初始化，对相关判断条件进行对接就能实现系统模型的算法功能。自动生成的代码中有两个重要的函数，即Line3DInterp_initialize和Line3DInterp_step，Line3DInterp_initialize函数完成插补前的参数的初始化设置，Line3DInterp_step函数进行插补运算得出每个方向轴本周期的脉冲增量，发给伺服驱动器，从而实现电机转动完成一次插补动作。

本研究在DSP BIOS中为插补模块创建一个周期模块对象Line3d_Interpo，当需要插补时先调用Line3DInterp_initialize函数进行插补参数初始化，接着调用PRD_start(&Line3d_Interpo)函数开始该周期模块对象，PRD开始计数后该周期模块对象周期性执行三轴插补函数TSK_Line3D_Interp()，如果插补没结束，该函数调用Line3DInterp_step函数计算每个轴的插补增量并将其发送出去，并进行其他标志位的标记，当插补结束后调用PRD_stop(&Line3d_Interpo)函数停止该周期模块对象[10-11]。

本研究对Matlab自动生成的代码进行简单的函数封装便可以实现将代码移植到DSP中，使其成为一个子函数，之后用CCS进行编译链接并下载到目标DSP中进行实物仿真验证。

5 实际生产加工验证

由于本研究加工的是螺旋槽，而前面开发的是三轴插补算法，笔者将该算法应用到螺旋槽加工中还需要进行参数的计算和转换。转换过程如下：

$$Zlength = Zlength \times Zgearratio \tag{1}$$

$$Lead = \pi \times D \times \tan(\pi \times (\frac{90-\theta}{180})) \times Zgearratio \tag{2}$$

$$Blength = \frac{Zlength}{Lead} \times 360 \times Bgearratio \tag{3}$$

式中：*Zlength*—加工螺旋槽的沟长，所在运动轴为*Z*轴；*Zgearratio*—*Z*轴的齿数比；*Lead*—螺旋槽的导程；*D*—所加工毛胚针的外径；θ—螺旋槽的螺旋角；*Blength*—旋转轴的转动量，旋转轴所在轴为*B*轴；*Bgearratio*—*B*轴的齿数比。

开始加工前，本研究根据输入的沟长、螺旋角等参数，通过以上公式计算得到两个轴的移动量，一个为沟长，即导程方向的移动距离，一个旋转轴的转动量，将这两个参数输入给三轴直线插补函数，并将第三轴的距离设为零，这样插补出来的形状便为加工所要求的螺旋槽。

加工出来的铣刀的效果如图6所示。由图6可见，事实上本研究确实联动插补出了螺旋槽，且通过检测可知其尺寸形状和精度是满足要求的。

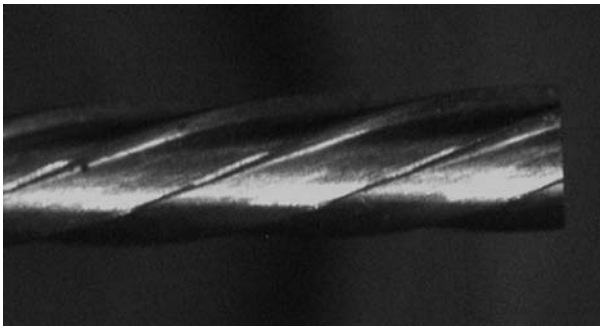

图6 加工出来的成品针

6 结束语

本研究结合Simulink和RTW工具的运用，采用代码混编的方式成功开发出面向DSP的铣刀开槽插补算法。在算法调试阶段，笔者利用Simulink工具强大的仿真功能，得以直观地观察插补轨迹图和速度规划曲线；在代码移植阶段，利用RTW工具自动生成DSP的实时C代码，使代码的编写变得更加简单快捷，提高了代码编写的效率；最后将该方法应用于铣刀开槽的实际生产加工中。

实际加工结果表明，该方法能够较好地解决直接在DSP中编写和调试插补算法的困难，所运用的方法具有较高的实际应用价值。

参考文献(References)：

[1] 范文康. 基于DSP+FPGA的高性能数控系统软硬件设计及实现[D]. 广州：华南理工大学机械与汽车工程学院，2009.

[2] 范影乐，杨胜天，李 轶. MATLAB仿真与应用详解[M]. 北京：人民邮电出版社，2001.

[3] 刘保柱，苏彦华，张宏林. MATLAB 7.0从入门到精通[M]. 北京：人民邮电出版社，2006.

[4] 吴基斌. 运动控制关键算法及嵌入式实现研究[D]. 广州：华南理工大学机械与汽车工程学院，2010

[5] 贾秋玲，袁冬莉，栾云凤. 基于MATLAB7.x/Simulink/Stateflow系统仿真分析及设计[M]. 西安：西北工业大学出版社，2006.

[6] 田 伟，熊晋魁. Simulink模型的C/C++代码实现[J]. 应用科技，2004，31(11)：16-18.

[7] CAO Yun-an，CHEN You-dong，WEI Hong-xing，et al. The algorithm of former s-shape acceleration/deceleration in cnc system[C] // 8th International Conference on Progress of Machining Technology，ICPMT2006，Matsue，Japan，2006. Kagamiyama：Hiroshima University，2006：165-168.

[8] 李 强，王民刚，杨 尧. 快速原型中Simulink模型的代码自动生成[J]. 电子测量技术，2009，32(2)：28-31

[9] 任传俊，蒋志文. Real-TimeWorkshop实时仿真研究与应用[J]. 计算机仿真，2007，24(8)：268-271.

[10] 徐永康，张 雷. 基于Matlab空间四连杆引纬机构运动仿真[J]. 轻工机械，2012，30(3)：17-21.

[11] 冯寿廷. 构建化的运动控制系统设计-实时性分析以及优化设计方法研究[D]. 广州：华南理工大学机械与汽车工程学院，2007.

[编辑：罗向阳]