

Matlab 环境下调用通用 DLL 的方法及应用

胡进 姚纯 苏敏

(苏州大学机电工程学院 江苏 苏州 215006)

摘要 介绍了一种在 Matlab 运行环境下直接调用通用 DLL 的方法。首先阐述了具体实现步骤,然后指出其适用场合,并给出了开发实例——针对开放式数控系统的加工监视软件开发。最后,总结了该方法与其它混合编程方法相比的优缺点。

关键词 Matlab 混合编程 动态链接库 开放式数控系统

CALLING GENERIC DLL FROM INSIDE THE MATLAB ENVIRONMENT

Hu Jin Yao Chun Su Min

(Mechanic and Electronic Engineering College, Soochow University, Suzhou 215006, Jiangsu, China)

Abstract A method for direct calling functions defined in the standard DLL from inside the Matlab environment is introduced. Main procedures of the approach are presented. Based on the hardware-access feature of DLL files, the feasibility of developing software to communicate between PC and some hardwares is shown by an example. The advantages and limitations of the method are summarized and compared to those of the other mixed programming methods.

Keywords Matlab Mixed programming DLL Open CNC system

0 引言

目前 Matlab 已在科学计算和软件编程等领域获得了广泛应用,与其它高级语言比较,其主要优点是代码效率高、库函数功能丰富,而缺点是程序运行效率低、用户界面相对单调。为了扬长避短,Matlab 提供了与高级语言的混合编程途径:①在 C 语言等源代码中调用 Matlab C/C++ 数学函数库,在 VC 等软件环境中编译;②在 Matlab 环境下,利用 MCC 命令实现 M 文件和 C 文件混合编译。两者均可生成独立执行程序。

最近 Matlab 编程又有了新的功能扩充:可以在 M 代码中加载高级语言编译生成的 DLL,进而调用其中的库函数。

将此功能应用到工程中,可简捷地实现 PC 与硬件通信并将数据可视化。在影像处理和系统控制等领域,已有许多公司和研究机构应用此法进行软件开发。例如德国 PCO 影像公司,就为其高速摄影系统提供了 M 代码形式的配套软件。

1 编程设置

在 MathWorks 主站下载扩展包 genericDll.exe,释放到 Matlab 的安装目录下。这一过程会在 toolbox\matlab\general 目录下添加 loadlibrary 等 8 个功能文件,并在 extern\examples 目录下添加示例程序 SHRLIB。

在 Matlab 命令行输入“rehash toolboxcache”,回车运行即完成了编程设置。另外,在编程之前必须获得 DLL 相应的头文件,其中包含必要的库函数和常量的定义。

2 相关函数介绍

在 M 代码中调用 DLL 主要通过新增的 8 个功能函数,下面

采用程序举例形式对其介绍。

2.1 基本功能函数

(1) loadlibrary

```
loadlibrary('LibName','HeadFile');
```

根据头文件 HeadFile.h 中的函数定义加载 LibName.dll。

此外,loadlibrary 还接受 'addheader' 和 'alias' 等选项,其作用分别为附加其它头文件、对已加载的 DLL 重命名等。

(2) calllib

```
arg1 = 20; arg2 = 35; ... ; argN = 64;
```

```
[x1, ..., xN] = calllib('LibName','Func', arg1, ..., argN)
```

加载 LibName.dll 后,向其中的库函数 Func 传递输入变量 arg1 ~ argN,并获得返回值 x1 ~ xN。

(3) unloadlibrary

```
unloadlibrary('LibName')
```

从内存中卸载相应 DLL。

2.2 查询函数

libisloaded 用以查询某个 DLL 是否已被加载。libmethods 和 libmethodsview 用以查询已加载 DLL 中的库函数信息,包括所有库函数名称、输入和输出变量数据类型。

2.3 类型匹配函数

DLL 中库函数的输入和输出变量通常采用 C 语言数据类型,Matlab 数据类型与之相比有较大区别,因此在此 M 代码中向 DLL 库函数传递变量之前,须保证数据类型相互匹配。

对于基本数据类型(表 1)和高级类型(表 2),匹配工作都

可交由 Matlab 自动完成。也即,可以按 Matlab 习惯方式定义变量、赋值,然后直接传递给库函数。注:表 1 和表 2 中的 [size] 代表数据位数,如 8、16 和 32。

但是对于高级数据类型,建议采用类型匹配函数将 Matlab 变量转换为与 C 数据类型匹配的变量,然后再传递给 DLL 中的库函数。被库函数多次使用的变量更应如此,否则 Matlab 将多次对其进行不必要的类型转换,耗费额外的内存空间和时间。

表 1 基本数据类型匹配

C 语言类型	(unsigned)char, byte, Short, ...	float	double	char *
Matlab 类型	(u)int[SIZE]	single	double	string

表 2 高级数据类型匹配

	指针			结构
C 语言类型	short *, int *, long *, ...	float *	double *	struct
Matlab 类型	int[SIZE]Ref	singleRef	doubleRef	libstruct

(1) libpointer

libpointer 用于实现各种指针类型的匹配,支持除 void * * 外的多重指针。例如,在 C 头文件和 DLL 中定义有指针函数:

```
int GetElement(int * Array, short Num) { ... }
```

可在 M 代码对其作如下调用:

```
num = 2; ary = [ 4, 6, 5 ]; % 赋初值
```

```
c_ary = libpointer('int32Ref', ary); % 创建匹配变量
```

```
elemt = calllib('LibName', 'GetElement', c_ary, num);
```

运算结果: elemt = 6。

(2) libstruct

libstruct 用于实现结构类型的匹配,尚不支持多重结构。

例如,在 C 头文件和 DLL 中定义有结构 Student:

```
struct Student { short Age; short Height; }
```

可在 M 文件对其作如下数据类型匹配:

```
st.Age = 18; st.Height = 168; c_st = libstruct('Student', st);
```

(3) 类型匹配的补充说明

1. Matlab 并不支持真正的指针寻址,其对指针和结构类型的匹配操作是通过“类”(class)间接实现的。2. 此法不支持 Windows API 类型声明,例如 C 头文件中的 MMRESULT 等字段须删除,而 PWCHAR 类型须改为 char *, 否则 DLL 无法被 Matlab 加载。3. DLL 被加载后,库函数输入和输出变量的数据类型和排列形式可能与头文件中的原始定义稍有区别,必要时须借助前述查询函数。

3 编程方法的应用

以 DLL 形式发布的应用程序,具有程序结构模块化、源码保密性好和节省内存资源等优点,因而被广泛采用。文中介绍的混合编程方法,使得 Matlab 能够轻松地利用已有的 DLL 形式的程序资源,并且与其它编程语言进行交互。

此外,设备驱动和硬件通信程序广泛采用 DLL 形式,利用此法通过调用相关 DLL 能够实现 Matlab 程序对硬件的控制。

据此,作者开发了针对开放式数控系统的加工监视软件。通信对象为 ISA 板卡 PMAC2-PC。该板卡通过板载 DSP 自动完成电机伺服更新,并对上位机提供 Pcomm32.dll(由厂商采用 C 语言针对板卡开发)以读取数据和控制电机。

编程思路:在 M 代码中,定义带有选项 'action' 的递归函数 monitor,利用 Matlab 定时器定期调用 DLL 库函数以读取各轴位置,并更新显示加工轨迹。主要程序代码及注释如下:

```
function monitor( action )
switch action,
case 'init', % 初始化
% 创建主窗口,并定义退出函数
fig = figure('deletFcn','monitor("close")');
axes('Units','normalized','Position',[0 0 1 1]); % 建立绘图区
[ 在此处创建菜单和文本框等控件 ]
% 创建定时器,周期为 0.05s
Timer_1 = timer('timerFcn','monitor("timer")','period',0.05);
% 加载 Pcomm32.dll,并重命名为 'pm'
loadlibrary('Pcomm32','pmaeu.h','alias','pm');
calllib('pm','OpenPmacDevice', Crd); % Crd 是板卡编号
% 开启“实时更新”,每 10 个伺服周期更新电机 1~5
calllib('pm','PmacDPRRealTimeEx', Crd,5,10,ON);
case 'timer', % 定时读取各轴位置,重绘三维轨迹
calllib('pm','PmacDPRUpdateRealTime', Crd); % 更新电机
xP = calllib('pm','PmacDPRPosition', Crd,1,1667);
yP = calllib('pm','PmacDPRPosition', Crd,2,1667);
.....
case 'close', % 退出
stop(Timer_1); % 停止定时器
set( fig, 'deletFcn',''); % 改退出函数为空,避免死循环
% 关闭“实时更新”
calllib('pm','PmacDPRRealTimeEx', Crd,5,0,OFF);
calllib('pm','ClosePmacDevice', Crd);
unloadlibrary('pm');
end;
```

软件的运行界面如图 1 所示。由于 M 代码可读性强、代码效率高,采用这种混合编程方法大大缩短了软件的开发时间。另外,Matlab 中三维图形编程较简单,减少了编程工作量。

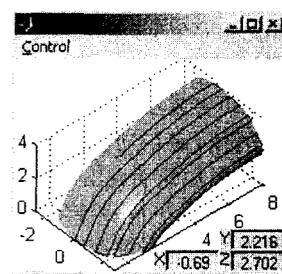


图 1 开放式数控系统加工监视软件界面

4 总结

文中介绍的方法实现了 Matlab 与其它高级语言的结合。相对其它混合编程方法,其编译设置非常简单,而且无需在 Matlab 或其它高级语言中引入中间数据类型。

此法兼顾了运行效率和易于编程。可利用运行效率较高的 VC 等高级语言开发核心算法或者硬件通信程序,编译生成 DLL,然后再用简单灵活的 M 代码开发上层软件对其调用。

其主要缺点是,程序不能编译生成独立执行程序而脱离 Matlab 环境运行,运行效率要低于 C/C++ 等高级语言。同时,程序运行时,Matlab 软件本身也占用一定的内存资源。因而此法不适用于对运行效率和实时性要求很高的场合。(下转第 180 页)

分别为沿 v, u 方向上的扰动次数, 对表面进行正弦干扰。经过反复的试凑, 测得当 $km = r/5$, 即干扰振幅大小为体积大小的 $1/5$ 时, 表面的凹凸幅度与体积的比例适中; 当 $kp = 2, kq = 4$, 即在 v 方向上进行两次正弦扰动以及在 u 方向干扰四次, 同时结合 $(|u| - \pi/2)$ 的 u 方向干扰, 能够得到真实的随机凹凸表面效果。图 1(b) 中的土豆数学模型为:

$$Q'(u, v) = Q(u, v) + r/5 * (|u| - \pi/2) * \sin(2 * v) * \sin(4 * u) N$$

(2) 把苹果作为例子, 其最小外接曲面也近似椭球。给出椭球面 $Q(u, v)$ 的数学模型:

$$\begin{cases} x = 0.8r * \cos(u) \cos(v) \\ y = r * \sin(u) \\ z = r * \cos(u) \sin(v) \end{cases}$$

其中 $u \in [-\pi/2, \pi/2], v \in [0, 2\pi]$, 效果如图 2(a) 所示。

因为苹果只是在两端处才向内凹, 故只需对其表面沿 u 参数方向干扰即可。给出干扰函数为:

$$P(u, v) = -km^{2u} - kj^{-2u}$$

式中 km, kj 分别用来控制沿 u 方向上的两端凹陷的干扰程度。在本实验中取 $km = 4, kj = 3$, 使得两端的凹陷幅度与体积之比显得恰当, 并且使上端比下端的凹陷幅度略大, 从而达到真实仿照的效果。扰动后的数学模型效果如图 2(b) 所示。



图 1 土豆造型过程图



图 2 苹果造型过程图

2.2 动态生长造型

在计算机中模拟作物生长, 已成为图形学研究的重要方向。如果通过对所有控制点连续移动来使曲线、曲面的形状不断发生改变, 从而达到生长的效果, 那算法的复杂度高且控制点的位置变化难以把握。

利用 Blinn 算法的思想首先记录分析各个生长阶段的数据及成长特点, 建立原图形的数学模型, 对其加以适当的扰动; 再根据其生长特点与规律在时间函数下适当改变数学模型及扰动模型中的参数, 使之产生变化达到生长的效果。

以良种南瓜生长后期(第 32 至 50 天)为例。该阶段南瓜的体积增长幅度较小, 外形刚开始类似为扁球体, 同时瓜瓣开始向外凸出并越来越明显, 致使两顶端呈凹陷状。

给出南瓜生长后期 $Q(u, v)$ 的数学模型:

$$\begin{cases} x = k * (dm + r * \sin(u)) * \cos(v) \\ y = (dm + r * \sin(u)) * \sin(v) \\ z = dm + r * \cos(u) \end{cases}$$

其中 $u \in [0, 2\pi], v \in [0, 2\pi]; r \in [h(32)/2, h(50)/2]$, 用来控制此阶段的高度增长 ($h(t)$ 表示第 t 天的高度); $dm \in [0, h(50)/2], k$ 从 $w(32)/h(32)$ 减少到 1 ($w(t)$ 表示第 t 天的宽度), 变量 dm 和 k 一同控制此阶段的宽度增长。在时间函数 $OnTimer()$ 下分别定义高度和宽度的日变化量, 令 $r = r + \Delta r, dm = dm + \Delta dm, k = k - \Delta k$ 。

此阶段瓜瓣逐渐向外凸出, 故沿 v 参数方向在表面各点的单位法向量上增加扰动函数:

$$P(u, v) = L * |\sin(km * v)|$$

其中, $L \in [0, h(50)/6]$, 表示南瓜瓣的突出振幅。 km 在此控制分瓣个数, 由于一般良种瓜有 10 片瓜瓣, 故在实验过程中取参数 $km = 5$ 。而要表现出凸出效果, 只要对扰动函数 $P(u, v)$ 中的变量 L 进行增加即可。在时间函数下定义 $L = L + \Delta L$, 其中 ΔL 为日平均增长量。在第 32 天时 L 取值为 0 表示刚开始出瓣(见图 3); 经测试当在第 50 天完全成熟时, L 取值为 $h(50)/6$ (即第 50 天高度的 $1/6$), 可以达到瓜瓣大小与南瓜体积的最佳比例。由于在生长后期, 良种瓜的体积与瓜瓣大小的日增长幅度基本不变, 则可以令 $\Delta L = (h(50)/6 - 0)/18$ 。根据 $Q'(u, v) = Q(u, v) + P(u, v)N$, 得到干扰后的数学模型效果如图 4 所示(该阶段南瓜表面由深绿色转变为成熟时的金黄色)。



图 3 刚出瓣时模型图



图 4 完全成熟时模型图

3 结 语

本文引入了一种基于 Blinn 算法思想的造型方法, 利用对原模型进行大幅度的适当扰动, 生成新的模型。该方法的关键点在于找到合适的干扰函数并确定其相关参数。目前为止主要是通过反复的实验以及有关经验来确定干扰函数及其参数。这种造型算法效率高, 具有良好的造型效果; 特别是针对瓜果模型及其生长模拟, 实现简单且对虚拟农作物提供了具有实际应用价值的研究方法。

参 考 文 献

- [1] 吴家铸, 党岗, 刘华峰, 等. 视景仿真技术及应用[M]. 西安电子科技大学出版社, 2001.
- [2] 和平鸽工作室. OpenGL 三维图形系统开发与实用技术[M]. 清华大学出版社, 重庆大学出版社, 2003.
- [3] 郭炎, 李保国. 虚拟植物的研究进展[J]. 科学通报, 2001, 46(4): 273-280.
- [4] 刘晓东, 楚丽平. 一个虚拟农作物实验平台的构架和实现[C]. 中国数字农业与农村信息化发展战略研讨会论文集, 中国农业出版社, 2003: 198-205.
- [5] 雷蕾, 郭新宇. 黄瓜果实的几何造型及可视化研究[J]. 计算机应用与软件, 2006, 23(5): 24-26.

(上接第 146 页)

参 考 文 献

- [1] The MathWorks Inc. Matlab Interface to Generic DLLs Manual [M]. 2002.
- [2] The MathWorks Inc. Is it possible to call a dynamic linked library from Matlab. <http://www.mathworks.com/support/solutions/data/33513.shtml>, 2003, 3.
- [3] 肖永韧, 解习农, 刘晓峰. VC 与 MATLAB 混合编程之 DLL 实现方法. 计算机工程与应用, 2001, 37(13): 174-176.
- [4] 张志涌, 刘瑞桢, 杨祖樱. 掌握和精通 MATLAB. 北京: 北京航空航天大学出版社, 1999.
- [5] The pco imaging company. Drivers and dlls which are needed for operation of pco cameras. http://www.pco.de/pco/php/support/drivers_1-en_01040202.html.