

文章编号:1671-7147(2003)05-0474-04

游戏杆在机器人虚拟示教系统中的应用编程

张爱红¹, 张秋菊²

(1. 无锡职业技术学院, 江苏 无锡 214073; 2. 江南大学 机械工程学院, 江苏 无锡 214063)

摘要: 分析了北通飞行摇杆 BTP2C313 的接口编程原理, 并将装置应用到 MOTOMAN 机器人虚拟示教系统中, 给出了验证实例, 实现了对机器人的虚拟示教等人机交互操作 1

关键词: 机器人; 虚拟现实; 游戏杆; 力反馈; 示教

中图分类号: TP 242

文献标识码: A

Application Programming of Joystick Used in Robot Virtual Teaching System

ZHANG AiZhong¹, ZHANG QiuJu²

(1. Wuxi Institute of Technology, Wuxi 214073, China; 2. School of Mechanical Engineering, Southern Yangtze University, Wuxi 214063, China)

Abstract: Programming theory for joystick BTP2C313 is presented in the paper. In MOTOMAN robot virtual teaching system, the joystick is used as an interaction device, through which the virtual teaching function can be realized.

Key words: robot; virtual reality; joystick; force feedback; teaching

工业机器人的示教方法分为在线示教和离线示教^[1], 以此为基础, 文献[2]介绍了 MOTOMAN 机器人微机虚拟示教系统的开发 1

在该系统中, 输入装置是重要的外围设备, 它的设计或选用可参考虚拟现实系统中的人机交互技术 1

目前应用于虚拟现实系统的人机交互设备主要有: 力反馈手控制器(游戏操纵杆)、数据手套、立体显示头盔、触感接口、力觉鼠标和力觉笔杆^[3]等 1 经比较, 作者采用市面上流行的力反馈手控制器(北通飞行摇杆 BTP2C313)作为输入装置, 通过 Visual C++ 接口编程, 实现了 MOTOMAN 机器人的虚拟示教等操作 1

1 游戏杆编程原理

北通飞行摇杆是一种适用于电脑游戏的游戏控制器, 在微软 Direct Input 或 Force Feedback API 510 以上版本支持的电脑游戏中, 可模拟出电脑震动摇杆的效果 1 在机器人虚拟示教系统中, 操作人员通过操纵游戏杆示教, 当虚拟机器人的关节角度超出设定的范围或直线示教过程中出现位姿奇异解时, 将有强烈的震动效果提示用户更改示教路径 1

1.1 游戏杆结构

BTP2C313 飞行摇杆为两轴(X轴、Y轴)游戏杆, 上面分布了 10 个按钮, 如图 1 所示 1 它有两个振动电机, 结合了数字、模拟两种控制方式, 安装时

收稿日期: 2003-04-22; 修订日期: 2003-10-221

作者简介: 张爱红(1971-), 男, 江苏盐城人, 工学硕士, 讲师 1

需要占用微机的并口、USB 口,以获取手柄、按钮的输入及电机的振动电源¹

图 1 北通飞行摇杆 BTP2C313 结构简图

Fig. 1 Sketch of BTP2C313 joystick

由于摇杆 BTP2C313 上共有 10 个有效按钮,因此除了用于控制机器人 6 个关节(或末端 6 个位姿分量)外还可定义按钮 10 为“示教”按钮、按钮 9 为“再现”按钮^{[2]1}

112 游戏杆编程

欲获取游戏杆数据,文中采用 joyGetPosEx 函数为核心的 Win32 API 编程,同时为了支持具有力反馈效果的游戏杆,增强输入操作的逼真度,作者在程序中采用了 DirectX 510 以上版本的 Direct2 Input 组件¹

DirectInput 与 DirectX 的其他模块一样,都基于 COM,其基本结构由支持 COM 界面的 DirectInput 对象和提供数据的外设对象组成。DirectInput 的 3 个对象分别为 DirectInput, DirectInputDevice, DirectInput2Effect1 其中 DirectInput 是一个高层对象,通过它可对输入设备进行基本的初始化和查找,并用于创建低层的 DirectInputDevice 对象,收集并设置设备状态信息的接口,最后创建 DirectInputEffect 对象,实现力反馈效果的启动、播放和停止¹

11211 游戏杆输入。

1) 创建 DirectInput 对象 为了创建 Direct2 Input 对象并得到其 IDirectInput 接口指针,在程序初始化阶段运用 DirectX 提供的助手函数 DirectInputCreate 创建并初始化 DirectInput 对象¹

函数原型为: HRESULT DirectInputCreate (HINSTANCE hinst, DWORD dwVersion, LPDIRECTINPUT 3 lpDirectInput, LPUNKNOWN punkOuter);

第一个参数是应用程序的实例;第二个参数是程序需要的 DirectInput 版本;第 3 个参数最为重要,是指向 IDirectInput 接口指针的地址,在程序中定义一个 LPDIRECTINPUT 类型的全局变量,并将其地址作为第 3 个参数传递给函数 Direct2 InputCreate¹ 第 4 个参数与 COM 中的聚合相关,程

序中置为 NULL¹

2) 使用 DirectInput 对象 拥有了 DirectInput 对象后,就可用其创建 DirectInputDevice 对象,以管理系统中特定的设备¹ 创建 DirectInputDevice 对象要使用函数 CreateDevice,它作为 IDirectInput 接口函数之一,需要所请求设备的 GUID,以返回 DirectInputDevice 对象的 IDirectInputDevice 接口指针¹

函数原型为: HRESULT CreateDevice (REFGUID rguid, LPDIRECTINPUTDEVICE 3 lpDirectInputDevice, LPUNKNOWN punkOuter);

在创建 DirectInputDevice 对象前需要该设备的 GUID,就游戏杆而言,只能通过列举方法获得¹

3) 列举游戏杆 列举设备需要调用 IDirect2 Input 接口函数 EnumDevices,原型为: HRESULT EnumDevices (DWORD dwDevType, LPDIENUM2 CALLBACK lpCallback, LPVOID pvRef, DWORD dwFlags);

第一个参数是指需列举的设备类型,对于游戏杆而言是指 DIDEVTYPE_JOYSTICK;第二个参数是一个回调函数;第 3 个参数是由程序定义的 32 位值,在回调函数调用时传入¹ 第 4 个参数是详细描述欲列举设备的标识(如 DIEDFL_ATTACHEDONLY, DIEDFL_FORCEFEEDBACK 等)¹ 当 EnumDevices 列举系统中的输入设备时,反复调用回调函数,而在回调函数中调用 CreateDevice 函数,直至创建满足需要的 DirectInputDevice 对象¹ 回调函数 BOOL CALLBACK EnumCallback (LPCDIDEVICEINSTANCE lpddi, LPVOID pvRef) 接受两个参数:第二个参数是程序定义传递给 EnumDevices 的 32 位值,第一个参数传递指向一个结构 DIDEVICEINSTANCE 的指针,设备 GUID 信息保存在成员 guidInstance 中¹

4) 使用 DirectInputDevice 对象 DirectInputDevice 对象的每个实例都与系统中的特定设备相关¹ 此对象具有对系统硬件更多的控制能力,可使 DirectX 实现预期功能¹ 因此,拥有 IDirectInputDevice 接口的指针后,需进一步使用 DirectInputDevice 对象,完成设备的设置,最终取得来自设备的数据¹

) 数据格式设置 通过调用接口成员函数 SetDataFormat 完成设备的数据格式设置¹ 设置数据格式包括轴信息、相对或绝对坐标信息等,这些细节通过 DIDATAFORMAT 结构传递给 SetDataFormat 函数¹ DirectInput 已定义了几个用于普通输入设备的 DIDATAFORMAT 结构变量: c_dfDIKeyboard, c_dfDIMouse, c_dfDIJoystick 和

c. `dfDIJoystick21` 设置普通力反馈游戏杆的数据格式,可调用 `pDIDeviceObject -> SetDataFormat (&c. dfDIJoystick)`,而 `pDIDeviceObject` 则是指向 `IdirectInputDevice` 接口的指针 1

)协作级别设置 设置好设备对象的数据格式后,则需调用接口成员函数 `SetCooperativeLevel` 设置设备的协作级别: `HRESULT SetCooperativeLevel (HWND hwnd, DWORD dwFlags)` 1 其中 `hwnd` 是程序的主窗口句柄,`dwFlags` 是以下值“或操作”的组合,即 `DISCL_BACKGROUND`、`DISCL_FOREGROUND`、`DISCL_EXCLUSIVE`、`DISCL_NONEXCLUSIVE` 1 为使程序在活动窗口中成为唯一允许访问该设备的进程,应选择 `DISCL_EXCLUSIVE | DISCL_FOREGROUND` 组合 1

)游戏杆属性设置 设备的属性设置包括 X 、 Y 轴范围以及游戏杆的中心点等细节 1 这项工作由 `SetProperty` 接口成员函数完成 1 为了设置设备的某个属性,首先要用相关的一些信息填写一个数据结构(如 `DIPROPDWORD`、`DIPROP RANGE` 等) 1 每一个结构的成员又都以 `DIPROPHEADER` 结构开始,填写其中需要改变的设置信息,然后填写上层结构中剩余部分,最后调用 `SetProperty` 函数: `HRESULT SetProperty (REFGUID rguidProp, LPCDIPROPHEADER pdiph)`;

参数 `rguidProp` 可以是预定义中的一个(`DIPROP_AUTOCENTER`、`DIPROP_RANGE` 等)、或是定义属性的 `GUID` 指针,参数 `pdiph` 为包含在上层结构中的 `DIPROPHEADER` 结构的地址 1

)获取游戏杆 为了获得对游戏杆的实际访问,获取游戏杆的数据,需要调用接口成员函数 `Acquire`,该函数不带任何参数 1 如果返回值为 `DIERR_INPUTLOST`,需要再次调用 1 在应用程序退出时,调用 `Unacquire` 函数释放设备 1

)获取来自游戏杆的数据 获得游戏杆之后,调用 `GetDeviceState` 接口函数轮流检测游戏杆输入的数据 1 由于并非所有的设备驱动程序在设备状态改变时都会通知 `DirectInput`,为此在调用检测函数之前,需要调用 `IdirectInputDevice2::Poll` 函数更新设备状态 1

结构 `DIJOYSTATE` 的成员包含了游戏操纵杆轴和按钮的输入状态,可通过如下程序检测 1

```
DIJOYSTATE js;
```

```
...
```

```
if ( js.lX > 0 && js.rgbButtons[0] & 0x80 ) // 检查 X 轴和按钮 1
```

```
...
```

```
if ( js.lY > 0 && js.rgbButtons[0] & 0x80 ) // 检查 Y 轴和按钮 1
```

```
...
```

条件语句 `if (js.lX > 0 && js.rgbButtons[0] & 0x80)` 将同时检查 X 轴和按钮 1 的状态,如果条件成立,将改变虚拟机器人的某一关节(或末端位姿)参数,然后调用重绘函数实时地更新虚拟机器人及其场景,完成人机交互操作 1

11212 力反馈的实现。

1) 创建力反馈设备 正常情况下,游戏杆输入编程中使用的接口为 `IdirectInputDevice1` 为了进行机器人虚拟示教,提高示教速度并增强使用者的临场感,力反馈功能的增加无疑是相当有意义的 1 为了支持查询设备和力反馈设备,需要将接口 `IdirectInputDevice` 升级到 `IdirectInputDevice2`,而新接口提供了原有接口的所有功能,得到新接口后再释放原有接口 1

```
// 返回新接口对象的指针 pDIDevice
```

```
pDIDeviceObject -> QueryInterface ( IID_ IdirectInputDevice2, (void**) (&pDIDevice));
```

```
// 释放原有接口指针:pDIDeviceObject
```

```
pDIDeviceObject -> Release();
```

设置协作级别、设置数据格式与上文介绍的相同,但是设置游戏杆属性却不一样 1 需要关闭设备,选取“自动中心”特性 1 程序如下:

```
DIPROPDWORD autoCenter;
```

```
autoCenter.diph.dwSize = sizeof(autoCenter);
```

```
autoCenter.diph.dwHeaderSize = sizeof
```

```
(DIPROPHEADER);
```

```
autoCenter.diph.dwObj = 0;
```

```
autoCenter.diph.dwHow = DIPH_DEVICE;
```

```
autoCenter.dwData = 0;
```

```
hresult = pDIDevice -> SetProperty (DIPROP_AUTOCENTER, &autoCenter.diph);
```

2) 创建力反馈效果 成功创建力反馈设备后,即可创建力反馈效果 1 为此,先创建 `DirectInputEffect` 对象实例 1 通过调用接口 `IdirectInputDevice2` 的成员函数 `CreateEffect` 完成 1 此函数需要效果的 `GUIDs`,以及指向 `DIEFFECT` 结构的指针,该结构中填写了效果细节 1 最后, `CreateEffect` 返回一个指向 `IdirectInputEffect` 接口的指针 `g_pEffect`,该指针的地址是 `CreateEffect` 的一个参数 1 本系统中使用的是恒力效果,编程时,可以省掉繁琐的效果列举 1

填写完 DIEFFECT 结构并调用 CreateEffect 函数,即获得指向 IDirectInputEffect 接口的指针 1

3) 播放并更改力反馈效果 IDirectInputEffect 接口成员函数 Start 和 Stop 可用于播放和停止力反馈效果 1 更改力反馈效果需调用 IDirectInputEffect 接口的 SetParameters 函数,程序调用为:

```
DIEFFECT eff;
LONG rgldirection[2] = {1, 1}; // 力反馈方向:45°
DICONSTANTFORCE cf;
// 设置 DICONSTANTFORCE 结构成员变量
cf.lMagnitude = magnitude;
// 设置 DIEFFECT 结构成员变量
eff.dwSize = sizeof(DIEFFECT);
eff.dwFlags = DIEFF_CARTESIAN | DIEFF_OBJECTOFFSETS;
eff.cAxes = 2;
eff.rgldirection = rgldirection;
eff.cbTypeSpecificParams = sizeof(DICONSTANTFORCE);
eff.lpvTypeSpecificParams = &cf;
g.pEffect->SetParameters(&eff, DIEP_DIRECTION |
DIEP_TYPESPECIFICPARAMS | DIEP_START);
```

SetParameters 函数的第二个参数反映了效果信息是如何被设置和处理装载的, DIEP_DIRECTION 标识位说明了将启用 DIEFFECT 结构的成员:cAxes 和 rgldirection 所定义的坐标轴数和反馈力方向;DIEP_TYPESPECIFICPARAMS 标识位说明了将启用 DIEFFECT 结构的成员:lpvTypeSpecificParams、cbTypeSpecificParams 所定义的效果类型的地址和大小;最后一个标识位 DIEP_START 意味着参数更新后,效果即被启动 1 若效果正在播放,那么将按新设的参数重新启动 1 因此通过设置不同的 DICONSTANTFORCE 结构成员 lMagnitude 可实现游戏杆力反馈效果的启动和停止 1 在关节示教过程中,如果某一关节角度超

参考文献:

- [1] 张爱红,张秋菊 1 机器人示教编程方法[J] 1 组合机床及自动化加工技术,2003,(4):47-491
- [2] 张爱红 1 机器人虚拟示教及远程控制研究:[硕士学位论文][D] 1 无锡:江南大学,20031
- [3] 殷跃红,尉忠信,黄晓曦 1 智能机器人系统力觉及力控制技术[M] 1 北京:国防工业出版社,20011
- [4] 王翠梅,王德强,李果 1 DirectX 6 多媒体技术参考手册[M] 1 成都:西南交通大学,19991
- [5] 赵春霞, Li Y F, 王树国,等 1 虚拟现实的发展及在机器人系统中的应用与研究[J] 1 机器人,1999,21(5):395-4001

(责任编辑:邢宝妹)

出设定范围,将有与角度成正比的数值赋予 lMagnitude,调用 SetParameters 函数,实施力反馈,提示用户更改示教路径;如果关节角度在设定范围内,lMagnitude 置 0,调用该函数停止力反馈 1

2 应用实例

按照游戏杆输入及力反馈的原理,作者在机器人虚拟示教系统源程序中,通过调用 DirectInput 组件接口函数并借助于游戏杆输入,实现了机器人对工作台上螺钉的抓取、移位、释放等,最终生成了机器人可执行的作业文件^{[2]1}

MOTOMAN 机器人的关节依次分别为腰关节(S 轴)、肩关节(L 轴)、肘关节(U 轴)以及腕关节(R 轴、B 轴、T 轴),末端位姿可以记为: X、Y、Z、R_x、R_y、R_z,本应用程序中建立起了“操纵杆、按钮”组合与机器人关节转轴 S、L、U、R、B、T 以及末端位姿 X、Y、Z、R_x、R_y、R_z 的对应关系,如图 2 所示 1 按下按钮 3 同时将操纵杆向右扳时,肘关节的角度在增大,反之肘关节的角度在减小,如果松开按钮或停止扳操纵杆,机器人关节角将保持不变 1

图 2 游戏杆(肘关节)输入

Fig. 2 Interaction of joystick front controller pattern

3 结论

作者在 MOTOMAN 机器人虚拟示教及远程控制系统中选用了北通飞行摇杆 BTP2C313 作为输入装置,利用关节、直线不同的示教方式实现了机器人作业的虚拟示教与再现,最终生成了机器人作业文件;实践表明它具有成本低、操作性好等优点,因而具有较高的实用价值与应用前景 1