

# 网络机器人控制系统中异构机器人接入技术的研究

Study on access technology for heterogeneous robots in control system of internet robot

辛洪兵<sup>1</sup>, 乔绪维<sup>1</sup>, 陈美钰<sup>1</sup>, 刘振宝<sup>1</sup>, 孙瑞涛<sup>2</sup>

XIN Hong-bing<sup>1</sup>, QIAO Xu-wei<sup>1</sup>, CHEN Mei-yu<sup>1</sup>, LIU Zhen-bao<sup>1</sup>, SUN Rui-tao<sup>2</sup>

(1. 北京工商大学 机械工程学院, 100037; 2. 机科发展科技股份有限公司 成套技术与装备事业部, 100044)

摘要: 网络控制机器人是传统机器人和网络技术的结合。研究了异构机器人接入网络机器人控制系统的要求和途径。采用多态和模板的设计方法, 实现了一种异构机器人动态接入方法。

关键词: 机器人; 网控机器人; 异构机器人

中图分类号: TP24

文献标识码: A

文章编号: 1009-0134(2009)07-0019-03

## 0 引言

Internet的迅速发展不仅使社会经济结构和人们的生活方式发生了巨大的变化,同时也开辟了网络控制机器人研究的新方向。把机器人接入网络,使网络拥有能够对环境进行操作的终端执行能力,可以极大地拓展网络和机器人的应用范围。

网控机器人的研究思想是把网络和机器人技术融合起来,从系统的角度来进行研究开发。通过计算机网络的发展来扩展机器人控制系统的多样性,使机器人的控制系统与网络中的其他成员(通过网络互连的其他智能设备)进行交互和协作,从而使信息和数据的获取与共享以及与人交互界面都将随之改变。机器人将可以越来越多的享受网络资源,利用网络的方便,使机器人成为网络的一员,这就可以突破距离的限制,真正实现跨区域、跨空间的远程控制。网络也可以借以机器人的特点来扩展它的外在延展性。

如今各国科学家在基于 Internet 的机器人控制进行了广泛的研究。但是每一种控制系统都针对特定机器人设计,系统功能相对单调,扩展性弱。本文以实现一种网络机器人控制系统可以接入不同类型的机器人为目标,开展异构机器人动态接入网络控制系统的研究<sup>[1-3]</sup>。

## 1 系统动态接入异构机器人的实现

### 1.1 异构机器人接入的瓶颈

为了实现针对不同类型的机器人都可以方便的接入系统,不能应用传统的单应用程序控制单个机器人的设计方式,必须采用一种新的设计方式进行系统设计,给不同机器人接入提供便捷的接口。

系统接入不同机器人的设计瓶颈在于:

- 1) 各种机器人的行为虽然具有相似性,但每一种机器人行为的具体实现都是不同的。
- 2) 机器人控制的状态数据和命令数据的差异性。
- 3) 机器人网络传输数据的差异性。

由此可见,要想实现系统对不同机器人的动态载入,这几个技术必须得以解决。

### 1.2 解决途径

解决多机器人动态载入的方法有如下几个方面:

- 1) 以机器人作为基本对象进行建立机器人的数据和行为模型,把机器人的一些共有的行为特征抽象出来,作为基本机器人对象的行为。
- 2) 基本机器人对象建立完成以后,考虑到不同机器人行为实现的具体差异,采用的是对象的继承和多态的设计思想。不同的机器人对象特征在子类中体现差异。不同机器人的行为在子类中得到具体

收稿日期:2008-12-12

基金项目:北京市自然科学基金项目(3082005)

作者简介:辛洪兵(1968-),男,辽宁庄河人,教授,工学博士,研究方向为机械制造、机器人学、谐波传动。

的实现。

3) 不同机器人的状态数据和命令数据的差异性, 采用 C++ 模板的技术。设计对象模型时, 不涉及到具体的数据类型, 只有在动态载入的时候才涉及到机器人数据类型。

通过上面的 3 种设计思想, 可以设计出每一种类型的机器人具体行为实现和数据结构的实现, 建立每个机器人的类库, 系统在设计时, 采用基本的机器人作为下位机进行实现, 这样在运行时动态载入需要接入系统的机器人库, 就可以使本系统方便的对机器人进行控制。下面详细的介绍机器人网络控制系统中多机器人动态载入的设计和实现。

## 1.3 实现

### 1.3.1 机器人模型基类的实现

本系统采用的技术面向对象的编程思想加上通用编程技术, 在程序设计中把机器人作为一个对象, 封装了机器人的操作行为和机器人的数据状态。这种实现方式不同于过程性编程, 它的重点不是实现任务的过程, 而是在概念上对对象的数据和行为进行设计, 对象模型的建立是整个程序设计的关键, 是系统稳定运行的前提, 采用面向对象的设计方式, 可以有效、方便的对系统进行功能扩展。

机器人基类的设计是实现不同机器人动态载入的基础和关键, 它是否具有高度的抽象性和行为的完备性, 决定了是否可以设计出有效的代替机器人的基类模型, 实现机器人动态载入。

机器人行为抽象可以简单的分为: 初始化机器人、得到机器人状态信息、定点行走、执行任务、停止任务、暂停任务、继续执行和删除任务等等。如下所示:

```
class BaseRobot
{
    ... ..
    virtual void Init ( HWND hwd ) // 机器人初始化
    virtual short BuildConnect(); // 建立和机器人的连接
    virtual short disConnect(); // 断开和机器人的连接 // 获得机器人的状态信息
    virtual void GetRobotStatus ( STATUSINFORMATION & status )
    virtual bool StartStep ( POSTION &pos )
    // 定点行走
```

```
    virtual bool StartJob ( CString tJobName )
    // 开始一个任务。
    virtual void ContinueRun(); // 任务暂时停止运行后, 重新运行任务
    virtual void StopRun(); // 暂时运行任务
    virtual void JobDelete ( CString jobname )
    // 任务删除。
    virtual bool JobDownload ( CString jobName )
    // 把任务下载到机器人上
    Robot();
    virtual ~Robot();
    ... ..
};
```

其中, 这些定义的都是机器人共有的动作行为, 不同机器人具有的特殊的行为可以在继承类里进行实现。

### 1.3.2 实现不同机器人动态载入

继承技术可以扩展和修改原有基类的行为, 所有待接入的机器人具有相同的特征, 因此可以从机器人基类 ( BaseRobot ) 中继承这些公有的行为特征, 然后根据本身特征进行相应的扩展和具体的实现, 下面是一个具体机器人 Motoman 的类模型, 它包括了机器人共有的行为方法, 和自己特有的行为。以继承的方式来达到不同机器人模型的公有和特有的属性。

```
class MotomanRobot : public BaseRobot
{
    ... ..
public:
    virtual void GetRobotStatus
    ( STATUSINFORMATION &status );
    virtual bool StartStep ( POSTION &pos );
    virtual void JobDelete ( CString jobname );
    virtual void Init ( HWND hwd );
    virtual void ContinueRun();
    virtual void StopRun();
    virtual bool StartJob ( CString jobName );
    Robot();
    virtual ~Robot();

private:
    bool ServoOn(); // 打开伺服电源
    HWND inputHwnd; // inputHwnd 是使用此 robot 类的窗口句柄
```

```

short nCid; //和控制柜连接的通道值
short JointMove(POSTION & pos);
//关节定点运动
... ..
};

```

继承实现的是不同机器人模型的建立,而多态技术实现的是机器人共同的行为特征的具体的不同实现,例如:移动机器人前进一步采用的是空间坐标,而工业机器人前进采用的可以是关节坐标。我们在基类BaseRobot和具体机器人MotomanRobot中都是用StartStep()方法来实现,但是不同机器人实际的实现代码可以是不同的。利用多态技术,可以实现机器人具有统一接口,但具体行为实现可以不同,因此网络控制系统就可以调用统一的接口函数,在运行时进行动态载入机器人进行特有的行为。

### 1.3.3 解决机器人状态数据差异

机器人种类不同,它本身的状态信息就会不同,状态信息的数据结构就会不同。实现不同的机器人接入必须解决和机器人数据通讯的问题。C++中的模块技术是很好的解决方案。

传统的和机器人用接口函数进行数据交换时,必须定义好数据格式,例如,移动机器人的状态信息可以是:

```

Class MoveRobot
{
    virtual void GetRobotStatus
(MOVESTATUSINFORMATION &status);
}
Struct MOVESTATUSINFORMATION
{
    ... ..
    int rightSpeed; //左轮转速
    int leftSpeed; //右轮转速
    POSITION pos; //位置坐标
    ... ..
}
而对工业机器人的状态信息是:
Struct MOTOMANSTATUSINFORMATION
{
    ... ..
    bool controlMode; //控制模式
    CString movtype; //关节移动方式
    CString vtype; //速度方式

```

```

double spd; //整体移动速度
short toolno; //机器人采用的工具坐标系
double p[12]; //每一个轴的位置信息
... ..
}

```

由此可以,我们在定义机器人具体的方法时,必须正确的使用状态数据结构,使用相应的机器人模型来调用机器人的方法。

本系统在继承和多态的技术上,使用模板的技术,这样,可以在定义机器人时采用通用的数据定义格式,而在具体的机器人载入时采把相应的机器人数据结构进行动态载入,实现统一的机器人接口方法。

例如:

```

Template <class T> //定义模板 通用数据类型T
class BaseRobot
{
    ... ..
    virtual void GetRobotStatus(T &status);
};
Template <class T> //定义模板 通用数据类型T
class MotomanRobot : public BaseRobot
{
    ... ..
public:
    virtual void GetRobotStatus(T &status);
};

```

如此,各个机器人在方法的接口就实现了统一,方便了不同机器人的调用。在实际调用的时候才实现具体数据类型的载入。

例如:

```

BaseRobot ( MOVESTATUSINFORMATION )
*moveRobot = new MoveRobot ( MOVESTATUSINF-
ORMATION ) ;
BaseRobot ( MOTOMANSTATUSINFSINFxT-
ION ) *motomanRobot = new MotomanRobot (
MOTOMANSTATUSINFORMATION ) ;
moveRobot 是机器人基类的指针,它的数据类
型是移动机器人的数据类型,它指向了移动机器人
(MoveRobot) 的具体实现。
motomanRobot是机器人基类的指针,它的数据
类型是Motoman工业机器人的数据类型,它指向了
Motoamn 机器人 ( MotomanRobot ) 的具体实现。

```

【下转第24页】

数进行模块组合,分别组成不同工况下的一组连贯操作。

工作装置的挖掘轨迹周期可以分解为:1)动臂和斗杆同时下降;2)下到一定高度,动臂停止,斗杆继续下降;3)当快接触地面时,斗杆继续运动,铲斗开始运动;4)接触地面后,斗杆和铲斗共同完成挖掘动作;5)动臂起升,斗杆和铲斗共同动作完成收拢;6)回转机构转动到一定角度后,斗杆和铲斗展开;7)再次转动回转机构,回转回到起始点。

运动控制程序采用C语言编写,将挖掘机工作装置的挖掘轨迹分解成以上7个动作,每个动作对应一组函数操作来实现,如:

Motion01 ( ); //动作1

Motion02 ( ); //动作2

.....

Motion07 ( ); //动作7

函数的控制流程如图3所示,此流程图为上述的某一函数执行时,内部数据运转示意图,根据函数传送来的数据要求选择相应的运动轴及模式,然后设置各轴的运动参数和位置参数,最后发出驱动信号。应用程序的编写仅仅是对函数写入要求的参数即可,因此极大的简化了应用程序的编写,方便应用和学习。

## 2 结束语

基于S3C2410微处理器和DSP专用运动控制芯片MCX314As的挖掘机运动控制系统的硬件设计方案,使工程机械运动控制器在软硬件设计上实现模块化,灵活性极大、大大提高开放性。它丰富的功能能够大量地减轻研发任务,提高研发速度,能在较短时间内得到高速度、高精度、高效率、高可靠、

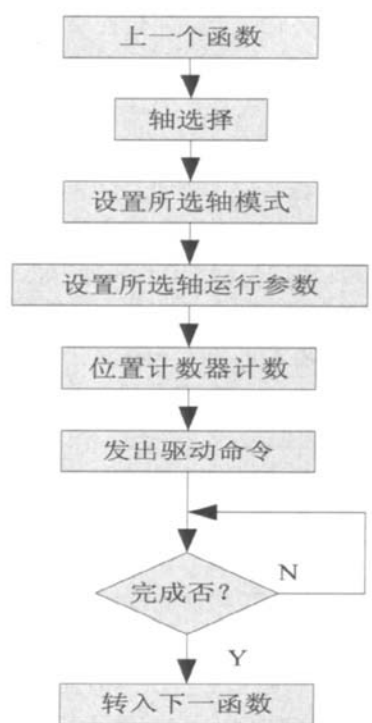


图3 函数控制流程图

低成本的挖掘机运动控制系统。

### 参考文献:

- [1] 叶佩青,汪劲松.MCX314 运动控制芯片与数控系统的设计[M].北京航空航天大学出版社,2002.
- [2] 于明.ARM9 嵌入式系统设计与开发教程[M].电子工业出版社,2006.
- [3] 吴明晖.基于 ARM 的嵌入式系统开发与应用[M].人民邮电出版社,2004.
- [4] 余张国.基于 ARM 和 MCX314A 的嵌入式运动控制器[J].微计算机信息,2005,21.
- [5] 王富春,蒋廷彪.采用 ARM9 的挖掘机工作装置智能控制[J].工程机械,2007,21.

【上接第 21 页】

## 2 总结

在机器人行为实现设计过程中,采用C++的继承机制和多态特性,建立了每一种机器人模型的行为类库,利用模板的特性,创建了机器人行为的统一接口,解决了异构机器人状态数据和命令数据的格式不一致对系统接入异构机器人的影响。在系统的运行中,可以动态的载入异构机器人的行为类库的 DLL 文件,从而实现系统动态载入机器人的功能。异构机器人接入技术的研究对基于网络的机器

人系统具有很重要的意义,对于分布式机器人技术、群体机器人协调等方面都具有重大影响。

### 参考文献

- [1] 刘轶,范宜洋,张晗.一种跨平台的机器人网络远程控制系统[J].计算机工程,2004(10):70-72.
- [2] 朱徐华.网络环境下移动机器人的建模与控制方法研究[D].南京理工大学,2006.
- [3] D.Westhoff,T.Scherer,J.Zhang.A flexible framework for task-oriented programming of service robots.VDI Verlag GMBH, dusseldorf,40001,Germany.2004:737-744.