基于神经网络的机器人碰撞检测系统研究开发

王晓伟 陈一民 沈 雄
(上海大学计算机工程与科学学院 上海 200072)

摘 要 要保证机器人本身具有比较高的健壮性、安全性和实用性,机器人必须具有碰撞检测功能,给出一种采用人工神经网络进行机器人碰撞检测的原理及实现方法。在分析机器人三轴多功能位置卡有关脉冲数据的基础上,总结机器人运动连续性的规律,并建立机器人碰撞检测问题的人工神经网络模型,设计并实现了相关的机器人碰撞检测算法及控制模块,最后介绍使用多线程方法,将碰撞检测模块加入到了机器人控制系统中。

关键词 碰撞检测 神经网络 BP算法 多线程

RESEARCH AND DEVELOPMENT OF ROBOT COLLISION DETECTION SYSTEM BASED ON ARTIFICIAL NEURAL NETWORK

Wang Xiaowei Chen Yimin Shen Xiong
(School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China)

Abstract Robot must have its collision detection function to ensure the higher robustness, safety and practicality of itself. In this paper, we introduced the principle and method of a kind of robot collision detection using artificial neural network. Based on the analysis of the pulse data collected from robot triple-axle multi-function control card, we got the law of continuity of robot's movement and built up the artificial neural network model for robot collision detection, and designed and realized the algorithm and control model of the robot collision detection. At last, we explained how to add the collision detection module to the robot control system using multi-thread technology.

Keywords Collision detection Neural network BP algorithm Multi-thread

0 引 言

机器人运行时需要防止碰撞到附近的物体或人员,避免损害机器人、其它设备或伤害到工作人员。目前国内外进行机器人碰撞检测的主要方法有两种[1-3]:(1)是利用机器人视觉,即在机器人上或工作空间中加装一个或几个摄像头,机器人控制软件对输入的机器人环境图像进行处理,当检测到有可能发生碰撞时,采取一定的措施。这种方法的不足在于:使用摄像头增加了系统成本,而且控制软件中要有图像处理和模式识别的模块,这方面的技术还有待进一步提高,另外,由于进行处理的运算量较大,相对来说很难实现实时避碰撞检测与控制;(2)是采用传感器,即在机器人的各个关节上安装相应的力检测传感器以及 A/D转换器,但它的成本相对较高,实现实时控制也不容易,因为同样要对不同的关节运动状态数据作不同的实时处理。由于存在上述这些困难和问题,因此我们提出了基于神经网络的机器人碰撞检测方法,即利用直接控制机器人控制系统中的主要部件:三轴位置中的脉冲数据,并结合神经网络来实现机器人碰撞检测,完成机器人避碰撞功能。

1 碰撞检测系统的结构

整个机器人控制系统的结构如图 1所示,主要由以下几部分:1)工业控制微机(IPC),在这上面主要运行机器人控制器软件;2)2块三轴多功能位置卡,用来输出机器人各关节(轴)的控制信号,同时接收机器人返回的位置脉冲信号;3)1块带光电隔离的 I/O卡,用来采集开关量输入信号和控制开关量输出信号;4)1块 A/D、D/A卡,用来采集焊接参数控制电流及电压输出信号;5)基于 8051单片机的示教器,包括键盘、液晶屏、串行通信线等附件;6)机器人本体为 PT600机器人。

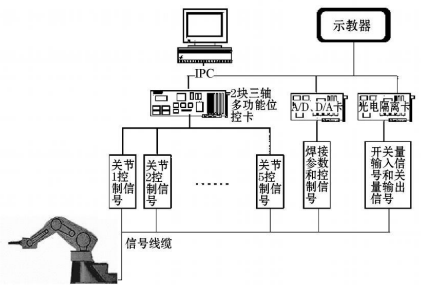

图 1 系统结构

其核心为三轴位置卡,亦称电机控制卡,通过它对机器人的关节进行控制。三轴位置卡能对三轴进行联动插补控制。每轴有专用位置芯片控制,构成一个伺服位置和速度环。三轴线性插补控制 DDA(Digital Differential Analysis)插补时间为 1毫秒至 2秒、1毫秒伺服参数更新时间,能用 C语言和其他高级语言

收稿日期:2008-02-03。上海市科技发展基金(015115042)。
王晓伟,讲师,主研领域:计算机控制技术。

对其编程。在机器人控制执行过程中,机器人的运动是由电机驱动的,而电机驱动所需要的位置信息是由计算机送往三轴位置卡的脉冲信号控制的,根据发送的脉冲的数量和正负决定机器人的运动方向和轨迹。由于可以知道发送的脉冲是多少,经过一个特定的转换关系,就可以从脉冲数得到相应的关节坐标和直角坐标,从而实现对机器人的运动控制。

2　碰撞检测的原理与实现方法

为实现对机器人运动的异常情况进行有效的检测,利用机器人的三轴位置卡提供的基本特性,即机器人运动过程中反馈给控制器的误差脉冲,进行碰撞检测。根据机器人运动的连续性原理,通过分析机器人运动过程中反馈给控制器中的误差脉冲数的变化情况,来判定有否碰撞发生。但由于误差脉冲数的变化随机器人的运行位置和速度的不同而不同,所以很难通过建立一个算法,根据误差脉冲数来判定是否发生碰撞。而神经网络具有处理常规数学方法难以解决的模糊信息处理的能力,因此尝试通过建立神经网络完成这一功能。神经网络碰撞检测功能框图如图 2所示。采用神经网络方法进行碰撞检测的思路为:在机器人正常运行和发生碰撞时,对机器人的运动轨迹进行检测,由机器人三轴位置卡获取 DDA周期中的与运动有关的信息,如插补脉冲数、误差脉冲数,及机器人的速度、加速度和位置等情况。通过对脉冲数和误差脉冲数的统计分析,总结出脉冲数和误差脉冲数对应于机器人运动速度、加速度等的规律,然后建立神经网络模型[4],利用机器人运动的数据对网络进行训练和仿真,得到合适的神经网络的结构,为了完成这些任务,我们采用 Visual C++ 6.0开发了神经网络训练仿真系统。利用神经网络训练仿真系统得到合适的神经网络的结构后,在机器人控制软件中实现此模型,编写碰撞检测线程,完成碰撞检测的功能。

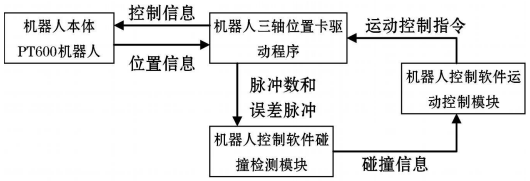

图 2　神经网络碰撞检测功能框

3　神经网络模型及算法

根据机器人运动连续性原理,即机器人每一插补周期(DDA周期)的运动的坐标位置、速度、加速度是由机器人上一插补周期的脉冲数,及该时刻的坐标位置、速度、加速度决定的。可通过在机器人正常工作情况和有碰撞发生情况下,分别对其运动轨迹进行检测,获取机器人在此运动轨迹的每一插补周期的插补脉冲数、误差脉冲数和机器人的位置信息,以及插补脉冲数、误差脉冲数对应于速度、加速度的变化情况,组成神经网络的输入、输出样本。我们选择五根轴的前一插补时间及当前插补时间的脉冲数,和前一插补时间的误差脉冲数作为输入节点,对应的输出节点是当前插补时间的误差脉冲数,利用神经网络的非线性映射能力和泛化能力,使训练好的网络能根据给出的输入数据得出合适的输出。这样设计的神经网络有 15个输入节点,5个输出节点。因为理论上,三层 BP神经网络可以任意精度逼近非线性映射关系,我们确定有一层隐藏层,并预先设定其隐藏层为 10个节点,然后根据训练的情况再进行调整。根据确定的方案,采用的神经网络结构如图 3所示。

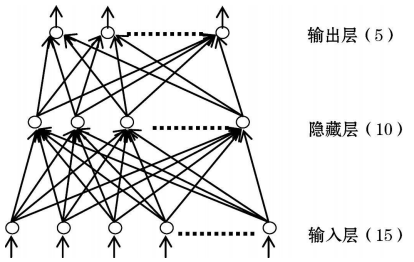

图 3　初步确定的神经网络结

采用的 BP算法中权值修正算法为:

$$w(k+1) = w(k) + \alpha D(k)$$

$$D(k) = \frac{-\partial E}{\partial w(k)}$$

其中 $w(k)$为第 k步时的权值,$D(k)$为第 k步时的负梯度,E为 k时刻的输出误差,α为学习率,$\alpha > 0$。

由于 BP算法在学习过程中收敛慢,因此在实际应用中,我们将动量法和自适应学习率法相结合[5],得到动量-自适应学习率 BP算法[6],其权值调整算法为:

$$\begin{cases} \Delta w(k) = \Delta w(k-1),\ \eta(k+1) = 0,\ \alpha(k+1) = \alpha(k) \cdot dm & \text{如 } E(k) > E(k-1) \cdot er \\ \Delta w(k) = \eta(k)\Delta w(k-1) + (1-\eta(k))\alpha(k)D(k-1),\ \eta(k+1) = \eta,\ \alpha(k+1) = \alpha(k) \cdot im & \text{如 } E(k) < E(k-1) \\ \Delta w(k) = \eta(k)\Delta w(k-1) + (1-\eta(k))\alpha(k)D(k-1),\ \eta(k+1) = \eta,\ \alpha(k+1) = \alpha(k) & \text{如 } E(k-1) \leqslant E(k) \leqslant E(k-1) \cdot er \end{cases}$$

其中,η为动量因子常数,dm为学习率减小率,im为学习率增加率,er为误差比率,它限制了单次训练中可能增加的误差。如果误差上升超过了误差比率,则舍弃新的权值,下一次的学习率减小,并暂时不使用动量;如果当前误差小于上一次的误差,则下一次的学习率增加,并使用动量;否则,学习率不变,并使用动量。

4　数据采集和神经网络训练

4.1　数据采集函数

我们对神经网络进行训练需要的数据是机器人运动过程中上一 DDA周期的插补脉冲数、误差脉冲数和当前 DDA周期的插补脉冲数、误差脉冲数,因此我们需要从三轴多功能位置卡的驱动程序中读取这些数据。三轴多功能位置卡驱动程序的编程接口为 a_PCL_Query(),利用它可以将机器人五个轴的运动信息读入到一个缓冲区 wBuffer中,以供分析训练用,具体代码如下:

```
BOOL a_PCL_Query(HANDLE hDevice, WORD wChannels, WORD wBuffer[12])
{
return(DeviceIoControl(hDevice, IOCTL_PCL_QUERY, (LPVOID)MAKELONG(wChannels, 0), 0, wBuffer, 24, NULL, NULL));
}
```

然后,利用函数 Transform()将缓冲区 wBuffer中的数据进行分析可得到每个轴的误差脉冲数和插补脉冲数,代码如下:

```
void Transform(WORD wChannels, WORD wBuffer[12], int error[6], int pulse[6])
{
```

```
int i;
DWORD dwBits;
for( i= 0, dwBits= 1; i< 6; i++, dwBits<<= 1)
{
    if(!(wChannels& dwBits)) continue;
    if(wBuffer[ i* 2] & 0x8000)
                    //转换各轴误差脉冲数,并保存到 error数组
      error[ i] = MAKELONG(wBuffer[ i* 2] | 0x0000, 0xffff);
    else
      error[ i] = MAKELONG(wBuffer[ i* 2] & 0x7fff, 0);
    pulse[ i] = wBuffer[ i* 2+ 1];
                    //转换各轴脉冲数,并保存到 pulse数组
}
}
```

其中, error数组保存每轴的误差脉冲数, pulse数组中保存每个轴的插补脉冲数。

4.2 网络训练

机器人运动的两个基本动作是直线和圆弧,所以在机器人走直线、走圆弧的两组动作下,分别取两组数据。在建立神经网络前首先对这两组数据进行处理,保存到文件中,然后从每个文件中分别取出一段数据组成输入、输出样本文件,并为每个输入节点找出样本的上下限,建立样本范围文件。

我们在神经网络训练仿真系统中建立一个输入层、隐藏层和输出层节点数分别为 15、10和 5的 BP网络。利用前面得到的范围文件对此神经网络进行初始化,然后用样本文件对其训练。

5 机器人碰撞检测功能的实现

有了训练好的神经网络的结构参数,就可以在机器人控制软件中,加入碰撞检测模块。具体思路为:在机器人控制软件启动时,利用给出的神经网络结构文件,对神经网络初始化,当机器人在执行运动命令后,首先启动碰撞检测线程,随后启动具体运动的函数,这样碰撞检测线程就可以监测机器人的运行,检测机器人碰撞情况的发生。程序框图如图 4所示。

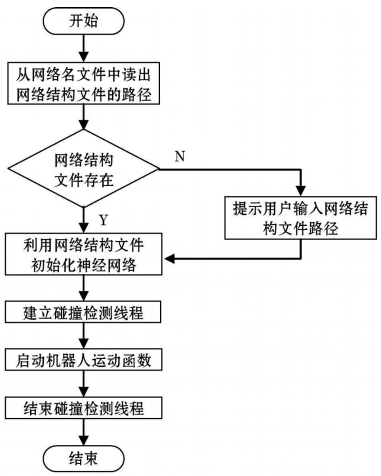

图 4 机器人碰撞检验程序流程图

为实现利用神经网络进行机器人碰撞检测,需要在机器人控制软件中实现神经网络类,根据给定的神经网络的结构文件,构造出神经网络,并对机器人三轴位置卡驱动程序中读取的数据进行仿真,将仿真结果与实际值进行比较,根据误差判断机器人运行是否正常。

另外,在机器人控制软件中,机器人的各种动作都是用单个函数来实现的,为保持这些函数的完整性,通过线程来实现碰撞检测功能。线程是 Win32分配 CPU时间的基本元素,通过多个进程与线程同时执行来实现多任务。在 Visual C++程序中,可以创建用户接口线程(User Interface thread)和工人(Worker thread)线程,其中用户接口线程用来处理用户输入和响应由用户产生的事件和消息,而工人线程不处理窗口消息,用于在后台执行任务。我们使用的线程属于后者。在碰撞检测线程中,循环读取机器人每一插补周期各轴的脉冲数、误差脉冲数,并保留上一周期的脉冲数、误差脉冲数,利用已经构造好的神经网络进行仿真,并作出判断。以下是线程中用到的处理函数 a_collisiondetect()的部分代码:

```
CTestDlg * pDlg= (CTestDlg *) pparam;
int lastpulse[ 6] = { 0, 0, 0, 0, 0, 0};
                    //保存每轴上一插补周期的脉冲数的数组
    int currentpulse[ 6];   //保存每轴当前插补周期的脉冲数的数组
……
    CString status[ 6];     //保存每轴当前插补周期的状态的数组
    ……
    while( (pDlg-> m_sys_status != S_OVERFLOW)&&! bruneover)
                    //机器人运动未结束
    {
a_PCL_Query(pDlg-> m_hDevice, -1, wRead);
                    //从三轴位置卡驱动程序中读取数据
Transform(-1, wRead, currenterrent, currentpulse);
                    //转换为脉冲数和误差脉冲数
      ……
    //下面四步构造神经网络输入矩阵
      mxin.SetVal(0, 0, lastpulse[ j] );
      mxout= ghNet.simu(mxin);  //神经网络进行仿真,输出结果矩阵
      for( j= 0; j< mxout.RowNo(); j++)
        for( k= 0; k< mxout.ColNo(); k++)
          if( (mxout.GetVal(, j, k) + ghNet.m_errange)>= 0)
          {
            pDlg-> OnPause();
ExitThread(10);
}   //如检测到有一轴的误差脉冲数超过允许值,则发出停止命令,强制
                    //机器人停止运动退出线程,并返回退出码 10
```

在机器人发出运动命令之后,先创建此线程,在创建开始时首先将它挂起,设置线程的 m_bAutoDelete成员变量为 false,这样线程在退出时就不会自动删除,然后可对线程的退出码进行分析,然后恢复碰撞检测线程,调用机器人动作函数。在机器人运动过程中,碰撞检测线程通过三轴位置卡驱动程序接口 a_PCL_Query和 Transform函数,取得每一轴的脉冲数和误差脉冲数,根据神经网络仿真计算的理论误差脉冲数和实际驱动程序得到的误差脉冲数,判断机器人运行是否发生异常。如果理论值与实际值之差超过神经网络训练时的目标误差,在线程内调用 OnPause()函数,强制机器人停下当前动作,然后退出,并返回一个 DWORD型值 10为退出码,机器人控制程序检查到此退出码,就可知道机器人运行发生了碰撞。碰撞检测线程代码如下(以机器人作直线运动为例,其他运动情况类似):

```
m_cdthread= AfxBeginThread( a_collisiondetect, this, THREAD_PRI-
ORITY_NORMAL, 0, CREATE_SUSPENDED);
                    //创建挂起的碰撞检测线程
```

```
m_cdthread->m_bAutoDelete = FALSE;
m_cdthread->ResumeThread();                    //恢复线程
if(m_sys_status == S_READY) {
iLPTP = line(m_hDevice,&m_totalpulse,&m_angle,Angle2Point(a),
SPL,&m_v,0.0,ACC1,ACC2,m_cycle/1000.0,TRUE);
                                               //机器人直线运动函数
if(iLPTP! =0) MessageBox("ErrorLPTP!");
}
brunover= true;                                //机器人运动结束
::GetExitCodeThread(m_cdthread->m_hThread,&ExitCode);
                                               //取碰撞检测线程退出码
if(ExitCode== STILL_ACTIVE)
                                //如碰撞检测线程尚在运行,强行结束它
{
  m_cdthread->SuspendThread();
  TerminateThread(m_cdthread->m_hThread,-1);
}
else if(ExitCode==10)
                        //碰撞检测线程返回退出码 10,表明发生碰撞
AfxMessageBox("发生碰撞!");
delete m_cdthread;
```

6　碰撞检测测试

采用模快化的编程思想,设计了如图 5所示的碰撞检测测试系统。为了控制机器人运行时确保机器人的运行安全,在碰撞检测测试系统中加入了上面设计的碰撞检测线程,检测机器人是否会与其它物体发生碰撞以及发生碰撞后的处理情况。

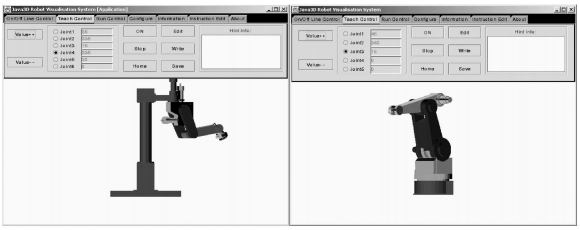

图 5　机器人碰撞检测测试系统

将设计的碰撞检测模块加入到了机器人控制器系统中,在实际联机调试时,分别采取人为阻挡和放置障碍物的方式,对机器人碰撞检测模块功能进行测试。机器人在受到碰撞后一个很小的延时约 6毫秒,大约 3DDA周期,碰撞检测模块就检测到发生了碰撞,机器人停止了当前的动作。结果表明:此模块能够实现我们碰撞检测的功能,反应时间为 6毫秒。

7　结　论

按照机器人运动规律建立神经网络模型,通过采集机器人三轴位置卡中的各轴的脉冲数和误差脉冲数,对神经网络进行训练和仿真,对各种网络模型分析比较,得到能适合碰撞检测任务的神经网络的结构,然后在已有机器人控制软件中加入碰撞检测线程,实现了机器人的碰撞检测功能。碰撞检测功能的实现,提高了机器人的健壮性、安全性。

参　考　文　献

[1] Jih-Gau Juang. Collision Avoidance Using Potential Fields[J]. Industrial Robot, 1998, 25(6): 408-415.

[2] Kron A, Schmidt G, et al. Disposal of Explosive Ordnances by Use of a Bimanual Haptic Telepresence System. IEEE International Conference on Robotics & Automation, 2004, 2: 1968-1973.

[3] 吉爱红,戴振东,周来水. 仿生机器人的研究进展[J]. 机器人, 2005, 27(3): 284-288.

[4] Chiping Zhang, Pingyuan Cui, Yingjun Zhang. An Algorithm of Data Fusion Combined Neural Networks with DS Evidential Theory. Proceedings of 1st International Symposium on Systems and Control in Aerospace and Astronautics, 19-21 Jan. 2006: 1141-1144.

[5] 徐春晖,徐向东. 前馈型神经网络新学习算法的研究[J]. 清华大学学报:自然科学版, 1999, 39(3): 1-3.

[6] 冯勤,王锁萍. 神经网络中 DLR-BP新算法的计算机模拟及性能分析[J]. 计算机应用研究, 1999, 2: 15-16.

(上接第 25页)

7　结　论

本文通过对流媒体数字版权的研究的基础上,提出了一个通用的流媒体数字版权保护系统的体系,实现了一个流媒体数字版权保护原型系统,进行了 SQDRM系统实验并有一定程度的实际应用。结果表明系统具有透明支持流媒体的本地回放和在线播放的能力。有效解决了数字版权领域的关键问题,并提供了标准的接口供用户扩展。实际系统的部署和初步实验表明系统实现了对流媒体数据的良好的保护。当然,要达到绝对的安全是不现实的,数字版权保护的解决方案将依赖更安全的软、硬件环境。将在后续的研究工作中进一步对系统进行扩展,完善系统的商业模型。

参　考　文　献

[1] Rosenblatt W, Trippe W, Mooney S. Digital Rights Management: Business and Technology. New York: M&T books, 2002.

[2] Garnett N. Digital rights management, copyright and napster[J]. ACM SIGecom Exchanges, 2001, 2(2): 1-5.

[3] 俞银燕,汤帆. 数字版权保护技术研究综述[J]. 计算机学报, 2005, 28(12): 1957-1968.

[4] 庄超. 一种新型的 Internet内容版权保护的计算机制[J]. 计算机学报, 2000, 23(10): 1088-1109.

[5] 张福学. 数字版权管理系统的功能和信息结构分析[J]. 情报杂志, 2002(6): 26-27.

[6] Kin Zwoll. Digital Document Delivery and Digital Rights Management[J]. Information Services & Use, 2001(21): 9-11.

(上接第 58页)

参　考　文　献

[1] Cristina G, Rogerio L. Structure for Dependability[M]. London: Springer, 2006.

[2] Virtual Socket Interface Alliance (VSIA). http://www.vsi.org.

[3] Ocp-IP. http://www.ocpip.org/.

[4] Spirit. http://www.spiritconsortium.org/.

[5] Accellera. Accellera Property Specification Language Reference Manual Rev1.1, 2004.

[6] 集成电路 IP核标准工作组. 数字软 IP核交付项规范(讨论稿).

[7] 集成电路 IP核标准工作组. 数字软 IP核文档结构规范(讨论稿).