

打磨单元设备保护控制系统 软件设计

文 | 单星¹ 郭丹蕊² 高狄¹ 尹文芳¹ 梁宪峰¹ 1. 长沙长泰机器人有限公司 2. 郑州电力高等专科学校

一、引言

在自动化铸件清理领域,由机器人抓取主轴,针对不同的清理特征选取相对应的刀具对铸件进行定点清理是打磨行业的发展趋势。考虑到清理过程中铸件的差异性 & 未知性,易造成清理不彻底或者过磨的情况,除此之外,由于机械误差、电气故障等也可能造成清理不达标的情况,考虑到持续过磨时会有刀具、主轴和机器人等设备损坏的风险,从而影响到机器人的连续生产等后果。

当前市场上应用的主要有三种方式对铸件打磨过程进行保护。

(1) 机械浮动式去毛刺执行器。执行器通过内部机构对来自外界的力进行相应的动作,从而避免打磨刀具由于受到较大的力磨损同时打磨工件也不会因为力太大被损坏。然而单独使用浮动式去毛刺执行器在加工过程中刀具出现发弹的现象严重,当碰到较大的清理特征时刀具将自动绕开,无法保证残渣完全去除,再者现阶段浮动式去毛刺执行器仍处于国外垄断状态,国内没有自主研发且商业化的产品,对国内中小型制造企业而言,无论是采购途径还是采购成本都是很大问题。

(2) 接触式力矩传感器。其属于被动补偿,在对铸件毛坯直接检测过程中存在干扰大、测量误差大以及传感器易损坏等技术问题。

(3) 通过设计高精度的工装。铸件清理属于铸件后处理,此阶段铸件为毛坯状态无准确的定位基准且铸造变形等因素要实现工件精确定位难度太大。

综合以上分析,本文设计了一种针对打磨单元闭环反馈

系统,该系统上位机用 Visual Studio 2013 开发环境,通过实时采集变频器参数信息及控制设备状态,将采集到的数据进行算法处理控制机器人运行方式,从而达到提高质量及保护设备的目的。

二、主要研究内容及原理

为了保证打磨质量同时保护打磨设备开发了打磨单元设备保护控制系统,该系统实现了通过变频器读取主轴运行参数,间接获取主轴承受力矩,实时调整机器人进给速度,保证主轴运行安全和延长打磨刀具使用寿命。当铸件某个位置毛刺较大时,降低机器人进给速度。待监测到变频器参数趋于正常之后,将机器人状态恢复到正常模式。通过以上方式完成一个标准的闭环控制,以此保证设备安全。在整个系统运行时软件需要通过 UDP 实时读取打磨时参数,使用 OPC 服务器获取整个打磨单元中各个设备的状态,上位软件通过模糊 PID 算法通过 Modbus Tcp 对机器人运行方式实时调整,达到保护设备及提高打磨质量的目的,软件结构图,如图 1 所示。

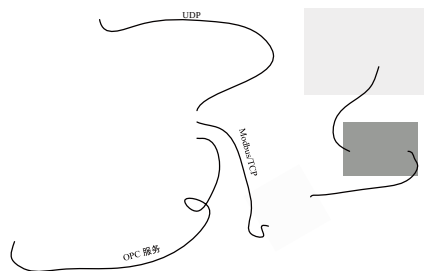


图 1 软件整体结构图

三、功能块介绍

采用 Visual Studio 2013 开发环境中的 C# 语言编写上位机软件。

Winform 中含有大量的可视化拖动控件,可以对空间控件完成快速的功能实现,除此之外,C# 是种完全面向对象的语言。本着函数内部耦合,外部解耦的原则,本次开发将各个功能分模块进行,各个模块相互独立,通过主界面调用完成相应功能,如图 2 所示。

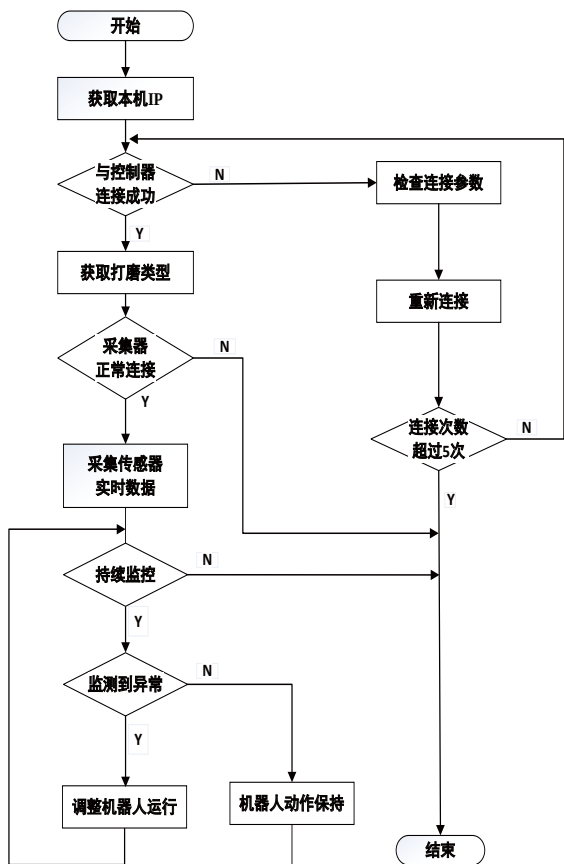


图 2 系统流程图

1、设备保护控制系统与控制器信息交互

本系统借助于 OPC 服务器作为上位机软件 and 控制器之间的桥梁,OPC 是为了连接数据提供源 (OPC 服务器) 和数据的使用者 (OPC 应用程序) 之间的软件接口标准。其数据提供源可以是 PLC、DCS 和条形码读取器等控制设备。随控制系统构成的不同,作为数据提供源的 OPC 服务器即可以是和 OPC 应用程序在同一台计算机上运行的本地 OPC 服务器,也可以是在另外的计算机上运行的

远程 OPC 服务器。

本次使用 Server 为组态软件 WinCC 自带的 OPCServer. WinCC.1, 使用之前首先将 serverName 与 IP 地址绑定,之后完成创建 Groups、Items 和 Tags 等。

```
List<OPCItem>KepItemList=new List<OPCItem>();
```

```
KepGroups = KepServer.OPCGroups;
```

```
KepGroup=KepGroups.Add(“OPCDOGROU”);
```

```
KepItems = KepGroup.OPCItems;
```

```
KepGroup.UpdateRate=100; // 刷新频率为 1 秒
```

```
KepGroup.IsActive = true;
```

```
KepGroup.IsSubscribed=true; // 使用订阅功能
```

```
for (int i = 1; i < distinctItems.Length; i++)
```

```
{
```

```
KepItemList.Add(KepItems.AddItem(distinctItems[i],  
i));
```

```
}
```

```
// 遍历所有变量并进行筛选
```

```
oPCBrowser.ShowBranches();
```

```
foreach (var item in oPCBrowser)
```

```
{
```

```
if (item.ToString().Contains(“ForControl”))
```

```
{
```

```
lbShowBranch.Items.Add(item);
```

```
}
```

```
}
```

至此可以将本系统和控制之间通信的变量已经获取成功并且存储到 lbShowBranch 中,在主界面中可以通过对 lbShowBranch 中的变量进行读取和写入操作。

本次上位软件主要完成从组态软件中读取整个打磨单元当前状态变量,根据实际需求读取需要的变量信息,考虑到需要获得状态信息才可以进行下一步操作,读取打磨单元变量通过同步方式实现,具体代码如下。

```
/// <summary>
```

```
/// 同步读取单个内容
```

```

///<summary>

public object SyncRead( OPCItem kepItemArray)
{
    object ItemValue;
    object Qualities;
    object TimeStamps;
    try
    {
        kepItemArray.Read(1, out ItemValue, out
Qualities, out TimeStamps);

        return ItemValue;
    }
    catch (Exception )
    {
        throw;
    }
}

```

2、设备保护控制系统与变频器信息交互

由于采集数据频率要求较高,且个别数据的丢失对整个系统的影响不大,所以本次选择 UDP 通信方式,与 TCP/IP 的三次握手不同,UDP 通信只需要设定 Port 或者 IP 之后就可以通信了,所以其通信会出现有丢包和数据缺失等情况,但同时 UDP 的信息传送速度是极快的。

在本系统中,实现 UDP 通信只需以下几行代码。

```

Socket socketUDP;
EndPoint endPoint;

public struct saveAndCheck
{
    public byte[] buffer;

    public int recv;
};

saveAndCheck saveAndCheckPower = newsaveAndCheck();

```

```

int port = 51234;// 本地 port
IPAddress ipAdd= IPAddress.Parse(tbIp.Text.Trim());
EndPoint point=new EndPoint(ipAdd,port);
endPoint = (EndPoint)point;

// 绑定完成之后,接收采集模块数据

socketUDP=new Socket(AddressFamily.InterNetwork,
SocketType.Dgram, ProtocolType.Udp);

// 创建 UDP

socketUDP.Bind(endPoint);

saveAndCheckPower.recv = socketUDP.
ReceiveFrom(saveAndCheckPower.buffer, ref endPoint);//
字节个数

string allReceive = Encoding.UTF8.
GetString(saveAndCheckPower.buffer, 0,
saveAndCheckPower.recv);

```

3、设备保护控制系统与工业机器人信息交互

对于整个打磨单元来说,除了此设备保护控制系统运行在工控机上,还会同时包含其他需要与机器人进行信息交互的控制软件,机器人本身带有以太网通信功能,考虑到本次开发的上位软件实时采集变频器参数,参数变化无规律性,为了保证工业机器人的正常运行不会被打断同时又能及时响应上位软件,本次控制软件使用输入输出 IO 模块与工业机器人通信。

外部设备与机器人硬接线主要借助于倍福模块,IO 模块的输出作为机器人的输入,上位机实时监测打磨过程中变频器参数变化,当有异常情况时发送相应的 IO 信号以此达到实时控制机器人运行速度的目的。本次选用的 IO 模块使用 Modbus Tcp 通信。

在使用模块时候,首先保证 IO 模块的 IP 地址和运行设备保护控制系统的工控机在同一网段内,之后将 IO 模块的 Connect 及 read 和 write 功能放置在单独的名字为 IOInstruction 类中方便主界面调用。

在主界面使用时只需完成连接 IO 模块,当需要控制机器人运行状态时,调用 IOInstruction 类的 write 函数按照与机器人的约定通过 IO 模块中发送命令。

```
IOInstruction ioInstruction = new IOInstruction();
ioInstruction.NetworkIsOk = ioInstruction.Connect();
ioInstruction.master.WriteMultipleCoils(1, 0, new bool[] {
true, true, true });
```

四、系统可行性验证及结果分析

首先启动设备保护控制系统,软件运行之前需要一系列的参数检查,包括与 OPC 服务器之间参数配置(保证与控制器之间通信正常),变频器参数获取通过 UDP 通信(IP 地址 + 端口号),IO 模块的 IP 地址等,尤其特别注意若 OPC 服务器参数配置,若服务器尚未开启则系统提示启动服务器,服务器正常启动后检查服务器中是否包含有软件中使用的所有标签,若软件中使用的标签服务器中并没有创建,会导致软件异常退出,所以启动前的各项检查是必不可少的环节。

软件启动完成之后,从控制器中获取当前生产状态,简单来说即工业机器人是否在工作,若在非工作状态则软件不会机器人控制,若在工作过程中,软件实时读取变频器参数信息并将参数信息绘制成波形直观显示,通过一定的算法对检测到的数据进行分析,判断整个系统工作状态是否正常,同时根据大量的数据分析正常工作范围。控制软件实时数据采集如图 3 所示。

打磨阶段的数据分析使用了后台处理,由于需要实时读取并分析数据,本系统使用线程池,相比手动创建的线程池可以复用,减少了创建线程的时间。

由图 4、图 5 可知,同种型号的不同工件打磨时传感器监测到的数据也会有不同,验证了设备保护控制系统提出的初衷。

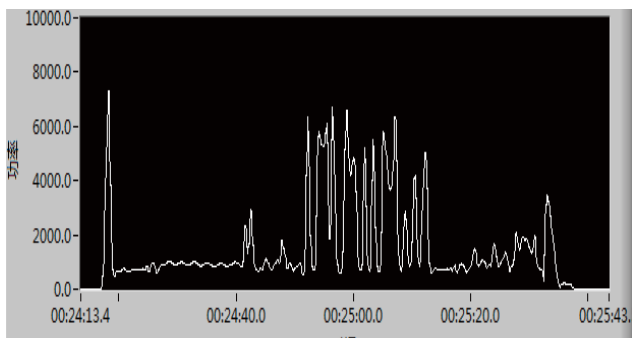


图 3 打磨过程变频器参数数据显示

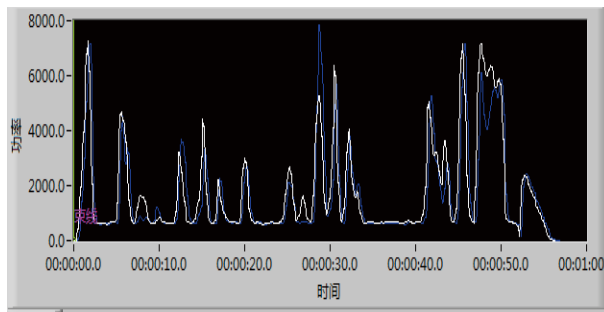


图 4 不同工件打磨数据对比图

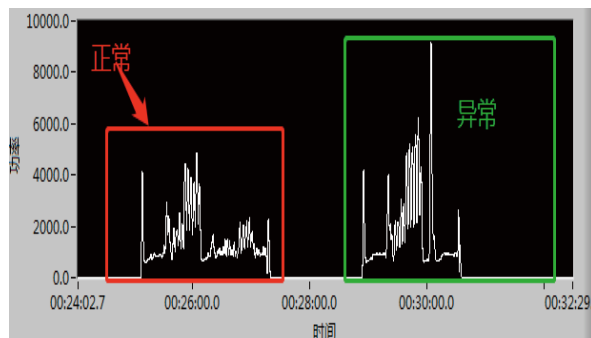


图 5 打磨异常对比图

五、总结

打磨单元设备保护控制系统创造性的解决了铸件毛坯异常特征随机出现的技术难题。可以有效解决因清理特征差异性、设备故障等原因造成打磨到工件本体和损坏设备的风险。可充分保证清理加工系统的可靠性,防止加工系统的损坏或者重大安全事故发生。同时当遇到较大的凸瘤等清理特征时,能够保证完全、可靠去除。对铸件自动清理系统的刚性以及在震动工况下的特性影响小。可以克服浮动工具的缺点,没有应用场合限制。所以本系统可以广泛应用于各种机械粗加工场合。IM