

单件生产实验系统控制及仿真的多线程实现

严 颖, 汪 峥

(东南大学 自动化学院, 江苏 南京 210096)

摘 要:文中采用多线程的方式实现了基于消息并结合仿真的单件生产实验系统的控制。其中工件线程通过在执行函数中下达工件到达指令,用来模拟工件进入生产系统的动作,机器线程则通过在线程的执行函数中向机械手发送工件加工请求来模拟机器。线程间通过消息队列进行通信。在介绍了该方法的思路后,阐述了结合仿真的生产过程控制的多线程的实现方法,并给出了实验结果。实验结果表明该方法能较好地完成单件生产实验系统的控制。通过对于单件生产实验系统的控制,可以灵活地设定生产计划,并完成。

关键词:多线程;并行;离散事件仿真;耦合度;控制

中图分类号:TP391.73

文献标识码:A

文章编号:1673-629X(2010)02-0197-05

Multi-Thread-Based Implementation of Control and Simulation for an Experimental One-of-a-Kind Production System

YAN Ying, WANG Zheng

(School of Automation, Southeast University, Nanjing 210096, China)

Abstract: In this paper, a message-based multi-thread method combined with simulation is implemented for the control of one-of-a-kind production system. The part thread simulates the action of the part entering into the production system by issuing instruction in execution function. The machine thread simulates the machine by sending requests to the manipulator. And the threads use the message queue to communicate with each other. After the introduction of its main idea, the implementation of the multi-threaded control method is elaborated in this paper, and an example of the implementation is given. The result shows that this method is helpful to the production control of one-of-a-kind production system. By the control of one-of-a-kind production system, the production plan can be flexibly made and accomplished.

Key words: multi-thread; parallel; discrete event simulation; coupling degree; control

0 引言

多线程技术在工业控制领域有着很好的应用,这是因为这一技术适合于编程实现多个并发过程的控制,文献[1]中采用多线程方法进行高精度的多通道数据采集,能够较好地满足实时性的要求。文献[2]中编写了一个基于多线程技术的工业控制系统,从现场采集数据,并根据现场数据来发出控制指令,使生产过程顺利进行。

多线程技术也可应用于离散事件仿真,仿真技术中的基于 Agent 的建模与仿真(Agent-Based Modeling and Simulation, ABMS^[3])的一种常用的实现方法就是

采用多线程的方式进行仿真。该仿真方式在制造系统调度中得到广泛应用^[4]。ABMS方法通过对复杂系统中的基本元素及其之间的交互关系的建模,可以将复杂系统的微观行为和宏观“涌现”现象有机地联系起来。当前很多文献提出将 ABMS 与当前流行的分布式仿真相结合,将复杂系统的仿真分为几个相对独立的部分,在多台计算机上同时进行仿真,并通过设置统一的时钟以及接口相互间保持耦合。对于单机仿真来说,可应用多线程技术实现 ABMS 的思想。

资源(resources)和处理(processes)是仿真程序的两个基本组成模块,资源是被动的对象,处理则是动态、独立的个体,它可以调用资源以达到仿真的目的。许多仿真方法中将它们视为两种不同的概念,然而两者的区别是相对的而不是绝对的^[5]。文献[6,7]和[8]中都提出以一个名为实体(entity)的程序模块来同时仿真资源和处理。由于线程间是独立并发的,每个线程就相当于一个 Agent,它本身应该既可以作为其他

收稿日期:2009-06-10;修回日期:2009-09-23

基金项目:国家自然科学基金资助项目(50505006)

作者简介:严颖(1985-),男,江苏南京人,硕士研究生,研究领域为计算机集成制造;汪峥,教授,硕士生导师,研究领域为制造系统控制,产品设计过程分析。

线程进行处理所需的资源,也应该具有对当前状态进行判断并做出相应处理的能力,所以将资源和处理合并为一体是非常合理的。

文中引入 ABMS 的方法,采用面向对象的方法,基于多线程技术将单件生产实验系统的控制与仿真相结合,通过仿真得到当前的部分生产状况,完成对单件生产实验系统的精确控制。由于条件所限,在本单件生产实验系统中并没有机器的存在,故而我们为机器创建类,并由生产现场的状况创建相应数量的对象,通过机器的类对象来对真正的机器进行模拟。对象中的资源部份和处理部分分别由属性和函数来体现。在类中包含有线程创建函数和线程执行函数,这样每个对象拥有自己的线程,并且可以相对其他对象独立执行,使得每个机器对象和工件对象都成为一个独立的 Agent,具有根据实时资源状况做出相应处理的能力。在对象中还有线程的句柄和线程 ID 可以对每个对象中的线程做唯一标示。通过以上措施,可以做到在生产线运行中对单个工件的生产进行精确控制,并且可以在仿真中通过并行的方式对生产过程实现精确的重现。因为考虑到本系统的复杂程度,通过单台计算机可以完成控制加仿真的任务,如若采用分布式的方法只会无谓地增加系统的复杂性,所以没有采用分布式仿真方法。

1 单件生产实验系统描述

一般离散制造单元由 4 部分组成:

- (1)加工中心;
- (2)物料自动传送系统;
- (3)计算机控制单元;
- (4)分布于各个加工中心前的缓冲区。

在离散制造单元中采用射频识别(Radio Frequency Identification,RFID)技术,可以自动跟踪在制品(正在制造的产品,包括构件、半成品、产成品等)的实时分布状况,进行产品和在制品的身份认证和标识。由于条件所限,本实验系统是对现实中的离散制造单元的模拟,本系统由可编程控制器(PLC)、上位机、基于 RFID 技术的数据采集终端设备和 GRB3014 四自由度机械手 4 部分组成。工件进入系统后,按照各自的工艺路线,完成各道工序的加工,然后离开系统。由 RFID 电子标签附着于工件的容器或工装上,读写器用于对电子标签的扫描,以确定工件当前的状态,包括工件当前所在机器以及当前所在缓冲区。而零件的搬运靠一个机械手来实现,机械手执行上位机中控制程序中下达的动作指令。PLC 用于采集生产现场的数据,并通过 ASM456 通信模块将采集到的数据每隔 1s 上

传至上位机。上位机通过获取的现场数据以及共享内存中的动态数据给机械手下达指令。

生产实验系统结构如图 1 所示。

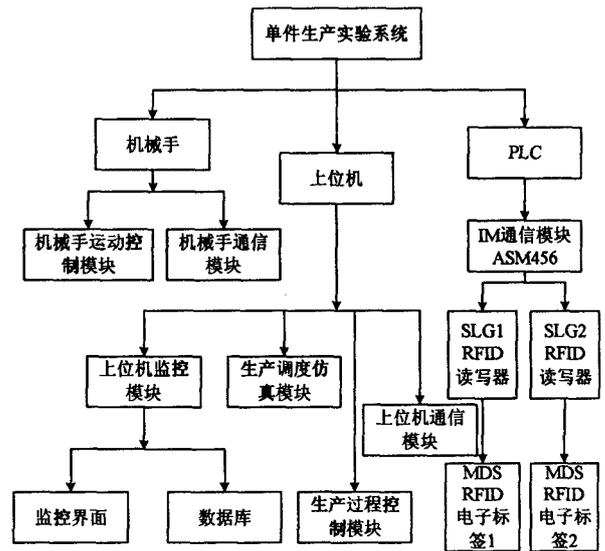


图 1 生产实验系统结构示意图

在图 1 中的生产过程控制模块以及生产调度仿真模块都是使用多线程技术来实现的。

在本单件生产实验系统中,由于条件所限,并没有真正的机器可供使用,通过建立机器线程来模拟机器。当需要加工工件时,由机器线程模拟真正的机器,向机械手下达动作指令,由机械手去完成相应的指令,因而机械手此时只是作为一个被控对象,并不为其建立独立的线程。工件本身对于机器来说其实也是一个被控对象,但是因为工件的到达动作是不由机器决定的,所以为了完成工件到达的动作,为工件建立了单独的线程,为此也为工件创建了类,并在类中实现了线程的创建以及线程的执行函数。通过以上的实物与仿真的结合,可以在现有的条件下比较好地对当前的单件生产实验系统加以描述。

2 基于多线程的生产过程仿真实现

线程是一组指令的集合,或者是程序的特殊段,它可以在程序里独立执行,也可以把它理解为代码运行的上下文。所以线程基本上是轻量级的进程,它负责在单个程序里执行多任务。通常由操作系统负责多个线程的调度和执行。

多线程是这样一种机制,它允许在程序中并发执行多个指令流,每个指令流都称为一个线程,彼此间互相独立。线程又称为轻量级进程,它和进程一样拥有独立的执行控制,由操作系统负责调度。线程状态如图 2 所示。

在本实验系统中,为了减少工件与工件之间,机器

与机器之间,以及机器与工件之间的耦合度,将每台机器或是每个工件都看作为一个独立的智能体(agent),并为其单独创建一个线程。线程与线程之间通过向消息队列发送消息以及接受消息的方法进行通信,而不是通过全局变量的方式通信,这样也是将线程间的相互影响降到最低,减少了相互间的耦合度。

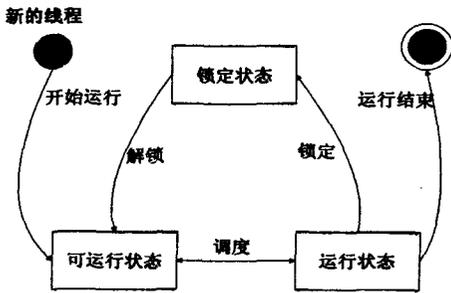


图 2 线程运行状态图

2.1 线程间的同步与互斥

在本实验系统中,通过共享内存单元的方式,实现了一个共享内存区,内存区中存放了生产现场的实时数据,因为一个线程就是一台机器,机器线程要下达工件的加工指令,则必须要访问共享内存区中的数据,为了能够对共享数据区中的生产现场实时数据的访问做到互斥进行,即不会出现多个线程同时访问同一数据区,采用了关键区(critical section)技术,关键区是一种具有排他性的共享资源,所有程序代码可以被同一个 critical section 保护起来。为了保证访问的排他性,一次只能有一个线程获准进入 critical section,通过将相应的 critical section 函数封装为类,可以在涉及到对数据区修改的操作中创建 critical section 对象,以确保对资源的互斥访问。

2.2 线程的流程

机器线程一旦开启就由挂起状态进入开启状态,然后通过 ChoosePart() 函数选出满足加工条件的工件,并向其对应的线程发送消息,表明该工件可以加工了,并改变机器的当前状态,即通过调用机器类中的相应函数将当前机器的状态由原来的空闲状态变为加工状态。工件线程在收到消息之后,若接受消息成功会反馈给机器线程一个消息,并会通过工件类的相关函数改变自身的状态,包括工件当前所在机器、当前所在缓冲区、当前工序,而机器线程在接受到反馈之后,通过 Sleep 函数让线程休眠(阻塞)一段时间以模仿加工过程,休眠完毕后会向工件线程发送加工完毕的消息。工件线程在接受到消息后会改变工件当前所在机器、当前所在缓冲区,然后反馈机器线程消息,机器线程在收到消息后,会继续下一个加工过程。线程与线程之间是并行的,也就是在同一时间的不同时间片轮流执

行。机器线程以及工件线程的流程图如图 3 和图 4 所示。

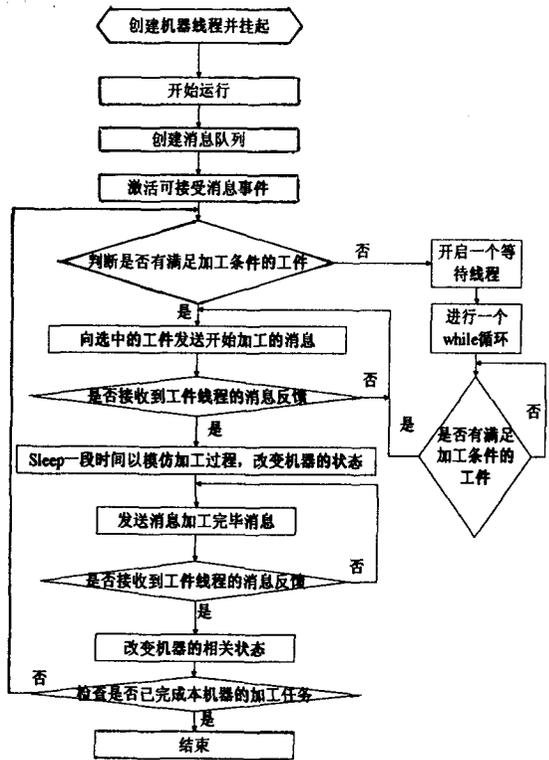


图 3 机器线程流程图

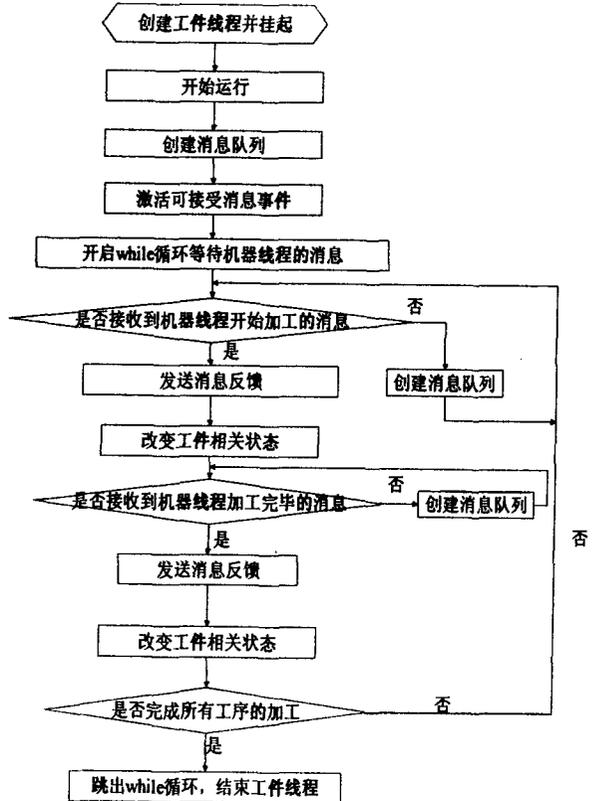


图 4 工件线程流程图

由于本调度仿真程序是在控制台下实现的,不同于 GUI 的是,在控制台下产生的线程是没有消息队列

的,为此必须通过 PeekMessage() 函数或是 GetMessage() 函数强制生成消息队列。为了保证在接收消息的线程的消息队列生成之后,发送线程再发送消息,所以采用事件来实现通信间的同步,即在消息队列生成后激活事件,而在发送线程中用 WaitForSingleObject() 函数等待该事件被激活才继续向下执行,以达到线程间通信的同步。采用消息队列的方式,可以实现异步通信,这样做不会造成程序的阻塞,将线程间的影响降到最低。

2.3 部分代码

为了每个对象对应一个线程,在机器类以及工件类中添加了线程的创建函数,以及线程的执行函数,一旦类对象中的线程创建函数被调用,线程就被激活开始并行地执行其执行函数。由于线程的执行函数必须是静态函数,所以通过向线程的创建函数中传递 this 指针,可以使用 this 指针完成对线程执行函数的调用。

2.3.1 机器线程的主体

先选中将要加工的工件,然后让线程休眠一段时间,以模拟工件的加工过程。而后改变机器和工件相关的状态。

```
unsigned Machine::MachineFunc()
{
    while(! m_machine_end)
    {
        int partID;
        //选择待加工工件
        partID=G_ChoosePart(m_machine_ID);
        if (partID==0)
        {
            //开启等待线程,等待直到有满足条件的工件,就结束线程
            _beginthreadex(NULL,0,WaitProc,(void *)m_machine_ID,0,
            NULL);
            //通知工件线程开始加工
            G_MachinePostMSG(m_machine_ID,partID,ps);
            //改变机器的相关状态
            g_machine[m_machine_ID].SetPart(partID);
            g_machine[m_machine_ID].SetIdleFlag(FALSE);
            //睡眠一段时间以模拟加工过程
            Sleep(g_machine_process_time[m_machine_ID-1]);
            //通知工件线程加工完毕
            G_MachinePostMSG(m_machine_ID,partID,ps);
        }
    }
}
```

2.3.2 工件线程的主体

接收到机器发送来的加工消息,改变当前状态,在等待到机器发送过来的当前工序加工完毕的消息则再次改变工件的状态。

```
unsigned Part::PartFunc()
```

```
{
    int machineID;
    //接受机器线程发送过来的消息
    G_PartRecvMSG(machineID,m_part_ID,ps)
    //改变工件相关状态
    g_part[m_part_ID].SetInBuffer(0);
    g_part[m_part_ID].SetInMachine(machineID)
    if(g_buffer[machineID].s_contain_part(m_part_ID))//若缓冲区中包含该工件,则工件离开该缓冲区,改变工件当前所在机器以及所在缓冲区
    {
        g_buffer[machineID].s_erase(m_part_ID);
        g_buffer[machineID].s_part_num--;
    }
    else if (ps==PROCESS_END)
    {
        g_part[m_part_ID].SetInBuffer(machineID+1);
        g_part[m_part_ID].SetInMachine(0);
        if(! g_buffer[machineID+1].s_contain_part(m_part_ID))//若缓冲区中不包含该工件,则工件进入该缓冲区,改变工件当前所在机器以及所在缓冲区
        {
            g_buffer[machineID+1].s_part_ID.push_back(m_part_ID);
            g_buffer[machineID+1].s_part_num++;
        }
    }
    return 0;
}
```

代码所占篇幅过多,只需给出主要部分,并加以必要的解释即可。

3 实验结果

以下是基于多线程技术的生产过程控制的实例。

实例 1:3 台机器,依次对一个工件进行加工。工件未到达时是在缓冲区 0 中,工件在并未完成所有工序时,一旦当前工序加工完毕进入公共缓冲区 1,之所以使用一个公共缓冲区是由于现场空间有限,为了防止读写器发生误读,所以所有的工件都共用一个缓冲区,如果工件完成其所有工序的加工会进入缓冲区 2。工件当前缓冲区为 300 时,表示当前工件不在缓冲区中。

当工件进入生产系统后,先要到读卡器上扫描一下,此时 PLC 上传给上位机的数据包中会发生相应的改变。到达的生产线的工件 ID 由 0 变为 1,即工件 1 到达生产系统。

上位机接收到数据包后会触发工件线程,工件线程会向机械手控制程序下达指令,将工件移入公共缓冲区。

如当工件1进入生产系统则此时共享内存区中的数据相应发生变化,工件1所在的缓冲区会由300变为1,这显示工件已经进入公共缓冲区。当上位机控制程序判别缓冲区中的工件满足加工条件,机器线程会向机械手控制程序下达加工指令,此时共享内存中的数据会发生相应变化,工件所在机器变为1,缓冲区变为300(即工件不在缓冲区中)。

当工件加工完毕,机器线程会向机器人下达加工完毕指令,由机械手将工件移动入公共缓冲区中,当前工件所有的工序都加工完毕,则将工件送入缓冲区2中。

当共享缓冲区中对应每个工件当前所在缓冲区都为2,即所有工件当前都已经进入缓冲区2,则说明所有的工件均已加工完毕。

4 结束语

文中基于单件生产实验系统的实际提出了用多线程的方法实现生产线的控制,鉴于单件生产实验系统没有机器的实际情况,通过将控制与仿真相结合,在没有机器的情况下用线程对机器进行模拟,实现了单件生产实验系统的控制。在实现过程中,通过采用多线程的方法真正实现控制与仿真的并发运行,比传统的

仿真方法,逻辑上更加清晰,相关变量的耦合度要小得多,不会因为局部的错误导致整个程序严重的逻辑错误,并且更加贴近生产线工作实际。通过采用文中的控制方法,可以较好地实现单件生产实验系统的实时控制。

参考文献:

- [1] 阎有运,郭晨宇,钟欠根.多线程技术在远程实验监控软件开发中的应用[J].河南理工大学学报,2007,26(1):56-59.
- [2] 张程,汪秉文.多线程技术在工业控制中的应用[J].工业控制计算机,2002,15(3):28-30.
- [3] 王维平,朱一凡,华雪倩,等.离散事件系统建模与仿真[M].长沙:国防科技大学出版社,1997.
- [4] 卫军胡,韩九强.离散事件系统仿真技术在制造系统调度中的应用[J].系统仿真学报,2000,12(1):27-30.
- [5] Kaubisch W H, Hoare C A R. Discrete event simulation based on communicating sequential processes [R]. Belfast, N. Ireland: Dep. Comput Sci The Queen's Univ, 1983: 39-65.
- [6] Bagrodia R L, Chandy K M, Isra J M. A message based approach to discrete event simulation [C]//IEEE SE213. [s. l.]: [s. n.], 1987: 654-665.
- [7] Misra J. Distributed discrete - event simulation [D]. Austin: Univ. of Texas at Austin, 1986: 39-85.
- [8] 曹阳,张维民.基于消息的离散事件仿真方法[J].小型微型计算机系统,2007,21(7):749-751.

(上接第196页)

造成高的时间损失率和低的生产效率。在当前经济危机的大背景下,这对于提高制造业企业国际竞争力是相当不利的。基于上述事实,有理由期待装配线平衡方法的下一步工程应用。

参考文献:

- [1] Hackman S T, Magazine M J, Wee T S. Fast, effective algorithms for simple assembly line balancing problems [J]. Journal of Operational Research, 1989, 37(6): 916-924.
- [2] Gutjahr A L, Nemhauser G L. An algorithm for the line balancing problem [J]. Management Science, 1964, 11(2): 308-315.
- [3] 梁燕,金焯.基于工位约束快速启发式算法的混合装配线分段优化[J].上海交通大学学报,2007,41(9):1501-1505.
- [4] Scholl A, Becker C. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing [J]. European Journal of Operational Research, 2006, 168(3): 666-693.
- [5] Lapierre S D, Ruiz A, Soriano P. Balancing assembly lines with tabu search [J]. European Journal of Operational Research, 2006, 168: 826-837.
- [6] 吴君华,夏巨湛,曹山河. ALB问题的数学模型及其优化算法的研究 [J]. 系统仿真学报, 1999, 11(5): 358-360.
- [7] Baybars I. A survey of exact algorithms for the simple assembly line balancing problem [J]. Management Science, 1986, 32: 900-932.
- [8] Johnson R V. Optimally balancing large assembly lines with FABLE [J]. Management Science, 1988, 34: 240-253.
- [9] Klein R, Scholl A. Maximizing the production rate in simple assembly line balancing - a branch and bound procedure [J]. European Journal of Operational Research, 1996, 91: 367-385.
- [10] 张文修,梁怡.遗传算法的数学基础[M].西安:西安交通大学出版社,2003.
- [11] 高建坤,周俊,许正良.自适应遗传算法在服装缝纫装配线平衡中应用[J].江苏纺织,2008(12):50-54.