

基于云控制系统的监控系统设计与实现

谭浩然^{1†}, 黄志武¹, 吴敏²

(1. 中南大学信息科学与工程学院, 长沙 410083; 2. 中国地质大学自动化学院, 武汉 430074)

摘要: 基于云计算的控制系统(简称云控制系统)由以云节点形式存在的控制器、传感器和执行器组成,各节点之间通过网络交换数据. 针对云控制系统中云节点分布式运行和矩阵类型数据被广泛使用使系统运行状态难以观测的问题,设计并实现了一种云监控系统. 该系统采用客户端/服务器架构,使用重构技术将性能稳定、功能丰富的控件以API的形式集成在系统中,满足了自定义监控界面的需求,提高了系统的可靠性. 基于TCP/IP协议实现同云节点的双向通信,在获得云节点数据的同时支持数据在线调整,以达到连续监控的目的. 通过设置线程池,以异步通信的方式实时获取并显示云节点的数据,达到监控与云节点数据同步. 针对云计算中被广泛使用的矩阵类型数据,提供全元素的监控支持. 最后,以直流电机系统为被控对象,基于阿里云平台的动态仿真和本地云平台的实际实验,验证了所提出云监控系统的可行性和实用性,促进和完善控制系统与云计算结合在实际中的应用.

关键词: 云控制系统; 监控系统; 异步通信; 双向通信; 矩阵监控; 自定义监控界面

中图分类号: TP277

文献标志码: A

Design and implementation of supervisory system based on cloud control systems

TAN Hao-ran^{1†}, HUANG Zhi-wu¹, WU Min²

(1. School of Information Science and Engineering, Central South University, Changsha 410083, China; 2. School of Automation, China University of Geosciences, Wuhan 430074, China)

Abstract: The control system based on cloud computing (cloud control systems) includes controllers, actuators and sensors which exist as cloud nodes. Information is exchanged among cloud nodes via the network. To solve the problem that the cloud control system is difficult to observe, which results from the wide application of the cloud node distributed running and matrix type data, a cloud supervisory system is designed and implemented. In this paper, the client/server architecture of this system is described. The stability and richly functional controls are integrated into the system as the API by reconstructing, which meet the requirement of customize supervisory interface and improve the reliability of systems. The system also supports tuning parameters online to achieve consecutive observation through the duplex channel based on the TCP/IP protocol. The data of cloud nodes is obtained and displayed in real time through asynchronous communication. All the elements of the matrix which is a common data type in cloud control systems are supported to supervise. Finally, for the motor plant, the Internet cloud-based dynamic simulation and Intranet cloud-based experiment demonstrate the feasibility and practicability of the designed system, and further promote and improve the control system combined with cloud computing in practical application.

Keywords: cloud control system; supervisory system; asynchronous communication; duplex communication; matrix supervise; custom supervisory interface

0 引言

随着互联网和计算机技术的迅速发展,云计算^[1-3]已经进入了我们的日常生活. 目前,云计算已经超越原来的产品概念,成为了一种服务. 它提供一个可动态伸缩和资源虚拟化的共享工具,如数据访问、计算、软件和存储服务等. 作为云计算的重要组

成部分,云监控是云计算中网络分析、系统管理、数据存储和复杂运算等服务的前提,对提高云计算的服务质量发挥着重要作用^[4-5],吸引了大批企业和学者对其进行研究. Google提供的App engine system status dashboard使开发商能够监控其云平台上的服务^[6],Hyperic公司开发的CloudStatus可以监控Amazon服

收稿日期: 2018-01-10; 修回日期: 2018-03-20.

基金项目: 国家自然科学基金项目(61672537).

责任编辑: 夏元清.

[†]通讯作者. E-mail: tanhaoran@csu.edu.cn.

务.文献[7]提出并实现了一种受信任的云监控系统框架.针对室内安全,文献[8]设计了一种智能嵌入式云监控系统.此外,云监控系统在城市资源、智能农业和医疗健康方面同样得到了应用^[9-11].

在控制领域,云计算卓越的计算能力、良好的通用性和高可扩展性为大规模控制系统的实现提供了解决方案.目前,云控制系统正作为网络控制系统的扩展被讨论^[12-13].应该看到云计算仍然存在许多的问题^[14-16],主要表现为数据在云网络传输过程中出现的延时、乱序和丢包^[17-18]现象对控制效果和稳定性的影响.针对这些问题,已有一些学者提出了自己的解决方法.夏元清教授采用基于数据的预测控制算法解决网络诱导时延对控制系统的影响^[19].文献[20]提出一种网络化预测控制策略,以补偿数据传输延时和丢包对多智能体云控制系统的影响.王彩路等^[21]将并行处理系统的时间周期、吞吐率和任务完成时间作为性能指标,采用极大-加代数方法分析和优化云控制系统并行处理能力.

然而,目前关于云控制系统的讨论大多为理论研究,工程实现较少,限制其应用的原因主要包括:从数据类型角度,为解决云控制系统中存在的延时、乱序和丢包问题,上述控制策略均引入了状态空间的概念,使向量或者矩阵类型数据成为了计算和传输的主体,全面监测控制系统运行数据对控制系统分析和控制策略设计至关重要,以监控标量数据设计的现有云监控系统,无法应用在云控制系统中,进而无法满足云控制系统的工程需要;在云平台选取方面,现有支持云控制系统的监控系统^[22],需针对指定云平台服务商进行二次开发,具有平台局限性且操作复杂.作为监控系统的重要部分,在传统监控系统^[23-25]和常见云监控系统中,所采用显示控件通常来自于个

人或者小企业,其功能简单,维护、更新能力有限,稳定性无法得到保证,不具有监控界面自定义、控制系统参数在线修改的功能.

与现有的监控系统相比,本文所设计实现的云监控系统具有以下特点:1)能够通过网络获取云控制系统中所有云节点的信息,实现基于网络的远程监控;2)支持阿里云和Google云等常见云平台,云环境更加宽松;3)支持云控制系统中矩阵类型数据全元素监控,达到云控制系统信息全面获取的工程实现;4)所采用显示控件来自National Instruments(NI)公司开发和维护的控件库,具有功能丰富、性能稳定的特点;5)允许监控界面自定义,支持控制系统参数在线修改.虽然部分技术并非本文首次提出,但是以现有的认知,鲜有运用在对云控制系统的监控中.

1 云监控系统设计

为了实现所提出云监控系统的上述特点,在云监控系统设计中采用了多线程和重构等技术.由于.NET系统组件伴随着Windows系统而普及,以及C#自身面向对象的优点,本系统采用C#进行开发,假设大部分读者对C#有基本的了解.

1.1 基于.NET框架的系统结构建立

本系统采用客户端/服务器架构,如图1所示,运行该监控系统的上位机作为客户端,云控制系统节点作为服务器端,两者之间通过网络建立通信通道,实现了基于网络的远程监控.本系统结构主要由GUI模块、数据工作区模块和监控工作区模块组成.各部分功能简介如表1所示.GUI模块与数据工作区模块和监控工作区模块之间不能直接进行信息交互,必须通过在GUI模块中实例化以及调用事件完成,交互过程将在后续部分阐述.

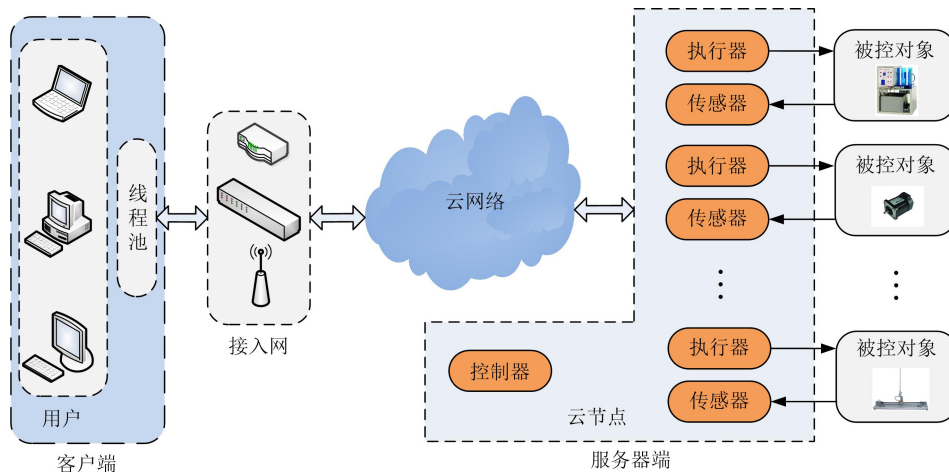


图1 系统架构

表1 系统组成

模块名称	功能
GUI模块	系统初始化;数据结构建立;数据包的处理;接收、拆分、打包和发送;通信通道建立;监控启动和停止与数据工作区和监控工作区模块交互
监控工作区模块	动态创建控件;传递控件修改显示数值事件
数据工作区模块	数据分类;树形多级列表显示信号和参数

1.2 控件设计与实现

允许用户通过动态创建NI控件和设置控件属性的方式自定义设计监控界面,是本监控系统的一个重要特点.为实现该特点,以API形式集成到GUI的NI控件必须经过重构,部分NI控件如图2监控区窗体区域所示.所有重构的NI控件均继承于.NET的Controls类,此外还需要继承NI控件库中对应的控件原生类.

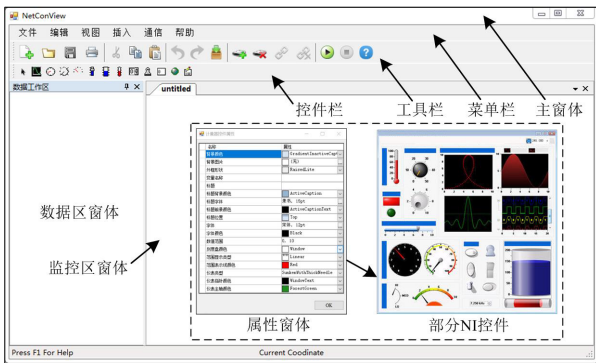


图2 云监控系统界面

以计量器控件的重构为例,首先,创建一个新计量器类,该类继承于NI控件库的计量器原生类,拥有该类公共和保护属性特性.为了实现对鼠标、新计量器控件的自定义操作,在新计量器类中,重写控件对鼠标的响应事件,例如将鼠标按下动作触发的MouseDown()事件重写成OMouseDown()事件.其次,为了区分控件对不同鼠标状态的响应,创建一个枚举类型数据表示鼠标不同状态,不同状态所代表的具体行为在MousePointPosition()方法中描述.为了实现对该控件颜色、显示范围和类型等属性的动态设置,创建一个窗体类GaugeProperty存放控件的属性.在窗体类中,通过PropertyEditorSource()方法获取计量器控件的一些常用属性,并将这些属性通过DataGridView控件以列表的形式显示在窗体中,新计量器控件的属性窗体如图2所示.在监控工作区动态创建该控件后,用户可以调用属性窗体设置控件的属性,设置完成后,控件将自动调用PropertyInfo.SetValue()方法保存新的属性值,实现属性的更新.为了实现数据在控件上的显示,在新计量器控件类中创

建UpdateData()方法,用来接收来自GUI的数据,并将该数据的数值赋给控件的value属性,从而达到显示的目的.其他NI控件的重构细节不完全相同,但重构方式相似.

1.3 监控界面和数据的保存

监控界面的保存、恢复及监控数据的单独存储给用户的使用带来很大便利.对监控界面的保存即对监控界面中控件的属性、尺寸和位置的保存,可通过序列化与反序列化技术实现.在控件类调用.NET库中可实现序列化行为的接口GetObjectData(),并在该接口中通过SerializationInfo.AddValue()方法将需要保存的属性逐一存储.监控界面保存时,首先创建一个序列化写入流,同时指定写入的文件;然后创建二进制格式器,在遍历监控界面控件的同时通过二进制格式器的Serialize()方法序列化属性,并写入所创建流中,实现监控界面保存在指定格式文件中.为加载已保存的监控界面,需要在控件类的构造方法中调用SerializationInfo.GetValue()方法逐一提取控件属性.监控界面恢复时,首先创建一个序列化读取流,同时指定读取的文件;然后创建二进制格式器,通过格式器的Deserialize()方法反序列化监控界面文件,将文件中存储的控件逐一恢复.

云控制系统运行过程中会产生大量的数据,这些数据可用于模型建立、控制器设计和系统分析等方面,具有很高的使用价值.为了实现数据的保存,首先在每个控件类中定义List<double>类型数据DataList,当数据在控件上显示的同时调用DataList的Add()方法将数据添加至DataList临时列表中;然后在线程池中创建新线程,将临时列表中的数据存储至数据库中.为了满足进一步处理所保存数据的需求,可将数据库中的数据单独保存到指定的.txt文件中.较为特殊的是,曲线类型控件中通常需要同时显示多条曲线,因此,在这类控件中使用List<List<double>>代替List<double>类型数据.

为满足NetCon实验平台^[26]基于网络监控的需求,根据该平台通信协议实现了一个云监控系统版本,此系统主界面如图2所示.

1.4 系统组成

由图2可见,系统主界面主要包括实例GUI模块的主窗体、实例数据工作区模块的数据区窗体和实例监控工作区模块的监控区窗体.主窗体所有按钮根据用途归类到菜单栏、工具栏和控件栏中.菜单栏设置有编辑、视图和通信等6个子菜单栏.每个子菜单栏包含一个下拉列表,通过下拉列表中的选项可实现

本系统所有功能. 工具栏包括新建设备、添加设备、连接、运行等常用功能按钮. 控件栏包含所有经过重构的NI控件,它们以按钮API的形式呈现. 将鼠标左键点击NI控件按钮,然后在监控区按住鼠标左键拖拽,可创建相应的控件. 若选中监控区中的控件并按住鼠标左键进行拖拽,可以拖动控件在监控区自由移动,进而实现监控界面的自定义. 鼠标右击所创建的控件,则会弹出图2中所示控件属性窗体.

1.5 通信通道建立

本系统通过建立监控界面与云节点之间的通信通道实现两者之间的数据交换. 通信通道结构如图3所示. 该通信通道有两个任务:建立NI控件与云节点之间的数据同步和传递用户远程操作指令(调整参数等)给云节点.

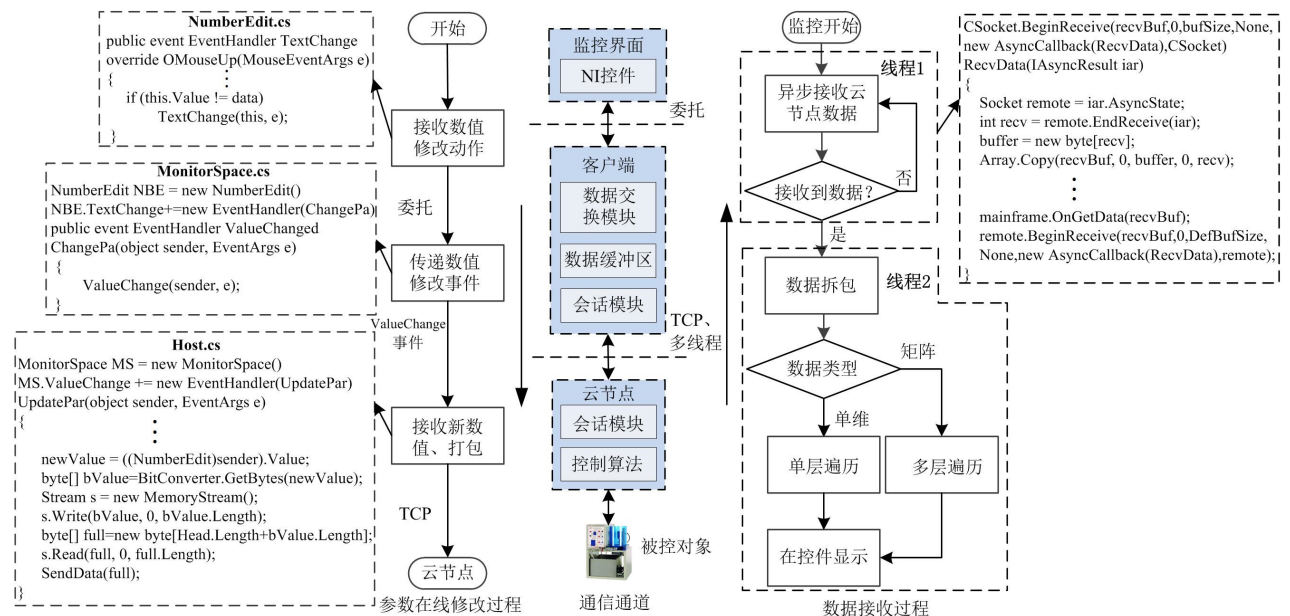


图3 通信过程

1.6 监控界面数据更新

当多个任务分配给单个线程时,任务将按照顺序依次执行,导致当前任务的响应变得“迟钝”,降低了用户的体验. 云计算强大的计算能力提高了云控制系统的更新频率,显然,单线程的处理方式难以满足对数据实时监控的需求. 为解决该问题,本系统设置了线程池,将系统常规任务与数据抓取、显示任务指派给不同的线程同时执行. 具体实现方式为,采用Socket异步通信的方式从线程池中调用两个线程,分别执行从云节点接收数据包和将数据进行拆分并显示在监控界面中,数据包接收和显示过程如图3所示. 为了实现对数据包的实时接收,采用递归循环的方式执行回调方法RecvData(),让其保持接收状

每个运行了控制算法的云节点,均会相应生成一个会话模块. 当本监控系统根据云节点的IP信息与之关联后,监控系统的会话模块与云节点的会话模块之间将自动建立数据传输通道,通信客户端通过该通道接收来自云节点的实时数据,并临时存储在数据缓冲区中. 当NI控件中关联了需要监控的参数或信号后,客户端和监控界面将自动建立数据传输通道,数据缓冲区中的数据通过数据交换模块以委托的方式传递给NI控件并显示. 为了将云控制系统中产生的大量数据实时显示在控件中,在监控系统和云节点之间设置有数据缓冲区,用以解决云节点通信环境的问题. 此缓冲区采用先进先出的结构临时存放实时数据. 当云节点通信环境良好时(如局部云),实时数据能够快速被控件获取. 允许通信环境出现较小的延时,但是需要保证数据流的连续性.

态. 借助云计算强大的计算能力,结合网络传输的优点,使得现代控制理论中比传递函数更适合控制系统描述的状态方程在云控制系统中得以实现. 状态方程基于矩阵数据类型,如果一个数据包被成功接收,则它将作为OnGetData()方法的实际参数传递给主窗体. 主窗体收到数据包后,对其进行拆分,首先根据每个数据的行列标识符判断数据类型,若为单维数据,则直接去除行列标识符等信息,并加上云节点名称和参数或信号名称成为绝对路径,再通过绝对路径寻找与之关联的NI控件,并将数据作为NI控件中UpdateData()方法的实际参数赋值给控件的显示属性,从而在控件中显示. 若为矩阵数据,则使用多层for循环分别遍历数据的行和列,将矩阵数据每个元

素拆分成单维数据类型,然后按照单维数据的方式传送给相应的NI控件进行显示。

在实际应用中,被控对象具有动态非线性特点,致使控制系统参数设定难以一步到位,虽然可通过反复离线调整参数后再将控制系统上传至云服务器中,但是这无疑增加了操作步骤,同时也无法连续观察控制系统对不同参数的响应过程。作为本系统的一个重要功能,允许在线调整控制系统参数。为此,本系统提供了具有数值修改功能的NI控件,如数值输入控件。控件显示数值被主动修改的过程与数值显示不同。监控区动态创建控件与主窗体程序处在不同线程,不同线程之间不能直接交互信息,因此采用委托作为两者之间的“桥梁”,将修改数值动作传送给主窗口并最终实现云节点中相应参数值的更改,整个传递过程如图3中相应部分所示。以数值输入控件的操作为例,在此控件类中添加一个TextChange事件(委托的一种形式),并在OMouseUp()方法中调用该事件,该方法将在修改控件显示数值时触发。在监控区类中,创建一个ChangePa()方法,并绑定数字输入控件实例的TextChange事件。数据包的封装和向云节点发送数据包的过程在主窗体中进行,因此需要建立监控区窗体与主窗体之间的联系。首先,在监控区窗体类中创建ValueChange事件,并在ChangePa()方法中调用;然后,在主窗体类中创建一个UpdatePar事件,并绑定监控区窗体实例的ValueChange事件;最后,在UpdatePar()中将修改后的数值封装成数据包,通过TCP发送给云节点完成相应参数值的更改。

2 实验验证和分析

为了验证云监控系统在实际应用中的可行性,以直流电机系统为被控对象,分别建立基于互联网云和局域网云的控制策略。选用文献[27]所提出的基于神经网络的网络化预测控制策略,补偿数据在网络传输中的诱导延时和丢包。该控制策略主要由预测控制序列生成器和网络延时补偿器两部分组成。SISO离散非线性模型可描述成

$$y(k+1) = f(y(k), \dots, y(k-n_y), u(k), \dots, u(k-n_u)). \quad (1)$$

2.1 控制策略

假设前向和反馈通道延时分别为 τ_{ca} 和 τ_{sc} ,且 $N \geq \tau_{ca} + \tau_{sc}$,所有在网络中传输的数据均带有时间戳。传感器将被控对象一个时间段内的历史输入输出数据通过反馈通道传送给预测控制序列生成器,预测序列生成器根据被控对象的历史数据、基础控

制器和被控对象模型预测出 N 步控制输出序列

$$U(k-\tau_{sc}|k-\tau_{sc}) = \begin{bmatrix} u(k-\tau_{sc}|k-\tau_{sc}) \\ u(k-\tau_{sc}+1|k-\tau_{sc}) \\ \vdots \\ u(k-\tau_{sc}+N|k-\tau_{sc}) \end{bmatrix}, \quad (2)$$

并通过前向通道传送给延时补偿器。

为了补偿延时,延时补偿器收到如下序列后:

$$U(k) = \begin{bmatrix} u(k-\tau_{sc}-\tau_{ca}|k-\tau_{sc}-\tau_{ca}) \\ u(k-\tau_{sc}-\tau_{ca}+1|k-\tau_{sc}-\tau_{ca}) \\ \vdots \\ u(k-\tau_{sc}-\tau_{ca}+N|k-\tau_{sc}-\tau_{ca}) \end{bmatrix}, \quad (3)$$

从中选取对当前时刻的预测值作为被控对象的控制输入,即

$$u(k) = u(k|k-\tau_{sc}-\tau_{ca}). \quad (4)$$

在实际中,被控对象具有动态和非线性的特点,其精确模型难以获得,因此采用多层神经网络在线辨识方法获得被控对象动态模型。用分段线性化和阶跃响应方法对电机系统进行建模,获得数学模型

$$G_p(z) = \frac{18.3296z^{-10} + 65.2067z^{-11}}{1 - 0.3383z^{-1} - 0.3737z^{-2}}. \quad (5)$$

该数学模型仅用于设计基础控制器和仿真时产生被控对象输出数据,不参与上述预测控制序列的计算。无网络环境时,设计一个满足条件的基础控制器

$$G_c(z) = \frac{0.001 - 0.0008z^{-1}}{1 - z^{-1}}. \quad (6)$$

2.2 互联网云监控

建立图4(a)所示基于互联网云的直流电机控制系统。互联网云由位于中国香港的阿里云服务器和位于美国硅谷的阿里云服务器组成。

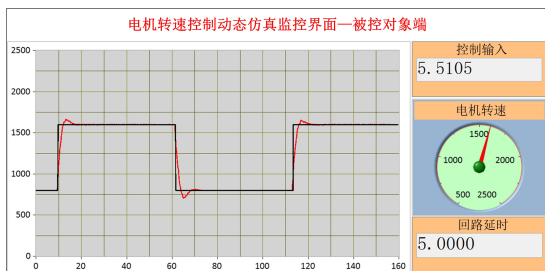
搭建电机控制系统simulink仿真框图,其中预测控制序列生成器位于控制器端框图,延时补偿器位于被控对象端框图,设置采样时间为0.1s,测得香港和硅谷阿里云服务器之间的回路延时为4~5步(0.4s~0.5s),设置预测步数 $N=9$,选择4-5-1的神经网络结构,选取sigmoid函数作为激活函数,通过离线训练选取一组输入层与隐含层以及隐含层与输出层之间的加权系数作为初始值。通过NetCon平台将控制器端和被控对象端框图分别编译下载至位于香港和硅谷阿里云服务器中运行,互联网云控制系统监控界面分别如图4(b)和图4(c)所示。图4(b)中阶跃信号曲线表示参考输入,动态曲线表示电机模型输出。图4(c)的曲线分别表示预测控制序列(3)的前两个元素值。



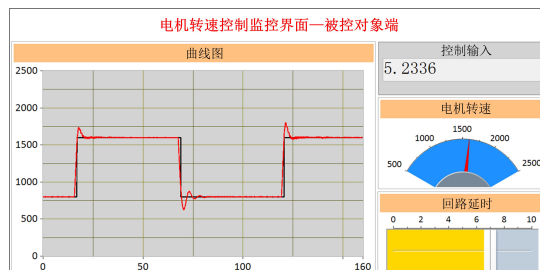
(a) 互联网云结构



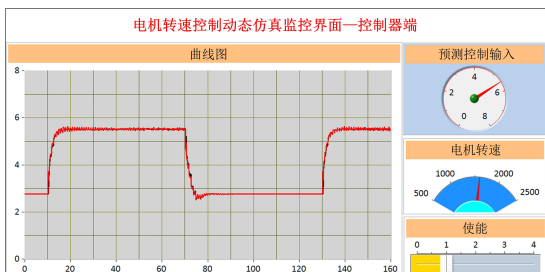
(a) 局域网云电机实际控制系统实物图



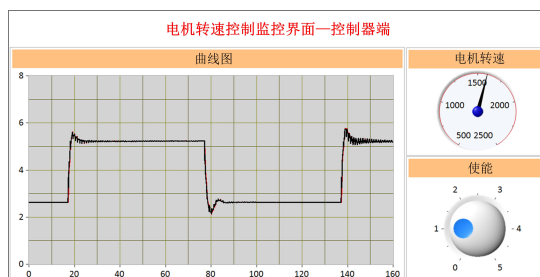
(b) 被控对象端监控界面



(b) 被控对象端监控界面



(c) 控制器端监控界面



(c) 控制器端监控界面

图4 基于互联网云监控

图5 基于局域网云监控

2.3 局域网云监控

建立如图5(a)所示基于局域网云的直流电机实际控制系统.局域网云主要由两台网络控制器组成,其中网络控制器1作为控制器,网络控制器2作为传感器/执行器,连接好电机系统和网络控制器之间的物理线路,构成实际控制系统.

使用NetCon平台搭建控制系统simulink实际框图.可看成在仿真框图的基础上将电机系统数学模型替换为网络控制器硬件接口模块,设置5步回路延时,其他参数与互联网云监控时相同.将被控对象端和控制器端框图分别编译下载至网络控制器1和网络控制器2中.局域网云控制系统监控界面分别如图5(b)和图5(c)所示.通过滑动条控件将回路延时调整为7步,图5(b)中阶跃信号曲线表示给定转速,动态曲线表示电机在回路延时变化下的实际转速.图5(c)的曲线分别表示网络控制器2实际接收预测控制序列前两个元素值的变化过程.

3 结论

本文针对云控制系统中云节点分布式运行、矩阵类型数据被广泛使用,使系统运行状态难以观测的

问题,设计并实现了一种云监控系统,阐述了系统的客户端/服务器结构,能够基于网络远程实时监控云节点数据的功能,支持矩阵类型数据全元素监控的特点,所集成NI控件性能稳定和功能丰富的优点,允许用户动态自定义监控界面的便捷性,支持在线调整参数的必要性,对监控数据单独存储的扩展性及相关部门的具体实现细节.通过基于互联网云的电机控制动态仿真和局域网云的电机控制实验,验证了本系统的可行性和实用性.本文不但为监控系统的开发提供了一种解决方案,更促进和完善了控制系统与云计算结合在实际中的应用.

随着云控制系统的发展,对监控系统提出了更高的要求,除对云节点进行二维监控外,开始寻求其3D形式和视频模式的监控,以获得更加直观和真实的监控效果,这些将作为下一步的研究内容.

参考文献(References)

[1] Qian L, Luo Z, Du Y, et al. Cloud computing: An overview[C]. Proc of the IEEE Int Conf on Cloud Computing. Berlin: Springer, 2009: 626-631.

[2] 陈康, 郑纬民. 云计算: 系统实例与研究现状[J]. 软件学报, 2009, 20(5): 1337-1348.

- (Chen K, Zheng W M. Cloud computing: System instances and current research[J]. *J of Software*, 2009, 20(5): 1337-1348.)
- [3] Armbrust M, Fox A, Griffith R, et al. A view of cloud computing[J]. *Communications of the ACM*, 2010, 53(4): 50-58.
- [4] Aceto G, Botta A, De Donato W, et al. Cloud monitoring: A survey[J]. *Computer Networks*, 2013, 57(9): 2093-2115.
- [5] Fahad A M, Ahmed A A, Kahar M N M. The importance of monitoring cloud computing: An intensive review[C]. *Proc of the TENCON 2017 IEEE Region 10 Conf. Penang: IEEE*, 2017: 2858-2863.
- [6] Abrahão S, Insfran E. Models@ runtime for monitoring cloud services in Google app engine[C]. *Proc of the 2017 IEEE World Congress on Services. Honolulu: IEEE*, 2017: 30-35.
- [7] Zou D, Zhang W, Qiang W, et al. Design and implementation of a trusted monitoring framework for cloud platforms[J]. *Future Generation Computer Systems*, 2013, 29(8): 2092-2102.
- [8] Dong B T, Lee C M, Chiu Y C, et al. An intelligent embedded cloud monitoring system design[C]. *Proc of the 2017 Int Automatic Control Conf. Pingtung: IEEE*, 2017: 1-4.
- [9] Pingali S. Cloud computing and crowdsourcing for monitoring lakes in developing countries[C]. *Proc of the 2016 IEEE Int Conf on Cloud Computing in Emerging Markets. Bangalore: IEEE*, 2016: 161-163.
- [10] Roopaei M, Rad P, Choo K K R. Cloud of things in smart agriculture: Intelligent irrigation monitoring by thermal imaging[J]. *IEEE Cloud Computing*, 2017, 4(1): 10-15.
- [11] Abawajy J H, Hassan M M. Federated internet of things and cloud computing pervasive patient health monitoring system[J]. *IEEE Communications Magazine*, 2017, 55(1): 48-53.
- [12] Yuanqing Xia. Cloud control systems[J]. *IEEE/CAA J of Automatica Sinica*, 2015, 2(2): 134-142.
- [13] Yuanqing Xia, Yongming Qin, Di-Hua Zhai, et al. Further results on cloud control systems[J]. *Science China: Information Sciences*, 2016, 59(7): 232-236.
- [14] Xiao Z, Xiao Y. Security and privacy in cloud computing[J]. *IEEE Communications Surveys and Tutorials*, 2013, 15(2): 843-859.
- [15] Kapil D, Tyagi P, Kumar S, et al. Cloud computing: Overview and research issues[C]. *Proc of the 2017 Int Conf on Green Informatics. Fuzhou: IEEE*, 2017: 71-76.
- [16] Subashini S, Kavitha V. A survey on security issues in service delivery models of cloud computing[J]. *J of Network & Computer Applications*, 2011, 34(1): 1-11.
- [17] Ma L, Xia Y, Ali Y, et al. Engineering problems in initial phase of cloud control system[C]. *Proc of the 36th Chinese Control Conf. Dalian: IEEE*, 2017: 7892-7896.
- [18] 夏元清. 云控制系统及其面临的挑战[J]. *自动化学报*, 2016, 42(1): 1-12.
- (Xia Y Q. Cloud control systems and their challenges[J]. *Acta Automatica Sinica*, 2016, 42(1): 1-12.)
- [19] Xia Y Q, Xie W, Liu B, et al. Data-driven predictive control for networked control systems[J]. *Information Sciences*, 2013, 235(6): 45-54.
- [20] Guo-Ping Liu. Predictive control of networked multiagent systems via cloud computing[J]. *IEEE Trans on Cybernetics*, 2017, 47(8): 1852-1859.
- [21] 王彩璐, 陶跃钢, 杨鹏, 等. 云控制系统并行任务分配优化算法与并联控制[J]. *自动化学报*, 2017, 43(11): 1973-1983.
- (Wang C L, Tao Y G, Yang P, et al. Parallel task assignment optimization algorithm and parallel control for cloud control systems[J]. *Acta Automatica Sinica*, 2017, 43(11): 1973-1983.)
- [22] Rossignaux F, Lefevre L, Gelas J P, et al. A generic and extensible framework for monitoring energy consumption of OpenStack clouds[C]. *Proc of the 2014 IEEE 4th Int Conf on Big Data and Cloud Computing. Sydney: IEEE*, 2014: 696-702.
- [23] Wang J, Zhang Y. Monitoring system of machine tools based on the InTouch[C]. *Proc of the 2013 Int Conf on Mechanical and Automation Engineering. Jiujiang: IEEE*, 2013: 70-72.
- [24] 桂卫华. 基于机器视觉的矿物浮选过程监控技术研究进展[J]. *自动化学报*, 2013, 39(11): 1879-1888.
- (Gui W H. Machine-vision-based online measuring and controlling technologies for mineral flotation-A review[J]. *Acta Automatica Sinica*, 2013, 39(11): 1879-1888.)
- [25] Yangang X, Han W, Xingqi L, et al. Monitor system design for machine electric spindle based on MCGS[C]. *Proc of the 2010 WASE Int Conf on Information Engineering. Beidaihe: IEEE*, 2010, 3: 248-252.
- [26] Baoran An, Liu G P. Networked real-time controller based on PC/104[C]. *Proc of the 2012 10th World Congress on Intelligent Control and Automation. Beijing: IEEE*, 2012: 831-834.
- [27] Pang Z H, Liu G P. Model-based recursive networked predictive control[C]. *Proc of the 2010 IEEE Int Conf on Systems Man and Cybernetics. Istanbul: IEEE*, 2010: 1665-1670.

作者简介

谭浩然(1989—), 男, 博士生, 从事网络化控制、数据驱动的研究, E-mail: tanhaoran@csu.edu.cn;

黄志武(1966—), 男, 教授, 博士生导师, 从事智能控制、故障诊断、机车控制系统等研究, E-mail: hzw@csu.edu.cn;

吴敏(1963—), 男, 教授, 博士生导师, 从事过程控制、鲁棒控制、智能系统等研究, E-mail: wumin@cug.edu.cn.

(责任编辑: 孙艺红)