

# 云控制系统不确定性分析与控制器设计方法

关守平<sup>1</sup> 王梁<sup>2</sup>

**摘要** 云控制系统 (Cloud control system, CCS) 是云计算与物理系统的融合, 由于云计算中资源是动态的, 因此云计算的加入使得云控制系统具有很大的不确定性. 本文给出一种典型的云控制系统结构, 通过将不确定性划分为云端不确定性和网络端不确定性, 有效简化了云控制系统不确定特性分析和建模. 针对典型的时延不确定性问题, 将云控制系统时延划分为云端时延和网络端时延, 进行了 MapReduce 模型下多计算节点云端时延分析, 同时进行了云控制结构下网络端时延分析, 两者结合实现了云控制系统的前向通道和反馈通道的时延建模. 基于所建立的云控制系统时延模型, 应用极点配置方法设计了云控制器算法, 包括观测器的设计和控制律的设计, 从而保证了闭环系统的稳定性. 对本文设计的云控制器算法进行了仿真验证, 结果表明考虑时延特性的控制器设计明显提升了云控制系统的控制性能.

**关键词** 云控制系统, 云计算, 时延特性, 极点配置

**引用格式** 关守平, 王梁. 云控制系统不确定性分析与控制器设计方法. 自动化学报, 2022, 48(11): 2677-2687

**DOI** 10.16383/j.aas.c190529

## Uncertainty Analysis of Cloud Control System With Its Controller Design

GUAN Shou-Ping<sup>1</sup> WANG Liang<sup>2</sup>

**Abstract** Cloud control system (CCS) is the integration of cloud computing and physical systems. Due to the dynamic resources added, the CCS possesses more uncertainties than other control systems. In this paper, a typical structure of CCS is presented, where the uncertainties are divided into the cloud uncertainty and the network uncertainty, to effectively simply the uncertainty analysis and modeling. Aiming at the typical uncertainty of time-delay, it is divided into the cloud-end time-delay and the network-end time-delay. The first one is carried out under the condition of multi-computing nodes in the MapReduce model, and the second one is also proceeded under the cloud control structure, then the CCS model with the forward channel time-delay and the feedback channel time-delay is realized by superimposing the above mentioned two parties. The controller is then designed based on the established time-delay model using the pole placement method, including the observer and the control law, thus the stability of closed-loop control for CCS is guaranteed. The simulation results show that the control performance of CCS is improved obviously by considering the time-delay characteristics.

**Key words** Cloud control system (CCS), cloud computing, time-delay characteristics, pole placement

**Citation** Guan Shou-Ping, Wang Liang. Uncertainty analysis of cloud control system with its controller design. *Acta Automatica Sinica*, 2022, 48(11): 2677-2687

新一代信息通信技术快速发展并与制造业的深度融合, 引发制造业制造模式、制造流程、制造手段、生态系统等的重大变革, 传统工业控制技术也面临新的变化. 目前, 工业控制上常用的控制系统有集中控制系统、分散控制系统、总线控制系统和网络控制系统, 这些控制系统有其各自的优势和适用范围<sup>[1-3]</sup>, 但是其共同的问题是建设和运行维护成本较高, 控制算法的更新替换不灵活. 近年来, 云计算 (Cloud computing) 的兴起给工业控制带来了新的思路.

云计算是通过网络提供可伸缩的廉价的分布式计算能力, 是并行计算 (Parallel computing)、分布式计算 (Distributed computing) 和网格计算 (Grid computing) 的发展<sup>[4]</sup>. 云计算是虚拟化 (Virtualization)、效用计算 (Utility computing)、将基础设施作为服务 IaaS (Infrastructure as a service)、将平台作为服务 PaaS (Platform as a service) 和将软件作为服务 SaaS (Software as a service) 等概念混合演进并跃升的结果<sup>[5]</sup>, 能达到 10 万亿次的运算能力. 在实际系统中, 云计算首先是将大量的计算机通过

收稿日期 2019-07-15 录用日期 2019-10-11  
Manuscript received July 15, 2019; accepted October 11, 2019  
国家自然科学基金 (62173072) 资助  
Supported by National Natural Science Foundation of China (62173072)  
本文责任编辑 郭戈  
Recommended by Associate Editor GUO Ge  
1. 东北大学信息科学与工程学院 沈阳 110819 2. 北京航空航天大学自动化科学与电气工程学院 北京 100191  
1. College of Information Science and Engineering, Northeastern University, Shenyang 110819 2. School of Automation Science and Electrical Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100191

网络连接起来,将其虚拟化成为可配置资源的共享资源池,其中包括计算、软件、数据访问和存储服务,然后把计算任务加载到资源池中完成计算<sup>[6-7]</sup>.终端用户不需要了解“云”中基础设施的细节,不必具有相应的专业知识,无需知道服务提供者的物理位置和具体配置就能进行使用.在这个系统中,电脑和其他设备通过网络提供了一种服务,使用户能够通过网络访问和使用,进行资源共享、软件使用和信息读取等,就如同这些设备安装在本地.云计算系统的处理能力不断提高,可以减少用户终端的处理负担,最终使用户终端简化成一个单纯的输入输出设备,并能够按需购买.云计算的虚拟化、多粒度(节点)、动态调度、软计算的特性一方面使其更具高效性、开放性和可扩展性,另一方面也使得云计算具有更大的不确定性,给基于云计算的控制系统设计带来了新的挑战.

在工业控制方案中引入云计算,形成“云控制系统(Cloud control system, CCS)”.在云端利用深度学习等智能算法,结合网络化预测控制、数据驱动控制等方法实现系统的自主智能控制,同时利用高速通信通道保证控制的实时性<sup>[8]</sup>.云控制系统可以在不增加硬件成本的前提下,工程师可以针对不同被控对象的不同状态,灵活地选择云控制器中相应的控制算法,实现多种被控对象的“定制化”控制.因此,云控制系统将是继集中控制系统、分散控制系统、总线控制系统和网络控制系统后的下一代控制系统新方式.

云控制系统最典型的特征是将控制器算法置于云端,从而形成基于云计算的远程控制系统.如果将云计算端用计算机来代替,就是典型的网络控制系统结构,因此云控制系统是网络控制系统的下一代远程控制系统.目前,针对云控制系统,夏元清等<sup>[2,8-9]</sup>提出了云控制系统的基本概念,但这些框架性质的概念有些抽象、模糊,缺少具体的控制系统结构与实现方法;而后虽然在现代智能交通控制网络的预测计算中有了云计算的应用<sup>[10]</sup>,但是并没有分析云特性加入对于控制系统带来的影响,不涉及控制算法的设计;国外可检索的文献<sup>[11]</sup>中,提出了一种视觉反馈云控制系统,利用云计算技术强大的计算能力进行图像的特征提取与分析,并且分析了云特性加入带来的时延,针对云特性影响提出了切换增益控制策略,取得了较好的控制效果;但该文所涉及的问题是云计算处理图像反馈信息带来的时延问题,控制系统结构本质上还是常规的反馈控制系统,与本文的云控制系统概念和结构有本质的

区别.文献<sup>[12]</sup>研究了基于云的控制系统的网络通信要求,分析了不同使用背景下的云控制系统通信要求并提出了解决方案;但该文并没有涉及针对通讯时延等特性的建模和控制器设计问题.在云计算应用方面,文献<sup>[13-14]</sup>提出了一种基于云的全新虚拟实验室教学平台,利用云存储平台提供学习资料的共享,学生远程控制虚拟机执行实验任务;但侧重于云平台的应用,并没有涉及到控制问题.因此,目前针对云控制系统的研究以概述性介绍和应用为主,建模与控制问题的研究很少,缺乏统一的云控制系统框架,缺乏对云控制系统不确定性问题的分析,也没有针对云控制系统建立一个普遍适用的模型,因此有必要基于本文所定义的云控制系统结构,从控制的角度对云控制系统进行深入的分析 and 研究.

本文从实际应用的角度出发,提出了一种云控制系统的通用结构,通过深入分析云控制系统内部特性,将云控制系统不确定性分解为云端不确定性和网络端不确定性,实现了云控制系统不确定性的有效分解;基于上述分解原则,针对典型的时延不确定性问题,分别对云端和网络端进行时延分析,然后将两者叠加获得了通用的云控制系统时延模型;基于建立的云控制系统时延特性模型,采用极点配置方法设计了云控制系统的控制器,从而保证了闭环系统的稳定性.典型环节的仿真证明了本文所提出方法的有效性.

## 1 云控制理念和典型系统结构

纵观目前控制系统的设计以及所使用的各种控制系统,即从集中控制系统,分散控制系统,总线控制系统到网络控制系统,主要围绕“控制器”这一核心进行系统设计和软硬件配置,工程师主要的焦点是运行各种控制算法的计算机,而被控对象(或过程)则一般置于远端,控制系统结构如图1所示.在这种控制系统结构中,为了提高控制器的性能,需要在计算机控制系统软硬件配置上不断提高;另外,信号的远距离传输也需要铺设大量的电缆等,因此控制系统的建设和运行维护成本较高.

而基于云计算的控制系统结构,则如图2所示,其核心理念是使控制系统设计围绕被控对象(或过程)进行,而将控制器置于远方,工程师的主要焦点是被控对象,而不是控制器(计算机),这有利于工程师对被控对象实时状态的把控.高速通讯信道和云计算的出现为这种控制系统结构提供了可能性,工程师可以将各种控制器(控制算法,优化算法等)

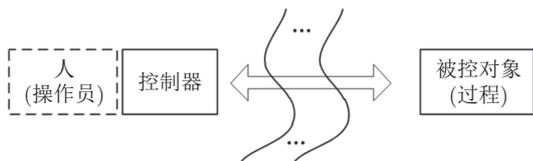


图 1 常规控制系统结构

Fig.1 The structure of conventional control system

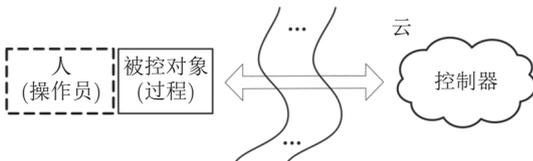


图 2 云控制系统结构

Fig.2 The structure of CCS

置于远方的“云”中,构成云控制器<sup>[15]</sup>,而被控对象端只需要通过高速通讯信道发送现场检测信号和接收远方控制信号即可。

云控制系统的一种具体实现结构如图 3 所示。在云端服务器建立数据库、控制算法库、优化算法库等,彼此互相连接,通过服务器软件进行云端管理。云端服务器通过互联网与各个被控对象端进行信息交换。被控对象端顶层为互联网通信接口,中层为检测器与执行器,底层为被控对象。检测器实时检测底层被控对象的各个状态变量,并通过通信端口提交至云端服务器;执行器依据通信端口接收从云端传来的控制信号,进而对被控对象的各操作变量进行实时控制。监控端可以是移动设备终端(如手机等)或者计算机,通过互联网与云端进行信息交换,供工程师实时监控被控对象的运行状态。

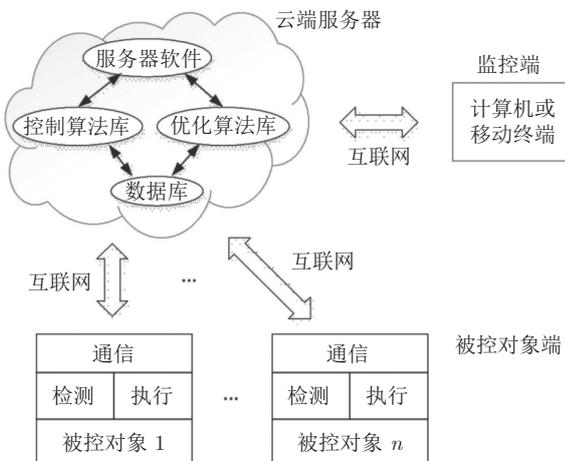


图 3 云控制系统实现结构

Fig.3 The realization structure of CCS

该结构可以有效克服现有控制系统控制算法更新替换不灵活、对于系统硬件要求高的问题,使控制系统设计更加灵活方便。在不增加硬件成本的前提下,工程师可以针对不同被控对象的不同状态,灵活地选择云控制器中相应的控制算法,实现多种被控对象的“定制化”控制。云控制器的提出和使用,有可能像本世纪初网络控制系统的出现一样,给控制理论的研究带来新的方向和课题,促进控制理论自身的不断完善和发展。

## 2 云控制系统不确定性分解及时延特性分析

### 2.1 不确定性分解策略

对图 3 所示的典型云控制系统结构进行简化,得到云控制系统理论结构模型如图 4 所示,该结构由云端、网络端和对象端三部分组成。实际上,如果将云端用一台计算机来代替,图 4 就是一种典型的网络控制系统理论模型<sup>[16]</sup>。

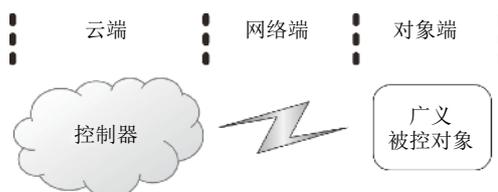


图 4 云控制系统理论结构模型

Fig.4 The theoretical structural model of CCS

从图 4 可以看出,云控制系统可以认为是云计算与网络控制系统的融合,包括了网络控制系统的全部特性。网络控制系统本身由于网络的存在而具有不确定性,云控制系统由于云计算的加入而进一步增加了系统的不确定性,其主要原因是云计算平台负载动态变化,计算资源也动态变化,多种资源动态变化导致云计算系统不确定性的存在。众多不确定性的混合给云控制系统的建模和控制研究增加了难度。

因此为了便于分析,本文对云控制系统不确定性进行分解,如图 5 所示,认为云控制系统存在两个不确定性:云端不确定性和网络端不确定性。二者结合形成整个云控制系统的不确定性。这种分解策略的优点是:可以充分利用网络控制系统的理论成果(如网络控制系统的时延模型),集中精力进行云控制系统的不确定分析;在此基础上,将两种不确定性模型按照某种原则有效结合,从而建立云控



图 5 云控制系统不确定性分解图

Fig.5 The uncertainty division diarggam of CCS

制系统的不确定性模型, 简化云控制系统不确定性建模的难度.

云控制系统中由于云计算这一动态资源的加入, 相较网络控制系统, 不确定性更多, 如时延不确定性<sup>[15]</sup>、丢包现象<sup>[1]</sup>、时序错乱现象<sup>[17]</sup>等, 其中时延的不确定性由于云计算中资源的动态调度更加突出, 是一种最主要的不确定特性. 因此本文重点对时延不确定性进行分析和建模.

对一个控制系统而言, 根据控制器的位置可以将时延分为两部分: 前向通道时延 (也称前向时延) 和反馈通道时延 (也称反馈时延). 前向通道指的是由控制器的输出到执行器的输入; 反馈通道指的是由传感器的输出到控制器的输入. 前向时延的存在使得被控对象不能实时地接收控制器发出的控制信息, 而反馈时延的存在使得系统状态不能得到及时的反馈, 两个时延对于系统的影响是不一样的.

与网络控制系统相比, 云控制系统时延分析更加复杂, 主要体现在如下几个方面:

1) 网络控制系统的组成中, 当传感器到控制器, 或者控制器到执行器的连接不是网络连接时, 则不必考虑网络特性 (网络时延), 这样就会大大简化控制器设计<sup>[18]</sup>. 但是在云控制系统设计中, 这种假定不存在, 因为传感器或者执行器与云端的连接必须通过网络, 因此网络特性更加复杂.

2) 网络控制系统中为了研究问题方便, 通常将前向通道时延和反馈通道时延合并, 成为一个总的时延, 置于前向通道当中<sup>[19]</sup>. 但是这种合并是有前提的, 即控制器的结构和参数不随时间改变. 这种假设在云控制系统中不一定成立, 因为云计算本身的不确定性使控制器模型也具有不确定性, 因此云控制系统前向通道时延和反馈通道时延要分开考虑; 进一步, 甚至网络端时延和云端时延也应该分开考虑, 因为网络端的时延是数据传输的时延, 而云端的时延是计算时延.

3) 一般来说, 云端的可靠性要远远大于网络端的可靠性, 因此云端的时延不确定性一般会服从某一概率分布, 较少会发生极端情况; 以丢包为例, 云端丢包的概率会很小, 而网络端的丢包概率会很大, 一旦发生丢包现象, 则网络端时延就会取极值 (最

大情况).

因此综合考虑云端和网络端不确定性的时延模型更加复杂. 本文基于云控制系统结构分解分别分析了云控制系统云端和网络端的前向时延和反馈时延产生的原因, 为了简单起见, 将两者叠加构成整个云控制系统的前向时延和反馈时延, 并分别考虑两个时延对于云控制系统中控制器设计的不同影响, 以更加适合云控制系统的结构特性.

### 2.2 云端时延分析

云端环境复杂, 云控制系统数据处理、传输、调度、存储等都是云端完成的, 云端任务执行的复杂性给云端时延分析带来很大难度. 一般来说, 云端任务执行原理采用的是 MapReduce 架构<sup>[20-21]</sup>, 近几年 Jstorm 架构逐渐兴起, 但是 Jstorm 也是基于 MapReduce 原理给出的, 而且目前没有得到广泛应用, 因此本文主要分析云端 MapReduce 任务执行架构.

为了便于时延分析, 本文基于 MapReduce 任务执行架构给出云端任务执行模型, 基于该执行模型进行云端时延分析.

MapReduce 任务执行架构如图 6 所示, 图中给出的是具有 5 个输入端口和 3 个输出端口的 MapReduce 架构, 实际应用中可以根据任务类型灵活选择 MapReduce 架构输入端口和输出端口数目. MapReduce 架构下对于任务处理主要包括图 6 所示的 5 个阶段. 任务输入 (Input) 到云端主节点后, 主节点负责接收任务并对任务进行分割 (Split), 然后把分割后的任务映射 (Map) 给从节点, 同时监控从节点的任务执行情况. 从节点执行分配的计算任务并将计算结果输出. 从节点的计算结果输出后会被按照一定的规则进行变换 (Shuffle), 然后送到相应的 Reduce 分区进行规约 (Reduce), 最终将规约结果输出 (Output), 任务执行完成.

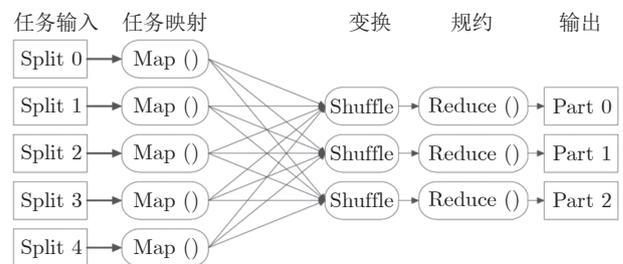


图 6 MapReduce 任务执行架构

Fig.6 The task execution architecture under MapReduce

因此, MapReduce 任务执行架构的主要思想就是对复杂任务进行分割后分配给多个从节点进行计算, 提高任务的执行效率. 它对于任务的执行主要包含两个阶段: 任务映射阶段和规约阶段<sup>[22]</sup>. 云端的时延也是在这两个阶段产生的. 为了便于时延的分析, 本文给出了云端基于 MapReduce 任务执行模型如图 7 所示.

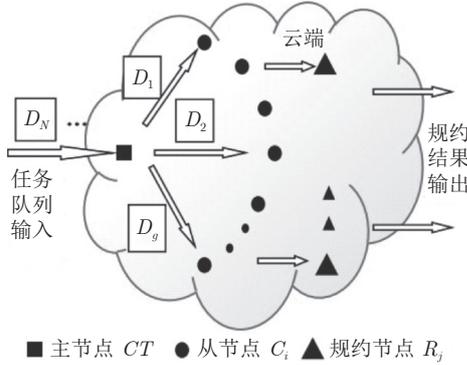


图 7 云端基于 MapReduce 任务执行模型

Fig.7 The task execution model under MapReduce in the cloud-end

在该模型中, 云端主要包含 3 类节点: 主节点  $CT$ , 从节点  $C_i$  ( $i = 1, 2, \dots, M$ ), 规约节点  $R_j$  ( $j = 1, 2, \dots, K$ ). 具体任务执行过程如下: 复杂任务以任务队列的形式进入云端主节点, 主节点接收任务并将复杂任务队列分割成  $N$  个任务块  $D_1, D_2, \dots, D_g$ , 任务块被均匀地映射到  $g$  个从节点执行, 其他任务块在主节点等待, 等  $D_1, D_2, \dots, D_g$  任务块执行完毕后,  $D_{g+1}, \dots, D_N$  任务块也会被依次均匀映射到  $g$  个从节点执行, 所有任务块执行完毕后, 任务映射阶段结束. 从节点执行结果均匀分配到规约节点  $R_j$  进行规约输出, 规约完毕后, 规约阶段结束. 至此云端任务执行过程结束.

云端基于 MapReduce 任务执行模型是在认为各种资源是静态的情况下进行分析的, 而实际上, 每个采样时刻输入云端的任务队列是动态的, 计算资源也是动态的. 因此, 将动态因素加入云端基于 MapReduce 任务执行模型, 建立基于 MapReduce 动态任务执行模型, 此时主节点的作用除了任务映射外还包括任务管理<sup>[23]</sup>. 任务队列传输到主节点后, 主节点会根据任务的大小将任务队列分割为  $n$  个任务块, 并在云端发送控制任务要求, 包括完成任务所需计算资源以及相应的控制算法, 云端所有从节点都可以接收这个要求.

$CT$  发送控制要求, 当有空闲从节点接收到要

求后, 对要求进行核实能够满足控制要求时,  $C_i$  会向主节点  $CT$  发送状态反馈, 反馈包含以下内容<sup>[21]</sup>: 1) 任务传输到从节点  $C_i$  的传输时延  $\tau_i$  以及可能丢包数  $P_i$ ; 2) 从节点  $C_i$  可用计算能力  $CCA_i$ .

当  $C_i$  的反馈信息到达  $CT$  后,  $CT$  会利用预先设定好的优先级评估函数  $S_i$  对  $C_i$  的优先级进行评估, 具体优先级评估函数见式 (1)<sup>[23]</sup>

$$S_i = \alpha f(\tau_i) + \beta g(P_i) + \gamma h(CCA_i) \quad (1)$$

式中,  $\alpha, \beta, \gamma$  参数值分别代表传输时延、可能丢包数以及可用计算能力在优先级评估中所占的权值, 参数值越大说明该参数代表的内容对优先级影响越大, 实际应用中可以根据优先级需要调整参数值.  $f(\tau_i), g(P_i), h(CCA_i)$  为离差标准化归一化函数, 以  $g(P_i)$  为例进行归一化函数介绍. 假设  $M$  个节点的可能丢包数分别为  $P_1, P_2, \dots, P_M$ , 其中最大丢包数为  $P_{\max}$ , 最小丢包数为  $P_{\min}$ , 则归一化后的第  $i$  个节点的可能丢包数为  $P'_i = (P_i - P_{\min}) / (P_{\max} - P_{\min})$ , 归一化后的  $P'_i$  在  $[0, 1]$  范围内取值. 对传输时延以及可用计算能力采用同种方法进行归一化处理. 总之, 传输时延  $\tau_i$  越短, 可能丢包数  $P_i$  越小, 可用计算能力  $CCA_i$  越高, 则  $C_i$  优先级越高.

主节点  $CT$  将收到的所有愿意从节点的优先级进行排序<sup>[23]</sup>, 具体排序内容见表 1. 排序后主节点会选择排在前  $g$  位的从节点分配任务并执行.

表 1 愿意节点列表  
Table 1 The list of willing modes

节点	IP 地址	优先级	排序
$C_1$	$Add_1$	$S_1$	1
$C_2$	$Add_2$	$S_1$	2
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$C_g$	$Add_g$	$S_g$	$g$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$C_M$	$Add_M$	$S_M$	$M$

每个采样时刻, 主节点  $CT$  都会根据当前任务队列需求, 决定该时刻任务队列分割数量  $n$  以及所需从节点数  $g$ , 并向所有从节点发送要求. 所有活动的从节点都会向  $CT$  发送反馈,  $CT$  依据从节点反馈信息根据优先级进行从节点的更替. 因此本文所给的基于 MapReduce 动态任务执行模型是一个动态管理过程, 主节点  $CT$  不断寻找愿意从节点, 删除并替换失效从节点, 该模型符合云端各种资源动态变化的特点.

云端资源动态变化给云端时延带来很大的不确定性. 为了便于分析, 在对基于 MapReduce 动态任务执行模型下信息传递时延进行分析前, 先做如下假设: 1) 每次任务执行中, 任务队列都是均匀分割的; 2) 每个从节点的计算能力是一致的; 3) 任务队列均匀地分配给从节点和规约节点.

接下来进行时延分析. 从图 7 可以看出, 假设某个采样时刻任务队列进入主节点后被分割为  $n$  个任务块, 并依次被分配到  $g$  个从节点. 如果一个任务队列在一个从节点计算所用的时间为  $T_C$ , 则在  $g$  个从节点计算所用时间为

$$T_{\text{compute}} = \frac{T_C}{g} \quad (2)$$

从节点将计算结果输送给规约节点进行任务计算结果的规约输出, 假设一个任务块的规约时间为  $T_I$ , 一共有  $r$  个规约节点, 则  $n$  个任务块的规约时间为

$$T_{\text{reduce}} = \frac{n \times T_I}{r} \quad (3)$$

假设任务队列的映射时间为  $T_{\text{map}}$ , 任务在云端的传输时间为  $T_{\text{cloud-transfer}}$ , 因此云控制系统云端部分前向时延  $\tau_{\text{cloud}}^{ca}$  和反馈时延  $\tau_{\text{cloud}}^{sc}$  分别为

$$\tau_{\text{cloud}}^{ca}(n, r) = T_{\text{reduce}} \quad (4)$$

$$\tau_{\text{cloud}}^{sc}(g) = T_{\text{map}} + T_{\text{compute}} + T_{\text{cloud-transfer}} \quad (5)$$

可以看出云端时延为任务块数  $n$ 、从节点数  $g$  以及任务规约节点数  $r$  的函数, 不同的  $n, g$  和  $r$  的取值会获得不同的云端时延. 本文所给出的云端基于 MapReduce 动态任务执行模型, 充分考虑了云端资源的动态变化, 基于此分析的时延也是动态不确定的, 符合云端的特征.

### 2.3 网络端时延分析

在一般网络传输过程中, 信息从源节点发送至目的节点主要包括 3 个过程: 源节点、网络传输、目的节点<sup>[24]</sup>. 具体时延组成如图 8 所示.

定义源节点内的处理器时延为  $T_{sp}$ , 源节点内通信等待时延为  $T_{sw}$ , 网络传输时延为  $T_{\text{net-transfer}}$ , 目的节点内通信等待时延为  $T_{dw}$ , 目的节点内处理器时延为  $T_{dp}$ .

在云控制系统中, 采集信息和控制信息的传输是通过网络实现的. 但是与一般网络不同, 云控制系统中反馈通道的目的节点和前向通道的源节点在云端, 相应的处理器时延以及通信等待时延已经在云端分析. 因此, 这里基于现有的一般网络传输时

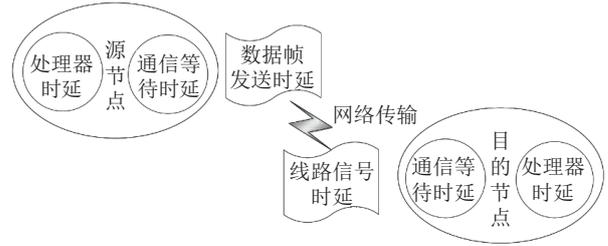


图 8 一般网络传输时延组成图

Fig.8 The delay composition graph of general network transmission

延组成, 给出云控制系统网络端时延组成如图 9 所示.

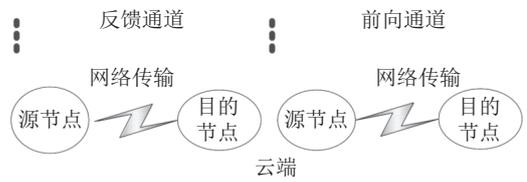


图 9 云控制系统网络端传输时延组成图

Fig.9 The delay composition graph of the network-end in the CCS

从图 9 可得出云控制系统网络部分前向时延  $\tau_{\text{net}}^{ca}$  和反馈时延  $\tau_{\text{net}}^{sc}$  分别为

$$\tau_{\text{net}}^{ca} = T_{\text{net-transfer}} + T_{dp} + T_{dw} \quad (6)$$

$$\tau_{\text{net}}^{sc} = T_{sp} + T_{sw} + T_{\text{net-transfer}} \quad (7)$$

云控制系统特性是云端特性和网络端特性的综合, 因此时延特性也是两者的结合. 这里采用叠加策略, 认为云端时延和网络端时延是相互独立的, 因此云控制系统前向时延  $\tau^{ca}$  和反馈时延  $\tau^{sc}$  分别为

$$\tau^{ca} = \tau_{\text{net}}^{ca} + \tau_{\text{cloud}}^{ca} \quad (8)$$

$$\tau^{sc} = \tau_{\text{net}}^{sc} + \tau_{\text{cloud}}^{sc} \quad (9)$$

基于以上分析, 得到具有前向通道时延和反馈通道时延, 综合考虑网络特性和云特性的云控制系统时延模型结构, 如图 10 所示.

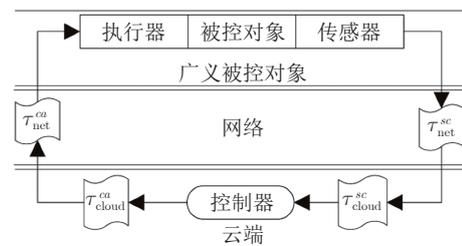


图 10 云控制系统时延模型结构

Fig.10 The structure of CCS time-delay model

### 3 极点配置法控制器设计

基于状态空间模型的极点配置设计方法, 其控制器设计包含两部分: 观测器和控制律. 观测器主要根据传感器输出进行状态重构, 然后基于重构的状态进行控制律设计. 分离性原理说明观测器和控制律可以独立设计, 综合应用, 可以保证系统的稳定性和控制系统的指标基本不变. 本文据此进行观测器和控制律设计, 并充分考虑云特性的加入对于观测器和控制律设计的影响.

假设包含传感器和执行器的广义被控对象的状态空间模型为

$$\begin{cases} \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \\ y(t) = C\mathbf{x}(t) \end{cases} \quad (10)$$

其中,  $\mathbf{x}(t) \in \mathbf{R}^n$ ,  $\mathbf{u}(t) \in \mathbf{R}$ ,  $y(t) \in \mathbf{R}$  分别为系统状态、系统输入和系统输出,  $A, B, C$  为具有相应维数的常矩阵.

基于图 10 所示系统设计控制器如图 11 所示. 这里将云控制系统的时延不确定性纳入到控制器的设计中, 通过控制器的设计来克服这些不确定性的影响, 而对于广义被控对象来说, 模型保持式 (10) 不变, 这是从被控对象的角度来审视控制器的. 如果从控制器的角度来审视被控对象, 则可以将云控制系统的时延不确定性纳入被控对象模型中, 对式 (10) 的模型进行修正, 纳入时延不确定性的影响, 然后针对修正后的不确定模型设计控制器.

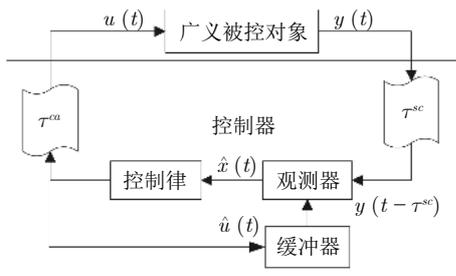


图 11 带有控制器的云控制系统结构  
Fig.11 The structure of CCS with controller

图 11 中, 传感器测量输出由于反馈通道时延的影响, 在  $t$  时刻输入到观测器的信号为  $y(t - \tau^{sc})$ . 为了使得观测器输入  $\hat{u}(t)$  与被控对象输入  $u(t)$  保持同步, 引入一个长度为  $\tau^{ca}$  的缓冲器 (实际应用当中缓冲器长度可取为  $\tau^{ca}$  的最大值  $\tau_{\max}^{ca}$ , 即前向通道时延的最大值). 利用  $y(t - \tau^{sc})$  和  $\hat{u}(t)$  构造观测器为

$$\dot{\hat{\mathbf{x}}}(t) = A\hat{\mathbf{x}}(t) + B\hat{u}(t) + K(y(t - \tau^{sc}) - C\hat{\mathbf{x}}(t - \tau^{sc})) \quad (11)$$

式中,  $\hat{\mathbf{x}}(t)$  为观测器重构状态向量,  $\hat{u}(t)$  为观测器输入,  $K = [k_1, k_2, \dots, k_n]^T$  为观测器增益矩阵.

观测器进行状态重构后会将其输入到控制律, 计算出的控制量通过网络到达执行器. 由于前向通道时延的影响,  $t$  时刻到达被控对象的控制量是基于  $t - \tau^{ca}$  时刻的重构状态获得的, 因此控制律设计为

$$u(t) = L\hat{\mathbf{x}}(t - \tau^{ca}) \quad (12)$$

式中,  $L = [l_1, l_2, \dots, l_n]$  为控制律增益矩阵. 不同于一般网络控制系统将前向时延和反馈时延叠加组成一个总的时延, 本文进行观测器和控制律设计时, 根据云控制系统实际情况分别考虑了前向时延和反馈时延的影响, 其中, 式 (11) 的状态观测器中考虑了反向时延, 式 (12) 的控制律中考虑了前向时延, 因此更加符合云控制系统的实际特性.

定义误差向量  $\mathbf{e}(t) = \mathbf{x}(t) - \hat{\mathbf{x}}(t)$ , 将式 (12) 代入状态空间表达式, 则有

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + BL\mathbf{x}(t - \tau^{ca}) - BL\mathbf{e}(t - \tau^{ca}) \quad (13)$$

$$\dot{\mathbf{e}}(t) = A\mathbf{x}(t) + BL\hat{\mathbf{x}}(t - \tau^{ca}) - A\hat{\mathbf{x}}(t) - B\hat{u}(t) - K(y(t - \tau^{sc}) - C\hat{\mathbf{x}}(t - \tau^{sc})) \quad (14)$$

又  $y(t - \tau^{sc}) = C\mathbf{x}(t - \tau^{sc})$ ,  $\hat{u}(t) = L\hat{\mathbf{x}}(t - \tau^{ca})$ , 代入式 (14) 可得

$$\dot{\mathbf{e}}(t) = A\mathbf{e}(t) - KCE\mathbf{e}(t - \tau^{sc}) \quad (15)$$

由式 (13) 和式 (15) 可得带有观测器和控制律的闭环系统方程式为

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{e}}(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{e}(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \begin{bmatrix} L \\ -L \end{bmatrix}^T \begin{bmatrix} \mathbf{x}(t - \tau^{ca}) \\ \mathbf{e}(t - \tau^{ca}) \end{bmatrix} + \begin{bmatrix} 0 \\ K \end{bmatrix} \begin{bmatrix} 0 \\ -C \end{bmatrix}^T \begin{bmatrix} \mathbf{x}(t - \tau^{sc}) \\ \mathbf{e}(t - \tau^{sc}) \end{bmatrix} \quad (16)$$

令  $\bar{A} = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix}$ ,  $A_{d1} = \begin{bmatrix} B \\ 0 \end{bmatrix} \begin{bmatrix} L \\ -L \end{bmatrix}^T$ ,  $A_{d2} = \begin{bmatrix} 0 \\ K \end{bmatrix} \begin{bmatrix} 0 \\ -C \end{bmatrix}^T$ ,  $\tau_1 = \tau^{ca}$ ,  $\tau_2 = \tau^{sc}$ . 并且定义增广矩阵  $\mathbf{z}(t) = [\mathbf{x}^T(t) \mathbf{e}^T(t)]^T$ , 则带有观测器和控制律的闭环系统表达式为

$$\dot{\mathbf{z}}(t) = \bar{A}\mathbf{z}(t) + A_{d1}\mathbf{z}(t - \tau_1) + A_{d2}\mathbf{z}(t - \tau_2) \quad (17)$$

其中,  $\bar{A}, A_{d1}, A_{d2} \in \mathbf{R}^{2n \times 2n}$ .

系统 (17) 为一个具有多时变时滞的非线性系

统. 假设时延  $\tau_1$  和  $\tau_2$  的上界已知, 即已知  $\tau_1^{\max}$  和  $\tau_2^{\max}$ , 对其进行拉普拉斯变换可得

$$sIZ(s) = \bar{A}Z(s) + A_{d1}e^{-\tau_1^{\max}s}Z(s) + A_{d2}e^{-\tau_2^{\max}s}Z(s) \tag{18}$$

式中,  $I$  为  $2n \times 2n$  的单位矩阵, 对式 (18) 中的  $e^{-\tau_1^{\max}s}$  和  $e^{-\tau_2^{\max}s}$  进行泰勒级数展开, 考虑到  $\tau_1^{\max}$ ,  $\tau_2^{\max}$  较小, 只取级数的前两项并代入式 (18), 可得系统 (17) 的闭环特征方程为

$$P(s) = \det(sI - \bar{A} - A_{d1}(1 - \tau_1^{\max}s) - A_{d2}(1 - \tau_2^{\max}s)) \tag{19}$$

将式 (19) 展开, 可得

$$P(s) = s^{2n} + f_1(K, L)s^{2n-1} + \dots + f_{2n-1}(K, L)s + f_{2n}(K, L) \tag{20}$$

其中,  $f_i(\cdot)$  ( $i = 1, 2, \dots, 2n$ ) 为  $K$  和  $L$  的线性函数, 其表达式为

$$f_i(K, L) = f_{i,1}k_1 + f_{i,2}k_2 + \dots + f_{i,n}k_n + f_{i,n+1}l_1 + f_{i,n+2}l_2 + \dots + f_{i,2n}l_n \tag{21}$$

控制系统的控制性能受系统极点分布影响, 本文利用极点配置进行系统控制器的设计. 从系统控制要求出发获得系统的  $2n$  个期望极点  $\lambda_1^*, \lambda_2^*, \dots, \lambda_{2n}^*$ , 根据求得的  $2n$  个期望极点可以获得期望的闭环特征方程, 即

$$(s - \lambda_1^*)(s - \lambda_2^*) \dots (s - \lambda_{2n}^*) = s^{2n} + a_1^*s^{2n-1} + \dots + a_{2n}^* \tag{22}$$

根据式 (20) 和式 (22) 的同次幂系数相等, 可得

$$\begin{cases} f_1(K, L) = a_1^* \\ \vdots \\ f_{2n-1}(K, L) = a_{2n-1}^* \\ f_{2n}(K, L) = a_{2n}^* \end{cases} \tag{23}$$

求解上述  $2n$  个方程组, 可以最终获得系统状态观测器增益  $K$  和控制律增益  $L$ .

式 (23) 存在唯一解是需要满足一定条件的. 进一步将式 (23) 整理, 可得

$$F_{2n \times 2n} X_{2n \times 1} = Q_{2n \times 1} \tag{24}$$

其中,

$$F_{2n \times 2n} = \begin{bmatrix} f_{1,1} & f_{1,2} & \dots & f_{1,2n} \\ f_{2,1} & f_{2,2} & \dots & f_{2,2n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{2n,1} & f_{2n,2} & \dots & f_{2n,2n} \end{bmatrix}$$

$$X_{2n \times 1} = [k_1, \dots, k_n, l_1, \dots, l_n]^T$$

$$Q_{2n \times 2n} = [a_1^*, \dots, a_{2n}^*]^T$$

由线性代数相关知识可知, 式 (24) 有唯一解的充分必要条件是  $2n$  个线性方程组线性独立, 即  $F_{2n \times 2n}$  行满秩:  $\text{rank}(F_{2n \times 2n}) = 2n$ .

### 4 典型环节仿真分析

为了检验本文设计的云控制器性能, 下面通过典型环节的仿真实验进行验证. 假设广义被控对象的状态空间表达式为

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \tag{25}$$

$$\text{其中, } A = \begin{bmatrix} -5 & 0 & 0 \\ 10 & -10 & 0 \\ 0 & 1 & 0 \end{bmatrix}, B = \begin{bmatrix} 5 \\ 0 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

实际云控制系统中时延是动态变化的, 本文仿真过程中假设随机时延服从高斯分布, 即  $\tau \sim N(\mu, \sigma^2)$ . 高斯曲线下  $P\{|\tau - \mu| < 3\sigma\} = 0.9974$ , 因此可以认为随机时延  $\tau$  主要在  $(\mu - 3\sigma, \mu + 3\sigma)$  范围内取值,  $\mu + 3\sigma$  为时延取值区间的最大值, 并且  $\mu - 3\sigma = 0$ , 即  $(\mu - 3\sigma, \mu + 3\sigma)$  等价于  $(0, \tau_{\max})$ .

为了检验云特性加入对于控制系统的影响, 首先对常规状态反馈控制系统进行仿真, 其次将云特性加入常规控制系统, 先分别设置云特性前向和反馈最大时延为 10 ms, 使两个时延均在 (0, 10) ms 内呈高斯分布变化; 再分别将云特性前向和反馈最大时延增大到 20 ms, 使两个时延均在 (0, 20) ms 内呈高斯分布变化. 应用常规极点配置进行常规控制器设计, 根据时域指标: 调节时间和超调量, 选定一对主导极点, 将其他 4 个辅助极点选定在远离主导极点的位置. 取调节时间 0.6 s, 超调量 5%, 从而可以确定一对主导极点  $s_{1,2} = -5 \pm 7.5j$ , 将其他 4 个辅助极点选定在远离主导极点的位置  $s_3 = -30, s_4 = -32, s_5 = -34, s_6 = -36$ . 由此求得系统期望的闭环特征方程. 设常规控制系统观测器和控制律增益矩阵为  $K = [k_1, k_2, k_3]^T, L = [l_1, l_2, l_3]$ , 另外根据式 (10) 利用常规状态观测器和控制律设计方法可以求得系统包含控制器的实际的闭环特征方程, 根据对应项系数相等可以求得常规观测器矩阵以及控制律矩阵分别为  $K = [128, 1\ 826, 87]^T, L = [5.8, 4.5, 56]$ . 将求得的观测器以及控制律分别代入常规控制系统、短时延云特性系统以及长时延云特

性系统. 对比常规控制器下不同时延云特性的加入对于控制效果的影响, 具体仿真效果如图 12 所示, 图中响应最平滑的曲线是常规控制系统的响应曲线, 振荡最剧烈的曲线是加入长时延云特性的响应曲线, 响应比较稳定但是带有毛刺的曲线是加入短时延云特性的响应曲线.

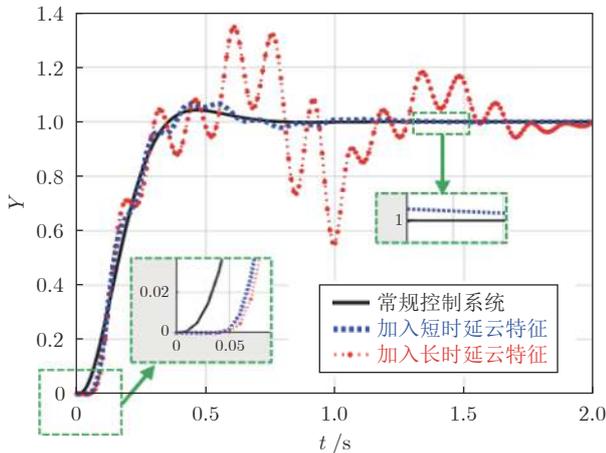


图 12 云特性影响控制效果对比图

Fig. 12 The contrast graph of cloud characteristics influencing control effect

为了使仿真效果更加明显, 对图 12 初始时刻以及稳态时刻的曲线进行了放大. 从图中可以看出, 无论是短时延云特性还是长时延云特性, 它们的加入延缓了系统的响应, 系统的响应速度明显变慢, 响应曲线也不再平滑. 另外, 短时延云特性的加入使得系统的稳态误差增大, 长时延云特性的加入使得系统响应难以达到稳态. 仿真结果说明云特性的加入对于系统控制效果的影响是比较大的, 有必要基于云特性进行控制器的设计.

分别针对短时延和长时延云特性进行云特性控制器的设计. 短时延云特性仿真实验中,  $\tau_1^{\max} = \tau_2^{\max} = 10$  ms; 长时延云特性仿真实验中,  $\tau_1^{\max} = \tau_2^{\max} = 20$  ms. 设短时延云特性系统观测器矩阵以及控制律增益矩阵分别为  $K' = [k'_1, k'_2, k'_3]^T$ ,  $L' = [l'_1, l'_2, l'_3]$ . 设长时延云特性系统观测器矩阵以及控制律增益矩阵分别为  $K'' = [k''_1, k''_2, k''_3]^T$ ,  $L'' = [l''_1, l''_2, l''_3]$ . 按照与常规控制器相同的时域指标对短时延云特性系统和长时延云特性系统进行极点配置. 此闭环系统为 6 阶系统, 极点配置方法如下: 将与常规控制系统相同的期望的闭环特征方程系数代入式 (24) 右侧, 将  $K'$ ,  $L'$  以及相应的系统和时延参数和  $K''$ ,  $L''$  以及相应的系统和时延参数分别代入式 (24) 左侧, 求解等式可以

分别求得  $K' = [3\ 712, 4\ 038, 177]^T$ ,  $L' = [9.8, 5.3, 100]$ ;  $K'' = [4\ 213, 38, 139]^T$ ,  $L'' = [6.3, 85, 10]$ .

分别将求得的参数代入前述控制系统中, 检验本文设计的控制器对于短时延和长时延云特性控制系统的控制效果, 如图 13 和图 14 所示, 两个图中较为平滑的曲线是加入控制器后的响应曲线. 为便于对比, 将图 12 中的不考虑云特性的控制系统响应曲线重画于图 13 和图 14 中, 两个图中较为粗糙的曲线是加入控制器前的曲线.

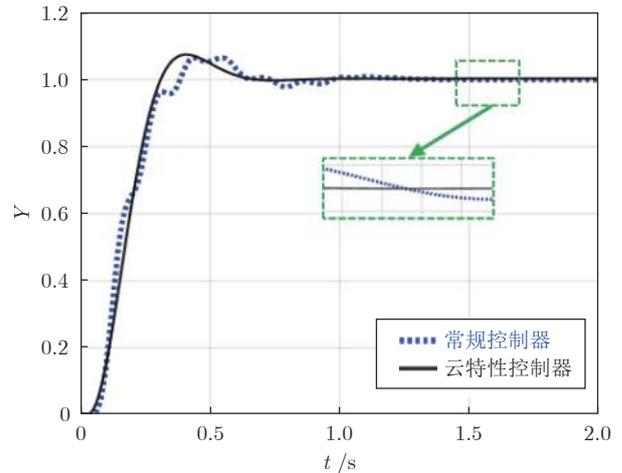


图 13 短时延云特性控制器控制效果对比图

Fig. 13 Contrast graph of control effect of short time-delay cloud characteristic controller

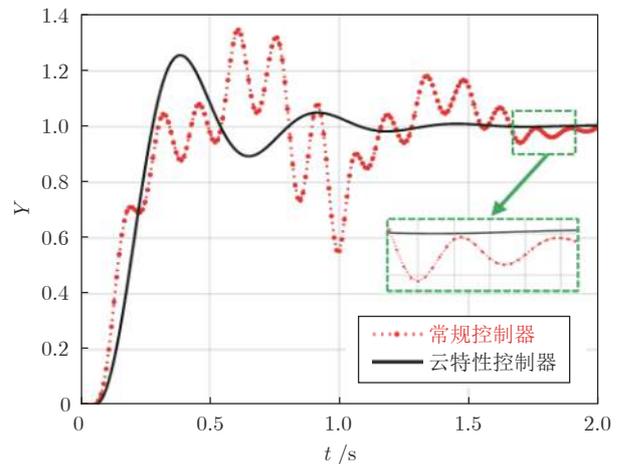


图 14 长时延云特性控制器控制效果对比图

Fig. 14 Contrast graph of control effect of long time-delay cloud characteristic controller

从图 13 中可以看出, 基于短时延云特性设计的控制器能够有效补偿短时延云特性的引入对于控制系统的影响, 与常规控制器相比, 云特性控制器

的响应曲线更加平滑,并且从局部放大图可以看出,对于频繁的云特性随机时延变化,云特性控制器响应曲线趋于稳定,说明它能够及时补偿时延的动态变化.从图 14 可以看出,基于长时延云特性设计的控制器使得原本发散的系统趋于稳定,并且有效地减小了系统的振荡程度.短时延云特性和长时延云特性对比仿真实验说明了本文提出的云特性控制器设计方法能够有效补偿长短时延云特性的加入对于控制系统的影响.

## 5 结束语

本文给出了一种典型云控制系统结构,提出了对云控制系统的不确定性分解策略,并据此进行了云端和网络端的时延分析,从而建立了云控制系统时延模型.在充分考虑前向时延和反馈时延特性对控制系统产生不同影响的情况下,应用极点配置方法进行了云控制系统控制器设计.典型环节的仿真对比实验证明了本文基于云特性所设计的控制器的有效性.

本文所做的研究工作属于云控制系统建模和控制问题的初步探索,仍然存在很多不足.例如,为了便于时延分析,本文对云端结构进行了简化,忽略了云端部分时延;本文基于时延进行控制器设计本质上是基于时延上界的前提下进行的,因此设计偏于保守;可以考虑云端时延和网络端时延在相同或不同随机分布条件下的动态建模和控制问题,以及其他不确定性(如丢包、时序错乱等)条件下的云控制系统建模和控制器设计问题.这些都是未来云控制系统所要解决的问题,也是我们下一步的研究方向.

## References

- 1 You Ke-you, Xie Li-hua. Survey of recent progress in networked control systems. *Acta Automatica Sinica*, 2013, **39**(2): 101-118  
(游科友, 谢立华. 网络控制系统的最新研究综述. 自动化学报, 2013, **39**(2): 101-118)
- 2 Xia Y. From networked control systems to cloud control systems. In: Proceedings of the 2012 Control Conference. Beijing, China: IEEE, 2012. 5878-5883
- 3 Schlechtendahl J, Kretschmer F, Sang Z Q, Lechler A, Xu X. Extended study of network capability for cloud based control systems. *Robotics and Computer Integrated Manufacturing*, 2017, **43**: 89-95
- 4 Chinacloud: The concept and connotation of cloud computing [Online], available: <http://www.chinacloud.cn/show.aspx?id=14668&cid=17>, March 3, 2019
- 5 Li Bo-Hu, Chai Xu-Dong, Zhang Lin, Lin Ting-Yu. Smart cloud manufacturing: Intelligent manufacturing model and means of industrial cloud. *China Industry Review*, 2016, (2): 58-66  
(李伯虎, 柴旭东, 张霖, 林廷宇. 智慧云制造: 工业云的智造模式和手段. 中国工业评论, 2016, (2): 58-66)
- 6 Luo Jun-Zhou, Jin Jia-Hui, Song Ai-Bo, Dong Fang. Cloud computing: architecture and key technologies. *Journal on Communications*, 2011, **37**(2): 3-21  
(罗军舟, 金嘉晖, 宋爱波, 东方. 云计算: 体系架构与关键技术. 通信学报, 2011, **37**(2): 3-21)
- 7 Hayes B. Cloud computing. *Communications of the ACM*, 2008, **51**(7): 9-11
- 8 Xia Yuan-Qing. Cloud control systems and their challenges. *Acta Automatica Sinica*, 2016, **42**(1): 1-12  
(夏元清. 云控制系统及其面临的挑战. 自动化学报, 2016, **42**(1): 1-12)
- 9 Xia Y Q, Qin Y M, Zhai D H, Chai S C. Further results on cloud control systems. *Science China Information Sciences*, 2016, **59**(7): 232-236
- 10 Xia Yuan-Qing, Yan Ce, Wang Xiao-Jing, Song Xiang-Hui. Intelligent transportation cyber-physical cloud control systems. *Acta Automatica Sinica*, 2019, **45**(1): 132-142  
(夏元清, 闫策, 王笑京, 宋向辉. 智能交通信息物理融合云控制系统. 自动化学报, 2019, **45**(1): 132-142)
- 11 Wu H Y, Lou L, Chen C C, Hirche S, Kühnlenz K. Cloud-based networked visual servo control. *IEEE Transactions on Industrial Electronics*, 2013, **60**(2): 554-566
- 12 Schlechtendahl J, Kretschmer F, Sang Z Q, Lechler A, Xu X. Extended study of network capability for cloud based control systems. *Robotics and Computer-Integrated Manufacturing*, 2017, **43**: 89-95
- 13 Ding Q, Cao S T. A cloud-based learning tool for graduate software engineering practice courses with remote tutor support. *IEEE Access*, 2017, **5**: 2262-2271
- 14 Xu L, Huang D J, Tsai W T. Cloud-based virtual laboratory for network security education. *IEEE Transactions on Education*, 2014, **57**(3): 145-150
- 15 Ma L, Xia Y Q, Ali Y, Zhan Y F. Engineering problems in initial phase of cloud control system. In: Proceedings of the 36th Chinese Control Conference. Dalian, China: IEEE, 2017. 7892-7896
- 16 Wang Fei-Yue, Wang Cheng-Hong. On some basic issues in network-based direct control systems. *Acta Automatica Sinica*, 2002, **28**(Z): 171-176  
(王飞跃, 王成红. 基于网络控制的若干基本问题的思考和分析. 自动化学报, 2002, **28**(Z): 171-176)
- 17 Wang Jian-Ping, Zhang Yun-Sheng, Zhang Guo, Zhang Jing. Analysis of signal timing sequence flow chart on parallel and distribute control network. *Control and Decision*, 2010, **25**(11): 1727-1731  
(王剑平, 张云生, 张果, 张晶. 并行分布控制网络的实时信号时序流图分析. 控制与决策, 2010, **25**(11): 1727-1731)
- 18 Tang Xiao-Ming, Deng Li, Yu Ji-Min, Qu Hong-Chun. Output feedback model predictive control for interval type-2 T-S fuzzy networked control systems. *Acta Automatica Sinica*, 2019, **45**(3):

- 604–616  
(唐晓铭, 邓梨, 虞继敏, 屈洪春. 基于区间二型T-S模糊模型的网络控制系统的输出反馈预测控制. *自动化学报*, 2019, **45**(3): 604–616)
- 19 Cloosterman M B G, Wouw N V D, Heemels W P M H, Nijmeijer H. Stability of networked control systems with uncertain time-varying delays. *IEEE Transactions on Automatic Control*, 2009, **54**(7): 1575–1580
- 20 Wang Cai-Lu, Tao Yue-Gang, Yang Peng, Liu Zuo-Jun, Zhou Ying. Parallel task assignment optimization algorithm and parallel control for cloud control systems. *Acta Automatica Sinica*, 2017, **43**(11): 1973–1983  
(王彩璐, 陶跃钢, 杨鹏, 刘作军, 周颖. 云控制系统并行任务分配优化算法与并联控制. *自动化学报*, 2017, **43**(11): 1973–1983)
- 21 Malekimajd M, Ardagna D, Ciavotta M, Gianniti E, Passacantando M, Rizzi A M. An optimization framework for the capacity allocation and admission control of MapReduce jobs in cloud systems. *The Journal of Supercomputing*, 2018, **74**(10): 5314–5348
- 22 Li Jian-Jiang, Cui Jian, Wang Dan, Yan Lin, Huang Yi-Shuang. Survey of MapReduce parallel programming model. *Acta Automatica Sinica*, 2011, **39**(11): 2635–2642  
(李建江, 崔健, 王聘, 严林, 黄义双. MapReduce并行编程模型研究综述. *电子学报*, 2011, **39**(11): 2635–2642)
- 23 Xia Y Q. Cloud control systems. *IEEE/CAA Journal of Automatica Sinica*, 2015, **2**(2): 134–142
- 24 Guan Shou-Ping, Zhou Wei, You Fu-Qiang. *Networked Control Systems and Applications*. Beijing: Publishing House of Electronics Industry, 2008. 15–18  
(关守平, 周玮, 尤富强. 网络控制系统与应用. 北京: 电子工业出版社, 2008. 15–18)



**关守平** 东北大学信息科学与工程学院教授. 1995 年获得东北大学工业自动化系博士学位. 主要研究方向为复杂工业过程建模, 优化与控制, 网络与云控制, 智能控制.

E-mail: guanshouping@ise.neu.edu.cn

(**GUAN Shou-Ping** Professor at the

College of Information Science and Engineering, Northeastern University. He received his Ph.D. degree from the Department of Industrial Automation, Northeastern University in 1995. His research interest covers industrial process modeling, optimization and control, networked and clouded control, and intelligent control.)



**王 梁** 北京航空航天大学自动化科学与电气工程学院博士研究生. 2019 年获得东北大学信息科学与工程学院自动化系学士学位. 主要研究方向为飞行器控制, 网络与云控制. 本文通信作者.

E-mail: wang\_liang@buaa.edu.cn

(**WANG Liang** Ph.D. candidate at the School of Automation Science and Electrical Engineering, Beijing University of Aeronautics and Astronautics. He received his bachelor degree from the College of Information Science and Engineering, Northeastern University in 2019. His research interest covers aircraft control, networked and clouded control. Corresponding author of this paper.)