



浙江工业大学

硕士学位论文

论文题目：通讯和计算资源受限系统的控制设计和分析

作者姓名 韩康

指导教师 赵云波 教授

学科专业 控制工程

培养类别 全日制专业学位硕士

所在学院 信息工程学院

提交日期 2019年1月

浙江工业大学硕士学位论文

通讯和计算资源受限系统的控制设计和分析

作者姓名：韩康

指导教师：赵云波教授

浙江工业大学信息工程学院

2019年1月

**Dissertation Submitted to Zhejiang University of Technology
for the Degree of Master**

**Control Design and Analysis for Systems Subject to
Communication or Computational Constraints**

Candidate: Kang Han

Advisor: Prof. Yun-Bo Zhao

**College of Information Engineering
Zhejiang University of Technology
Jan 2019**

浙江工业大学

学位论文原创性声明

本人郑重声明：所提交的学位论文是本人在导师的指导下，独立进行研究工作所取得的研究成果。除文中已经加以标注引用的内容外，本论文不包含其他个人或集体已经发表或撰写过的研究成果，也不含为获得浙江工业大学或其它教育机构的学位证书而使用过的材料。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人承担本声明的法律责任。

作者签名： 韩康 日期：2019年 1 月

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权浙江工业大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于

- 1、保密 ，在一年解密后适用本授权书。
- 2、保密 ，在二年解密后适用本授权书。
- 3、保密 ，在三年解密后适用本授权书。
- 4、不保密 。

(请在以上相应方框内打“√”)

作者签名： 韩康 日期：2019年 1 月

导师签名： 韩康 日期：2019年 1 月

通讯和计算资源受限系统的控制设计和分析

摘 要

通讯和计算技术的飞速变革使得控制系统的发展呈现出新的趋势，在此过程中涌现出的大量新型系统，如物联网、信息物理融合系统等，都依赖复杂的通信网络和各种嵌入式计算设备闭合控制系统。通信和计算技术的大量使用拓宽了控制系统的适用范围，但与此同时通信与计算的各种限制也引入到控制系统中，使得对该类系统的设计和分析需要新想法新思路。

本文研究了两类特殊通信和计算受限系统的设计和分析问题。针对特定场景下的通讯和计算资源受限系统分别给出了问题的严格定义，提出了相应的解决方案并利用仿真实验验证了方法的有效性。本文主要工作包括：

1. 研究了通信网络多包传输引发的多时延通信限制问题。无线通信技术的广泛使用使得多个传感器采集系统信息的情况变得很普遍，但多传感器的使用造成了控制系统输出由独立的多个数据包传输的情况，即“多包传输”，这引发了系统输出各个部分时延不一致的问题，造成了设计的困难。对该问题给出了严格的数学定义，在基于状态重建的基础上利用预测控制思想给出了一种基于数据包的主动补偿方案，并分析了方案的闭环稳定性，最后分别使用纯数值仿真和基于 TrueTime 工具箱的仿真对方法的有效性进行了验证。
2. 研究了由于控制系统和其他组件共享有限计算资源而可能引起的控制系统计算资源不足的计算资源受限问题，该问题在物联网等大规模使用共享嵌入式计算设备的系统中较易出现。针对该问题提出了一种全资源预测控制方法，在不占用更多系统计算资源的前提下优化了控制系统性能。对闭环系统的稳定性和方法的可扩展性做了进一步的分析，并对方法的有效性基于 Matlab 仿真进行了验证。

关键词：资源受限，控制系统，多包传输，主动补偿，预测控制

CONTROL DESIGN AND ANALYSIS FOR SYSTEM SUBJECT TO COMMUNICATION OR COMPUTATIONAL CONSTRAINTS

ABSTRACT

The rapid revolution of the communication and computational technologies enable a new trend for the development of control systems, where many emerging systems of new form such as the Internet of Things, cyber-physical systems, all depend on complex communication networks and various embedded computing devices to close the control system. Here, the wide applications of the communication and computational technologies have greatly broadened the applicability of control systems, while at the same time the constraints induced by communication and computation are also introduced to control systems which require new approaches to deal with.

This thesis investigates the design and analysis of two special classes of communication/computation constrained systems. The mathematical definition of the problems is given, solutions are proposed, and the effectiveness are verified by numerical examples. The works contained in this thesis include,

1. The communication constraint problem of multi-delay caused by multi-packet transmission is investigated. The wide use of the wireless communication makes it popular for the control systems to use multiple sensors, while the latter causes the problem of “multi-packet transmission”, i.e., the system output is transmitted via multiple independent data packets, which then consequently causes the partially inconsistent delay for the system output, a new challenge for the design. The mathematical definition of the problem is given, a packet-based active compensation scheme is proposed based on the predictive control idea, the closed-loop stability is analyzed, and the effectiveness of the approach is finally verified using numerical and TrueTime based simulations.
2. The computational constraint problem of lacking computational resources due to the sharing of these resources is investigated, which is common in systems such as the Internet of Things where shared embedded computing devices are broadly used. A full

resource predictive control method is proposed following the co-design idea, which optimizes the system performance at no extra cost of the computational resources. The closed-loop stability is analyzed with discussions on the expendability of the approach, and the effectiveness of the approach is verified numerically.

Key Words: resource constrained, control system, multi-packet transmission, active compensation, predictive control

目 录

摘要.....	i
第 1 章 绪论.....	1
1.1 课题的背景和意义.....	1
1.2 国内外研究现状.....	2
1.2.1 通讯资源受限控制系统在国内外的研究现状.....	2
1.2.2 计算资源受限控制系统在国内外的研究现状.....	7
1.3 本文主要工作.....	7
第 2 章 通讯资源受限控制系统的多时延主动补偿方法.....	9
2.1 问题描述.....	9
2.1.1 多包传输产生的原因和负面影响.....	9
2.1.2 系统描述.....	10
2.2 基于模型的系统状态重建.....	11
2.2.1 系统状态重建的原因.....	11
2.2.2 系统状态重建的方法.....	12
2.3 基于数据包的补偿.....	14
2.3.1 基于数据包的补偿方案.....	14
2.3.2 基于数据包的控制器设计.....	16
2.4 闭环稳定性分析.....	17
2.5 仿真验证.....	18
2.5.1 基于 Matlab 的数值仿真.....	19
2.5.2 基于 TrueTime 工具箱的仿真.....	20
2.6 本章小结.....	21
第 3 章 计算资源受限控制系统的全资源预测控制方法.....	23
3.1 问题描述.....	23
3.2 全资源预测控制策略.....	25
3.2.1 可用计算资源与控制预测步长.....	25
3.2.2 全资源预测控制策略及算法.....	26
3.3 系统性能讨论和稳定性分析.....	28
3.3.1 系统性能讨论.....	28
3.3.2 闭环稳定性分析.....	29
3.4 基于改进模型预测控制的前向控制预测序列计算.....	30
3.5 仿真验证.....	32
3.6 本章小结.....	34
第 4 章 总结与展望.....	35
4.1 总结.....	35
4.2 展望.....	35

参考文献.....	37
致谢.....	40
攻读学位期间参加的科研项目和成果.....	41

第 1 章 绪 论

1.1 课题的背景和意义

随着信息技术的飞速发展，学科之间的交叉运用成为一种趋势，通讯技术和计算机技术对控制系统的影响也越来越明显。从通讯技术的发展角度看，随着网络和通讯技术的快速发展，新兴的各种形式的通讯网络使得复杂控制系统内部的数据交换通过通讯信道进行传递变得可能，进而催生了控制和通讯领域的交叉运用，比如网络化控制系统（Networked Control Systems）^[1]，从计算技术的发展角度来看，得益于过去几十年间微电子技术和新兴计算技术的进步，计算机的形态也经历了从巨大到微小的转变，这种巨变使得计算机技术在控制领域中特别在一类实际应用场景中（比如嵌入式系统）发挥着举足轻重的作用。大规模通讯技术的运用使相关领域人员能够以更低廉的成本、更便利的维护性和更灵活的配置能力来构建大型控制系统，计算机的小型化满足了各种可移动、便携的系统对计算能力的需求，在这个理论的基础上，Internet of Things（IOT）领域的一些应用场景（比如智能家居、自动驾驶、无人机和楼宇自动化、可穿戴设备等）^[2-5]已经产生了较大的实际应用价值。由于上述新技术的引入形成的新系统具有传统控制系统无法比拟的优点，因此通讯技术、计算技术和控制技术的交叉领域已经成为国际控制领域的一个热门课题^[6-10]。

从通讯技术和控制技术结合的角度来看，这类控制系统和传统控制系统的最大区别在于其控制系统的各个组成部分之间的闭环连接是通过一定形式的通讯网络来实现的。网络化控制系统的特征是它在结构组成上将网络空间层和物理层连接起来，以便在非本地的区域执行多个任务。因此，和传统的控制系统相比，网络化控制系统有许多传统控制系统没有的优势。得益于其网络空间层和物理层相互连接的这种特点，一方面，网络化控制系统可以免去各个系统组件之间不必要的物理连接，这也就意味着在设计和实现同样的一个控制系统时，网络化控制系统相较于传统控制系统既能降低系统的复杂性还可以显著降低经济成本和空间成本；另一方面，网络化控制系统可以在不对系统进行重大修改的前提下额外添加或者删除一部分系统组件，而传统的控制系统可能就会存在“牵一发而动全身”这种情况。因此，早期的网络化控制系统曾被广泛地应用于工业控制领域中^[11]。早期网络化控制的一个典型应用场景即是其在汽车行业中的运用，通过引入网络化控制技术，极大减少了布线成本^[12]。同时，通讯和控制结合的技术还被广泛应用于智能交通、智慧城市、智能

电网等诸多新兴领域^[13]。此外，随着物联网浪潮的到来和云计算技术的日趋成熟，当海量的物理对象被连接成一个巨大的系统的同时还需要对其进行实时的监管和控制，这其中还有大量的理论问题尚未解决^[9]，其中最为广泛的一个问题就是通讯本身的约束对控制系统的影响，因此通讯资源受限的控制系统不仅在理论上有着巨大的研究潜力，在实际中也有着广泛的应用价值。

从计算技术和控制技术结合的角度看，现代控制系统和计算技术一直是不可分割的整体。近年来，随着处理器计算能力的进步，特别是微型嵌入式计算芯片的发展，使得更多的微型设备也具有了一定的计算能力，这为系统的智能化发展提供了物质基础。然而在一些实际应用场景下，各种新型的嵌入式设备和控制系统结合形成的嵌入式系统（比如自主移动小车和无人机等）对控制性能和计算能力的要求也越来越高，但是由于受限于现有的能量管理和电池技术，对这些控制系统而言，如何充分使用现有资源为控制系统提供更好的控制效果在实际应用场景下中具有极大的价值。

1.2 国内外研究现状

1.2.1 通讯资源受限控制系统在国内外的研究现状

一个简单的网络化控制系统的结构如图 1-1 所示。从图中可以看出，虽然网络的引入为控制系统带来了诸多好处，但是由于控制系统的各个元器件之间的数据交换是通过通讯信道来实现的，因此数据在采集和传输过程中就会不可避免的受到网络结构、通讯协议和网络负载等多方面因素带来的影响。这些负面的影响会影响数据的正常传送，进而可能会导致诸如数据传输时滞、数据包丢失、数据包时续错乱等一系列的问题。本小节将会对上述的问题展开进一步的讨论。

1) 时钟同步问题。在控制系统中，在一些场景下各个系统组件之间需要时间同步才能正常工作，但是由于网络化控制系统采用了分布式的结构，因此时钟同步问题对这类网络化控制系统来说是首先要处理的一个问题。在一些条件下，如果控制信号的计算不需要使用反向信道中的网络诱导时延，或控制动作的执行不需要前向信道中的网络诱导时延，这种情况下网络化控制系统中的时间同步可能就不是太必要。然而，对另一些情况来说，使用时间标签计算得到前向通道和反馈通道的通讯时延，然后根据这些时延来进行控制量的计算，对控制系统来说使用这种方法可能就会有更好的控制效果^[14,15]。此外，在时间同步的基础上还可以扩展出更好的控制架构^[16]。

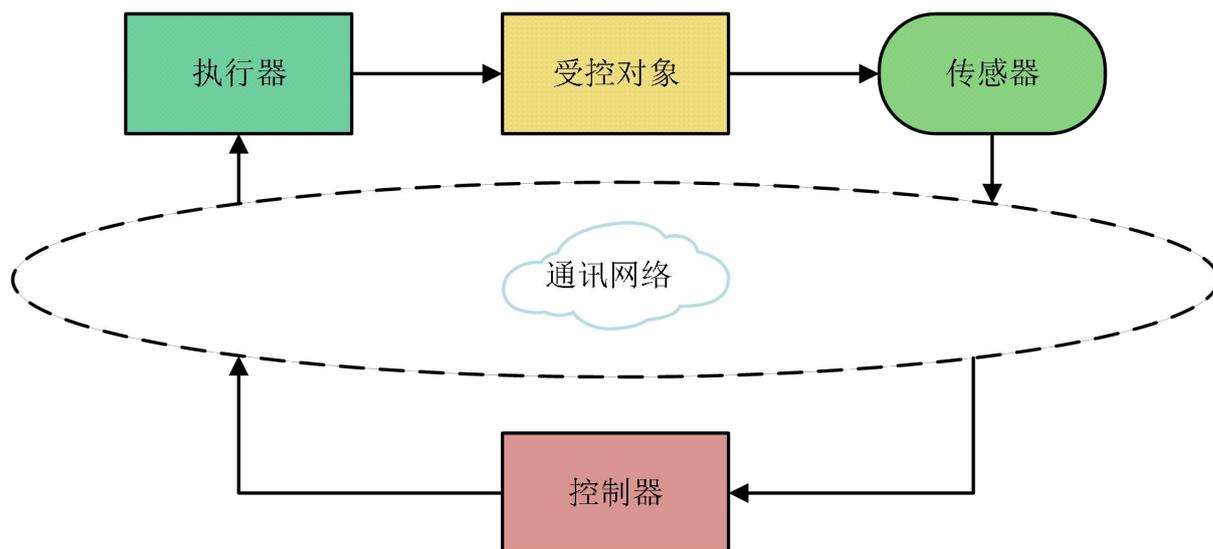


图 1-1 简易网络化控制系统结构框图

2) 节点驱动方式。连续时间信号在通过通信网络传输之前需要先进行采样，采样的方式有两种：时间触发模式和事件触发模式。因此对于控制系统的元件（节点）来说也就存在上述的两种工作模式。时间触发驱动的特征是当一个事件被触发之后，下一个触发发生在一个固定的时间间隔之后，因此这种驱动方式要求比较严格的时间同步策略。在早期的数字控制系统中，由于采用固定时间的的时间触发模式在系统的分析和设计上比较简单，因此被广泛应用于数字控制系统。对于一个离散的控制系统的而言，当采样周期缩小至接近于零时，系统节点的工作间隔时间减少，这时候就可以近似于一个连续时间系统。理论上来说，采样周期越小，系统的性能就越好。事件触发模式描述了节点被触发的原因，其特点是两次事件的动作上没有直接的时间联系，只和特定的事件相关，这对系统的时间同步的要求也就没那么高。对于网络化控制系统来说，采样周期变小意味着需要传输的数据量增大，这在一定程度上会引起网络负载增大进而会导致通讯信道堵塞，这个时候周期较小的事件驱动模式就不能保证更好的控制性能，而事件触发的驱动模式可能就助于降低通讯网络的负载。一般而言，在网络化控制系统的参数设置中，传感器通常是时间驱动的，控制器和执行器可以是时间驱动的，也可以是事件驱动的^[17-19]。

3) 量化误差。在网络化控制系统中，由于数据的传送是通过通讯网络实现的，而通讯网络的带宽是有限的，因此需要对系统的信号量进行量化。由于单个编码段的字长也是有限的，对信号进行量化的过程中就不可避免地会出现量化误差，进而也会对网络化控制系统的系统性能产生负面影响。一般情况下，量化误差可以用区域有界不确定性或非线性的方法来处理^[20]，也就是意味着可以用鲁棒控制分析的方法来研究量化误差对网络化控制

系统的影响。此外，采用合适的量化方法也能在一定程度上降低系统带宽的占用，进而避免产生网络拥塞提高网络化控制系统的系统性能。针对量化对系统系统性能的影响这个问题，现有的不少文献都对这个问题进行了比较详细的研究^[21,22]。

4) 网络诱导时延。在网络化控制系统中，网络导致的延迟通常由几个因素引起，比如带宽、网络占用情况和传输协议等。考虑到上述的几个因素，网络延时可以细分为数据处理上的延时、路由等待引起的延时和传输延时等多种情况。对于一个闭环的控制系统来讲，根据网络时延发生的位置又可以将其分为以下两类。

τ_{sc} ：反馈通道中存在的传感器到控制器的传输时延；

τ_{ca} ：前向通道中存在的控制器到执行器的传输时延。

从时间的连续性上来讲，传感器-控制器延迟表示从采样物理信号到生成相应控制信号的时间间隔，控制器-执行器延迟表示控制信号从控制器传输到执行器的时间间隔。由于网络诱导时延严重依赖网络状况^[23]，因此，网络时延通常是时变的、随机的且不确定的。在大多数情况下，网络时延通常被认为是系统性能下降甚至不稳定的重要原因。

5) 数据包丢失。数据包丢失现象也是网络化控制系统中一个重要的问题。我们都知道网络化控制系统中的数据是先被编码到数据包中，然后通过通信网络来传输的。由于通信网络在进行数据传输过程中不可避免地会出错，就会造成一些数据包在传输过程中发生丢失。当网络状况不好时，如网络流量过载，通信网络出现传输超时和传输错误等，这个时候即使某些通讯协议规定了丢包重传策略，但是也会造成数据包的丢失，这种现象我们称之为由网络引起的丢包或者称为被动丢包；在另一种情况下，当一个数据包发送的时间比较早但是到达的时间比较晚，即在这个数据包到达之前已有时延更小的更新的数据包到达了，这个时候系统可能会主动丢弃过时的数据包，这种现象被称为主动丢包现象，这类情况多和数据包的时序错乱有关。

通常来讲网络的动态性被认为是一种随机过程，因此数据包丢失现象表现为一定的随机性。基于此种理论，一些研究人员把数据包丢失视为一个一定形式的马尔可夫或者伯努利过程^[24-27]。

6) 数据包乱序。在大多数通信网络中，由于每个数据包的时延都是时变的、随机的且不确定的，因此单个数据包的时延也是不同的，这就有可能造成在时间发送比较早的数据包比较晚的到达目的地，或者晚发的数据包比较早的到达了目的地，这种情况通常被称为数据包的乱序。数据包的时序错乱意味着一个新到达的网络化控制系统控制信号可能不

是最新的，一般来说，过时的数据包对控制系统分析和设计并不是太有用。因此在网络化控制系统中，在系统的分析和设计上也要对此种现象进行考虑。

7) 单包传输和多包传输。据上所述我们已经得知网络化控制系统的传感器数据和控制器数据是通过网络以数据包的形式进行传送时，因此对于同样的数据也就会存在两种不同的传输方式，即通过单个数据包传输和分成多个多数据包分别进行传输。通常情况下单数据包传输是比较常见的也是比较容易理解的，而多包传输的情况来说并不是太常见。这里对多包传输的原因进行简要的陈述，一般来说，多包传输的原因有两个，一种情况是单个数据包的有效编码位不能完全编码一个信号量，这时候就需要把一个信号拆分成多个数据包来分别传送，比如在控制器局域网（CAN）中，一个数据帧最多只能容纳八个字节的数据量，当传感器的测量量或者控制的信号大于八个字节的时候就必须将其拆分成多个数据包来分别传输；另一种情况是网络化控制系统的传感器节点不止一个，而且地理上分布的也不集中，这种分布式的网络化控制系统在进行传感器数据或者其他信号量传递时，就不得不使用多个数据包通过多个通讯信道来传输，这种情况也是造成多包传输现象的主要原因。随着控制系统的结构越来越复杂、受控对象的日益增多，单一受控对象的传感数据维度也不断增多，这种多包传输的网络化控制系统引起了越来越多的研究者的兴趣。针对上面两种情况，不管是拆分成多个数据包来传递还是地理隔离造成的多数据包传输都会引起多个时延，进而就造成了多个数据包不能同时到达控制器或执行器的情况，于是系统只能得到系统的部分更新信息，从而给网络化控制系统在分析和设计带来了极大的麻烦。

从前面可知，控制系统中由于引入了通讯网络进而存在通讯资源受限情况的一类系统被大家称之为网络化控制系统。在近些年里，由于网络化控制的迅猛发展，网络化控制这个主题也渐渐成为了国内外控制和自动化领域学者们的研究热点，国内外的一些学者为此贡献了丰硕的科研成果和智慧。早期的对于控制系统的研究主要集中于控制方法和策略上的改进与创新，而对于网络化控制系统来说，由于网络传输过程会产生传输时延、数据包丢失、量化误差和通信受限等多方面的问题，这也无法使得把传统的控制方法和策略直接应用到网络化控制系统中，进而产生了通讯和控制联合设计策略。从近些年发表在 IEEE Trans Automatic Control 和 IEEE Systems Magazine 等国际专业控制领域的文章上来看，网络化控制系统已经成为了控制领域一个相当热门的主题之一。

本世纪初，Tipsuwa Y^[28]和 Richard J P^[29]等人从控制理论的角度分析了网络化控制系统的控制性能，其中对于网络的分析是先通过预设参数再对控制系统进行建模。在这类的研究中，网络化控制系统的一些通信特性例如网络时延等可以作为一些参数被整合进系统的

模型中，从而构建出一个常规的控制系统的进一步分析和设计。这种类型的研究简化了网络化控制系统的建模和分析过程，这种方法的优点是可以保证现有的一些控制方法可以很容易地被应用于网络化控制系统。Yue D^[30] Tipsuwan Y^[28] Mazenc F^[31] Guan Z H^[32] 等人在处理有时延的网络化控制系统时将其处理为一类特殊的时滞系统。对时延进行处理时会不可避免的要涉及到最大时延上界（MADB）的问题，这里的最大时延上界指的是能确保网络化控制系统的稳定性或其他性能目标的最大传输时延的上限，为此 Kim D S^[33] 关于此问题做了大量的调查和研究。

如上所述，考虑到网络化控制系统中的通信约束从本质上可以说是随机的，因此可以使用随机控制的方法对网络化控制系统进行分析。在关于此问题研究中，Hu S S^[34] 等人的工作将 Halevi Y^[35] 等人的工作扩展到随机最优控制框架，并给出了随机最优状态反馈和输出反馈控制器的设计。Nilsson J^[36] 明确了网络时延的特征，并初步得到了具有网络诱导时延特征的随机稳定性条件。Schenato L^[37]、Imer O C^[38]、Seiler P^[39]、Sahebsara M^[40] 等人认为可以在网络控制环境中使用传统的最优控制方法进行网络化控制系统控制器的设计，并在此方面做了大量的工作。上述的这些方法中，由于都假设通讯上的特性是已知的，因此在对系统进行分析和设计时就有一定的局限性，因为上述的大多数工作都只在现有控制方法上对网络化控制系统进行扩展，没有充分利用网络化控制系统特有的通信特性。

正如前面我们提到的，由于通信网络取代了传统控制系统中控制组件之间的物理上的直接连接，这使得网络化控制系统在结构上和传统控制系统存在差异。因此，一种新兴的控制与通信相结合的方法，是近年来控制界的一种新趋势。在这种方案中，由通信约束引起的一些系统参数（时延等）不再被假定为预先确定的值，而是被当作一种系统设计时必须考虑的因素参与到控制器的设计中，因此这种联合设计的方案能够取得比较好的控制效果^[41-44]。由于数据在通讯网络中是以数据包的形式进行的，因此可以利用这种特性来对网络化控制系统进行设计和分析，这种通过数据包的形式来分析和设计网络化控制系统的方案又被称为基于数据包的控制架构。基于这种框架，Liu G P^[41] 和 Zhao Y B^[42-44] 等人在使用通用的预测控制方法的基础上，采用主动补偿的方案有效的降低了通讯约束带给控制系统的影响。在控制和调度算法层面上的研究上，Walsh G C 等人^[45] 提出了一种称为“Try-Once-Discard”（TOD）的动态调度算法，该算法分配网络资源的方式是在上一个报告期间内出现最大错误的节点可以访问网络资源。Nešić D^[46] 基于 TOD 的基础上给出了一个具有李雅普诺夫一致全局渐近稳定特性的方案。Branicky M^[47] 提出了一种将控制器与执行器的通道直接连接起来的方法，并采用速率单调调度算法对各子系统的传感数据进行调度。

1.2.2 计算资源受限控制系统在国内外的研究现状

最近几年,随着通讯技术特别是无线通讯技术的发展,越来越多的系统采用嵌入式或者移动式的工作模式(无人机、移动机器人等),对这类控制系统来说,用于执行控制器计算的处理器不是专用的而是和系统的其他部分共享的,而计算资源本身是有限的,因此对于这类控制系统来说如何在有限的计算资源(能量)下到达比较好的控制效果也是不得不考虑的一个因素。

针对这类采用了和其他设备共享的嵌入式微处理器作为承载控制器运行和计算的物理主体的这样一类智能系统,一般来说这类智能系统以多个具有有限计算和通信能力的设备的互联互通为基本特征。如果从计算能力的角度来看,运行控制器的设备的计算能力本身有限,又要在互联的设备间共享计算资源,这对其中的某些特定的控制系统而言,用于执行控制器功能的部分可用计算资源成为一种有限且不可预测的变量^[48-50]。文献^[51]在资源有限的前提下提出了一种基于事件的网络控制和调度联合设计的方法保证系统的性能。在这种方法中,作者将具有一组相互作用的连续时间线性时不变对象建模为一个离散时域的切换线性系统,进一步地将其表述为一个线性矩阵不等式(LMI)的优化问题加以解决。Sardellitti S^[52]等人在处理具有复杂功能的控制系统时将系统额外的一些需要大量计算能力的功能上传到固定的云端,借助云端较强的计算和分析能力对系统串串的数据进行分析和计算,并将计算结果返回到本地,进而做到节省本地计算资源的效果。

1.3 本文主要工作

本文主要考虑了资源受限控制系统的相关问题,并着重考虑了控制系统中通讯资源受限和计算资源受限的问题。在本文的第二章内容中考虑了网络化控制系统中存在由多数据包传输引起的多个随机时延进而给控制系统带来的通讯受限问题,并针对此问题给出了一种基于模型的主动补偿方法,从设计框架上考虑了多包传输对控制系统的影响,进而针对此问题给出了一个有效的降低了多时延对控制系统影响的解决方案。在解决上述多包传输引起的通讯受限对控制系统的负面影响这个问题上,本文在方法上采用了两步走的策略,首先在控制器端重构了系统状态,然后使用基于包的控制方法来补偿由通信网络带来的约束。在第三章的内容中考虑了实际应用的控制系统中存在的计算资源受限的问题,针对这种问题,在仅使用系统分配给控制系统的有限的计算资源的前提下基于预测控制的思想提出了一种全资源预测控制的方法。

本文接下来的具体内容和行文结构可以归纳如下:

在第二章的内容里，研究了多包传输引起的通讯资源受限对控制系统的影响，在对此问题进行了分析的同时给出了此类控制系统的控制器设计方案。步骤可以归纳如下：首先分析了实际应用中存在的多包传输情况（比如系统使用了地理上隔离的多个传感器的控制系统），并针对网络化控制系统中存在的由多数据包传输引起的多个随机时延引起的通讯受限问题给出了一种基于数据包的主动补偿方法，然后对此方案进行了稳定性验证，在第二章的最后一部分内容中，分别使用了基于 Matlab 的纯数值仿真和基于 TrueTime 工具箱的仿真针对本文给出的基于包的主动补偿方法进行了验证。

第三章的内容针对计算资源受限的系统进行了分析。针对一些实际应用场景下存在的控制系统和其他设备共用一个计算资源池的现象，可能存在分配给控制系统的计算资源不足的问题，进而给出了一种全资源预测控制的方法，并对其进行了性能分析和仿真验证。

在本文的最后一个章节，总结了全文的工作并对以后的工作进行了展望。

第 2 章 通讯资源受限控制系统的多时延主动补偿方法

本章的内容中，主要围绕由于多个传感器的存在而引入了多个时延的这类系统进行了控制设计和分析。作者首先对这类问题进行了描述并进一步分析了产生这种问题的原因，然后使用基于数据包的控制方法进行了控制器的设计，并在本章节的最后部分通过仿真实验验证了使用这种方法去处理这类问题相比于使用传统控制方法可以使系统达到更好的控制性能。

2.1 问题描述

对于存在通讯资源受限的且具有多个传感器的网络化控制系统而言，由于其在组成结构上和一般的网络化控制系统有区别，每个传感器发送到控制器的数据可能会存在需要经由不同的数据包进行传输，多个数据包进行数据传输时可能就会存在每个数据包的时延都不同的情况。在这种情况下，就需要使用不同于处理单一数据包传送的方法去处理多数据包传输的问题，因此，对于这类问题需要认真对待^[53-55]。

本章的这部分内容首先讨论了网络化控制系统中为什么会多包传输这种情况以及它对控制系统的潜在影响，然后在介绍了本章会用到的一些相关知识的基础上给出了本章所要讨论的问题的系统描述。

2.1.1 多包传输产生的原因和负面影响

“多包传输”这个专业术语用于描述网络化控制系统中一类特定的应用场景，在这种应用场景下，传感器数据或者控制器数据是以被封装成多个数据包的形式来进行传递的。这种情景的产生可能会有以下两种原因：

第一种原因可能是因为传感器由于地理上的隔离，传感器分别分布在不同的地方，从传感器发送到控制器的数据无法打包成一个数据包，因此这些数据包不得不以通过不同的通讯信道向控制器发送的形式进行传递，而对于这种结构的控制系统来说，由于控制器发送的控制量是被打包成一个数据包发送的，因此控制信号的传输并不受这种多包传输架构的约束，系统框图如图 2-1 所示。

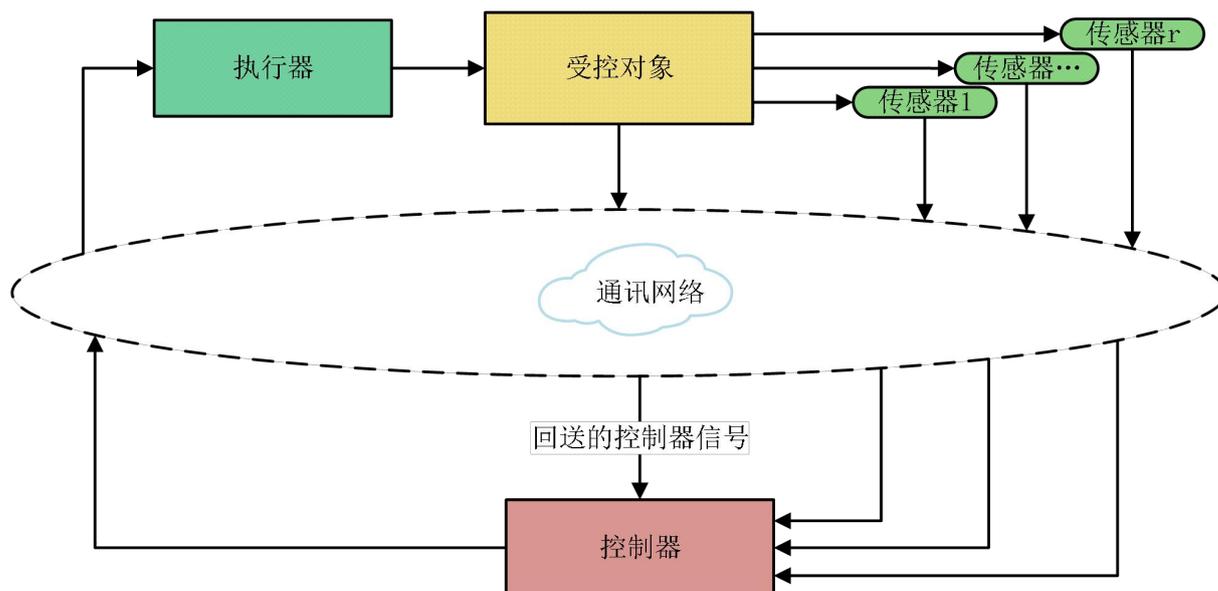


图 2-1 由地理隔离造成的多包传输框图

第二种原因可能是一些其他系统设置的影响（比如通信网络本身使用了不同的编码机制），使得系统将一些可以打包成一个数据包的数据拆分成了比较小的多个数据包。这样的话，无论是传感器数据还是控制器数据，都必须根据数据包的大小被划分成多个单独的数据包分别进行传输。

这两种情况可能都是存在的，但是对于第二种类型的多包传输来说，在基于以太网的网络化控制系统中，这类情况并不常见。这里拿一个应用比较广的以太网传输协议 IEEE802.3 协议作为例子来加以说明。一个以太网数据包包含 368 位有效编码位，从理论上讲，一个数据包会包含有 2^{368} 种不同的控制信号，因此一个数据包所包含的有效信息个数在数量上远远大于控制器的需求。因此考虑到实际情况，本章内容中只考虑第一种类型的多包传输系统，即由于地理上的隔离造成了传感器的分散，数据包不得不以通过单独的通讯信道进行传输的控制系统。这种类型的多包传输系统使得不同的传感器数据（由不同传感器产生的数据）具有不同且相互独立的时变时延，由于这些时延的存在使得系统的设计和分析变得更加复杂，因此在处理这类系统时，并不能将多个相互独立的时延简单地看作一个统一的时延进行处理。

2.1.2 系统描述

对于下面的一个离散的线性系统：

$$x(k+1) = Ax(k) + Bu(k) \quad (2-1)$$

其中, $x \in R^n, u \in R^n, A \in R^{n \times n}, B \in R^{n \times m}$ 。这里, 假设系统状态是直接可用的, 或者是可以在执行器一侧通过估计得到的。无论上面假设中的哪一种, 系统的状态量都是通过不同的传感器以单独的方式进行传输的, 因此, 系统的控制信号 $u(k)$ 是控制器在 k 时刻基于传感器数据通过计算得到的, 计算过程可以表示为:

$$x_{\tau_{sc,k}}(k) = [y_1^T(k - \tau_{sc,k}^1) \cdots y_r^T(k - \tau_{sc,k}^r)]^T \quad (2-2)$$

其中, $y_i(k - \tau_{sc,k}^i)$, $i=1,2,3,\dots$, r 代表 k 时刻控制器从第 i 个传感器获得的状态信息, $\tau_{sc,k}^i$ 是状态信息 $y_i(k - \tau_{sc,k}^i)$ 的时延写成矩阵的形式即为 $\tau_{sc,k} = [\tau_{sc,k}^1 \cdots \tau_{sc,k}^r]^T$ 。在这样一个离散系统中, $\tau_{sc,k}^i, i=1,2,3,\dots, r$ 均是非负的整数, 接下来会对这些符号做进一步的解释。

$x(k)$ 表示系统的状态量, 通过适当的变换和对每个元素的适当调整, 总可以写成下面的这种形式,

$$x(k) = [y_1^T(k) \cdots y_i^T(k) \cdots y_r^T(k)]^T = [x_1(k) \cdots x_j(k) \cdots x_n(k)]^T \quad (2-3)$$

即系统状态矩阵的组成顺序和传感器的顺序一致。因此

$$y_1(k) = [x_1(k) \cdots x_{n_1}(k)]^T, y_2(k) = [x_{n_1+1}(k) \cdots x_{n_1+n_2}(k)]^T, \cdots, \quad (2-4)$$

这里的 n_i 是传感器 i 的维度且 $\sum_{i=1}^r n_i = n$ 。本章接下来的内容也是在假设 (2-3) 成立的基础上进行的。

2.2 基于模型的系统状态重建

基于模型的多包传输主动补偿方法首先需要在控制器端根据部分时延不同的传感器数据对系统进行状态重建。本节的主要内容时讨论为什么要进行系统状态重构, 以及系统状态重建的方法。

2.2.1 系统状态重建的原因

在 2.1 节的式 (2-2) 里写出了控制系统可以直接用于计算控制信号的关于传感器时延的数学表达式。对大多数传统的控制方法而言, 控制器计算控制量时都需要完整的状态信息, 然而在多包传输的网络化控制系统中, 完整的状态量意味着全部传感器的数据都必须要在同一时刻采集完成, 这也就引入额外的人为的为了计算方便而带入的时延, 即控制

器在计算控制量时使用的时延为 $x(k - \tau_{sc,k}^{\max})$ ，这里

$$\tau_{sc,k}^{\max} = \max\{\tau_{sc,k}^i : i = 1, \dots, r\} \quad (2-5)$$

可能这种使用最大时延来计算控制量的方式在处理网络化控制系统的多包传输问题上更为简单和方便，因为使用了这种把多个时延简化成一个时延的处理方法之后，许多控制方法都可以套用。但是这种简化方式明显是不合适的，因为对控制系统来说， $\tau_{sc,k}^i < \tau_{sc,k}^{\max}$ ，也就意味着控制系统可用的状态量的时延是小于计算时所用的实际时延的。此外，任何传感器数据的丢失也都意味着在这一时刻内采集的所有的状态数据都不可用，因为这个时刻的系统状态量是不完整的，从而无法进一步运用传统的方法计算控制量。这种简化的处理方法对控制系统来说无疑是很大的资源浪费。

针对上面提到的可能存在的问题，这里的策略是使用当前所有可用的传感器信息 $x_{\tau_{sc,k}}(k)$ 去代替上述方案中人为简化后的时延比较大的状态 $x(k - \tau_{sc,k}^{\max})$ 。在本文的方案中，使用了预测控制的方法在状态量的最小时延时刻即 $\bar{x}(k - \tau_{sc,k}^{\min})$ 时刻重建了系统的状态，这里的 $\bar{x}(k - \tau_{sc,k}^{\min})$ 表示的状态量是包含了预测值和测量值的矩阵，这里

$$\tau_{sc,k}^{\min} = \min\{\tau_{sc,k}^i : i = 1, \dots, r\} \quad (2-6)$$

为了简单起见，这里定义 $\tau_{sc,k}^* = \tau_{sc,k}^{\min}$ ，同样地也定义 $\bar{x}(k - \tau_{sc,k}^*) = \bar{x}(k - \tau_{sc,k}^{\min})$ 。

由于控制器到执行器的时延是时变的，控制器本身并不知道实际应用于执行器的控制信号是哪一个，在本文的方法中需要把执行器实际使用的控制量回馈到控制器，如图 2-1 所示，同时假设回馈的时延比传感器的时延要小，即：

$$\tau_{sc,k}^u \leq \tau_{sc,k}^*, \forall k \quad (2-7)$$

其中， $\tau_{sc,k}^u$ 表示在 k 时刻执行器回馈到控制器的时延。在实际情况中，上述假设也是有道理的，因为这些控制信号比相应的传感数据发送的更早，而传感器数据则必须在控制信号被应用于受控对象之后且系统输出被采样之后才能被发送，此外如果从信道资源的占用角度来考虑，错时发送的分发方式也会优化上述两种数据到达的先后顺序。

2.2.2 系统状态重建的方法

在式 (2-1) 引入了系统矩阵 A ， B ，这里把 A 和 B 拆分成 $r \times r$ 和 $r \times 1$ 的两个块矩阵， r 表示传感器的个数，拆分的具体过程如下：

$$A = \begin{pmatrix} A^{11} & A^{12} & \cdots & A^{1r} \\ A^{21} & A^{22} & \cdots & A^{2r} \\ \vdots & \vdots & \ddots & \vdots \\ A^{r1} & A^{r2} & \cdots & A^{rr} \end{pmatrix}, B = \begin{pmatrix} B^1 \\ B^2 \\ \cdots \\ B^r \end{pmatrix} \quad (2-8)$$

这里 $A^{ij}, i, j = 1, 2, \dots, r$ 是 $n_i \times n_j$ 的维度, B^i 是 $n_i \times m$ 的维度。把矩阵分块以后可以很方便地写出当前时刻每一个传感器数据更新的数学描述:

$$y_i(k+1) = \sum_{j=1}^r A^{ij} y_j(k) + B^i u(k) \quad (2-9)$$

由于在 k 时刻, 系统的 $k - \tau_{sc,k}^{\max}$ 时刻的所有的系统状态信息都是可得到的, 因此传感器数据预测一步的数学描述可以写成下面的形式:

$$\hat{y}_i(k - \tau_{sc,k}^{\max} + 1) = \sum_{j=1}^r A^{ij} y_j(k - \tau_{sc,k}^{\max}) + B^i u(k - \tau_{sc,k}^{\max}) \quad (2-10)$$

这里的 $k \geq k_0$, k_0 是控制器第一次得到全部状态信息的时刻。在 k_0 时刻之前, 系统由于无法得到任何完整的状态量因此无法计算控制量, 此时的系统处于开环状态。

在 (2-6) 中定义了 $\tau_{sc,k}^{\max}$, 但是对控制器而言, 在 $k - \tau_{sc,k}^{\max} + 1$ 时刻之前, 系统的状态量都是不完整的, 因此, 上面说的系统状态重建的方法无法直接运用到实际系统中。若深究其原因可以发现造成上述现象的原因可能有以下两种, 第一种可能的原因是因为传感器 i 在 k 时刻的时延比较大, 如果传感器 i 的时延大于一个值的时候, 这里可以表示为 $\tau_{sc,k}^i > \tau_{sc,k}^{\max} - l$, 那么在 $k - \tau_{sc,k}^{\max} + l$ 时刻之后就不会有新的传感器 i 的系统数据传到控制器。然而, 在 (2-10) 中已经看到在系统重建的过程中, 如果系统的状态重建能够正常进行的话, 这就需要 $\hat{y}_i(k - \tau_{sc,k}^{\max} + 1)$ 时的状态必须是一直存在于控制器端的, 这明显和上面的数学分析是不相符合的。第二种可能的原因是数据包的丢失对控制系统造成了影响, 如果传感器时延小于一个定值 $\tau_{sc,k}^i \leq \tau_{sc,k}^{\max} - l$ 且发生了数据包的丢失现象, 可能控制器能接收到这一时刻之后的传感器发送过来的新状态, 但是对于丢失信息的这个传感器 i 来说, 在 $k - \tau_{sc,k}^{\max} + l$ 时刻, 控制器是接受不到这个传感器的信息的, 在这种情况下, 可以使用特殊的处理方法估计出 $k - \tau_{sc,k}^{\max} + l$ 时刻的传感器数据, 当实际的测量值可用的时候再使用测量值进行系统状态的重建。

上面的估计方法可以表述成以下的形式，当 $1 \leq l \leq \tau_{sc,k}^{\max} - \tau_{sc,k}^* - 1$ 时：

$$\hat{y}_i(k - \tau_{sc,k}^{\max} + l + 1) = \sum_{j=1}^r A^{ij} \bar{y}_j(k - \tau_{sc,k}^{\max} + l) + B^i u(k - \tau_{sc,k}^{\max} + l) \quad (2-11)$$

这里 $\bar{y}_j(k - \tau_{sc,k}^{\max} + l)$ 的定义有两种不同的形式，当传感器发送过来的系统的实际测量状态可用时，使用系统的实际测量状态来定义 $\bar{y}_j(k - \tau_{sc,k}^{\max} + l)$ ，如果系统实际可用的测量值不完整的时候，使用测量值和预测值共同组成的混合状态量作为控制器计算下一步控制信号的系统状态，为了表述不产生歧义这里用下面的式子来表示 $\bar{y}_j(k - \tau_{sc,k}^{\max} + l)$ ：

$$\begin{cases} y_j(k - \tau_{sc,k}^{\max} + l), & \text{如果}(k - \tau_{sc,k}^{\max} + l) \text{可用} \\ \hat{y}_i(k - \tau_{sc,k}^{\max} + l), & \text{其它情况} \end{cases} \quad (2-12)$$

根据 (2-1) 可知，当 $1 \leq l \leq \tau_{sc,k}^{\max} - \tau_{sc,k}^* - 1$ 时，系统的控制量 $u(k - \tau_{sc,k}^{\max} + l)$ 总是可用的。

上式 (2-11) 中的估计过程不仅可用于单独的某个传感器，也可用于所有的传感器，这样的话，对于系统来说，在 $\tau_{sc,k}^*$ 时刻，系统就能知道所有传感器的信息量（测量值和估计值的结合）。这个过程可以描述如下：

$$\bar{x}(k - \tau_{sc,k}^*) = [\bar{y}_1(k - \tau_{sc,k}^*) \cdots \bar{y}_r(k - \tau_{sc,k}^*)]^T \quad (2-13)$$

这里的 $\bar{y}_j(k - \tau_{sc,k}^*)$ 也可以用下面的式子表达：

$$\begin{cases} y_j(k - \tau_{sc,k}^*), & \text{如果}(k - \tau_{sc,k}^*) \text{可用} \\ \hat{y}_i(k - \tau_{sc,k}^*), & \text{其它情况} \end{cases} \quad (2-14)$$

总的来说，在多包传输的网络化控制系统中，由于每个传感器数据发送到控制器端的网络时延并不一样，使用基于模型的系统重建方法对丢失的部分传感器数据或者尚未到来的传感器数据进行状态重建，这样控制器就可以使用完整的传感器数据（收到的测量值和预测的值）计算控制量。相比较于传统的使用最大时延来计算控制量的方法，本章所提出的方法在系统结构上有着明显的优越性。

2.3 基于数据包的补偿

2.3.1 基于数据包的补偿方案

这个小结所讨论的主要内容是基于前面所提出的系统架构的基础上阐述控制器的设计。众所周知，在网络化控制系统中，一个数据包可以用来编码不止一个控制信号，其中

控制信号的数量可以写成

$$N = \left\lfloor \frac{B_p}{B_c} \right\rfloor = \max \left\{ \zeta : \zeta \in N, \zeta \leq \frac{B_p}{B_c} \right\} \quad (2-15)$$

上式中的 B_p 表示一个数据包的有效编码位， B_c 代表一个控制信号实际占用的有效编码位。通常来讲，这里的 N 是一个大于 1 的整数。

在控制器的设计之前，首先给出下列的两种假设。

假设 1：控制器-执行器通道中网络的延迟和连续丢包数之和小于 N ，这里用 \aleph_{ca}^{\max} 表示连续丢包数，于是上面的文字描述又可以写成下面的这种形式：

$$\tau_{ca}^{\max} := \max_{k \geq 1} \{ \tau_{ca,k} + \aleph_{ca}^{\max} \} < N \quad (2-16)$$

这里 $\tau_{ca,k}$ 表示 k 时刻控制器到执行器信道中的网络诱导时延。

假设 2：在整个网络化控制系统中，控制器、执行器和传感器之间应当是时钟同步的且相互之间发送的数据包上需要打上时间标签。

为了使上面的假设更便于读者理解，这里简单对此加以说明。通常来讲，假设 1 中的内容在实际应用中是比较有意义的，原因有以下三点：

- (1) 如果通讯信道能正常传输信息的话，(2-16) 中的 \aleph_{ca}^{\max} 就不会太大。
- (2) 网络化控制系统在进行数据包的传输过程中通常都会使用 UDP（用户数据报协议）作为传输控制协议，根据此协议，当传输时延过大时就自动认为数据包丢失。因此，能成功发送和接受的数据包一般来说都是时延比较小的数据包。
- (3) 拿前面提到的以太网为例， $B_p = 368$ ，如果每个控制信号的有效编码位是 8 位且每 8 位只能编码一个控制量的话，可以得出 $N = 46$ ，对于 (2-16) 来说，也是远远足够的。

从上面假设 1 和 2 的角度来看，执行器总会接受到从控制器发送来的带有 $\tau_{ca}^{\max} + 1$ 个控制信号的数据包，即：

$$U(k | k - \tau_{sc,k}^*) := [u(k | k - \tau_{sc,k}^*) \cdots u(k + \tau_{ca}^{\max} | k - \tau_{sc,k}^*)] \quad (2-17)$$

当执行器接收到 $U(k | k - \tau_{sc,k}^*)$ 之后就会根据当前控制器到执行器的时延从控制序列中选择一个控制量作用于执行器。执行器选择控制序列的选择过程可以表示如下：

$$u(k) = u(k | k - \tau_k^*) \quad (2-18)$$

这里 $\tau_k^* := \tau_{sc,k}^* + \tau_{ca,k}^*$ 。

具体的控制结构如图 2-2 所示，从图中可以看出，在网络化控制系统中，由多包传输带来的通讯约束进而给控制系统带来的负面影响可以使用主动补偿的方法进行处理。

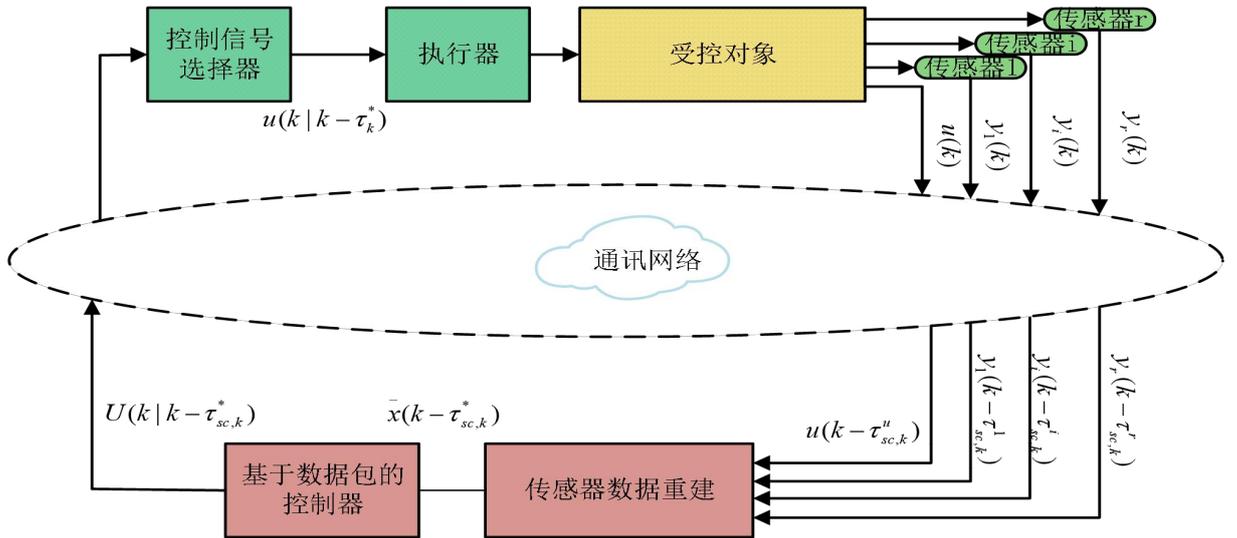


图 3-2 基于模型的多包传输网络化控制系统主动补偿方案示意图

2.3.2 基于数据包的控制设计

在本章的式 (2-18) 为网络化控制系统提供了一种通用的解决方案，任何可能改善控制系统性能的设计方法都可以采用上述方案。为了进一步验证本章节中提出的这种方法，在这部分内容中使用基于模型的预测控制方法 (MPC) 来验证本章节提出的这种框架的合理性。

在本章提出的这种基于主动补偿的方案中，目标函数的定义如下：

$$J_{k, \tau_{sc,k}^*} = \bar{X}^T(k | k - \tau_{sc,k}^*) Q \bar{X}(k | k - \tau_{sc,k}^*) + \bar{U}^T(k | k - \tau_{sc,k}^*) R \bar{U}(k | k - \tau_{sc,k}^*) \quad (2-19)$$

这里 $J_{k, \tau_{sc,k}^*}$ 是系统在时刻 k 的目标函数， Q 、 R 是系统的加权矩阵， $\bar{U}(k | k - \tau_{sc,k}^*)$ 是系统的控制信号表示如下：

$$\bar{U}(k | k - \tau_{sc,k}^*) = [u(k - \tau_{sc,k}^* | k - \tau_{sc,k}^*) \cdots u(k + N_u - 1 | k - \tau_{sc,k}^*)]^T \quad (2-20)$$

系统状态轨迹的预测量用 $\bar{X}(k | k - \tau_{sc,k}^*)$ 表示：

$$\bar{X}(k | k - \tau_{sc,k}^*) = [x(k + 1 | k - \tau_{sc,k}^*) \cdots x(k + N_p | k - \tau_{sc,k}^*)]^T \quad (2-21)$$

其中 N_p 是预测时域, N_u 是控制时域, 通常 $N_p \geq N_u$, 这里假设 $N_u > \tau_{ca}^{\max}$ 。

对于本方案中的预测的控制信号序列将在 (2-22) 给出, 有关预测控制的详细情况, 感兴趣的读者可以参考^[43]。

$$U(k | k - \tau_{sc,k}^*) = K_{\tau_{sc,k}^*}^* \bar{x}(k - \tau_{sc,k}^*) \quad (2-22)$$

这里

$$K_{\tau_{sc,k}^*}^* (k | k - \tau_{sc,k}^*) = -M_{\tau_{sc,k}^*}^* (F_{\tau_{sc,k}^*}^{*T} Q F_{\tau_{sc,k}^*}^* + R)^{-1} F_{\tau_{sc,k}^*}^{*T} Q E_{\tau_{sc,k}^*}^* \quad (2-23)$$

上面的 $E_{\tau_{sc,k}^*}^* = [(A^{\tau_{sc,k}^*+1})^T \cdots (A^{\tau_{sc,k}^*+N_p})^T]^T$, $F_{\tau_{sc,k}^*}^*$ 是由 $(F_{\tau_{sc,k}^*}^*)_{ij}$ 和 $M_{\tau_{sc,k}^*}^*$ 定义的下三角非负矩阵, 这里 $(F_{\tau_{sc,k}^*}^*)_{ij} = A^{\tau_{sc,k}^*+i-j} B$, $j-i \leq \tau_{sc,k}^*$ 且 $M_{\tau_{sc,k}^*}^* = [0_{m(\tau_{ca}^{\max}+1) \times m\tau_{sc,k}^*} \quad I_{m(\tau_{ca}^{\max}+1) \times m(\tau_{ca}^{\max}+1)} \quad 0_{m(\tau_{ca}^{\max}+1) \times m(N_u - \tau_{ca}^{\max})}]$ 。

2.4 闭环稳定性分析

在本小节的内容中, 主要讨论了基于模型的主动补偿方法的系统稳定性, 以及基于此方法的一般性归纳。

在 2.3.2 中讨论了系统对传感器数据的重建过程, 在 $k - \tau_{sc,k}^*$ 时刻, 估计出来的传感器数据 $\bar{x}(k - \tau_{sc,k}^*)$ 由两部分数据计算得到, 一部分指 $k - \tau_{sc,k}^{\max}$ 到 $k - \tau_{sc,k}^*$ 这段时刻之内控制器接收到的控制量, 另一部分指的是在这段时间之内控制器接收到的所有可用的但是不完整的传感器数据。系统收到的不完整的传感器数据可用数学语言描述成如下形式

$$\{y_j(k - \tau_{sc,k}^{\max}), \dots, y_j(k - \tau_{sc,k}^j) : j = 1, 2, 3 \dots r\} \quad (2-24)$$

基于上面的讨论这里构造出下列的系统增广状态

$$X(k) := [x(k)x(k-1) \cdots x(k - \bar{\tau}_{sc})u(k-1)u(k-1) \cdots u(k - \bar{\tau}_{sc})] \quad (2-25)$$

由 (2-19)、(2-11)、(2-14) 可以构造出 $\tau_{sc,k}$ 时刻的增广矩阵

$$\bar{x}(k - \tau_{sc,k}^*) = \Lambda_{\tau_{sc,k}^*} X(k) \quad (2-26)$$

由 (2-18)、(2-22) 系统控制信号的计算过程可以得出:

$$u(k) = d_{\tau_{ca,k}^*} K_{\tau_{sc,k}^*}^* \Lambda_{\tau_{sc,k}^*} X(k) \quad (2-27)$$

这里的 $d_{\tau_{ca,k}^*}$ 是系统根据控制器到执行器的时延 $\tau_{ca,k}^*$ 从控制序列 $U(k | k - \tau_{sc,k}^*)$ 中选取控制量时所用的矩阵。

根据 (2-26) 和 (2-27) 可以把闭环系统写成下面的形式

$$X(k+1) = \Pi_{\tau_{sc,k}} X(k) \quad (2-28)$$

这里的系统矩阵 $\Pi_{\tau_{sc,k}}$ 由 $\Lambda_{\tau_{sc,k}}$ 和 $K_{\tau_{sc,k}}^*$ 定义。

基于上面的讨论内容, 根据李雅普诺夫稳定性理论可以很容易得出下列的系统稳定性条件:

对于一个 (2-1)、(2-2) 所描述的闭环系统, 如果存在一个正定矩阵 $P = P^T > 0$ 使得 (2-25) 表示的线性矩阵不等式有解, 则在 (2-13)、(2-18) 和 (2-22) 中所讨论的系统状态重建过程和基于数据包的控制方法是稳定的。

$$\Pi_{\tau_{sc,k}}^T P \Pi_{\tau_{sc,k}} - P < 0, \Pi_{\tau_{sc,k}} \in \Pi \quad (2-29)$$

由于这个过程的证明可以很容易地在了一本标准教科书中找到, 这里就不再加以证明。

虽然这章节所讨论和分析的内容是基于一个简单的线性系统模型来进行的, 但是文中所用的系统状态重建的基本思想对于其他的某些系统来说却是通用的, 特别地, 对于控制器端存在数学模型的这类系统。因为有了这样一个数学模型, 控制器就可以根据它去预测系统的状态。这就使基于模型的主动补偿方案成为一种处理网络化控制系统中多包传输问题的通用框架。这种通用的解决方案可以为任何一个使用了这种框架的控制系统提供“主动补偿”。

在 (2-7) 中做了系统时延比较小的假设, 如果将本章节提出的框架的作用范围进一步扩大, 即不考虑 (2-7) 的假设的话, 广义上的系统状态重建是完全不受影响的, 这种情况只需要保证系统在没有收到控制信号使用零控制信号 (或者其他合适的方案)。除了这种情况之外, 这时候就不用人为地去假设系统延时的界限, 因为从理论上讲, 不管时延是多少总可以估计出系统的状态。本章的内容中, 由于考虑到太大的时延可能导致估计值和实际值相差太远这种实际的情况才人为限定了系统时延的大小。

2.5 仿真验证

在本小节中, 分别使用了基于 Matlab 和基于 TrueTime 工具箱的仿真方式和去验证本章所使用方法的可行性, 并分别使用了不同的系统去验证本章节方法的可行性, 进而给出了此方法和其它方法的系统状态相应的对比图。

2.5.1 基于 Matlab 的数值仿真

为了验证本章节所提方法的有效性，这里作者使用 Matlab 做了一个数值验证。在本小节里，从文献^[43]中借用如下一个系统，系统的参数设置如下：

$$\begin{aligned} A &= \begin{pmatrix} 0.98 & 0.1 \\ 0 & 1 \end{pmatrix} \\ B &= \begin{pmatrix} 0.04 \\ 0.1 \end{pmatrix} \end{aligned} \quad (2-30)$$

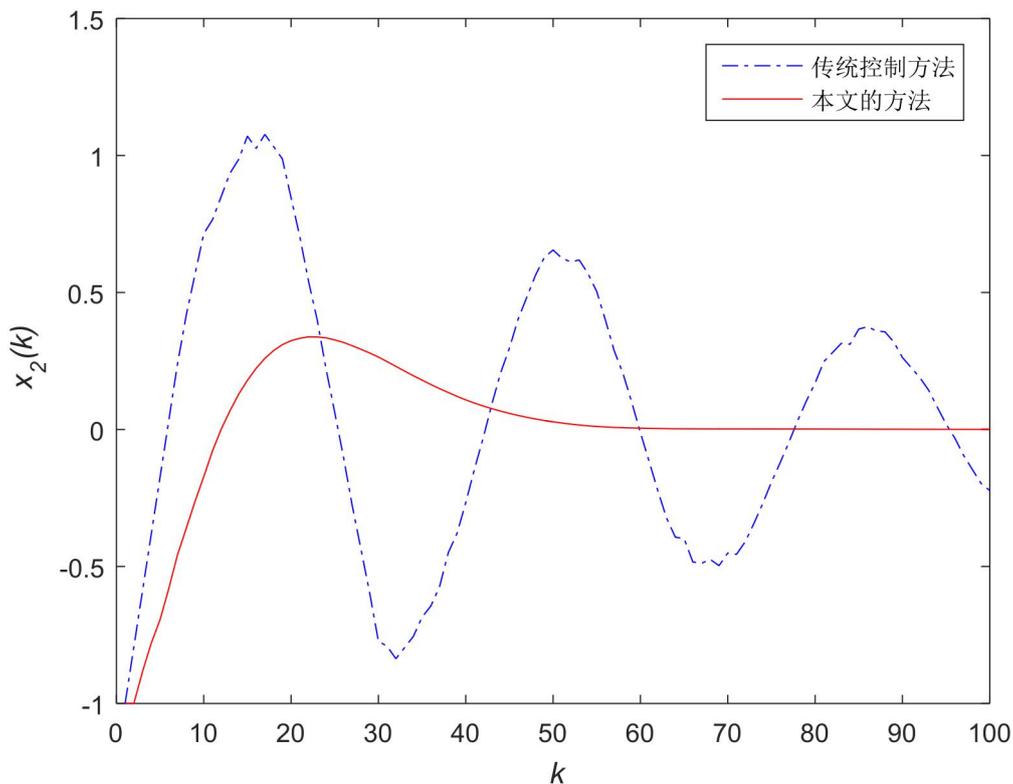


图 2-3 系统状态对比图

为了使仿真情况更加符合本文的方案，这里选取了一个没有考虑通讯约束的 LQR 控制器和本章的方法做对比，这里 LQR 的反馈矩阵的参数设置为： $K_{LQR} = [0.7044, 1.3611]$ 。此外还对程序进行了处理，目的是为了使系统的两个状态量分别由不同的传感器采集且通过不同的通讯信道分别传输到控制器。系统的传感器的最大时延上界分别设置为 4 个步长和 8 个步长， $\tau_{ca}^{\max} = 3$ ，且系统的两个初始状态均设置为 -1，同时设置 N_p 、 N_u 分别为 20 个步长和 15 个步长。两种方法的系统状态量的变化图如图 2-3 所示。从图中的曲线可以看出，对于同一个系统来说，使用本章节方法能在保证系统稳定性的前提下使系统达到更好

的控制性能。

2.5.2 基于 TrueTime 工具箱的仿真

在本小节的内容中使用了一种专业的可以用于仿真网络化控制系统的 Matlab 工具箱 TrueTime，使用这种平台的优势在于其可以模拟出一个通讯网络，对随机时延的处理上更加符合真实的应用场景。和上面的完全基于 Matlab 的数值实验相比，此工具箱的网络时延不再是由简单是随机数生成器产生，而是和其模拟出来的一个网络相关。因而这种模拟出来的网络在时延的处理上能更接近实际的应用场景。其它关于 TrueTime 的更加详细的内容请参阅这篇文献^[56]。

在本文的这次数值仿真中，使用 TrueTime 搭建了一个模拟网络，各个网络节点之间的数据传递，比如传感器到控制器、控制器到执行器之间的数据都是在这个模拟出来的网络上进行传递的。本次实验选择的对比方法为 LQR 方法，在实验参数的处理上，为了控制单一变量，即除了控制方法以外的所有条件都相同，这里把两个使用了不同方法的控制器搭建进了同一网络。

在本次仿真中考虑了如下一个系统^[57]：

$$\begin{aligned} \dot{X} &= \begin{pmatrix} -3 & -7 & -5 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} X + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} u \\ Y &= \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} X \end{aligned} \quad (2-31)$$

将其以 5ms 为采样周期进行离散化之后得到如下一个状态空间方程（这里只考虑了系统的状态量）：

$$x(k+1) = \begin{pmatrix} 0.9850 & -0.0348 & -0.248 \\ 0.0050 & 0.9999 & -0.0001 \\ 0 & 0.0050 & 1 \end{pmatrix} x(k) + \begin{pmatrix} 0.0050 \\ 0 \\ 0 \end{pmatrix} u(k) \quad (2-32)$$

从上述的状态空间方程可以看出本次实验的状态空间方程中含有三个状态信息。在本次仿真中，系统的三个状态数据分别由不同的传感器采集并通过单独的数据通道发送到控制器节点，每个传感器的采样周期均为 5ms。在传感器发送状态数据的同时还会将当前的系统时间以时间戳的形式打包进一个数据包。控制器节点在每次收到传感器的数据时都会记录当前的系统时间，并会将当前时间和传感器的时间戳相减以获取数据从传感器到控制器的

传输延时；然后根据这个时延进行系统状态的重建的控制量的计算。因为考虑的是离散系统，这里将时延做了处理以获取延时的离散步长。具体的处理方式为：系统的步长 t_{step} 等于时延除以采样周期，即传感器到控制器的时延步长为 $t_{step,sc} = \frac{\tau_{sc}}{T}$ ，控制器到执行器的采样步长为 $t_{step,ca} = \frac{\tau_{ca}}{T}$ 。

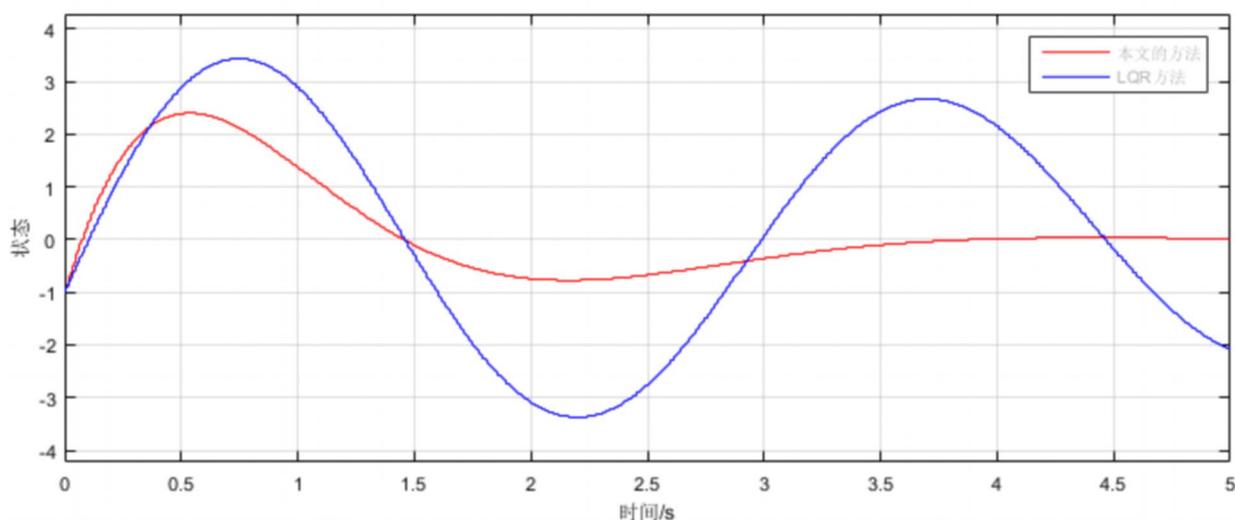


图 2-4 系统状态图

在系统的参数处理上，设定系统的三个初始状态都是-1，传感器工作模式设置为周期性的采样，控制器和执行器都设置为事件触发模式。通讯网络的网络类型设置为比较常用的 WLAN (802.11b) 网络，考虑到实际应用时的网络情况，这里将丢包率也考虑进去并设其设为 5%，数据的传输速率为 800000bits/s。图 2-4 是使用了两种方法的系统状态量随时间变化的对比图，图 2-5 为两种方法在当前网络情况下的系统传感器到控制器的时延 τ_{sc} ，图 2-6 为两种方法在当前网络情况下的系统控制器到执行器的时延 τ_{ca} 。从图中的曲线可以看出，使用本章节方法能在保证系统稳定性的前提下使系统达到更好的控制性能。

2.6 本章小结

本章介绍了一种处理网络化控制系统多包传输问题的通用框架，此框架使用基于数据包的主动补偿方法来应对多包传输过程中存在多个时延的现象。在此框架中包含的两个步骤：系统状态重建和基于数据包的补偿。本章的主要篇幅都是围绕这两个步骤展开的，在

本章的最后部分内容中，分别运用了基于 Matlab 和 TrueTime 工具箱的仿真实验去验证本章所提的这种方案的有效性。通过图形可以得出，使用本方法去处理多数据包传输时延的问题和使用传统的方法相比能明显的改善系统的控制性能。

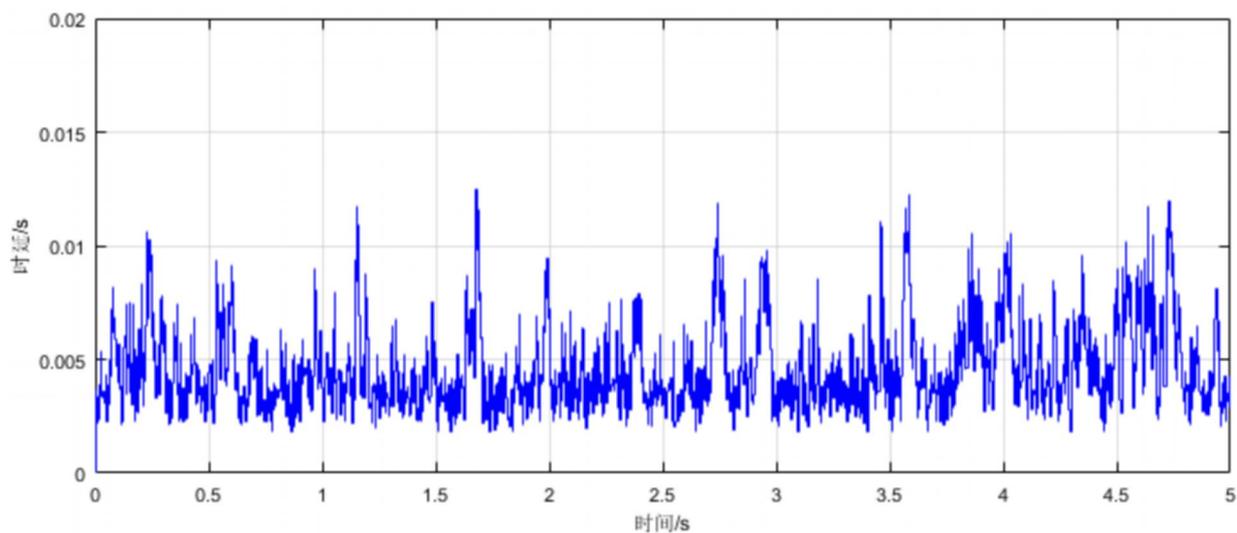


图 2-5 传感器到控制器的时延分布图

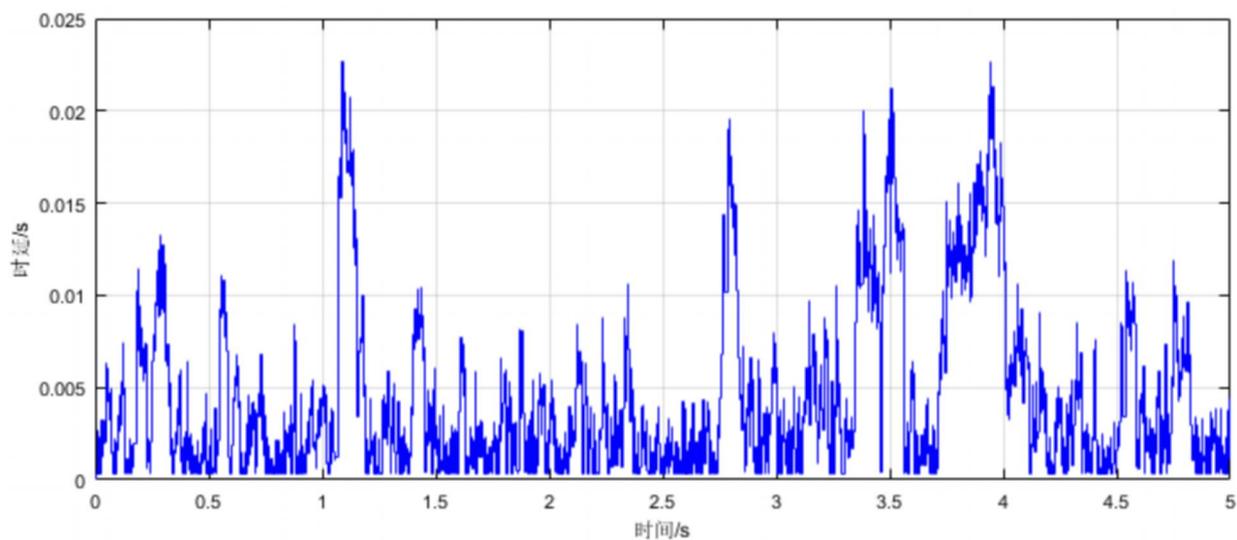


图 2-6 控制器到执行器的时延分布图

第3章 计算资源受限控制系统的全资源预测控制方法

在本章节中，提出了一个面向计算受限网络控制系统的全资源预测控制策略及其控制器的设计方法。该策略注意到了计算资源的这种时变特点，即在某些时刻可用计算资源可能极为充足，除去控制器的计算之外还有大量的剩余，因此就可以使用这些剩余的计算资源来进行控制量的前向预测，这一策略仅使用已分配给控制系统的计算资源，但通过对其更充分的利用使其能达到了更好的控制系统性能。

针对上述的问题，首先严格定义了本章所要考虑的问题；接着提出了全资源预测的控制策略；然后对所提的方案进行了性能分析并使用了一般意义上的李雅普诺夫函数分析了这种方案的稳定性；最后通过仿真对提出的策略和算法进行了验证。

3.1 问题描述

在本章的内容中，主要考虑了一类控制系统和整个系统的其他组件共享有限的计算资源而可能引起的控制系统计算资源不足的情况。这里对此加以详细说明，如图 3-1 所示的一个控制系统，其控制对象经离散后得到如下一个系统描述：

$$x(k+1) = Ax(k) + Bu(k) \quad (3-1)$$

这里的 $x(k) \in R^n, u(k) \in R^m, A \in R^{n \times n}, B \in R^{n \times m}$ 。

在图 3-1 中，嵌入式除了负责控制系统的控制器计算任务之外，还要处理其他系统发送过来的各种信号。由于处理器的计算能力本身有限，加之有限的计算资源在某些情况下需要和其他的设备进行共享，也就会造成该计算能力在控制系统的边界之外使控制系统本身无法管理，这也就意味着分配给所考虑的控制系统的计算资源表现为时变且不可预知的特性。

为了表述不产生歧义，可以通过下面的一系列数学描述进行阐释。如果使用 $R(k)$ 表示 k 时刻处理器分配给控制系统的计算资源，用 $R^*(k)$ 表示其总的计算资源，则会有如下的关系：

$$R(k) < R^*(k) \quad (3-2)$$

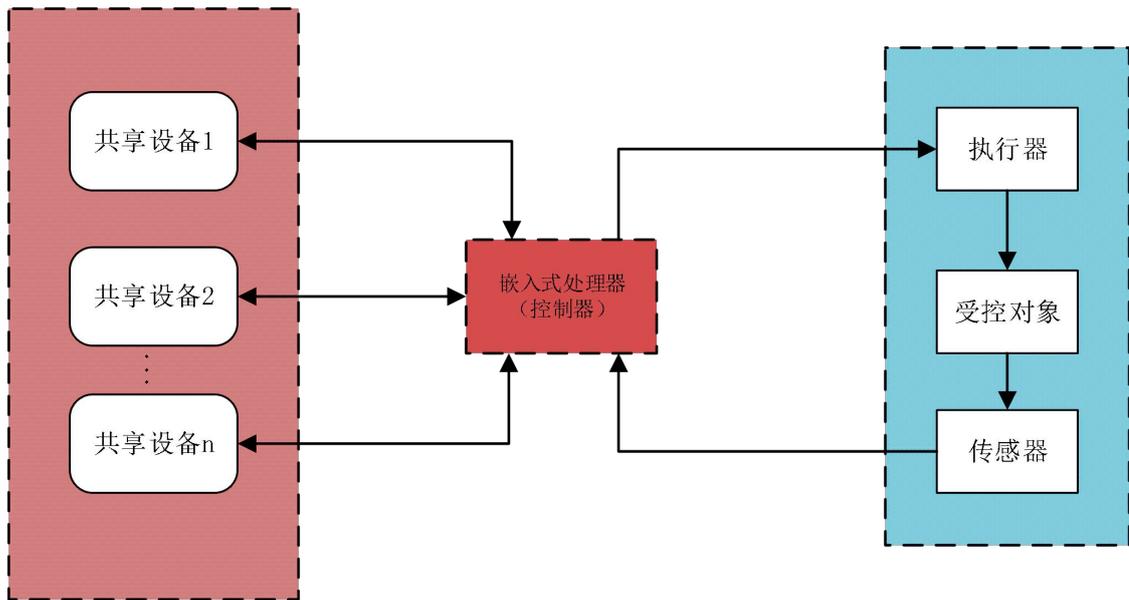


图 3-1 控制器和其它设备共享计算资源的系统结构图

如果将控制系统计算控制信号 $u(k)$ 所需的最小计算资源记为 R_u^* ，则需满足 $R_u^* < R^*$ ，因为如果该处理器的最大计算能力都不足以运行控制器，这对控制系统而言也就意味着会因计算能力不足而出现停摆的现象，这无疑使毫无意义的。在时刻 k 控制量可被计算的条件是在此时刻控制系统可使用的计算资源不小于所要求的最小计算资源，即：

$$R_u^* \leq R(k) \quad (3-3)$$

上述的 (3-2) 和 (3-3) 可合并如下：

$$R_u^* \leq R(k) \leq R^*, \forall k \quad (3-4)$$

然而，可用计算资源的时变特性意味着在某些时刻系统分配给控制系统的计算资源 $R(k)$ 是有限的，即：

$$\exists k, s.t. R(k) < R_u^* < R^* \quad (3-5)$$

为了解决上述的问题，一般意义上的可能的解决方案是对计算资源进行全局调度。然而，调度算法在这类问题中通常难以见效，原因有以下两点：① 在处理器的计算资源是有限的并且共享的系统和设备对处理器计算资源要求较高的前提下（这是本文考虑问题的出发点），上式 (3-4) 中的计算资源要求不可能对所有时刻都满足（否则传统控制方法即可满足要求），这也就意味着式 (3-5) 中的情形总会出现，而在这些情形下，传统的控制方

法便失效了；②调度算法通常无法精确地将控制系统对实时计算资源的需求考虑在内，因此在进行资源调度时可能会对其分配计算资源过少而造成式(3-5)中的情形，或者太过保守将非必需的计算资源分配给控制系统进而造成不必要的浪费，也就是会出现如下情形：

$$\exists k, s.t. R_u^* \leq R(k) \leq R^* \quad (3-6)$$

综上所述，考虑如下的问题：

问题 1：考虑如图(3-1)所示的控制系统，其中控制系统可用计算资源是有限时变且不可预知的，即在某些时刻会太少而不能执行控制器的计算(式(3-5))，而在其他一些时刻在执行完控制器计算后还有大量剩余(式(3-6))。针对这个问题设计控制策略以有效使用计算资源，提升控制系统性能。

值得说明的是，图 3-1 中的嵌入式处理器的有限计算资源在控制系统的控制器和其他可能的共享设备中通过某种调度算法进行分配，但本章内容中并不研究这些计算资源的调度算法，而是在一定的计算资源调度算法下，研究如何对分配给控制系统的有限时变而不可预知的计算资源进行高效的使用。从这个意义上说，本文所做的研究并非是计算资源调度算法的替代，而是其在控制系统应用场景下的补充。

3.2 全资源预测控制策略

3.2.1 可用计算资源与控制预测步长

在本节的主要内容中，首先讨论了控制系统可用计算资源与控制预测步长的一个基本关系，随后给出了全资源预测控制策略的描述。

本小节所要讨论的内容主要基于如下事实：控制器所需的计算资源随着控制预测步长的增加而严格增加。注意到计算资源一般可通过分配的对处理器的占用时间来衡量，在不引起误解的前提下，用 $R(k)$ 本身来表示在 k 时刻分配给控制系统的处理器时间，从而 $R(k) \in \mathbb{R}_0^+$ ，这里 \mathbb{R}_0^+ 是非负实数集。这里，把计算控制步长 i 所需要的最小资源记为 R_i^* (从而 $R_0^* = 0$ 且 $R_1^* = R_u^*$)。有如下的从可用计算资源到控制预测步长的映射， $f: \mathbb{R}_0^+ \rightarrow \mathbb{N}_0^+$ ，其中 \mathbb{N}_0^+ 是非负整数集：

$$f(R(k)) = N_k, \text{ 如果 } R_{N_k}^* \leq R(k) < R_{N_k+1}^* \quad (3-7)$$

从上式可以看出，对任一时刻 k 的任意可用计算资源，将有唯一的控制预测步长 N_k 与之相

对应，其中 $f(\bullet)$ 是一个右连续的分段常数值函数。

上式 (3-7) 中的映射给出了一个看待控制系统可用资源 $R(k)$ 的另一个视角，对于系统的可用资源 $R(k)$ 可以把它等价的看作计算资源可以处理的控制预测步长数这里用 N_k 表示。除此之外，从上面的 (3-5) 和 (3-6) 也可以得出：(1) 如果式子 (3-5) 成立，那么 $N_k = 0$ ，(2) 如果 (3-6) 成立，则 $N_k \geq 1$ 。前一个条件 $N_k = 0$ 使得传统的控制算法在这里失效了，因为在这种情况下，可用的计算资源连一步控制信号都无法计算出（要求 $N_k \geq 1, \forall k$ ），而后一个条件 $N_k \geq 1$ 则提供了更为有效利用可用计算资源的可能性。因此在这种情况下，可以把其余的计算资源用于计算多步控制信号。

应当说明的是，上式 (3-7) 的映射是依赖具体的控制算法的：若控制算法本身复杂度比较高，则由 $R(k)$ 映射到的 N_k 就比较小，反之亦然。但这并不影响这一映射的基本形式以及上述函数的性质，因此也并不影响后续的关于此问题的深入讨论。

3.2.2 全资源预测控制策略及算法

本文针对计算受限控制系统的全资源预测控制策略的基本架构如图 3-2 所示：

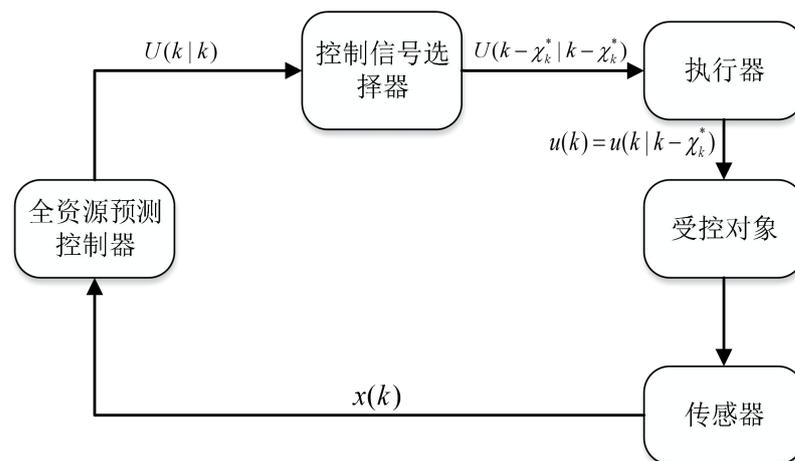


图 3-2 全预测控制策略的结构图

传统控制方法仅计算当前时刻的控制量，而在本章的全资源预测控制策略中，分配给控制系统的计算资源被充分使用于计算尽可能多的前向预测控制量。由公式 (3-7) 中的映射，并记基于 k 时刻信息计算的 $k+i$ 时刻的预测控制量为 $u(k+i|k), i=0,1,2,\dots,N_k$ ，则时刻 k

的前向预测控制序列可构建如下：

$$U(k|k)=[u(k|k)u(k+1|k)u(k+N_k-1|k)] \quad (3-8)$$

根据本章关于此类问题的描述，图 3-2 中的控制信号选择器设计为两个部分：一个为能够存储最大可能长度的前向预测控制序列的寄存器，另一个是用来选择合适控制信号的信号选择器。且控制信号选择器按照如下方式工作：

(1) 在新的前向控制预测序列到达时更新寄存器，保证寄存器中的前向控制预测序列总是最新的。从式子 (3-5) 可知，寄存器中 k 时刻的前向控制预测序列并不一定是在当前时刻计算的。这里，记 k 时刻寄存器中的前向控制预测序列已经在寄存器中存储了 χ_k^* 步，这样， k 时刻的前向控制预测序列则是在 $k-\chi_k^*$ 时刻计算的，可写作 $U(k-\chi_k^*|k-\chi_k^*)$ 。

(2) 若在当前资源无法计算控制信号时，从前向控制预测序列中选取合适的数来替代控制量。若使用上述的符号来表示，则 k 时刻从 $U(k-\chi_k^*|k-\chi_k^*)$ 中选取的控制量可表示为：

$$u(k)=u(k|k-\chi_k^*) \quad (3-9)$$

从上面的描述中可知，尽管在计算资源受限时系统选择的 $u(k|k-\chi_k^*)$ 时基于之后的系统状态 ($k-\chi_k^*$ 时刻) 计算得到的，但是这一控制量是专门针对 k 时刻来计算的，如果在此基础设计合适的前向控制预测序列，其效果会好于传统的在无法获得当前控制量时采用零控制的策略。

为使得在任何时刻从前向控制预测序列中都有控制量可用，可用计算资源需满足如推论 1 所述的前提条件。

推论 1 若下式成立：

$$N_{k-\chi_k^*} \geq \chi_k^* + 1, \forall k > \chi_k^* \quad (3-10)$$

则 (3-9) 中的控制量总在控制信号中可供信号选择器选择。

对上述的推论在这里给出简易的证明：由于 k 时刻的前向预测控制序列 $U(k-\chi_k^*|k-\chi_k^*)$ 是在 $k-\chi_k^*$ 时刻计算的。为了使得控制信号选择器能按照式子 (3-9) 的规则从 $U(k-\chi_k^*|k-\chi_k^*)$ 中选择控制量，则需要满足前向控制预测序列的长度 $N_{k-\chi_k^*}$ ，不小于

$k - (k - \chi_k^*) + 1$ ，这正是上述推论给出的条件。

然而上述的推论中可能会存在如下的两种特殊情况：

1) $\chi_k^* = 0$ 的情况。这一条意味着控制系统的可用计算资源 $R(k)$ 总能满足至少计算一步控制量，即 $N_k \geq 1, \forall k$ 。在这种情况下，传统的控制方法也可使用，因此，在这种情况下传统控制的方法可以作为本文方法的一种特例。

2) $\exists k, \chi_k^* > 0$ 。这一条件意味着控制系统的可用计算资源 $R(k)$ 在时刻 $k - \chi_k^* + 1$ 到 k 内不能满足任何控制量的计算，即 $N_i = 0, k - \chi_k^* + 1 \leq i \leq k$ 。在这种情况下，传统的控制方法将无法工作，而如果采用了本文的方法，总会有预测控制量 $u(k | k - \chi_k^*)$ 作用于执行器。

根据上面的描述，可以把本章所提的算法总结成以下步骤，算法 1：

- 1) 传感器采样系统状态并将其传至控制器；
- 2) 由当前可用的计算资源 $R(k)$ 按照 (3-7) 获得可计算的控制序列步长 N_k ，进而按照数学表达式 (3-8) 计算前向控制预测序列并将其传至执行器；
- 3) 控制信号选择器更新寄存器，并按照 (3-9) 选择控制量供执行器使用。

这里对上述的算法 1 加以简要的说明。首先，上述算法 1 中并未考虑计算时间所引起的时延，这是因为和控制系统的时间尺度相比较这一计算时间往往是足够小且是可以忽略的。但是应当说明的是，如果系统有必要考虑这个时延可以将其归入执行器的等待时延 χ_k^* 中，因此这种处理方式可以在不更改系统的架构下进行有效的处理。其次，上述的算法 1 仅对已经分配给控制系统的计算资源做有利于控制系统性能的合理使用，其权限局限于控制系统本身，对整个嵌入式处理器的计算资源的分配需由上一级机构，即由计算资源的调度算法做出，本章的全资源预测控制策略是在此意义下对计算资源调度的一个有效补充。

3.3 系统性能讨论和稳定性分析

3.3.1 系统性能讨论

传统控制方法在无法获得当前控制信号时，往往会使用零阶保持（即保持先前的控制输入），或者采用零控制（使系统临时开环）。在上述的这两种情况下，对本文所考虑的计算资源有限的控制系统来说，使用传统控制方法实际所采用的控制信号如下：

$$\tilde{u}(k) = \begin{cases} u(k - \chi_k^* | k - \chi_k^*) \text{或} 0 & \text{如果式子 (3-5) 成立} \\ u(k | k) & \text{其它情况} \end{cases} \quad (3-11)$$

与之相对应的,采用全资源预测控制策略的实际控制信可通过 (3-9) 给出, 记作:

$$u(k) = \begin{cases} u(k | k - \chi_k^*) & \text{如果式子 (3-5) 成立} \\ u(k | k) & \text{其它情况} \end{cases} \quad (3-12)$$

从上面 (3-11) 和 (3-12) 可以看出, 在控制系统被分配的计算资源充足时, 就实际使用的控制信号而言, 使用或不使用本文提出的全资源预测控制策略并无区别; 但是, 在当前计算资源不足以进行控制计算时 (即 (3-5) 成立), 相比较于传统的控制方法, 根据本文的策略可以针对具体情境根据 (3-8) 来设计前向预测控制序列对本来缺少的控制信号进行代替补偿。

从这一角度来看, 本章节中提出的全资源预测控制策略通过充分使用分配给控制系统的计算资源 (而并不要求更多), 可以实现至少不差于传统控制所带来的控制性能, 除此之外, 通过合理的设计 (3-8) 中的前向预测控制序列 $U(k|k)$, 提供了提升系统性能的自由度和可能性。

另外, (3-8) 中的前向预测控制序列 $U(k|k)$ 仅是一种一般形式, 针对具体形式和目标系统还需要具体的设计。事实上, 只要可以产生所需的前向控制预测序列, 这种框架允许所有可能的控制器设计方法。比如在第 3.3 节中给出了一种基于改进的模型预测控制的设计方法。更进一步的说, 从方法的适用性和 $U(k|k)$ 的一般性上说, 本章所提出的全资源预测控制策略可用于任意受控对象, 而并不局限于线性系统模型, 本章中采用线性对象仅为了后续系统分析和具体前向预测控制序列实现的方便。

3.3.2 闭环稳定性分析

由算法 1 计算得到的控制量是由 (3-8) 和 (3-9) 得到的。但是 (3-8) 中预测控制序列 $U(k|k)$ 的设计是一个开放的问题, 并没有局限于任何一种特定的形式, 因此闭环系统的具体形式和相应的闭环稳定性分析就难以用一般性的量化给出。从另一方面看, 如上面 3.2.2 中所述, 控制系统的可用计算资源由上层调度机构给出, 在很多情况下可能无法事先估计, 这导致了对可用计算资源无法用定量的形式描述, 也使得一般性的系统闭环稳定性分析较难进行。

为了能够以量化的形式探讨闭环系统的稳定性, 这里以一种简单情况为例对其加以说

明。首先假设预测控制序列 $U(k|k)$ 采用状态反馈。此时实际使用的控制信号 $u(k|k - \chi_k^*)$ 也具有状态反馈的形式，即：

$$u(k|k - \chi_k^*) = K_{\chi_k^*} x(k - \chi_k^*) \quad (3-13)$$

其中状态反馈增益 $K_{\chi_k^*}$ 显式依赖 χ_k^* 。进而闭环系统具有如下的形式：

$$x(k+1) = Ax(k) + BK_{\chi_k^*} x(k - \chi_k^*) \quad (3-14)$$

其次，由于分配给控制系统的计算资源并不是一直缺乏的，那么存储在寄存器中的前向预测控制序列就总可以在有限时间内得到更新。也就是说，存在一个上界 N_R ，使得 $0 \leq \chi_k^* \leq N_R$ 。进而，可以构建出如下增广系统状态 $X(k)$ ：

$$X(k) = [x(k)x(k+1)\cdots x(k+N_R)]^T \quad (3-15)$$

闭环系统 (4-14) 进而可以写成如下的增广形式：

$$x(k+1) = \Lambda_{\chi_k^*} X(k) \quad (3-16)$$

其中 $\Lambda_{\chi_k^*}$ 可由 A, B, χ_k^* 和 $K_{\chi_k^*}$ 等参数确定，在此不再详细写出。

将上述系统视为一般的切换系统，则可以利用标准切换系统方法在李雅普诺夫理论和线性矩阵不等式框架下得到其稳定性的条件。这种方法已经是比较成熟的，这里就不再给出详细的证明过程。

应当指出的是，上述稳定性分析是基于对控制律的状态反馈形式的假设，并不具有普适性，所得结论也很难对前向控制预测序列的设计提供好的指导意义。所以，在实际应用中闭环系统的稳定性尚需在具体的控制律设计方法给出后另行具体分析。

3.4 基于改进模型预测控制的前向控制预测序列计算

如前所述，前向预测控制序列 $U(k|k)$ 需针对所要考虑的系统进行进一步的具体化。针对本文考虑的 (3-1) 的系统，这小结的主要内容是在此基础上提出一种改进的模型预测控制设计方法作为 $U(k|k)$ 的一种具体实现方法。经典模型预测控制是一种有限时域优化控制方法：它通过优化有限步长的系统轨迹来求取一串控制序列，然后其中的第一步用于实际控制。通过在每一步的滚动优化，模型预测控制方法可以处理噪声以及其他的一些不确定性限制。

模型预测控制的目标函数一般定义如下：

$$J_k(N) = X^T(k|k)Q(N)X(k|k) + U^T(k|k)R(N)U(k|k) \quad (3-17)$$

其中 $J_k(N)$ 是 k 时刻的目标函数，定义于 (3-8) 中的控制序列 $U(k|k) = [u(k|k) \cdots u(k+N-1|k)]^T$ 是需要确定的控制预测序列， $Q(N)$ 和 $R(N)$ 是系统的权重矩阵， N 为预测时域。

上述优化问题的计算复杂性主要决定于预测时域 N 。如果把解决预测时域为 N 的优化问题所需要的最小计算资源为 R_{j_N} ，则根据式子 (3-7) 中所定义的函数 $f(\bullet)$ 可对目标函数 (3-17) 进一步具体化成如下的形式：

$$\text{如果 } R_{j_{N_k}} \leq R(k) < R_{j_{N_{k+1}}}, f(R(k)) = N_k \quad (3-18)$$

依照上式得到的具体形式的函数 $f(\bullet)$ 意味着带目标函数 (3-17) 和 $N = N_k$ 的优化问题可以在时刻 k 使用可用的计算资源进行求解。为了解决优化问题， k 时刻的预测状态可以递归求解如下：

$$x(k+j|k) = A^j x(k) + \sum_{l=0}^{j-1} A^{j-l-1} B u(k+l|k) \quad (3-19)$$

定义 $E(N_k) = [A^T \cdots (A^{N_k})^T]^T$ ， $F(N_k)$ 是 $N_k \times N_k$ 的下三角的矩阵块，其非零项定义为 $F(N_k)_{ij} = A^{i-j} B, j < i$ 。则上式的向量形式可以表示如下：

$$X(k|k) = E(N_k)x(k) + F(N_k)U(k|k) \quad (3-20)$$

最优的前向控制预测序列可以通过将 (3-20) 带入 (3-17) 并将 $J_k(N_k)$ 最小化之后得到，其解具有如下的状态反馈形式：

$$U(k|k) = K(N_k)x(k) \quad (3-21)$$

其中 $K(N_k) = R(N_k)^{-1} F^T(N_k) Q(N_k) E(N_k) - F^T(N_k) Q(N_k) F(N_k)$ 。

根据上面的分析可以把本小节的内容以算法形式写成如下形式，算法 2：

- 1) 传感器采样系统输出并传至控制器；
- 2) 基于分配的计算资源 $R(k)$ ，依照式 (3-18) 选择出前向控制预测序列的步长 N_k ，并通过 (3-21) 计算出前向控制预测序列并将其传至执行器；
- 3) 控制选择器更新其寄存器并依照式 (3-9) 选择合适的控制量并令执行器应用至系统。

在本节中的改进模型预测控制算法在两个方面不同于传统的模型预测控制。首先，算法 2 中的预测时域是时变的并取决于可用的计算资源 $R(k)$ ，其次，传统模型预测控制只使用计算相互来的控制序列中的一个，而算法 2 则利用了所有可能的预测出来的控制量。

3.5 仿真实验验证

在这个小节中使用了常用的 Matlab 仿真软件基于一个倒立摆的模型对所提出的全资源预测控制算法进行验证。该系统可以建模成 (3-1) 中的一个线性系统，其系统矩阵 A 和 B 分别为：

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -m^*g/M & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & (M+m)^*g/Ml & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1 \\ 1/M \\ 0 \\ -1/Ml \end{pmatrix} \quad (3-22)$$

其中状态变量 $x(k) = [x_1(k) x_2(k) x_3(k) x_4(k)]$ 且 $x_1(k)$ 为小车的位移， $x_2(k)$ 为小车的速度， $x_3(k)$ 为倒立摆与垂直方向的夹角或偏离角度， $x_4(k)$ 为刚性摆杆的角速度。

为了验证提出的控制策略的有效性，这里考虑了两种情况的比较，两种情况都同样使用同样的模型预测控制器，不同的地方在于其中一种使用本章节提出的全资源预测控制策略，另一种则不使用。更详细地说，其中一种利用存储在向前控制预测序列中的预测控制量，而后者在可用计算资源无法进行控制计算的时候只能使用零控制，而在其他时刻则如传统模型预测控制一样使用预测序列的第一个控制量。

首先，将可用的计算资源用整数表示并设处理器的最大计算资源 $R^* = 100$ 。设控制系统进行每步控制预测所需要s的最小计算资源如下：

$$[R_1^* R_2^* R_3^* R_4^* R_5^* R_6^* R_7^*] = [25 \ 31 \ 40 \ 51 \ 68 \ 84 \ 105] \quad (3-23)$$

按照本章节中的定义， $R_u^* = R_1^*$ ，并注意到 $R_6^* < R^* < R_7^*$ ，即处理器的最大计算资源可进行 6 步控制序列的计算。式子 (3-18) 中的函数 $f(\bullet)$ 可以按照上面的式子进行定义。

在这里假设控制系统的可用的计算资源 $R(k)$ 是一个马氏过程，并且具有较高的概率小于 R_u^* 或者接近 R^* 。在实际中，已分配的计算资源在短时间内的骤变较为少见，因此在下一时刻控制系统的可用计算资源有较高概率仍保持在原有或相近水平。在 $R(k)$ 的转移概率矩阵中也将此特性考虑在内。

在上述系统的设置下，控制系统可用计算资源 $R(k)$ 的典型时间演化如图 3-3 所示，由式子 (3-21) 可知，前向控制预测序列的长度 N_k 可由计算资源 $R(k)$ 来确定，其随离散时间步长的变化如图 3-4 所示。图 3-5 是 χ_k^* 随离散时间 k 的演化，从图可知系统的可用计算资源经常会太少而不足以计算任何控制量，因而传统的控制方法无法在这种情况下工作。图 3-6 是系统的状态响应对比图，从图中可以看出在不增加任何计算资源消耗的前提下，使用了本章控制策略的系统状态响应曲线能更快的到达稳定，且波动也比较少。

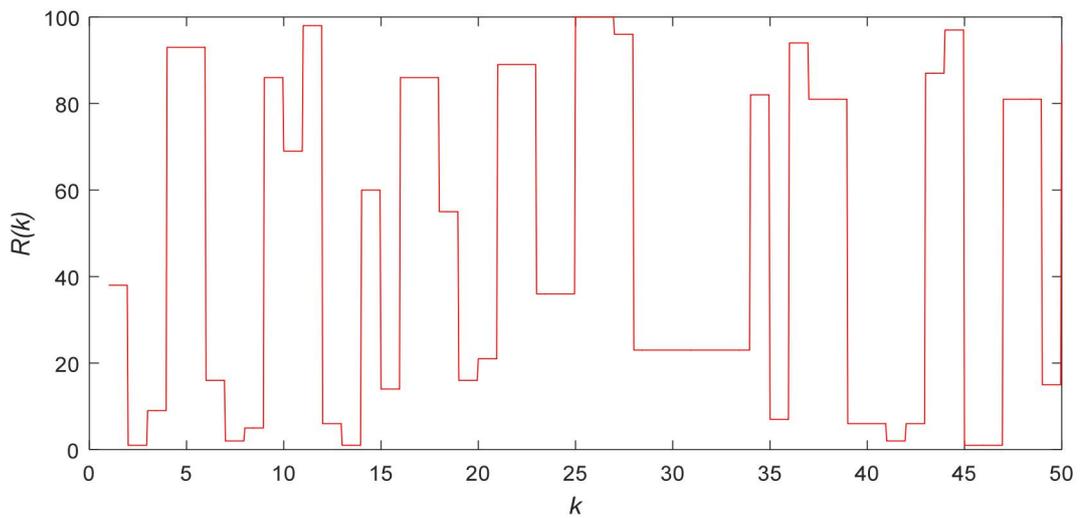
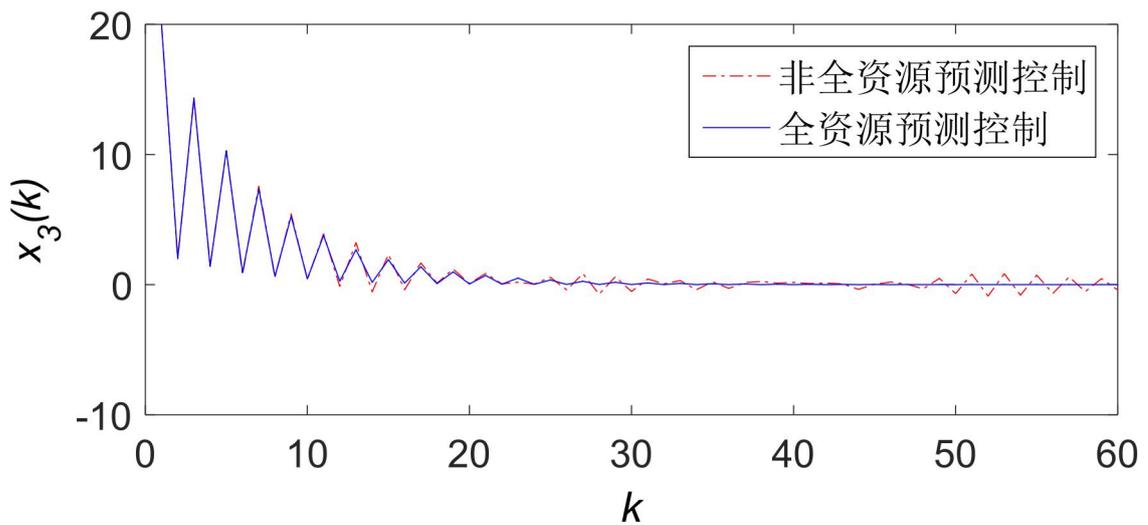
图 3-3 $R(k)$ 随时间 k 的典型演化曲线

图 3-6 系统状态响应对比图

3.6 本章小结

在本章的内容中，针对有时变有限且不可知计算资源的控制系统提出了一种充分利用可用计算资源的预测控制策略和相应的控制器设计方法。该策略的思想时控制系统可用计算资源充足时计算多步前向预测控制量以供控制系统计算资源不足时选择，在本章的最后一个小节通过使用基于 Matlab 的仿真验证了这种策略的有效性。

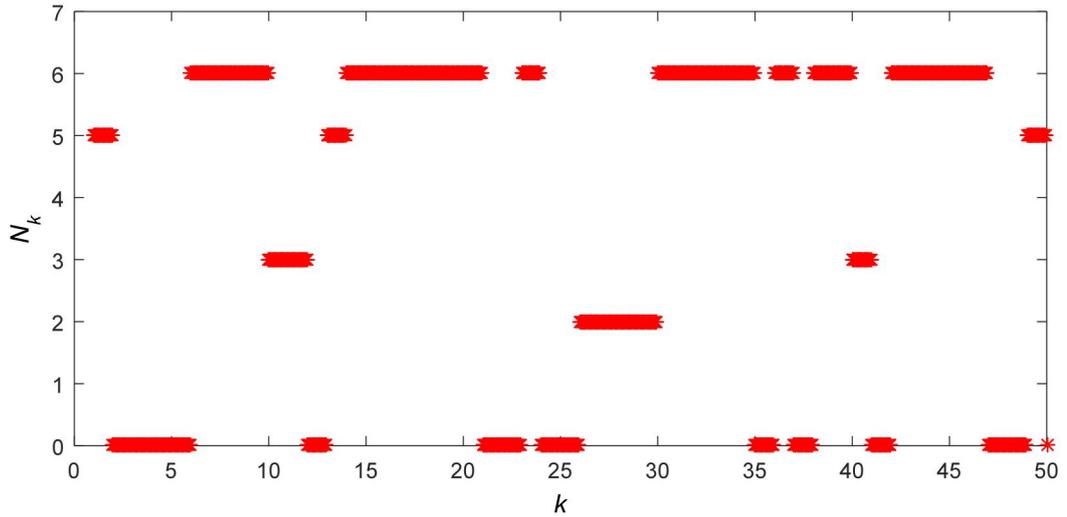


图 3-4 N_k 随时间 k 的典型演化曲线

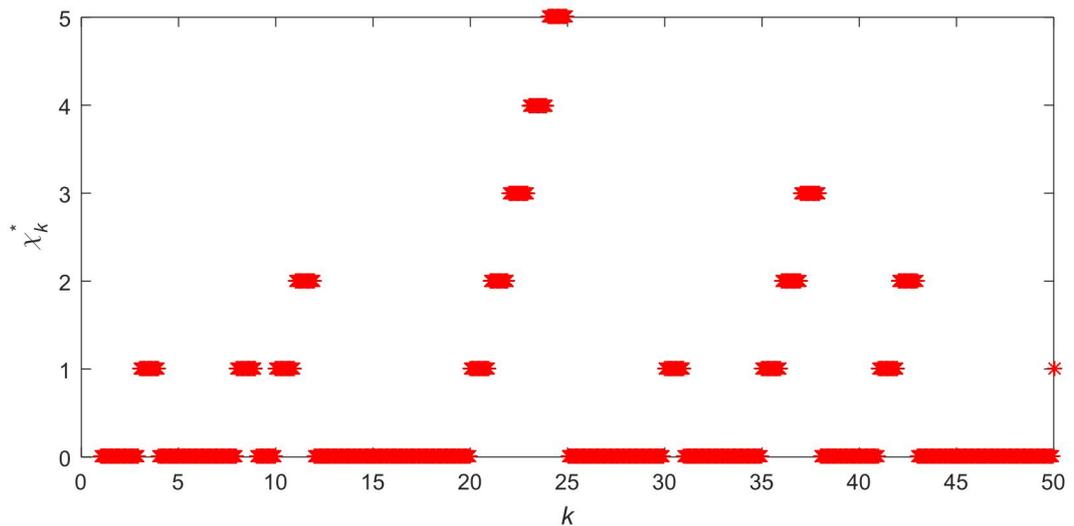


图 3-5 χ_k^* 随时间 k 的典型演化曲线

第4章 总结与展望

4.1 总结

本文研究了通信资源和计算资源受限的控制系统的分析和设计问题。在通信受限问题中，特别研究了由多包传输引起的通讯约束问题；在计算资源受限的情况中，本文特别考虑了一类控制器和其他系统或应用共用一个计算资源池的情况。本文的主要研究成果可总结如下：

1) 针对网络化控制系统中由多包传输带来的通讯资源受限的问题，考虑了控制系统中多个传感器由于地理上的隔离不得不使用打包成多个数据包的形式进行系统状态的传输这种问题，在这类问题中，传感器到控制器通道中多个数据包的传输自然会引入多个传输时延，多个传输时延的存在进而会影响控制系统的性能。在本文中（第二章），使用一种主动补偿的方法对系统的时延进行补偿，在很大程度上降低了由上述通讯约束对控制系统的影响。

2) 在另外一方面，考虑了实际应用中广泛存在的计算资源受限的一类问题，特别针对嵌入式或者移动式的系统，在这类系统中，控制系统和其他子系统共用一个计算资源池的问题，在这种情况下，整体系统分配给控制系统的计算资源可能是有限的。本文提出了一种全资源预测控制策略处理这类问题，这种方案在尽量使用总系统已经分配给控制系统的计算资源而不占用更多计算资源的情况下，提高了控制系统的表现性能。

4.2 展望

随着物联网浪潮的到来和云计算技术的日趋成熟，海量的物理对象之间的管理和控制都需要使用不同形式的控制技术，通讯问题和计算问题一直是控制领域中长期存在的一些问题。但从目前来说，这里面还有很多问题需要解决。本文针对通讯资源受限和计算资源受限的控制系统进行了研究，并取得了一定的成果，但是总的来看还有许多方面需要进一步深入研究：

1) 在通讯资源受限的情况下，只考虑了网络化控制系统的时延问题，特别是传感器到控制器的时延问题，对于由通讯引入的其他问题比如丢包等问题，本文并未做具体的深入

研究；

2) 在计算资源受限这类问题中，本文只结合一种情况给出了系统稳定性分析的证明过程，因此如何对闭环系统稳定性进行证明也是后续的研究工作之一；

3) 从研究方法上来说，本文的预测控制思想是需要基于数学模型的，但就目前的情况来说，在实际应用中，对于一个复杂的控制系统，如何精确的给出一个数学模型仍然是一个极具挑战的问题；

4) 从理论和实际结合的角度来讲，理论研究的最终目的是为了更好的服务于实践，本文只给出了一般的通用的方法，并没有将本文的方法深入地结合到各种类型的应用场景中加以分析。

参 考 文 献

- [1] Nilsson J, Bernhardsson B, Wittenmark B. Stochastic analysis and control of real-time systems with random time delays[J]. Automatica, 1998, 34(1):57-64.
- [2] Kranz M, Holleis P, Schmidt A. Embedded interaction: Interacting with the internet of things[J]. IEEE Internet Computing, 2010, 14(2): 46-53.
- [3] 陈幼平, 陈冰, 谢经明. 网络化控制系统的科学问题与应用展望[J]. 控制与决策, 2004, 19(9): 961-966.
- [4] 何飞跃. 电力系统网络化控制中的关键问题[J]. 电网技术, 2009 (1): 80-86.
- [5] Han D M, Lim J H. Smart home energy management system using IEEE 802.15.4 and zigbee[J]. IEEE Transactions on Consumer Electronics, 2010, 56(3):1403-1410.
- [6] Wong W S, Brockett R W. Systems with finite communication bandwidth constraints. II. stabilization with limited information feedback[J]. IEEE Transactions on Automatic Control, 1999, 44(5):1049-1053.
- [7] 夏元清, 李慧芳, 张金会. 控制与计算理论的交互: 云控制[J]. 指挥与控制学报, 2017, 3(2): 99-118.
- [8] Yang T C. Networked control system: A brief survey[J]. IEE Proceedings-Control Theory and Applications, 2006, 153(4): 403-412.
- [9] 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.
- [10] 游科友, 谢立华. 网络控制系统的最新研究综述[J]. 自动化学报, 2013, 39(2): 101-118.
- [11] Abidi K, Xu J X. Iterative learning control for sampled-data systems: from theory to practice[J]. IEEE Transactions on Industrial Electronics, 2011, 58(7): 3002-3015.
- [12] Zhang L, Gao H, Kaynak O. Network-induced constraints in networked control systems-a survey[J]. IEEE Transactions on Industrial Informatics, 2013, 9(1): 403-416.
- [13] Pratl G, Dietrich D, Hancke G P, et al. A new model for autonomous networked control systems[J]. IEEE Transactions on Industrial Informatics, 2007, 3(1):21-32.
- [14] Zhang D, Yu L, Zhang W A. Delay-dependent fault detection for switched linear systems with time-varying delays-the average dwell time approach[J]. Signal Processing, 2011, 91(4):832-840.
- [15] Chen B, Yu, Li, Zhang, Wen An. Robust kalman filtering for uncertain discrete time-delay systems with missing measurement[J]. Acta Automatica Sinica, 2011, 37(1):123-128.
- [16] Zhang W, Branicky M S, Phillips S M. Stability of networked control systems[J]. IEEE Control Systems Magazine, 2001, 21(1):84-99.
- [17] 查利娟. 基于事件触发机制网络控制系统的若干问题研究[D]. 上海: 东华大学, 2017.
- [18] 杨飞生, 汪璟, 潘泉. 基于事件触发机制的网络控制研究综述[J]. 控制与决策, 2018(6):65-71.
- [19] Dong Y, Tian E, Han Q L. A delay system method for designing event-triggered controllers of networked control systems[J]. IEEE Transactions on Automatic Control, 2013, 58(2):475-481.
- [20] Fu M, Xie L. The sector bound approach to quantized feedback control[J]. IEEE Transactions on Automatic Control, 2005, 50(11):1698-1711.
- [21] 王硕, 禹梅, 谭文. 具有丢包补偿网络化控制系统的量化反馈稳定性分析[J]. 系统科学与数学,

- 2015, 35(3):287-297.
- [22] Nesic D, Liberzon D. A unified framework for design and analysis of networked and quantized control systems[J]. IEEE Transactions on Automatic Control, 2009, 54(4):732-747.
- [23] Cloosterman M B G, Hetel L, Vande Wouw N, et al. Controller synthesis for networked control systems[J]. Automatica, 2010, 46(10): 1584-1594.
- [24] Jiang X, Han Q L, Liu S, et al. A new H_{∞} stabilization criterion for networked control systems[J]. IEEE Transactions on Automatic Control, 2008, 53(4): 1025-1032.
- [25] Hristu V D. Short-period communication and the role of zero-order holding in networked control systems[J]. IEEE Transactions on Automatic Control, 2008, 53(5): 1285-1290.
- [26] Donkers M C F, Heemels W P M H, Bernardini D, et al. Stability analysis of stochastic networked control systems, American Control Conference[C]. Baltimore: 2010:917-925.
- [27] Gao H, Meng X, Chen T. Stabilization of networked control systems with a new delay characterization[J]. IEEE Transactions on Automatic Control, 2008, 53(9): 2142-2148.
- [28] Tipsuwan Y, Chow M Y. Control methodologies in networked control systems[J]. Control Engineering Practice, 2003, 11(10): 1099-1111.
- [29] Richard J P. Time-delay systems: An overview of some recent advances and open problems[J]. Automatica, 2003, 39(10): 1667-1694.
- [30] Yue D, Han Q L. Delayed feedback control of uncertain systems with time-varying input delay[J]. Automatica, 2005, 41(2): 233-240.
- [31] Mazenc F, Niculescu S I. Lyapunov stability analysis for nonlinear delay systems[J]. Systems & Control Letters, 2001, 42(4): 245-251.
- [32] Guan Z H, Chen W H, Xu J. Delay-dependent stability and stabilizability of uncertain jump bilinear stochastic systems with mode-dependent time-delays[J]. International Journal of Systems Science, 2005, 36(5): 275-285.
- [33] Kim D S, Lee Y S, Kwon W H, et al. Maximum allowable delay bounds of networked control systems[J]. Control Engineering Practice, 2003, 11(11): 1301-1313.
- [34] Halevi Y, Ray A. Integrated communication and control systems: Part I-Analysis[J]. Journal of Dynamic Systems, Measurement, and Control, 1988, 110(4): 367-373.
- [35] Hu S S, Zhong Q X. Stochastic optimal control and analysis of stability of networked control systems with long delay[J]. Automatica, 2003, 39(11): 1877-1884.
- [36] Nilsson J, Bernhardsson B. Analysis of real-time control systems with time delays[C]. Japan: Proceedings of 35th IEEE Conference on Decision and Control, 1996, 3: 3173-3178.
- [37] Schenato L, Sinopoli B, Franceschetti M, et al. Foundations of control and estimation over lossy networks[J]. Proceedings of the IEEE, 2007, 95(1): 163-187.
- [38] Imer O C, Yüksel S, Başar T. Optimal control of LTI systems over unreliable communication links[J]. Automatica, 2006, 42(9): 1429-1439.
- [39] Seiler P, Sengupta R. An H_{∞} approach to networked control[J]. IEEE Transactions on Automatic Control, 2005, 50(3): 356-364.
- [40] Sahebsara M. Optimal filtering in multiple channel networked control systems with multiple packet dropout[C]. Cancun: IEEE Conference on Decision and Control, 2008, 3366-3371.
- [41] Liu G P, Mu J X, Rees D, et al. Design and stability analysis of networked control systems with random communication time delay using the modified MPC[J]. International Journal of Control, 2006, 79(4):

- 288-297.
- [42] Zhao, Y B, Liu, et al. Integrated predictive control and scheduling co-design for networked control system[J]. *IET Control Theory & Applications*, 2008, 2(1):7-15.
- [43] Zhao Y B, Liu G P, Rees D. Design of a packet-based control framework for networked control systems[J]. *IEEE Transactions on Control Systems Technology*, 2009, 17(4): 859-865.
- [44] Zhao Y B, Liu G P, Rees D. Packet-based deadband control for internet-based networked control systems[J]. *IEEE Transactions on Control Systems Technology*, 2010, 18(5): 1057-1067.
- [45] Walsh G C, Beldiman O, Bushnell L. Asymptotic behavior of networked control systems[C]. San Diego: IEEE International Conference on Control Applications, 1999, 2: 1448-1453.
- [46] Nešić D, Teel A R. Input-to-state stability of networked control systems[J]. *Automatica*, 2004, 40(12): 2121-2128.
- [47] Branicky M S, Phillips S M, Zhang W. Scheduling and feedback co-design for networked control systems[C]. Nevada: Proceedings of IEEE Conference on Decision and Control, 2002, 2: 1211-1217.
- [48] Adimoolam A, Dang T, Donzé A, et al. Classification and coverage-based falsification for embedded control systems[C]. Heidelberg: International Conference on Computer Aided Verification. Springer, 2017,483-503.
- [49] Martí P, Velasco M, Fuertes J M, et al. Design of an embedded control system laboratory experiment[J]. *IEEE Transactions on Industrial Electronics*, 2010, 57(10): 3297-3307.
- [50] Bhattacharya R, Balas G J. Anytime control algorithm: Model reduction approach[J]. *Journal of Guidance, Control, and Dynamics*, 2004, 27(5): 767-776.
- [51] Al-Areqi S, Görges D, Liu S. Event-based networked control and scheduling co-design with guaranteed performance[J]. *Automatica*, 2015, 57: 128-134.
- [52] Sardellitti S, Scutari G, Barbarossa S. Joint optimization of radio and computational resources for multicell mobile cloud computing[C],Toronto: IEEE 15th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC),2014: 354-358.
- [53] 王天宝, 吴成东, 王玉龙. 基于多包传输的网络控制系统控制器设计[J]. *东北大学学报(自然科学版)*, 2013, 34(2):157-161.
- [54] 党向东. 时变时延多包传输网络控制系统的稳定性[J]. *中南大学学报: 自然科学版*, 2009, 13(S1):158-163.
- [55] Guan Z H, Chen C Y, Feng G, et al. Optimal tracking performance limitation of networked control systems with limited bandwidth and additive colored white Gaussian noise[J]. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2013, 60(1): 189-198.
- [56] Cervin A, Henriksson D, Lincoln B, et al. How does control timing affect performance? Analysis and simulation of timing using Jitterbug and TrueTime[J]. *IEEE Transactions on Control Systems*, 2003, 23(3): 16-30.
- [57] Chen C T. *Linear system theory and design*[M]. Holt: Rinehart and Winston, 1984.

致 谢

转眼间研究生生涯已经进入了尾声，在这两年的时间里我认识了许多优秀的人，在他们的身上许多值得我学习的地方，同时也遇到很多帮助过我的人，在此向他们表示由衷的感谢。首先我要感谢的是浙江工业大学的各位老师，是你们春风化雨般的教诲让我一步一步走完了这两年多的历程。特别地，我要感谢我的导师赵云波教授。赵老师渊博的学识和开阔的科研视野带领我推开了新世界大门；每当和赵老师交谈，他温文尔雅的谈吐都使我如沐春风；从赵老师身上我寻得了看待世界的另一种方式。赵老师严谨的工作态度和科学高效的工作方式也已经成了学生们学习的榜样，每周一次的组会除了在纲领上和方法上指导着我们不断前行之外，还为刚刚组建的实验室营造了良好的学术氛围。

除此之外我还要感谢实验室的各位同届的小伙伴，作为赵老师的第一届学生，尽管以后我们会各自追逐自己的梦想，走不同的路，但我们之间的情谊我会一直铭记于心。

当然，我还要深深地感谢自己的家人，感谢你们对我无条件的支持以及无微不至的关爱；感谢女友何明明同学，感谢你八年来的一路相伴，陪我度过我生命中最好的年华；是你们让我有了不断前行的动力。

最后再一次向所有曾经帮助、关心、支持和鼓励过我的老师、家人、朋友和同学致以我最诚挚的感谢！

攻读学位期间参加的科研项目和成果

参加的科研项目

[1] 国家自然科学基金面上项目：基于资源调度和预测控制的无线网络化控制系统的联合设计（61673350）

录用和发表的论文

[1] 马翠芹, 姚俊毅, 韩康, 赵云波. 计算受限控制系统的一种全资源预测控制方法[J]. 自动化学报 (3/4, 已录用)

专利

[1] 赵云波, 韩康, 黄涛. 一种针对多传感器网络化控制系统的主动时延补偿方法. (除导师外第一作者, 201811196434.8, 受理)

[2] 赵云波, 韩康, 黄涛. 一种针对多传感网络化控制系统的数据包丢失的补偿方法. (除导师外第一作者, 201811196447.5, 受理)