



浙江工业大学

本科毕业设计论文

题目：基于微小型机器人集群的
协同控制研究

作者姓名 张求熠

指导教师 赵云波

专业班级 自动化 1304 班

学 院 信息工程学院

提交日期 2017 年 6 月 1 日

浙江工业大学本科毕业设计论文

基于微小型机器人集群的协同控制研究

作者姓名：张求熠

指导教师：赵云波

浙江工业大学信息工程学院

2017年6月

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

**Research on Collaborative Control Based on
Micro - Robot Cluster**

Student: Zhang Qiuyi

Advisor: Zhao Yunbo

**College of Information Engineering
Zhejiang University of Technology
June 2017**

浙江工业大学

本科生毕业设计(论文、创作)诚信承诺书

本人慎重承诺和声明：

1. 本人在毕业设计（论文、创作）撰写过程中，严格遵守学校有关规定，恪守学术规范，所呈交的毕业设计（论文、创作）是在指导教师指导下独立完成的；

2. 毕业设计（论文、创作）中无抄袭、剽窃或不正当引用他人学术观点、思想和学术成果，无虚构、篡改试验结果、统计资料、伪造数据和运算程序等情况；

3. 若有违反学术纪律的行为，本人愿意承担一切责任，并接受学校按有关规定给予的处理。

学生（签名）：

年 月 日

本科毕业设计任务书

(用盖过红章的任务书替换)

基于微小型机器人集群的协同控制研究

摘 要

协同控制指的是在没有集中控制和全局通信的情况下,主要靠个体之间局部的通信来实现各种集体行为。对协同控制的研究包括了对编队控制、群集运动、一致性控制等基本问题的讨论。

本文以微小型机器人 Kilobot 为实验平台,针对编队控制和一致性控制的一些典型算法进行研究,并将算法在该平台上实现。

论文的主要工作如下:

- 1.介绍模拟器 V-rep 和 Kilobot 实验平台,以及 Kilobot 所提供的一些主要的 API。
- 2.针对编队控制,选取了一种应用广泛的领导者-跟随法对该问题进行研究。结合 Kilobot 实物建立运动学模型。对于领导者,自定义合适的参数进行控制。对于跟随者,选取了合适的控制律使其与领导者的相对位置和相对角度收敛到期望值,从而实现编队控制。将该方法在模拟器上进行仿真验证其可行性。
- 3.针对一致性控制,选取了一种存在通信时延的一致性协议对该问题进行研究。基于图论法,选取适当的参数和模型,对机器人的某一状态进行控制,使其达到该状态一致。将方法在模拟器上仿真,并在 Kilobot 实物上实验,验证方法的可行性。

关键词: Kilobot, 编队控制, 一致性控制, V-rep 模拟器

Research on Collaborative Control Based on Micro – Robot Cluster

ABSTRACT

Cooperative control refers to the absence of centralized control and global communication in the case, only through the local communication between individuals to achieve a variety of collective behavior. The study of collaborative control includes some discussion of basic problems such as formation control, cluster motion, and consistency control.

In this paper, I study the typical algorithms of formation control and consistency control with the micro-robot-Kilobot as the experimental platform, and implement the algorithm on the platform. The main works of the paper are as follows:

The main work of the paper is as follows:

1. Introduce the simulator V-rep and Kilobot experimental platforms, as well as some of the major APIs provided by Kilobot.

2. For the formation control, select a widely used leader-follow the law to study the problem. Combine Kilobot physical establishment kinematics model. For the leader, customize the appropriate parameters to control. For the followers, the appropriate control law is chosen so that its relative position and relative angle to the leader converge to the desired value, thus achieving formation control. The method is simulated on the simulator to verify its feasibility.

3. Aiming at the consistency control, a coherence protocol with communication delay is selected to study the problem. Based on the graph theory, select the appropriate parameters and models to control a state of the robot to achieve the same state. The method is simulated on the simulator and tested on the Kilobot object to verify the feasibility of the method.

Key Words: kilobot, formation control, consistency control, v-rep simulator

目 录

摘要.....	I
ABSTRACT.....	II
第 1 章 绪论.....	1
1.1 课题研究背景及意义.....	1
1.2 协同控制的基本问题.....	1
1.2.1 一致性控制.....	2
1.2.2 编队控制.....	2
1.3 本文研究内容.....	3
1.4 本章小结.....	3
第 2 章 V-rep 模拟器及 Kilobot 实验平台.....	4
2.1 Kilobot 实验平台.....	4
2.1.1 Kilobot 的特点及功能.....	4
2.1.2 Kilobot 提供的 API.....	6
2.2 V-rep 模拟器.....	6
2.3 本章小结.....	7
第 3 章 领导-跟随者编队控制方法.....	8
3.1 编队控制综述.....	8
3.1.1 基于行为法的编队控制.....	8
3.1.2 基于图论法的编队控制.....	8
3.1.3 基于领导-跟随者法的编队控制.....	8
3.2 运动学模型简介.....	9
3.2.1 领导者运动模型.....	9
3.2.2 跟随者运动模型.....	10
3.3 跟随者编队控制策略.....	11
3.4 算法实现.....	12
3.5 仿真结果展示.....	13
3.6 本章小结.....	15
第 4 章 不同延时下离散时间多智能体系统的一致性.....	17
4.1 选择一致性协议.....	17
4.2 图论法.....	17

4.3	一致性模型.....	18
4.4	一致性分析.....	18
4.5	算法实现.....	21
4.6	仿真结果展示.....	21
4.7	Kilobot 实物实验结果展示	24
4.8	本章小结.....	25
第 5 章	总结与展望	26
5.1	总结.....	26
5.2	展望.....	26
参考文献	28	
附录 1: 领导-跟随者法程序	30	
附录 2: 具有通信延时的一致性协议程序	35	
致谢	40	

第 1 章 绪 论

本章主要是对描述该课题的背景进行了简要的描述，交代了其研究的意义。并且针对该课题的几个主要问题，对他们展开了重点介绍。

1.1 课题研究背景及意义

从 20 世纪 90 年代开始，随着计算机、网络、通信等技术的发展，多智能体系统^[1]的研究就已经成为控制领域的一个新方向。多智能体系统也可以被叫做是自主体系统、群体系统，目前对于该系统还没有一个严格的统一的定义，但最具有代表性的一种定义称：多智能体系统是由一群具备传感、计算、执行和通信能力的智能体通过通信等方式关联成为的一个网络系统。多智能体系统具有较好的鲁棒性，较好的容错性等等。它的主要作用就是代替人去处理一些复杂的、难度较大的任务，协同检验和优化数据等。

而到目前为止，对多智能体系统的研究已经成为研究人工智能的一个热点问题了，其广泛的应用前景已经引起了计算机科学、生物学、系统与控制科学等领域的关注和兴趣。因为，随着人工智能在生活、工作等各个领域的普及，许多问题仅仅依靠单个机器人去完成就变得十分困难，所以研究多机器人来解决这些问题就成了必然。

在研究多智能体系统的过程中，对其协同控制的研究非常重要的问题之一。因为，通过机器人个体之间的相互作用来实现一些群体活动，解决一些超出单个机器人能力范围的问题是研究多智能体系统的一个核心思想。目前，已有国际电气与电子工程师学会 IEEE，国际自动控制联合会 IFAC 等等一些国际组织对协同控制问题进行着研究；同时国内外的许多大学和研究员们也成立各自的研究小组，并且已经将协同控制应用于交通、侦查、航空航天等各个领域。

因此本文也将针对多智能体的协同控制的方法进行研究和实验。

1.2 协同控制的基本问题

所谓多智能体的协同控制指的是：在没有集中式控制和全局通信的情况下，仅靠个体之间局部的通信协调来实现各种集体行为。对其研究中的一些基本问题主要包括：编队控制^[2]、群集运动、一致性控制^[3]等。下面，主要对编队控制和一致性控制进行介绍。

1.2.1 一致性控制

一致性问题是在群体机器人协同控制研究中的一个重要问题，它是各个智能体之间协同合作的基础问题，对理论研究和实践都有重要的意义。一致性问题是指：随着时间的变化，群体机器人通过局部通信协同，最终使得整个集体的状态趋于一致。一致性问题关键在于设计适当的一致性协议或算法使整个网络中的个体的状态趋向于同一个值。

一致性问题的研究通常以图论法和矩阵论法为基础，用它们来表示拓扑结构中各个节点与节点，节点与边之间的关系。其中设计一致性协议是对其研究的一个重点，一致性协议要求多智能体系统同时满足实时性、鲁棒性等特性。现在已有的一些一致性协议有：基于连续时间线性一致性协议；基于离散时间线性一致性协议；基于切换拓扑结构的一致性协议；带时滞一致性协议；一致性滤波等。

含有领导者的一致性跟踪问题是一致性控制中一个重要的问题。可分为真实的领导者与跟随者的一致性跟踪问题和虚拟的领导者与跟随者的一致性跟踪问题两类。前者表示领导者是真实的目标跟踪对象，其主要出现在无线传感器网络和分布式跟踪问题中，由于在实际中，无法获取真实的领导者的一些状态变量，所以 Hong 等在文献^[4]中提出了基于观测器的一致性跟踪算法。而后者主要出现在编队、群集算法等控制问题中，又被称为模型参考一致性控制，因为领导者的动态或者模型是预先指定的，所以是虚拟的。

为了更好的实现一致性协议对机器人系统的控制，未来的研究将集中于以下几个方面：在切换拓扑环境下传感器网络分布式估计问题，在切换拓扑环境下一致性协议的设计和应用等。

1.2.2 编队控制

编队控制是指群体机器人在向着特定目标运动时，群体中的个体需要通过保持一定的队形并且适应环境约束的控制问题。编队控制是一致性控制的一个扩展，各个智能体之间不但要保持固定的队形而且还要避免个体之间发生碰撞。近十几年来，对编队控制的研究内容可作如下的总结：队形生成、队形保持、队形切换、编队避障和自适应。其中前 4 种是研究的重点，而对自适应的涉及相对较少。群体机器人编队控制的方法主要有：领导-跟随者法、基于行为法、虚拟结构法和基于图论法。目前，这几种方法已经逐步混合在一起了，尤其是前三种方法被很好地融合在一起，成为编队控制的主流方法。如何选取最有效的队形来执行协调任务是编队控制中的一个基本问题，常见的队形有线

型、柱形、菱形、V型四种。

目前，编队控制已经在工业、军事等众多领域中有着广泛的应用。例如利用群体机器人以特定队形搬运货物；利用群体机器人编队执行侦察和搜索任务；使用群体机器人进行舞蹈表演和足球竞赛等。除了这些，它还被应用于国防，美国国防部已将飞行器队形控制技术和卫星编队控制技术确定为 21 世纪的关键技术。总之，编队控制对各种群体机器人系统完成指定的任务有着有力的帮助。

1.3 本文研究内容

本课题研究的主要内容是对协同控制算法的研究和实验。针对编队控制和一致性控制这两个基本问题，分别选取了一种典型的控制方法进行讨论，并基于 Kilobot^[5]实验平台进行实验验证。本文分成五章，每章的具体内容如下：

第一章是绪论部分，它概述了课题的研究背景和意义，分析了工多智能体协同控制的研究背景及其基本研究问题，并给出了本文的主要研究内容。

第二章对 V-rep 模拟器和 Kilobot 实验平台做基本介绍。

第三章介绍了领导-跟随者编队控制算法，并展示在模拟器上的仿真效果。

第四章基于图论法介绍了一种具有不同延时的离散时间多智能体系统的一致性协议，在模拟器上实现仿真，并且展示在 Kilobot 实验平台上验证的成果。

第五章对所做工作进行进一步的总结，并对未来发展做出一定的期望。

1.4 本章小结

本章简要介绍了课题的研究背景及意义，简单介绍了协同控制的两个基本问题，以及接下来将要讨论的两种控制方法。最后，介绍了本文的主要研究工作和论文的组织结构。

第 2 章 V-rep 模拟器及 Kilobot 实验平台

本章对实验中所采用实验平台 Kilobot 做了详细的描述，介绍了它的一些主要特点和功能，以及它的工作原理和一些必要的 API。之后，也对 V-rep 模拟器的功能做了简要的介绍。

2.1 Kilobot 实验平台

Kilobot 是哈佛大学生物工程研究所的研究者在 2014 年公布的一项成果，是一种低成本且完全可扩展的机器人。下面介绍 Kilobot 的主要功能特点及其 API。

2.1.1 Kilobot 的特点及功能

Kilobot 虽然成本低，但依然拥有以下基本功能：前进，旋转，邻居间的通信，测量和邻居之间的距离。此外，为了提高 Kilobot 集体行动的能力，以及得到一个更通用的机器人研究平台，它还具备环境光测量的能力，能显示一些内部状态来协助调试，并允许可扩展操作。接下来，将结合 Kilobot 的硬件对其主要功能做简单的介绍。图 2-1 就是 Kilobot 机器人的硬件描述。

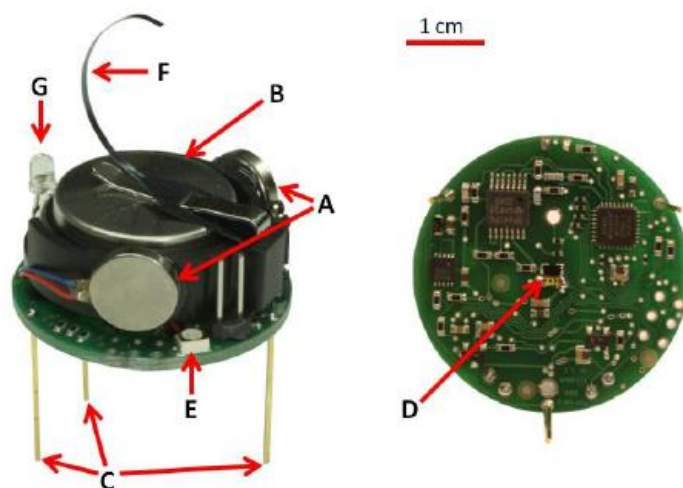


图2-1 Kilobot的等距（左）和下（右）视图。一些主要特点是：（A）振动电机，（B）锂离子电池，（C）刚性支撑腿，（D）红外发射器/接收器，（E）三原色（RGB）LED，（F）充电片，和（G）环境光传感器。

1. 运动

能在环境中移动是每个机器人最重要的一个功能之一。对于集群机器人，最常见的运动策略是使用两轮差速驱动器，其中每个轮子是由电动齿轮马达供能。这种传统轮式的运动虽然非常有效，但它相对较贵。为了降低成本，Kilobot使用两个密封的硬币型振动电机用于运动（见图1中的A）。当电动机中的一个被启动时，由振动电动机产生的向心力将在Kilobot安装电动机的位置上转换为向前力。由于两个振动电机的偏心安装，单独一个电动机的振动可以使Kilobot绕其垂直轴的旋转，而另一个电动机的振动将引起相反方向的旋转。通过以差动驱动方式独立地控制两个电动机的振动的大小，机器人可以在从顺时针旋转，直线前进到逆时针旋转的连续范围内移动。

但使用这种基于滑动杆的运动的一个主要缺点是它没有实际形式的里程计。这就无法做到精确地长距离或长时间移动。同时还有另一个限制，就是它不能在较粗糙的表面上移动，需要在光滑表面上工作。本次实验在进行Kilobot实物调试时确实证实了以上两个问题。

2. 通信和感知

每个Kilobot都有一个红外LED发射器和红外光电二极管接收器，它们位于PCB的中心（见图1中的D），及机器人底部中心。红外发射器和接收器都具有等方向性的发射或接收模式，使得机器人能从各个方向均等地接收消息。当发射器发出信号时，半径10cm范围内的机器人都可以通过桌面反射的红外光来接收信息。而在实际实验时发现，Kilobot的通信距离大约为7cm。

因为所有机器人都使用相同的红外通道进行通信，所以存在两个或多个机器人同时发送的可能性。为了减轻这个问题造成的影响，Kilobot使用了具有冲突避免的标准载波来侦听多路访问（CSMA/CA）。

3. 控制器

Kilobot机器人的控制器具有两个功能。第一，它与所有底层电子设备如电机，电源电路和RGB LED相互作用。第二，它能运行用户编写的程序。它所使用的控制器是一个Atmega328微处理器，频率为8 Mhz，有32K的内存。

在Kilobot中使用该控制器的一些关键特征有：用于控制振动电机速度的两个脉冲宽度调制通道（PWM），用于测量入射红外光强度的10位模数转换器，用于更新机器人程序的可编程存储器和低功耗睡眠模式。机器人的程序是用C语言编写的，这使得

研究人员能够快速轻松地开发机器人的行为。

4. 供电系统

为了给机器人供电，每个 Kilobot 都有一个 3.4 V ， 160 mAh 的锂离子电池（见图 1 中的 B）。该电池可以根据机器人的运动情况向机器人供电 3-24 小时。与这个电池连接的有三个电压调节器和一个电池充电器。其中两个电压调节器向电机和通信系统提供电力。这两个调节器都可以通过微控制器打开和关闭，使电机和通信系统关闭，以节省功耗。第三个电压调节器持续向微控制器供电，在低功耗状态下，仅消耗 30 μ A。当电池充电器接收 6 V 直流电时，充电器将开始对板载电池充电；当电池充满电时，充电将停止。

2.1.2 Kilobot 提供的 API

Kilobot 的开发环境是由 AVR-C 语言搭建的，即它使用了 WinAVR 在 eclipse 上进行开发，是将编程语言转换成机器语言的一个过程。接下来列举几个 Kilobot 提供的核心 API。

`kilo_init`: 用来初始化 kilobots 的硬件和功能

`kilo_start`: 表示它使用的是一个基本的事件循环编程范式，这依赖于 `setup` 和 `loop` 的回调。

`get_ambientlight`, `get_voltage`, `get_temperature`: 是各种传感器的功能，能获取环境光、机器人的电压值和温度。

`set_motors`, `set_color`: 用来控制机器人的速度和 RGB LED。

用户可以注册回调以与消息传递子系统进行交互。其中包括有消息接收事件 `kilo_message_rx`，消息传输 `kilo_message_tx` 和成功发送通知 `kilo_message_tx_success` 的回调。

同时还定义了两个数据结构：`distance_messurement_t` 和 `message_t`。前者用来存放与接收者之间的距离，后者为一个长度为 12 字节的数组，用来存放需要传输的数据，其中数据类型为无符号整型。

2.2 V-rep 模拟器

V-rep 被称作是机器人仿真器里的“瑞士军刀”，它拥有的功能众多，而且它提供

的应用编程接口非常详尽。下面列举它一些主要的优点：

1. V-rep 是一个跨平台的软件，在 Windows、MacOS、Linux 上都是兼容的。他拥有 6 种编程方法，包括嵌入式脚本、插件、附加组件、ROS 节点、远程客户端应用编程接口、或自定义的解决方案；而且它有超过 400 种不同的应用编程接口函数，可以支持 7 种不同的语言进行编程，分别是 C/C++、Python、Java、Lua、MATLAB、Octave、和 Urbi。本次实验为了方便起见，选用了它自带的脚本语言，Lua 语言进行编程仿真。
2. V-rep 拥有 ODE、Bullet、Vortex 和 Newton ， 4 个物理引擎和对于任何机构的逆运动学和正运动学完整的运动学解算器。物理引擎是一种基于牛顿力学的数学算法，也就是说，通过这种算法能够模拟真实的物理世界，它是通过为刚性物体赋予真实的物理属性的方式来计算运动、旋转和碰撞反映，是包含了力的效应的动力学，而不是简单的运动学。这样就使得仿真具有更高的真实性，对实际情况具有更高的还原度。
3. V-rep 可以自定义用户界面，整合了图形编辑模式，可以制作自己想要的场景。他还包含众多公司提供的机器人模型用于实验，当然也包含了本次实验所用到的 Kilobot，自然地就选择了 V-rep 作为仿真软件。

当然 V-rep 还有许多其他优点，如拥有 100 项 ROS 服务、30 个发布类型、25 个 ROS 订户类型、可拓展；拥有 Mesh, octree, point cloud-网孔干扰检测；拥有完全可拓展的嵌入图像处理的视觉传感器和现实的接近传感器；拥有完全集成的第四类 Reflexxes 运动库+RRS-1 interface specifications；它还能记录数据并可视化，有多层、取消或重做、影像记录、油漆的仿真等功能。

2.3 本章小结

V-rep 模拟器拥有强大的功能，对真实环境有很好的还原度，是进行此类仿真实验的一个好的选择。Kilobot 机器人拥有一些基本的功能，可以完成许多的任务，而且其成本低、体积小、数量大，是进行实物实验的不错之选。

第 3 章 领导-跟随者编队控制方法

3.1 编队控制综述

本课题研究的机器人编队主要是利用分布式控制策略,通过各个机器人之间的通讯信息的互通有无,降低了单个机器人所需获取大量信息的要求。与此同时,为了解决编队控制中存在的某些不完美的问題,比如需要优化编队控制的稳定性与不确定性,在这里我们选取比较了几种不同的编队控制模型

3.1.1 基于行为法的编队控制

基于行为法的编队控制的核心思想是定义一系列较为基本的行为动作的控制器,例如定义某些躲避障碍等的基本动作。各个动作都有作为自身需要完成的基本目标,随机选取其中一种基本动作后,作为控制器的输出又可以作为其他控制器的输入。在选取动作,各个动作相互协调这一方面,往往影响编队控制器的作用效果。

基于行为法的控制器能直观给出对于各个机器人的行为动作,队列的反馈效果良好。但如果针对要求高准确性高精度的队列时,局限性比较大甚至出现通讯量过载的情况。

3.1.2 基于图论法的编队控制

如果将编队的整体结构映射成有向的加权图或是无向的加权图,将已经编好队的机器人视作图内的各个顶点,将编好队的机器人之间的联络关系映射成刚才图中的边缘,将编好队的机器人之中所要求的相对距离作为图中的权值。近年来,相关学者研究了大量关于图的编队理论,其中有图的稳定性分析,最小刚性的定义与判断准则。另外,图编队完以后,合并与撕裂也是处理优化编队图的重要手段。

按经验来说,编队结构一般都是使用图的形式,如果各个机器人之间的能够相互传递信息,那么可以用无向图解决编队方式,如果各个机器人的信息是单向流通获得信息的话,那么往往是采用有向图的形式来表示。

3.1.3 基于领导-跟随者法的编队控制

领导-跟随的编队控制法,其含义正如名字,核心思想是在多机器人编队中,将选

中的某一个机器人作为领导者，对整个队伍的运动起着引领的作用，除此之外的机器人则作为跟随者，跟着领导者动作。作为实现方式，一般是计算两者之间的相对距离和旋转角度来进行编队。此种方法的优越之处在于便于控制，实现方便，是一种极为经典的编队算法。

综上所述，考虑到实现的可行性与易操作性，本文采用的编队算法为领导-跟随编队控制^[6]。

3.2 运动学模型简介

在上述所选定的领导-跟随模型中，我们需要对领导者和跟随者这两类不同角色建立运动学模型。主要依据便是两者之间相对位置和相对角度来描述互相的运动位置状态。

3.2.1 领导者运动模型

我们将机器人的外形进行简化，如果假设在运动的某一时刻质心坐标点为 $N(x,y)$ ， t 时刻轮式机器人线速度方向 W 和 x 轴正方向的夹角为 θ 。将此时刻的位置信息由向量 $q = (x, y, \theta)^T$ 确定，运动信息由向量 $p = (v, w)^T$ 确定，其中 $v(t)$ 和 $\omega(t)$ 分别表示机器人在 t 时刻的线速度和角速度。

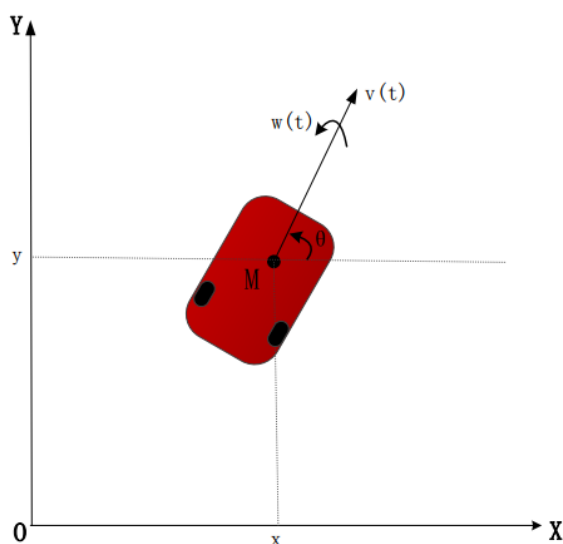


图 3-1 移动机器人运动学模型

由图易得，领导机器人的运动学方程可表示成：

$$\begin{cases} \dot{x} = v \cdot \cos(\theta) \\ \dot{y} = v \cdot \sin(\theta) \\ \dot{\theta} = w \end{cases} \quad (3.1)$$

若将式(3.1)用向量形式重新表示为:

$$\dot{q} = R(\theta) \cdot p \quad (3.2)$$

其中

$$R(\theta) = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \quad (3.3)$$

3.2.2 跟随者运动模型

在建立跟随者的运动模型上, 需要同时参考领导者的运动信息, 如图 3-2 所示。领导机器人与跟随机器人在 t 时刻的线速度分布为 $v_1(t)$ 和 $v_2(t)$, 其运动方向与 x 轴正方向夹角为 θ_1 和 θ_2 。将简化后的模型中两轮连线的中点作为参考点, 两者的旋转点距离为 l , 跟随机器人前进方向与旋转点的连线夹角为 φ 。

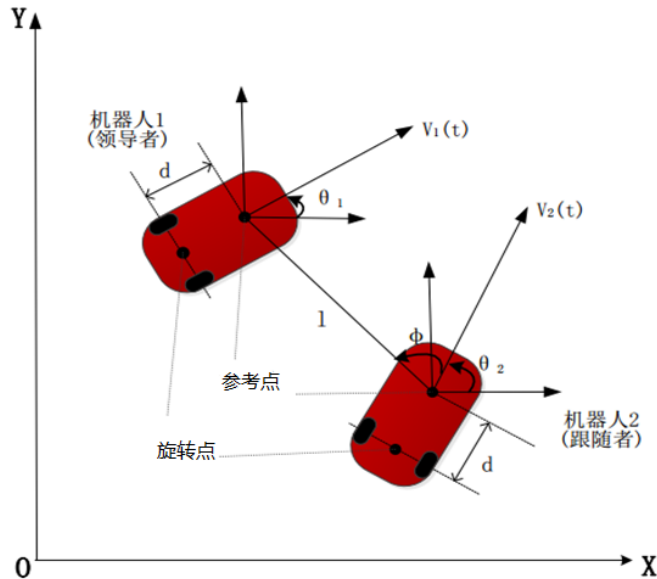


图 3-2 跟随者保持队形模型图

对于跟随者的运动状态, 采用参考类似 SBC 控制中跟随者的运动学特性的描述方法, 建立跟随者的运动学特性方程为:

$$\begin{cases} \dot{l} = v_1 \cos \gamma + d\omega_1 \sin \gamma - v_2 \cos \varphi - d\omega_2 \sin \varphi \\ \dot{\varphi} = \frac{1}{l}(v_2 \sin \varphi - d\omega_2 \cos \varphi - v_1 \sin \gamma + d\omega_1 \cos \gamma - l\omega_2) \\ \dot{\theta}_2 = \omega_2 \end{cases} \quad (3.4)$$

其中

$$\gamma = \theta_2 - \theta_1 + \varphi \quad (3.5)$$

3.3 跟随者编队控制策略

在上述所选定的领导-跟随模型中，我们需要对领导者和跟随者这两类不同角色建立运动学模型。主要依据便是两者之间相对位置和相对角度来描述互相的运动位置状态。首先根据图 3-3 描述的运动学模型画出闭环控制框图如下：

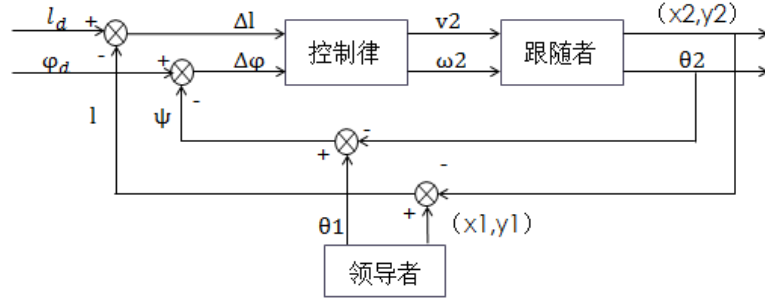


图 3-3 1-φ 闭环控制框图

在编队过程中，如果每个跟随者与领导者的相对距离 l 和角度 φ 能收敛到给定的期望值 l_d 和 φ_d ，即满足：

$$\lim_{t \rightarrow \infty}(\Delta l) = 0, \quad \lim_{t \rightarrow \infty}(\Delta \varphi) = 0 \quad (3.6)$$

就说明实现了编队控制效果。

用差分法将式(3.6)表示为以下形式：

$$\begin{cases} \dot{l} = k_1(l_d - l) \\ \dot{\varphi} = k_2(\varphi_d - \varphi) \end{cases} \quad (3.7)$$

k_1 和 k_2 为比例控制系数。由式(3.4)和式(3.7)可以求得跟随者的控制律，即得到它的控制输出 v_2 和 ω_2 如下：

$$\begin{cases} \omega_2 = (\rho_1 \sin \varphi + \rho_2 \cos \varphi) / \rho_6 \\ v_2 = (\rho_3 \sin \varphi + \rho_4 \cos \varphi - \rho_5) / \rho_6 \end{cases} \quad (3.8)$$

其中，

$$\begin{cases} \rho_1 = v_1 \cos \gamma + d\omega_1 \sin \gamma - k_1(l_d - l) \\ \rho_2 = d\omega_1 \cos \gamma - v_1 \sin \gamma - k_2(\varphi_d - \varphi) \\ \rho_3 = dv_1 \sin \gamma - d^2\omega_1 \cos \gamma + dlk_2(\varphi_d - \varphi) \\ \rho_4 = dv_1 \cos \gamma + d^2\omega_1 \sin \gamma - dk_1(l_d - l) \\ \rho_5 = lv_1 \cos \gamma + dl\omega_1 \sin \gamma - lk_1(l_d - l) \\ \rho_6 = d + l \cos \varphi \end{cases} \quad (3.9)$$

为了证明此控制律的稳定性，将式(3.8)改写为以下形式：

$$\begin{cases} v_1 = \beta_1 \cos \gamma - \beta_2 \sin \gamma \\ \omega_1 = (\beta_2 \cos \gamma + \beta_1 \sin \gamma) / d \end{cases} \quad (3.10)$$

其中，

$$\begin{cases} \beta_1 = d\omega_2 \sin \varphi + v_2 \cos \varphi + k_1(l_d - l) \\ \beta_2 = d\omega_2 \cos \varphi - v_2 \sin \varphi + k_2 l(\varphi_d - \varphi) + l\omega_2 \end{cases} \quad (3.11)$$

令稳态误差 $e_l = \Delta l$, $e_\varphi = \Delta \varphi$, 下面用李雅普诺夫稳定性定理证明系统在 $e_l = 0$ 且 $e_\varphi = 0$ 处的稳定性, 构造 Lyapunov 能量函数如下:

$$V = \frac{1}{2} e_l^2 + \frac{1}{2} e_\varphi^2$$

可以看出 V 是正定的, 对 V 求导并将运动学方程式(3.4)带入得:

$$\begin{aligned} \dot{V} &= e_l \dot{e}_l + e_\varphi \dot{e}_\varphi \\ &= -(l_d - l)\dot{l} - (\varphi_d - \varphi)\dot{\varphi} \\ &= -(l_d - l)(v_1 \cos \gamma + d\omega_1 \sin \gamma - v_2 \cos \varphi - d\omega_2 \sin \varphi) \\ &\quad - (\varphi_d - \varphi) \frac{1}{l} (v_2 \sin \varphi - d\omega_2 \cos \varphi - v_1 \sin \gamma + d\omega_1 \cos \gamma - l\omega_2) \end{aligned}$$

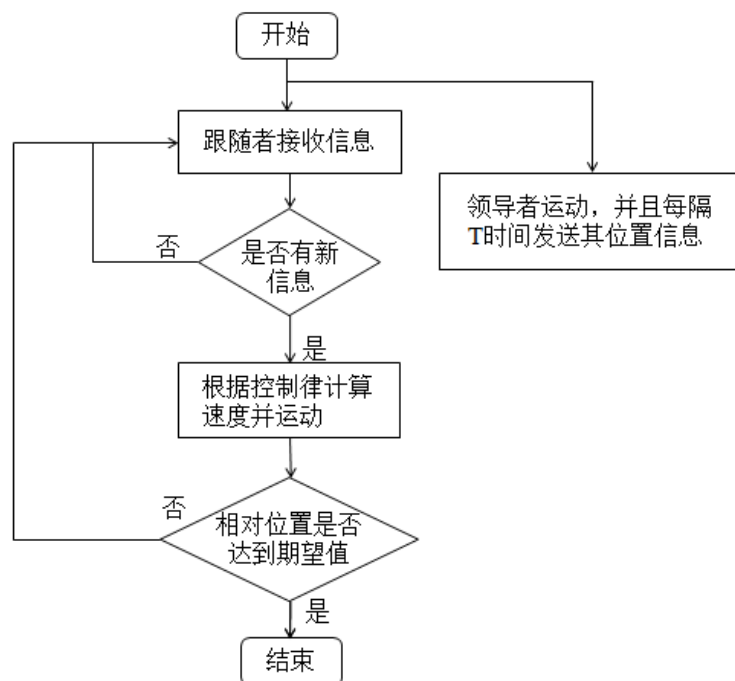
将控制律(3.10)和(3.11)带入上式得:

$$\begin{aligned} \dot{V} &= -(l_d - l)(\beta_1 \cos^2 \gamma - \beta_2 \sin \gamma \cos \gamma + \beta_2 \sin \gamma \cos \gamma + \beta_1 \sin^2 \gamma - v_2 \cos \varphi \\ &\quad - d\omega_2 \sin \varphi) \\ &\quad - (\varphi_d - \varphi) \frac{1}{l} (v_2 \sin \varphi - d\omega_2 \cos \varphi - \beta_1 \sin \gamma \cos \gamma + \beta_2 \sin^2 \gamma + \beta_2 \cos^2 \gamma \\ &\quad + \beta_1 \sin \gamma \cos \gamma - l\omega_2) \\ &= -(l_d - l)(\beta_1 - v_2 \cos \varphi - d\omega_2 \sin \varphi) \\ &\quad - (\varphi_d - \varphi) \frac{1}{l} (\beta_2 + v_2 \sin \varphi - d\omega_2 \cos \varphi - l\omega_2) \\ &= -(l_d - l)[k_1(l_d - l)] - (\varphi_d - \varphi) \frac{1}{l} [k_2 l(\varphi_d - \varphi)] \\ &= -k_1(l_d - l)^2 - k_2(\varphi_d - \varphi)^2 \\ &= -k_1 e_l^2 - k_2 e_\varphi^2 \end{aligned}$$

当 $k_1, k_2 > 0$ 时, \dot{V} 是负定的, 满足李雅普诺夫稳定性定理, 所以系统在 $e_l = 0$ 且 $e_\varphi = 0$ 处是渐进稳定的。

3.4 算法实现

根据流程图 3-4 分别对领导者和跟随者编写程序, 完整代码见附录 1。程序的关键点在于跟随者对接收到的信息的处理。我在程序中将处理信息部分封装成一个函数 `get_vw(message_rx1, message_rx2, message_rx3)`, 参数 1 和 2 分别是领导者的横、纵坐标, 第 3 个参数是领导者前进方向与 x 轴的夹角。在一次通信之后, 跟随者处理信息, 并根据计算得到的线速度和角速度开始运动。接下来, 每个周期 T 进行一次通信, 直至跟随者和领导者的相对位置达到期望值, 完成编队。

图 3-4 1- φ 编队控制算法流程图

3.5 仿真结果展示

Kilobot 机器人的电机是通过 PWM 脉冲驱动电路来控制转速的, `set_motors(left_v, right_v)` 中参数可调范围是 0~255, 它对应的实际速度是 0~0.01m/s。根据两轮差速驱动模型得到线速度 v 和角速度 ω 与左右两轮速度 v_r, v_l 的关系:

$$\begin{cases} v = (v_r + v_l)/2 * 2550 \\ \omega = \Delta * (v_r - v_l) \end{cases}$$

通过在 V-rep 模拟器上调试可推出上式中的 $\Delta = 0.09$, 并由此得到了 v_r, v_l 的计算公式:

$$\begin{cases} v_r = 5100v + 50\omega/9 \\ v_l = 5100v - 50\omega/9 \end{cases}$$

为了将速度控制在 Kilobot 的可调范围内, 先根据设定的参数在 MATLAB 上进行仿真。下面分别是领导者和两个跟随者的初值设定: 领导者, 初始位置(0,0), $v=0.3\text{cm/s}$, $\omega=0.5\text{rad/s}$, 初始方向 $\theta=90^\circ$; 1 号跟随者, 初始位置(-0.01, -0.06), 初始方向 $\theta=-10^\circ$; 期望距离和角度 $l=0.04, \varphi=90^\circ$; 2 号跟随者, 初始位置(-0.04, -0.04), 初始方向 $\theta=40^\circ$; 期望距离和角度 $l=0.04, \varphi=-90^\circ$ 。MATLAB 仿真结果图如下:

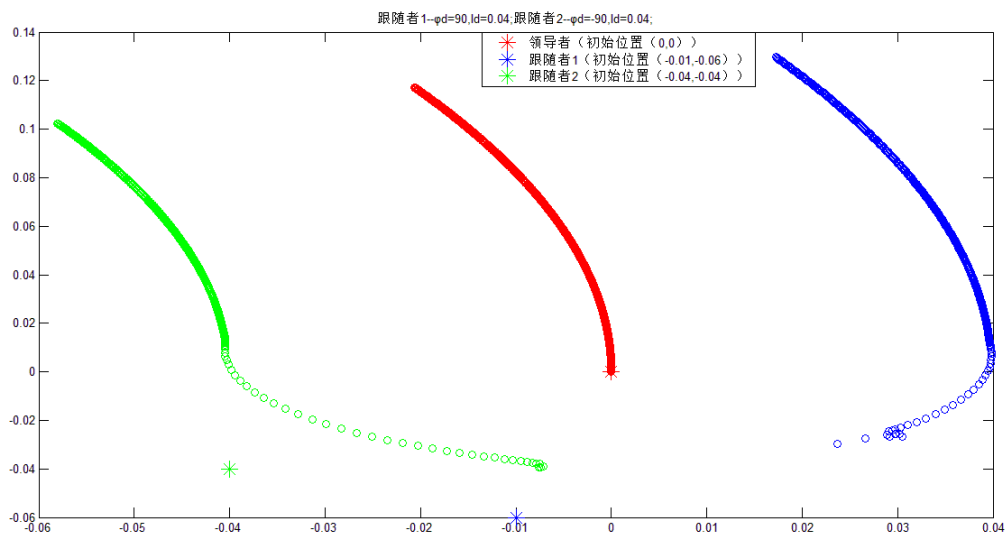


图 3-5 MATLAB 仿真效果图

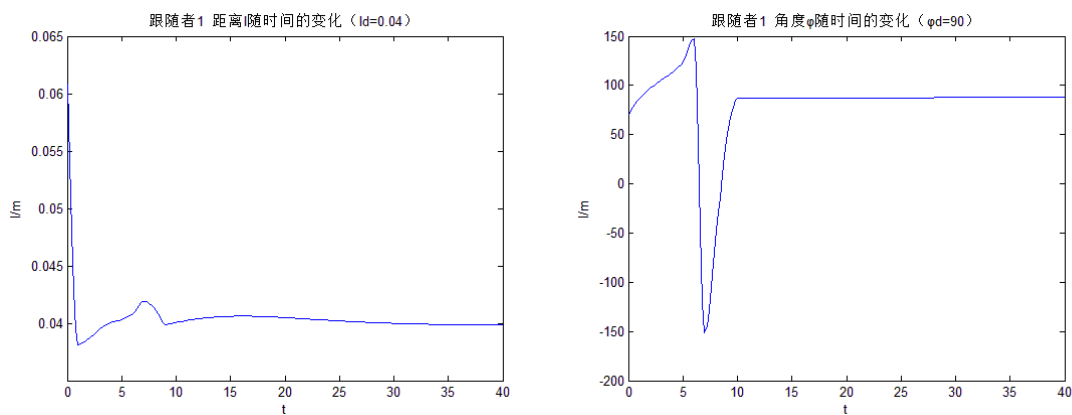


图 3-6 跟随者 1 号 l , φ 随时间变化的曲线图

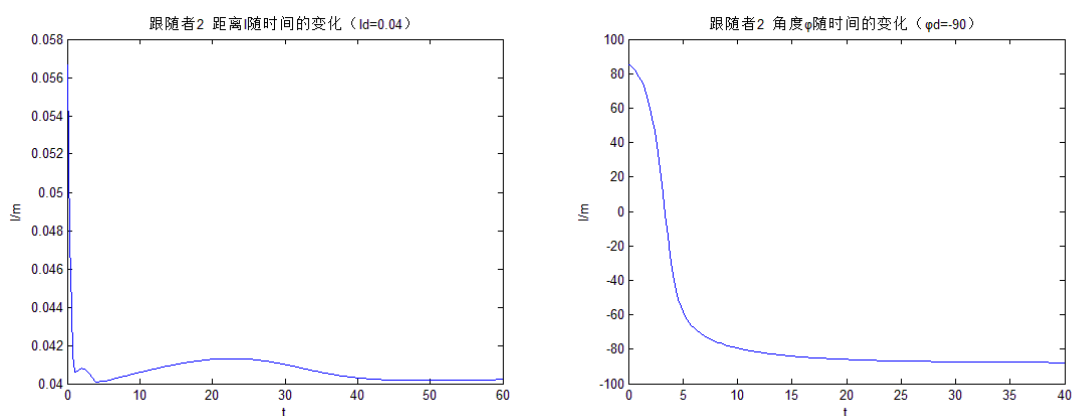


图 3-7 跟随者 2 号 l , φ 随时间变化的曲线图

由图 3-6 和图 3-7 可以看出，变量 l , φ 成功收敛到期望值，且稳态误差小，系统稳定。

但是计算得到 v 和 ω 的数据偏大，换算成 v_r ， v_l 远远超出了 Kilobot 的可调范围。因为 ω 是控制旋转的，为了尽量保证收敛速度，只对 v 进行处理，使 v_r ， v_l 保持在可调范围内。但是由于对 v 的调节没有一个好的办法，所以在模拟仿真时会存在跟随者速度过大而翻车的情况。下面展示在 V-rep 模拟器上的仿真结果。

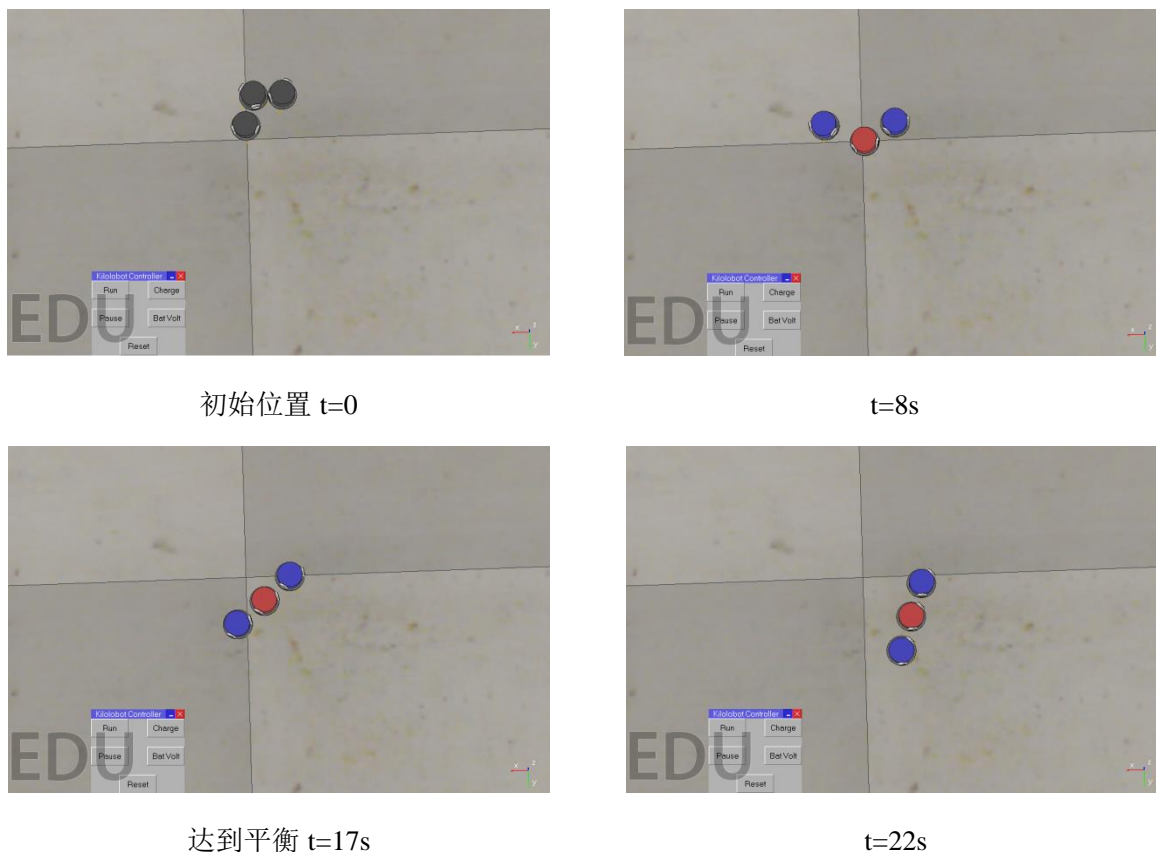


图 3-8 V-rep 模拟仿真截图

图 3-8 中，红色机器人为领导者，两个蓝色机器人为跟随者，它们的初始信息与 MATLAB 仿真时一致。左上图为实验初始时刻的截图，右上图为系统达到稳定前某时刻的截图，左下图为系统达到稳定时刻的截图，右下图为系统达到稳定后某时刻的截图，3 个机器人将保持这样的队形持续运动。

因为在实际中，Kilobot 无法像在模拟器中获得定位，所以要将此方法在实物上验证会十分困难，所以只进行了仿真验证。

3.6 本章小结

本次实验研究的领导-跟随者法($l - \phi$ 法)实现方法简单，便于使用和接受。在编队过程中，只需要考虑期望的相对距离和角度，就能使系统达到编队的效果，使用起来非常

灵活。而且每个跟随者都只和领导者进行通信，所以此方法对跟随者的数量没有太大的限制，扩展性较好；但是由于考虑到 Kilobot 机器人的最大通信范围在 7cm 左右，所以在仿真验证时只用了 3 个机器人来实现。

第 4 章 不同延时下离散时间多智能体系统的一致性

4.1 选择一致性协议

如绪论中所描述的，对一致性问题的研究主要以图论法和矩阵法为基础，此问题的核心是设计一致性协议，而对于这些协议的研究和分析在不断地继续，这也对我们这些初学者带来了极大的便利。而在阅读这些文章后发现，系统在进行通信交互时，通信延时是无法避免的，因此研究存在通信时延的多智能体系统的一致性控制协议具有很好的现实意义。

本次研究将选取的方法是——不同延时下离散时间多智能体系统的一致性协议^[7]。此协议可以有效地解决通信时延问题。因为计算机在系统分析和控制设计中的快速发展，离散时间模型和控制器在实际应用中变得越来越方便，所以研究离散时间模型也具有重要的意义。

接下来就以图论法为基础，提出次协议，并基于广义的 Nyquist 稳定性判据和盖尔圆理论分析其一致性。

4.2 图论法

多智能体之间的通信用有向图 $G = (V, E, A)$ 来表示，其中 $V = \{1, \dots, N\}$ 是节点集， $E \subseteq V \times V$ 是边的集合，每条边用 (i, j) 表示， $A = (a_{ij}) \in R^{N \times N}$ 是具有权重的加权邻接矩阵。若智能体 i 有一条边指向智能体 j ，那么 $a_{ij} \neq 0, \forall i \neq j$ ；否则， $a_{ij} = 0$ 。 $N_i = \{j \in V \mid (j, i) \in E\}$ 表示智能体 i 的邻居集合。在本次研究者，总是假设 $a_{ii} = 0, a_{ij} \geq 0, i, j = 1, \dots, N$ ，即权重不带符号。令 D 为图 G 的出度对角矩阵， $D = \text{diag}(\sum_{j=1}^N a_{1j}, \dots, \sum_{j=1}^N a_{Nj})$ ， $L_G = D - A$ 是 G 的拉普拉斯矩阵，其中， l_{ij} 表示为：

$$l_{ij} = \begin{cases} \sum_{j \in N_i} a_{ij}, & i = j \\ -a_{ij}, & i \neq j \end{cases}$$

4.3 一致性模型

令由 N 个智能体组成的系统，其索引为 1 到 N 。用 $x_i(k)$ 来表示第 i 个智能体在 k 时刻的状态，将其连续时间一阶系统的状态方程 $\dot{x}_i(t) = u_i(t)$ 离散化后得：

$$x_i(k+1) = x_i(k) + Tu_i(k) \quad (4.1)$$

其中， $x_i(k) \in R^n$ ， $u_i(k) \in R^n$ 。 N 个智能体之间的通信由带有符号的有向图 $G = (V, E, A)$ 表示。 T 为采样周期。

具有通信时延的控制输入为：

$$u_i(k) = -\sum_{j \in N_i} a_{ij} [x_i(k - \tau_{ij}) - x_j(k - \tau_{ij})] \quad (4.2)$$

其中， τ_{ij} 表示信息从智能体 j 传到 i 时消耗的采样周期数，即通信时延， τ_{ij} 为大于 0 的整数，它在 $x_i(k - \tau_{ij})$ 中被称为自延迟。

将协议(4.2)带入状态方程(4.1)得：

$$x_i(k+1) = x_i(k) - T \sum_{j \in N_i} a_{ij} [x_i(k - \tau_{ij}) - x_j(k - \tau_{ij})] \quad (4.3)$$

式 (4.3) 就是不同延时下离散时间多智能体系统的一致性模型。因为 a_{ij} 不随时间变化，其输出 $y_i(k) = \sum_{j \in N_i} x_i(k)$ 满足叠加定理，所以该系统是一个具有通信时延的线性定常系统。若 $\lim_{k \rightarrow \infty} \|x_i(k - \tau_{ij}) - x_j(k - \tau_{ij})\| = 0$ ，则表示系统是收敛的。

4.4 一致性分析

本节中将基于广义的 Nyquist 稳定性判据和盖尔圆理论对系统进行一致性分析。所谓广义的 Nyquist 稳定性判据是指：广义 Nyquist 曲线不绕点 $(-1, j0)$ ，即全部特征根都位于复平面的左半平面上，是线性定常系统具有稳定性的充分必要条件。而盖尔圆定理是指：设 $A = (a_{ij}) \in C^{n \times n}$ ，则它的特征根都落在复平面上的 n 个圆盘 $D_i(A) = \{z \mid |z - a_{ii}| \leq P_i\} (i = 1, \dots, n)$ 的并中，其中 $P_i = \sum_{j \neq i} |a_{ij}|$ 。

在进行一致性分析之前，先给出在接下来证明过程中需要用到的引理。

假设该系统的开环传递函数为 $G(z)$ ，转化到频率域中表示为 $G(z) = G(e^{j\omega})$ ， $\omega \in [-\pi, \pi]$ 。令 $G(e^{j\omega}) = c + jd$ ， $c, d \in R$ ， $\lambda(G(e^{j\omega}))$ 表示广义 Nyquist 特性曲线。

引理：如果系统的 $\lambda(G(e^{j\omega}))$ 在圆形区域 $\lambda(G(e^{j\omega})) \in \{\theta: \theta \in C, |\theta - (c + jd)| \leq |c + jd|\}$ 内部，那么它不环绕点 $(-1, j0)$ 的充分必要条件为：当 ω 在 $[-\pi, \pi]$ 上变化时，总有 $c > -1/2$ 。

证明上述引理如下：假设复平面上有一点 $(-b, j0)$ ， $b \in R$ 且 $b \geq 1$ ，如果系统稳定，即开环系统的广义 Nyquist 特性曲线不环绕点 $(-1, j0)$ ，那也说明曲线也不环绕点 $(-b, j0)$ 。根据上述引理的条件，换句话说就是点 $(-b, j0)$ 不在圆形区域 $\{\theta: \theta \in C, |\theta - (c + jd)| \leq |c + jd|\}$ 的内部，将点 $(-b, j0)$ 带入不等式得：

$$\begin{aligned} |(-b + j0) - (c + jd)| &> |c + jd| \\ \Rightarrow |-b - c - jd| &> |c + jd| \\ \Rightarrow |b + c + jd| &> |c + jd| \end{aligned}$$

上式中不等式两边均表示到原点的距离，因为其虚部一致，所以只需要考虑实部，即：

$$|b + c| > |c|$$

易得 $b(b + 2c) > 0$ 。又因为 $b \geq 1$ ，所以 $1 + 2c > 0$ ，解得 $c > -1/2$ 。证毕。

下面给出系统(4.3)能实现一致性收敛的充分必要条件：

$$\tau_{ij} < \frac{2-2T \sum_{j \in N_i} a_{ij}}{\pi^2 T \sum_{j \in N_i} a_{ij}} \quad (4.4)$$

证明：根据上述引理对条件(4.4)进行证明，即只需系统(4.3)满足以下两点，(1)系统的开环传递函数的奈奎斯特特性曲线在一个圆形区域内。(2)若存在圆形区域，则其圆心的实部必须大于 $-1/2$ 。因此，求出系统的开环传递函数是证明的关键。

首先将式(4.3)两边进行 Z 变换， $x_i(k)$ 的 Z 变换用 $X_i(z)$ 来表示，得到：

$$\begin{cases} zX_i(z) - zx_i(0) = X_i(z) - T \sum_{j \in N_i} a_{ij} (X_i(z)z^{-\tau_{ij}} - X_j(z)z^{-\tau_{ij}}) \\ \left(\frac{z-1}{T}\right)X_i(z) = -\sum_{j \in N_i} a_{ij} (X_i(z)z^{-\tau_{ij}} - X_j(z)z^{-\tau_{ij}}) + \frac{x_i(0)}{T} \end{cases} \quad (4.5)$$

定义 $n \times n$ 的矩阵 $L(z) = [l_{ij}^-(z)]$ ，由式(4.5)可以得到：

$$l_{ij}^-(z) = \begin{cases} \sum_{j \in N_i} a_{ij} z^{-\tau_{ij}}, & i = j \\ -a_{ij} z^{-\tau_{ij}}, & i \neq j \end{cases} \quad (4.6)$$

从上式中可以看出，当 $\tau_{ij} = 0$ 时， $L(z)$ 为系统的拉普拉斯矩阵 L_G 的 Z 变换。将式(4.5)用矩阵的形式表示如下：

$$\left(\frac{z-1}{T}I + \bar{L}(z)\right)X(z) = \frac{x(0)}{T} \quad (4.7)$$

其中, $X(z) = [X_1(z), X_2(z), \dots, X_n(z)]^T$, $x(0) = [x_1(0), x_2(0), \dots, x_n(0)]^T$ 。特征方程的根决定了系统式(4.3)的稳定性, 关于 $x(k) = [x_1(k), x_2(k), \dots, x_n(k)]^T$ 的特征方程可以表示为:

$$\det\left(I + \frac{T}{z-1}\bar{L}(z)\right) = 0 \quad (4.8)$$

上式为系统闭环特征方程。对特征方程做频域分析, 由上式可得到开环传递函数为:

$$G(z) = \frac{T}{z-1}\bar{L}(z) \quad (4.9)$$

接下来基于盖尔圆定理, 得 $\lambda(G(e^{j\omega}))$ 的所在区域为:

$$\lambda(G(e^{j\omega})) \in \bigcup_{i \in V} \left\{ \theta: \theta \in \mathbb{C}, \left| \theta - \frac{e^{-\tau_{ij}\omega T} \sum_{j \in N_i} a_{ij}}{e^{j\omega} - 1} \right| \leq \sum_{j \in N_i} \left| \frac{a_{ij} e^{-\tau_{ij}\omega T}}{e^{j\omega} - 1} \right| \right\}$$

其中,

$$\frac{e^{-\tau_{ij}\omega T} \sum_{j \in N_i} a_{ij}}{e^{j\omega} - 1} = \frac{(\cos \omega \tau_{ij} - j \sin \omega \tau_{ij}) T \sum_{j \in N_i} a_{ij}}{\cos \omega + j \sin \omega - 1}$$

上式中实部为 $-\frac{1}{2} T \sum_{j \in N_i} a_{ij} \left(\cos \omega \tau_{ij} + \frac{\sin \omega \tau_{ij} \sin \omega}{1 - \cos \omega} \right)$ 。由引理可知: 当

$-\frac{1}{2} T \sum_{j \in N_i} a_{ij} \left(\cos \omega \tau_{ij} + \frac{\sin \omega \tau_{ij} \sin \omega}{1 - \cos \omega} \right) > -\frac{1}{2}$ 时, 系统的广义 Nyquist 特性曲线不环绕点

$(-1, j0)$, 即达到收敛一致性。所以, 解不等式如下:

$$-\frac{1}{2} T \sum_{j \in N_i} a_{ij} \left(\cos \omega \tau_{ij} + \frac{\sin \omega \tau_{ij} \sin \omega}{1 - \cos \omega} \right) > -\frac{1}{2}$$

$$T \sum_{j \in N_i} a_{ij} \left(\cos \omega \tau_{ij} + \frac{\sin \omega \tau_{ij} \sin \omega}{1 - \cos \omega} \right) < 1$$

$$T \sum_{j \in N_i} a_{ij} \left(\cos \omega \tau_{ij} + \frac{\tau_{ij}}{2} \left(\frac{\sin \omega \tau_{ij}}{\omega \tau_{ij}} \right) \left(\frac{\sin \omega}{\omega} \right) \left(\frac{\omega}{\sin(\omega/2)} \right)^2 \right) < 1$$

因为 $\cos \omega \tau_{ij} < 1$, $\frac{\sin \omega \tau_{ij}}{\omega \tau_{ij}} < 1$, $\frac{\sin \omega}{\omega} < 1$, $\frac{\omega}{\sin(\omega/2)} < \pi$, 所以有:

$$T \sum_{j \in N_i} a_{ij} \left(1 + \frac{\pi^2}{2} \tau_{ij} \right) < 1$$

解得,

$$\tau_{ij} < \frac{2-2T \sum_{j \in N_i} a_{ij}}{\pi^2 T \sum_{j \in N_i} a_{ij}} \quad (4.10)$$

所以，当系统(4.3)满足条件(4.10)时，其全部特征根都位于复平面的左半平面上，且所有智能体的状态达到一致，证毕。

4.5 算法实现

因为该一致性控制是基于分布式控制的，所以各个智能体除了传输的数据不同，其他条件都是一致的，可以根据流程图 4-1 对每个智能体编写程序。程序以其中一个智能体为例，完整的代码见附录 2。

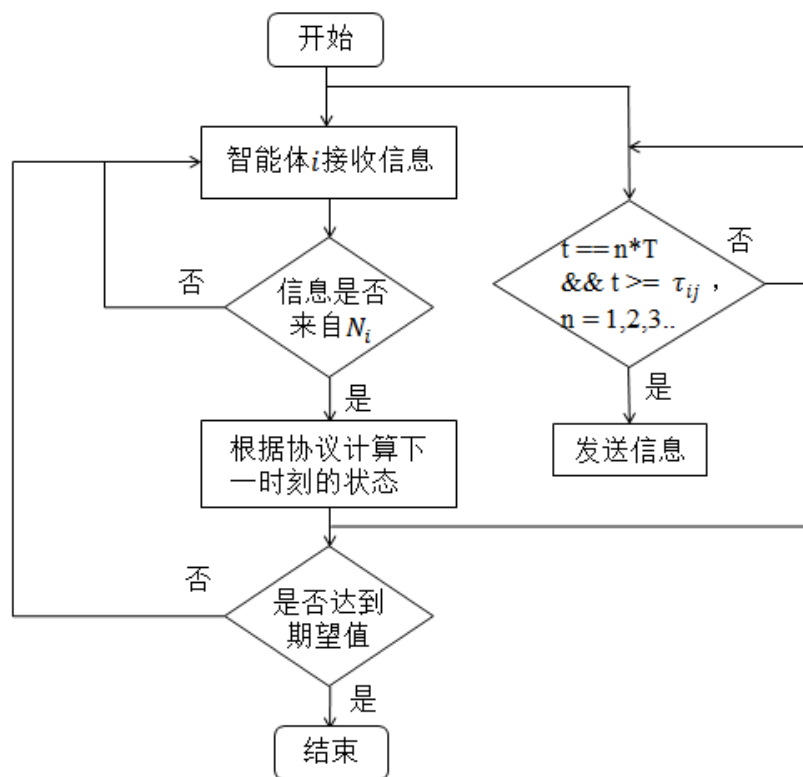


图 4-1 一致性控制算法流程图(以智能体*i*为例)

编写程序的关键在于智能体和其邻居之间的通信过程。因为存在通信延时，所以我定义了一个数据结构 `array_x` 用来存放从延时时刻起到当前时刻的所有状态量。解决了这个关键点，接下来就是根据协议使智能体的状态实现一致性。

4.6 仿真结果展示

定义图 4-2 所示的拓扑结构，用该拓扑结构来验证一致性协议(4.2)的有效性。

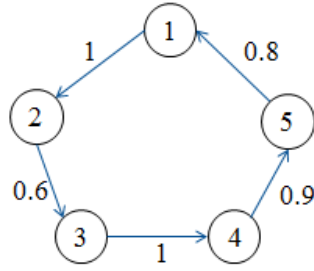


图 4-2 单项环形拓扑

上图中已经标出各个节点与邻居之间的权重，表示成加权邻接矩阵为：

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0.8 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0.6 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.9 & 0 \end{bmatrix}$$

取采样周期 $T=0.05s$ ，由式 $\tau_{ij} < \frac{2-2T \sum_{j \in N_i} a_{ij}}{\pi^2 T \sum_{j \in N_i} a_{ij}}$ 可以解得：

$$\tau_{15} < 4.86, \tau_{21} < 3.85, \tau_{32} < 6.55, \tau_{43} < 3.85, \tau_{54} < 4.3$$

因为 τ_{ij} 为非负整数，所以取延时矩阵如下：

$$\tau = \begin{bmatrix} 0 & 0 & 0 & 0 & 2 \\ 3 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \end{bmatrix}$$

在数值仿真时，给定 5 个智能体的初始状态为 $x_1(0) = 1$ ， $x_2(0) = 2$ ， $x_3(0) = 3$ ， $x_4(0) = 4$ ， $x_5(0) = 5$ ，图 4-3 位单向环形系统在 MATLAB 上的仿真结果。由图可知，系统在 180 个采样周期后稳定，且状态达到一致，收敛于 2.9539。对于期望值的计算，并没有在上述方法分析中提出。因此我通过选取不同的加权邻接矩阵和初始状态，进行了多次的数值仿真后，发现期望值与各个权重是反比关系。即得到期望的计算公式如下：

$$x = \frac{x_1(0)/a_{51} + x_2(0)/a_{12} + x_3(0)/a_{23} + x_4(0)/a_{34} + x_5(0)/a_{45}}{1/a_{51} + 1/a_{12} + 1/a_{23} + 1/a_{34} + 1/a_{45}} \quad (4.11)$$

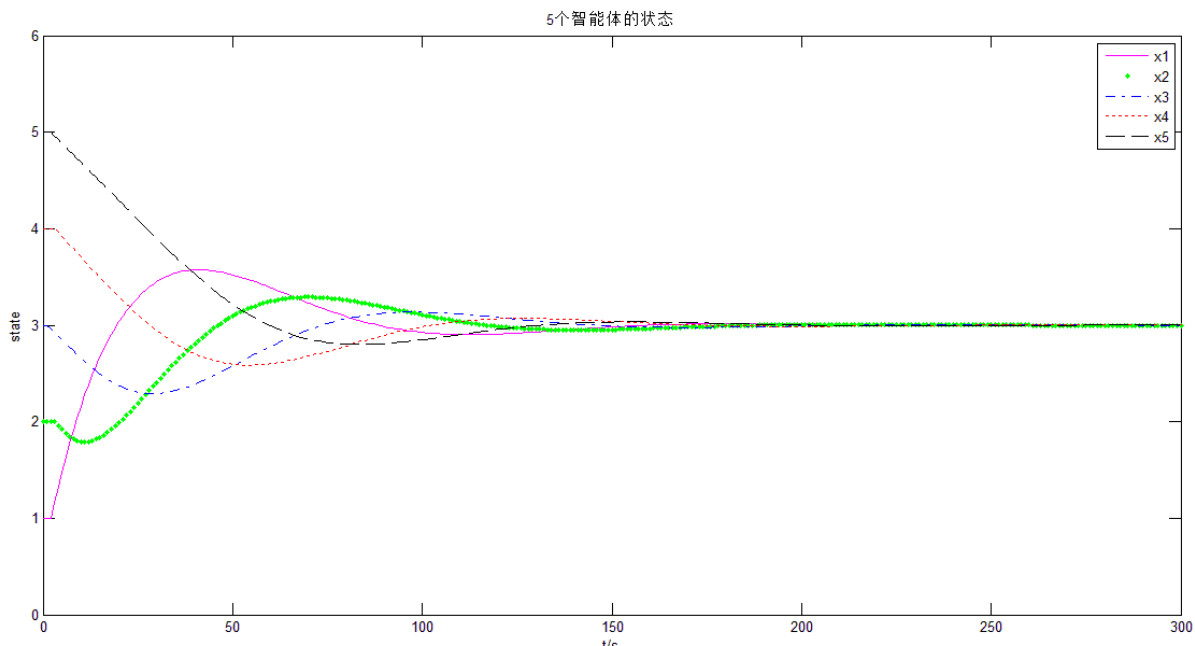
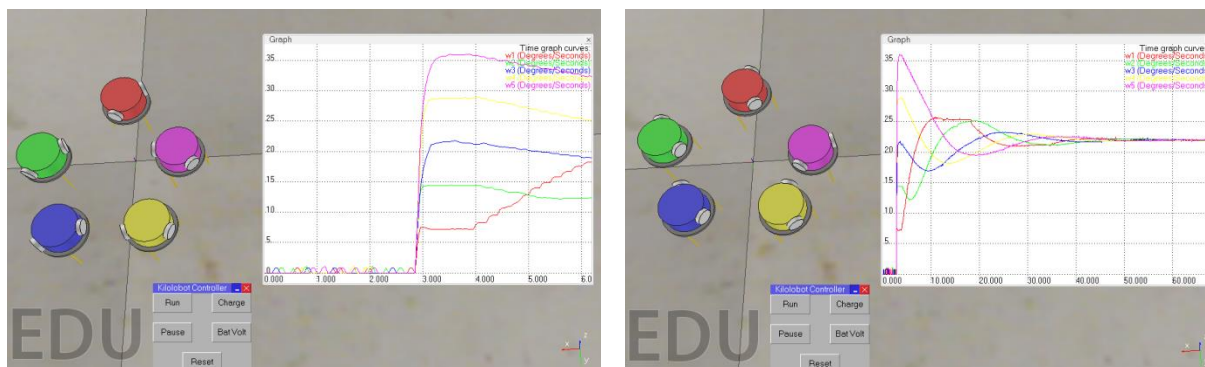


图 4-3 单向环形一阶系统一致性

依然使用图 4-2 所示的拓扑结构，在 V-rep 模拟器上进行了 Kilobot 机器人的模拟仿真，仿真结果见图 4-4。图中上方两张图为运行过程中的截图，红、绿、蓝、黄、紫这 5 个 Kilobot 模拟机器人分别对应拓扑结构中的 1~5 号；实验中选取 Kilobot 的角速度为其状态，初始值分别为： $\omega_1(0) = 7.2rad/s$ ， $\omega_2(0) = 14.4rad/s$ ， $\omega_3(0) = 21.6rad/s$ ， $\omega_4(0) = 28.8rad/s$ ， $\omega_5(0) = 36rad/s$ ，Kilobot 右边的表格用来记录其状态随时间变化的过程，下方的图是整个变化过程。



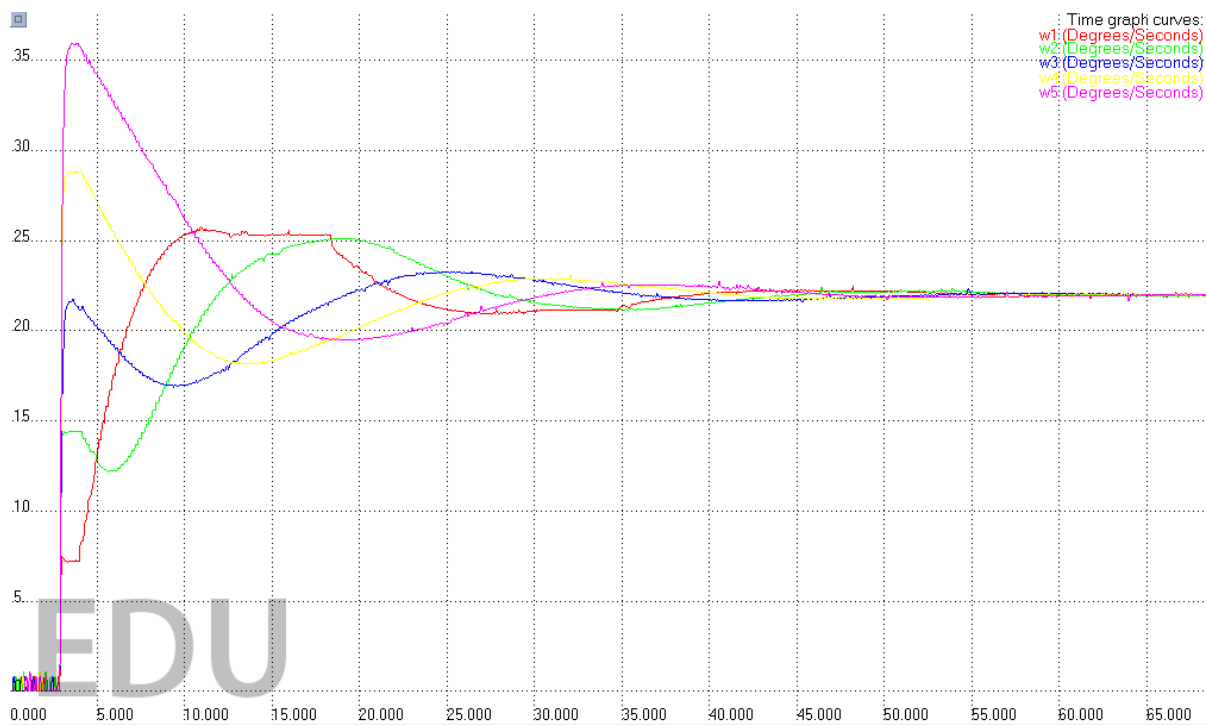


图 4-4 Kilobot 模拟仿真结果

4.7 Kilobot 实物实验结果展示

采用上节中拓扑结构和参数选取，进行实物实验，过程中遇到了以下问题：

1. 第 3 节中已讲到 Kilobot 机器人的电机是通过 PWM 脉冲驱动电路来控制转速的，`set_motors(left_v, right_v)`中参数可调范围是 0~255。而实际中发现，当参数设置为 0~60 左右时，电机虽然在工作，但它无法带动机器人运动；而当参数设置为 70 以上时，机器人不能根据设定条件正常运动，而是开始无规则地跑动。
2. 即使将参数设置在 60~70 的范围内，由于机器人的支撑腿受运动表面的影响较大，也时常出现不规则运动的现象。
3. 由于每个机器人之间有差异，当给他们设置相同的参数值时，出现运动情况不同的情况。

为了解决以上问题，在实验中做出以下调整：

1. 改变了角速度初值的选取，保证在实验过程中，参数值控制在 60~70 之间。
2. 由于无法克服运动表面对支撑腿的影响，将所有机器人倒置（见图 4-5 右）来进行实验，效果确实比之前更好。

通过以上调整后再进行实验，虽然实验效果有所好转，但是由于其速度可调范

围不大，而且各个机器人之间存在差异，导致了在实验过程中仍然会出现有机器人的速度超出可调范围乱跑的现象，这只能采用手动调整解决。



图 4-5 Kilobot 实验实物图 左(正立) 右(倒立)

4.8 本章小结

本次实验研究的不同延时下的离散时间多智能体系统的一致性协议，思想简单，便于使用和接受，而且具有实际的应用价值。虽然在模拟器中可以顺利仿真，且很好的验证了该协议的有效性，但是由于 Kilobot 机器人在实际中的种种因素，致使实物验证的效果不明显。

第 5 章 总结与展望

5.1 总结

多智能体的协同控制问题已经成为控制领域的一个热点问题。随着科技的进步和人们生活水平的提高，人工智能已经渐渐地进入了各个领域、行业乃至我们的生活。因此对于多智能体的协同控制问题的研究将有利于人工智能的发展。本文主要探讨了基于 Kilobot 微型机器人集群的协同控制问题，着重研究了其中编队控制和一致性控制两个问题，先将本文工作总结如下：

(1) 分析了课题研究的背景及意义，并概述了协同控制的几个基本问题。

(2) 介绍了 Kilobot 实验平台和 V-rep 模拟器。总结介绍了 Kilobot 的主要特点、工作原理及其 API。介绍了 V-rep 模拟器的基本信息和特点，以及选择该模拟器进行仿真实验的原因。

(3) 介绍了领导-跟随者法的模型及原理，分析了方法的稳定性，并展示了仿真结果，同时对方法优缺点做了简单的分析。

(4) 简要描述了研究一致性协议的一种常用方法图论法，介绍了具有通信时延的离散时间多智能体一致性通信协议，并分析其使用时的一些条件，最后展示了仿真结果和在 Kilobot 平台上的实验结果。

以上两种方法都进行了仿真验证，并结合 Kilobot 的特性，选择了后者做了实物验证，总的来讲，本次研究还是比较成功的，达到了预期的要求。在整个实验过程中也学到了很多新的知识，比如学习了使用 V-rep 模拟器的部分功能，掌握了 Lua 脚本语言，同时在实物调试过程中对于遇到的问题，学会了从不同角度考虑并解决，也提高了动手能力。

5.2 展望

本文经过认真的实验和研究，基本达到了预期的研究目标，完成了基于 Kilobot 微型机器人的协同控制的研究。但是本次实验仅仅选择了两种相对简单且适用范围广的方法进行讨论，而且在实物实验时针对 Kilobot 的特性，做了一些特殊处理，某些算法中要体现的特性并不能再实际中很好地展现，因此无论在研究何种算法时，有一个好的

实验平台是极其重要的。

接下来，对于各种算法的研究当然重要，同时研究能够用于实验研究的好的平台也是一个大方向。

参 考 文 献

- [1] 刘学良. 多智能体系统协调控制中的若干问题研究[J]. 华南理工大学,2012.06.
- [2] 王文明, 周帅. 多机器人协同控制与编队方法研究[J]. 北京理工大学计算机学院, 2014.06.
- [3] 袁坤. 多智能体网络一致性问题的分布式算法研究[J]. 中国科学技术大学,2014.05.
- [4] Hong Y. G., Chen G. R., Bushnell L. Distributed observers design for leader-following control of multi-agent networks[J]. Automatica, 2008, 44(3): 846-850.
- [5] Rubenstein M, Ahler C, Nagpal R. Kilobot: A low cost scalable robot system for collective behaviors[C]. IEEE International Conference on Robotics and Automation. IEEE, 2012:3293-3298.
- [6] 吴垠. 多智能体系统领导者路径规划与编队控制研究[D]. 南开大学, 2015.
- [7] 朱旭, 闫建国, 屈耀红. 不同延迟下离散多智能体系统的一致性[J]. 电子与信息学报, 2012, 34(6):1516-1520.
- [8] 刘佳, 陈增强, 刘忠信. 多智能体系统及其协同控制研究进展[J]. 南开大学,信息技术科学学院, 2010.01.
- [9] 巩俐. AmigoBot 机器人协同操作控制研究[J]. 济南大学,2014.05.
- [10] Ma C Q, Qin Z Y. Bipartite consensus on networks of agents with antagonistic interactions and measurement noises[J]. Iet Control Theory & Applications, 2016, 10(17):2306-2313.
- [11] 刘宗春. 群体机器人系统协同一致性行为控制算法设计与仿真研究[J]. 吉林大学,2011.05.
- [12] Reynolds C. W. Flocks, herds, and schools: a distributed behavioral model[C]. Poceedings of the 14th Annual Conference on Computer Grophics and Interactive Technique, Institute of Electrical and Electronics Engineers Inc., 1987, 21: 25-34.
- [13] T Vicsek, A Czirák, E Ben-Jacob, II Cohen, O Shochet. Novel type of phase transition in a system of self-driven particles[J]. Physical Review Letters, 1995, 75(6):1226.
- [14] A Jadbabaie, J Lin, AS Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules[J]. IEEE Transactions on Automatic Control, 2003,

48(6):988-1001.

- [15]REN W, BEARD R W. Consensus seeking in multi—Agent systems under dynamically changing interaction topologies [J]. IEEE Transactions on Automatic Control, 2005。 50 (5): 655-661.
- [16]Song Q, Cao J, Yu W. Second-order leader-following consensus of nonlinear multi-agent systems via pinning control[J]. Systems & Control Letters, 2010, 59(9): 553-562.
- [17]Das A, Lewis F. Cooperative adaptive control for synchronization of sencond-order systems with unknown nonlinearities[J]. Int J of Robust and Nonlinear Control, 2011, 21(13): 1509-1524.

附录 1：领导-跟随者法程序

该程序用 lua 脚本语言编写，用于 V-rep 模拟仿真。

```
// 期望值 ld 和 cd, 机器人编号 robot_id
ld = 0.04; cd = -90; robot_id = 1;

// 常量 d, 比例系数 k1 和 k2, 周期 t
d = 0.0165; k1 = 40; k2 = 0.01; t = 0.05;

// 分别对应式 (3.5) 中的变量 (从左到右)
e = 0; a = 90; b = 0; c = 0;

// 领导者和跟随者的初始速度、初始角速度
v1 = 0.003; v2 = 0; w1 = 0.5; w2 = 0;

function user_prgm()

    // id=0 表示领导者
    if(robot_id == 0) then

        // 设置 LED 灯
        set_color(3,0,0)

        // 获取自身绝对位置
        position = simGetObjectPosition(KilobotHandle,-1)

        // 获取自身绝对角度
        angle = simGetObjectOrientation(KilobotHandle,-1)

        // x,y 坐标和绕 z 轴的角度, 面向 x 轴正方向为-180°
        position_x = position[1]
        position_y = position[2]
        angle_z = angle[3] * 180 / math.pi
        if(angle_z > 0) then
            angle_z = angle_z - 180
```

```
else
    angle_z = angle_z + 180
end

// 给跟随者发送自身位置信息
message_out(position_x*100000,position_y*100000,angle_z*
    100000)

// 发送标志位
enable_tx=1

// 设置自身速度,使运动
set_motor(5100*v1+50*w1/9,5100*v1-50*w1/9)

// id>0 表示跟随者
elseif (robot_id > 0) then
    set_color(0,0,3)

    // 接收信息
    get_message()

    // 表示收到新的信息
    if(message_rx[6] == 1) then

        // 以接收到的信息为参数,调用函数 get_vw()
        get_vw(message_rx[1]/100000,message_rx[2]/100000,
            message_rx[3]/100000)
    end
end

end

function get_vw(message_rx1,message_rx2,message_rx3)

    // 领导者 x,y
    master_position_x = message_rx1
    master_position_y = message_rx2
```



```
position = simGetObjectPosition(KilobotHandle,-1)
angle = simGetObjectOrientation(KilobotHandle,-1)

// 跟随者 x,y 和绕 z 轴的角度

position_x = position[1]
position_y = position[2]
angle_z = angle[3] * 180 / math.pi
if(angle_z > 0) then
    angle_z = angle_z - 180
else
    angle_z = angle_z + 180
end

// 领导者绕 z 轴的角度
a = message_rx3

// 跟随者绕 z 轴的角度
b = angle_z

// 跟随者和领导者相对距离
l = math.sqrt((position_x - master_position_x) ^ 2
              + (position_y - master_position_y) ^ 2)

// 根据跟随者和领导者的相对位置关系, 求他们的相对角度 c (即  $\varphi$ )
local qq = math.atan((master_position_y - position_y) /
                     (master_position_x - position_x))
qq = qq * 180 / math.pi
if(master_position_x >= position_x) then
    c = qq - b
else
    c = 180 - b + qq
end

// 保证 c 的范围在 (-180,180]
```

```
if(c > 180) then
    c = c - 360
elseif(c <= -180) then
    c = c + 360;
end
kprints("l: "..l.." c: "..c)
local cc = c * math.pi / 180
// 求出 e 的值 (即  $\gamma$ )
e = b - a + c;
local ee = e * math.pi / 180;
// 根据式 (3.8) 和 (3.9) 求出跟随者的速度和角速度
p1 = v1*math.cos(ee)+d*w1*math.sin(ee)-k1*(ld-l)
p2 = d*w1*math.cos(ee)-v1*math.sin(ee)-k2*(cd-c)
p3 = v1*math.sin(ee)-d*w1*math.cos(ee)+l*k2*(cd-c)
p4 = v1*math.cos(ee)+d*w1*math.sin(ee)-k1*(ld-l)
p5 = l*v1*math.cos(ee)+d*l*w1*math.sin(ee)-l*k1*(ld-l)
p6 = d+l*math.cos(ee)
w2 = (math.sin(cc) * p1 + math.cos(cc) * p2) / p6
v2 = (math.sin(cc) * p3 + math.cos(cc) * p4 + p5) / p6
kprints("v21: "..v2.." w21: "..w2)
if(v2 > 0.1) then
    v2 = v2 / 200
elseif(v2 > 0.05) then
    v2 = v2 / 50
elseif(v2 > 0.01) then
    v2 = v2 / 10
end
kprints("v2: "..v2.." w2: "..w2)
local v1 = 5100*v2-50*w2/9
```

```
local vr = 5100*v2+50*w2/9  
set_motor(vl,vr)  
end
```

附录 2：具有通信延时的一致性协议程序

该程序用C语言编写，用于Kilobot实物验证。

```
message_t message      // 用于存储发送信息，长度为9个字节

int message_sent = 0   // 发送信息标志位

int new_message = 0    // 收到新信息标志位

float Mess0 = 0        // 存放接收到的第1个信息

float Mess1 = 0        // 存放接收到的第2个信息

int my_id = 1          // 智能体的编号

int x = 100            // 初始状态值

float a = 0.8          // 权重

float T = 0.05         // 采样周期

int t15 = 2            // 智能体收到信息的延时时间

int t21 = 3            // 智能体发送信息的延时时间

int delay_x = 0        // 延时时刻的状态值

int text_delay_x = 0   // 发送的状态值

int array_x[4]         // 存放从延时时刻到当前时刻的状态值

int k = 1;

void setup()           // 初始的设置

{

    array_x[0] = x

    spinup_motors()    // 启动电机

    delay(15)
```

```
set_motors(60+x*2/100,0) // 设置速度

set_color(RGB(0,3,0)) // 设置LED灯

delay(5000) // 延时5s

message.type = NORMAL // 设置类型为normal

message.data[0] = my_id // 赋值要传输的数据

message.data[1] = text_delay_x

message.crc = message_crc(&message)

}

void loop() {

// 收到新消息

if (new_message == 1)

{

// 确定消息来自正确的智能体

if (Mess0 == 5)

{

set_color(RGB(0,3,3))

delay(5000)

// 根据协议求出下一时刻状态值

 $x = x - 100 * T * a * (delay\_x / 100 - Mess1 / 100)$ 

// 将信息存入数组array_x

if (k <= 3)

{

array_x[k] = x

if (k - t15 < 0)

{

delay_x = 0

}

}

}

}

}
```

```
else
{
    delay_x = array_x[k - t15]
}
if (k - t21 < 0)
{
    delay_x = 0
}
else
{
    delay_x = array_x[k - t21]
}
}
else
{
    for (int i = 0; i++; i < k - 1)
    {
        array_x[i] = array_x[i + 1]
    }
    array_x[k - 1] = x
    k = 3
}
k = k + 1
spinup_motors()
delay(15)
set_motors(62 + x * 2 / 100, 0)
set_color(RGB(0, 3, 0))
delay(5000)
message.data[0] = my_id
message.data[1] = text_delay_x
```

```
        message.crc = message_crc (&message)
    }
}
message_t *message_tx()
{
    return &message
}
void message_tx_success ()
{
    message_sent = 1
}
void message_rx(message_t *message, distance_measurement_t
*distance_measurement)
{
    new_message = 1
    distance = estimate_distance (distance_measurement)
    Mess0 = message->data[0]
    Mess1 = message->data[1]
}
int main() {
    // 初始化硬件
    kilo_init()
    debug_init()
    // 注册message_tx回调函数
    kilo_message_tx = message_tx
    // 注册message_tx_success回调函数
    kilo_message_tx_success = message_tx_success
```

```
// 注册message_rx回调函数  
kilo_message_rx = message_rx  
  
// 开始程序  
kilo_start(setup, loop)  
return 0  
}
```


致 谢

四年，说短不短，说长不长。期间有太多的欢笑、汗水、遗憾，但一切终将结束。借此，我由衷的感谢这四年里给过我帮助、关心和支持的老师和同学，是他们带给了我一个独一无二的大学。

首先我要感谢的是我的毕业设计指导老师赵云波老师，赵老师有着丰富的教学经验和科研经验。从开始选题、开题、中期和最后的答辩，赵老师一直对我认真负责。每次周会，都仔细讨论目前的工作，以及接下来要完成的事情，给了我许多建议，开拓了我的思路。这是一个培养自己各方面能力的过程，无论将来做什么工作，这都将成为宝贵财富。

同样也感谢袁征学长，每当我毕设过程中遇到困难求助于学长时，他都会耐心地解答我的疑问，在软件操作以及实物调试方面都给予了我极大的帮助。非常感谢学长牺牲牺牲自己的时间为我解惑。

最后，特别感谢的是我的父母，是他们辛苦地工作和支持，让我可以安心地学习。他们是我前行的动力。

这个四年是唯一的，它是我人生中一份无可替代的财富。