



## 引言

车联网技术源于物联网技术和移动互联技术,随着通信技术的发展,特别是5G技术的出现,车联网领域已成为当今社会发展的新热点。作为车辆与交通技术发展的重要方向,同时作为现代智能交通系统的重要组成部分,车联网的发展将给汽车产业带来巨大的革新与颠覆。

本章首先对车联网技术做简单的概述,其中包括车联网的概念及内涵、车联网的要素和关键技术,以及车联网的发展历史;然后展开介绍车联网的关键技术:网联通信技术和关键的理论知识。在网联通信技术小节,主要介绍 DSRC (dedicated short-range communication, 专用短程通信)技术和 C-V2X (cellular V2X, 蜂窝车联网)技术以及具体的应用;在关键理论知识小节,介绍信息论、图论、博弈论这三种对于车联网领域十分重要的基础理论。信息论将主要讲述信息论的基础知识以及如何建立网联信道模型,图论将介绍图论的基础知识以及如何用图论来建立网联车辆的通信模型以及冲突模型,博弈论将介绍博弈论的基础知识以及如何建立网联车辆的博弈模型。本章的框架结构如图 3-1 所示。

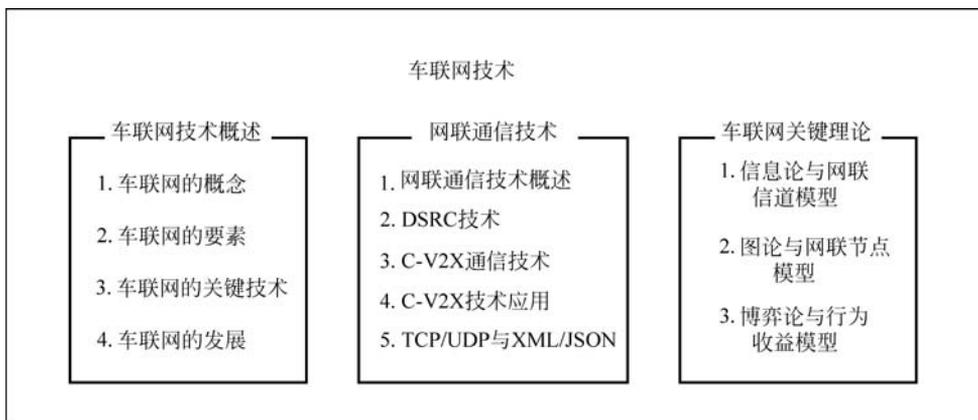


图 3-1 本章框架结构

## 学习目标

- 了解车联网技术的概念、内涵、要素、关键技术以及发展历史。
- 了解车联网中网联信道与通信技术,对比中西方车联网发展路线与优缺点。
- 了解信息论的基础知识以及如何建立非理想通信下的信道模型与控制方程。
- 了解图论基础知识以及如何建立网联车辆通信、冲突模型。
- 了解博弈论基础知识以及如何建立网联车辆的行为收益模型。



第 3 章学习素材

### 3.1 车联网技术概述

车联网能够通过车-车通信、车-路通信、车-服务平台实时联通以及网联协同技术,方便有效地实现交通管理、规划与控制,从而提高驾驶人与行人的安全性,提高交通运行效率,减少环境污染。总而言之,车联网作为一个有潜力的新兴产业,对智慧交通的构建与产业升级具有非常重要的意义。本节首先介绍车联网的概念、与几个常见相关概念的关系与异同点;其次,详细介绍车联网的组成要素与功能;最后,阐述车联网的发展与趋势。

#### 3.1.1 车联网的概念

车联网是以车内网、车载移动互联网和车际网为基础,按照约定的通信协议和数据交互标准,在车-X(X:车、路、行人及互联网等)之间,进行无线通信和信息交换的大系统网络。其中,车内网是指通过应用成熟的总线技术建立一个标准化的整车网络;车载移动互联网是指车载终端通过 4G/5G 等通信技术与互联网进行无线连接;车际网是指基于 DSRC/LTE-V 技术的“车-车”“车-路”无线局域网。三网通过光纤等媒介与基础数据中心相连接,以能够实现智能交通管理、智能动态信息服务和车辆智能化控制。

#### 3.1.2 车联网的要素

##### 1. 整体架构

车联网主要由终端、中间层和云平台三大要素构成,其整体架构如图 3-2 所示。终端是具有车内通信、车-车通信、车-路通信等的泛在通信终端,通过车载和路侧智能传感器采集与获取车辆的智能信息,感知行车状态与环境,基于通信与中心计算实现车与路的协同控制;中间层主要包括高精地图与定位功能,以及软件、硬件的中间件,用于解决车与车、车与



路、车与网、车与人等的互联互通,实现车辆自组网及多种异构网络之间的通信与漫游,在功能和性能上保障实时性、可服务性与网络泛在性,同时它也是公网与专网的统一体;云平台是一个云架构的信息平台,是多源海量信息的汇聚,包括虚拟化、安全认证、实时交互、海量存储等云计算功能,其应用系统也是围绕车辆的数据汇聚、计算、调度、监控、管理与应用的复合体系。



图 3-2 车联网整体架构

## 2. 路侧基础设施

路侧基础设施通常布置于路侧杆件上,主要包括路侧单元(road side unit,RSU)、路侧计算单元(roadside computing unit,RCU)和路侧感知设备(如摄像头、毫米波雷达、激光雷达)、交通信号设施(如红绿灯)等,以实现车路互联互通、环境感知、局部辅助定位、交通信号实时获取等功能。

## 3. 通信网

车联网系统的通信网包括无线接入网、承载网和核心网等。云控系统集成异构通信网络,使用标准化通信机制,实现智能网联汽车、路侧设备与三级云的广泛互联通信。无线接入网提供车辆与周边环境的多样化通信能力,实现车与路侧基础设施(V2I)通信、车-车直通信(V2V)通信、车与人(V2P)通信、车与网络(V2N)/边缘云的通信。路侧设备与云控基础平台各级云由多级有线网络承载。利用5G、软件定义网络、时间敏感网络、高精度定位网络等先进通信技术手段实现互联的高可靠性、高性能与高灵活性。

#### 4. 车辆及其他交通参与者

不同网联化和智能化等级车辆是数据采集对象和服务对象。在数据采集方面,对于具有联网能力的车辆,系统既可以直接通过车辆网联设备采集车辆动态基础数据,也可以间接通过路侧智能感知获得车辆动态数据;对于不具有网联能力的车辆,则间接通过路侧智能感知获得车辆动态数据。在云控服务方面,对于 L3 级及以上驾驶自动化等级的车辆,可以直接接收云控平台输出的协同决策与控制数据,再由其车载智能计算平台或控制器做出响应;对于 L2 级及以下驾驶自动化等级的车辆,间接接收云控平台输出的协同决策数据,再由其车载人机交互平台接收决策,并由单车或驾驶人完成控制;与车辆类似地,其他交通参与者包括行人、骑行者等。可以通过路侧智能系统采集其他交通参与者位置与速度信息,也可以通过在基础平台已注册的其他交通参与者所携带的定位设备采集其位置与速度信息,并基于云端融合感知向这些已注册的其他交通参与者提供安全预警服务。

### 3.1.3 车联网的关键技术

车联网技术是智能网联汽车的技术支撑,实现 V2X 信息交互,将无人驾驶的概念与车联网有效结合对智能网联汽车的发展是十分有帮助的。这就要求车联网需要包括如下关键技术。

#### 1. 融合感知技术

智能网联汽车需要大量的数据支持,这些数据由车载的各种传感器进行采集。传统意义上自动驾驶通常使用车载计算单元对单车采集数据进行处理。而在车联网环境下,车辆以及路侧传感器共同采集数据,最终传输给中央处理器进行信息的融合、筛选、处理。各类传感器通过采集系统组成一个庞大的数据采集系统,动态采集一切车联网所需要的数据,如车辆位置、交通状况、障碍信息环境参数等,经过计算机处理后传输到后续功能模块。

#### 2. 边缘云技术

低时延与高可靠性是车联网技术最主要的两个需求。在时延需求上,辅助驾驶要求 20~100ms,而自动驾驶要求时延低至 3ms。边缘云是在现有移动网络中能够实现低时延需求的技术之一。边缘云通过将本地云平台下沉在基站侧,为移动终端提供低时延业务。在紧急情况时,通过 LTE 蜂窝网络和 MEC(multi-access edge computing,多接入边缘计算)车联网平台的本地计算,给车载单元(on board unit, OBU)发出警告等信息,与现有网络相比,车-车时延可降低至 20ms 以内,大幅度减少车主反应时间。此外,通过 MEC 车联网平台还可实现路径优化分析、行车与停车引导、安全辅助信息推送和区域交通服务指引等功能。

#### 3. 通信技术

以 5G 网络为代表的通信弥补了传统移动通信网络存在的传输带宽不足、网络时延较大等缺陷,具有高速度、低时延等优点。5G 车联网与自动驾驶结合,可显著降低系统响应的时间,进一步提升整车的性能,提高信息传输的精准性,以及降低对高精度传感器的依赖,从



而降低成本。同时 5G 网络为无人驾驶和车联网技术提供了更广阔的平台,能够有效提高无人驾驶的智能化和探测的精准度,从而降低交通事故的发生率。

#### 4. 信息安全技术

作为低时延、高可靠通信的重要应用,车联网的信息安全问题同样受到重视。随着车联网应用范围的不断扩大,安全攻击也相应增多。在车联网“端-管-云”的基本网络架构下,每个环节都是信息安全的防护重点。车联网产业链较长,涉及终端设备、通信设备以及云端管理和服务平台,涉及的厂商有元器件供应商、设备生产商、整车厂商、软硬件技术提供商、通信服务商、信息服务提供商等,包括控制安全、数据安全、功能安全等各个方面。

#### 5. 资源调度技术

车联网系统需要运行大量应用以服务于智能网联汽车及交通系统各种场景。为消解高并发下各应用在资源使用上的冲突和物理世界车辆行为的冲突,系统要根据应用对实时性、通信方式、资源使用与运行方式等方面的要求,选择服务的运行地点及所分配的资源,保障服务按需、实时可靠地运行,保障所服务车辆的行车安全。相关技术工作包括:以平台统一管理或自行管理的方式进行负载均衡、生命周期管理,并利用领域特定的规则引擎按需调用云端车辆感知共享、增强安全预警、车辆在线诊断、高精度动态地图、辅助驾驶、车载信息增强以及全局协同等资源。

### 3.1.4 车联网的发展

从时间上来看,车联网的发展最早可追溯到 20 世纪 60 年代,日本率先开始车间通信的研究,并于 20 世纪 80 年代开发了车路通信系统(road automobile communication system, RACS)。随后美国和欧洲也迅速跟进,在 20 世纪 90 年代末期和 21 世纪初展开了多个车联网项目,如 2000 年的欧洲 FleetNet 项目、2004 年美国 VII 项目、2006 年欧洲车路协同 CVIS 项目等。我国在 2007 年颁布了基于 DSRC 的中华人民共和国国家标准《电子收费 专用短程通信 第 1 部分:物理层》(GB/T 20851.1—2019),是国内最早的车联网相关文件,随后清华大学智能车路协同 863 项目于 2009 年正式启动,同年,多家企业推出了车载信息服务系统,因此 2009 年也被称为车联网元年。2010 年我国首次提出了“车联网”概念,成立了车载信息服务产业应用联盟。在随后的十年内,中国车联网产业发展迅速,在北京、上海、广州、重庆、长沙等地区接连建立了智能网联汽车示范基地。目前,结合先进的 5G 通信技术和广大的内需市场,国内车联网产业仍有巨大的发展潜力。

车联网演进大致可分为三个阶段:第一阶段是车企主导的功能性车载信息服务阶段,基于无线网络通信技术,以车为主体,利用无线通信技术,与外部网络相连接,实现车与外在信息资源之间单向或双向传输的互动系统,从而实现远距离和车辆的信息交互与业务管理。第二阶段是智能网联服务阶段,主要参与者为车辆驾驶人、整车厂和图商。如图 3-3 所示,车载传感器信息上传云端,更新地理信息,同地图+地理+交通信息相互匹配,控制车辆将采取适应前方道路的驾驶模式和驾驶速度。第三阶段是智慧出行服务阶段,在该阶段将实现完全自动驾驶,网联化技术用于自动驾驶将成为行业共识。目前正处于智能网联服务阶

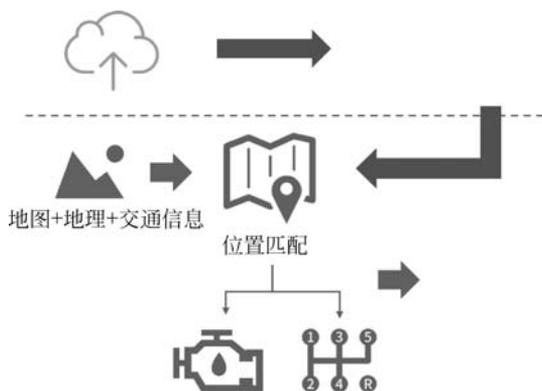


图 3-3 智能网联服务示意图

段,随着 LTE-V2X 技术的不断突破,有望实现 L3/L4 级的自动驾驶。

总而言之,目前车联网的发展已经初具规模,并且形成了完整的发展路径。在未来,车联网发展将呈现如下三大趋势。

(1) 智能网联汽车将成为主流产品。车-路、车-车网联协同系统已经成为现阶段各国发展的重点,也已成为市场竞争制胜的关键因素。我国在智能汽车领域的基础技术、研发水平、相关产业链基础还相对薄弱,目前仍处于辅助驾驶阶段,未来将逐步向部分自动驾驶、高度自动驾驶和无人驾驶过渡。

(2) 车联网与大数据云计算相融合。车联网能够将车辆本身信息、车辆位置信息、驾驶人信息、天气情况、交通状况等数据搜集起来,通过大数据分析,能够获取深层次的洞察。此外,基于车联网对于数据处理实时性的要求,车联网移动云服务也将得到广泛应用。

(3) 跨界合作和服务创新日益显著。随着生态系统的健全,除了智能网联汽车本身的技术之外,车联网将提供更加多样化的服务,包括先进的人机交互技术等,并向 O2O(online to offline)与汽车后市场渗透,跨界合作和服务创新将日益显著。

## 3.2 网联通信技术

### 3.2.1 网联通信技术概述

目前,车联网的网联通信标准主要包括专用短程通信(dedicated short-range communication, DSRC)和蜂窝车联网(cellular V2X, C-V2X)两大标准。

DSRC 是一种高效、专用的车辆无线通信技术,于 1998 年由美国提出,它是以 IEEE 802.11p 为基础扩充而来。DSRC 协议栈如图 3-4 所示。DSRC 是智能交通系统领域中专门用于 OBU 与 RSU 的通信技术,其中,RSU 通过车-路信息传输与 OBU 进行双向通信,以获取当前交通信息。同时,RSU 还与云端平台相连接,通过光纤或者行动网络进行信息传输。在车联网环境中,通信速度和通信质量将直接影响行车安全,而由于车联网环境十分复杂,通信速度和质量往往受到汽车数量、通信距离、恶劣天气等影响,这就要求车联网安全应用相关通信网络必须具备高移动性和低延迟率。根据 IEEE 规定,安全应用通信延迟容许



范围在 50ms 内,最多不超过 100ms,允许接收信息后有足够的反应时间。美国高速公路安全管理局力推 DSRC,目标是为消费者提供安全、效率、便捷三大方面的优质服务。安全方面,中轻型车辆将避免 80% 的交通事故,重型车辆避免 71% 的事故;效率方面,交通堵塞将减少 60%,短途运输效率提高 70%,现有道路通行能力提高 2~3 倍;便捷方面,停车次数可减少 30%,行车时间降低 13%~45%,实现降低油耗 15%。

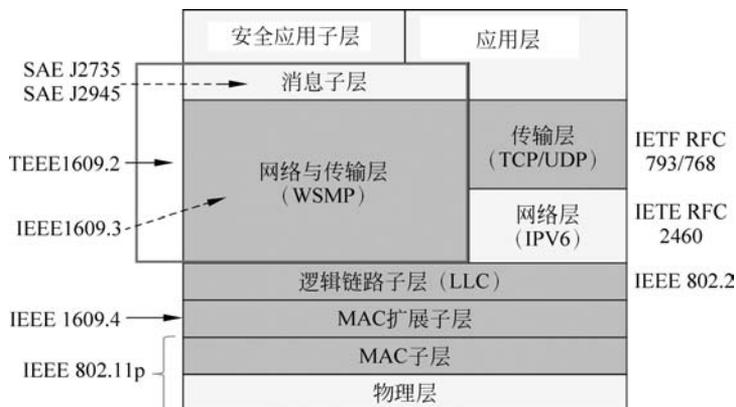


图 3-4 DSRC 协议栈示意图

在标准进程与导入方面,DSRC 发展较成熟,美国、欧洲等国家和地区已提出相关标准规格,LTE-V 目前已在 3GPP 进入标准制定流程,在布建上 DSRC 由于需要安装新的路侧设备,将增加导入成本与时间。LTE-V 则能够整合既有的基地台装置,不需要大量布建新基础建设,可缩短导入时间,两者之间互有优势。在技术层面上,根据福特在 2018 年 4 月的 5GAA 会议上发布的测试报告,福特联合大唐和高通两家企业进行了实际道路性能测试。结果表明,保证测试环境相同的情况下,当通信距离大于 400m 小于 1200m 时,LTE-V 相比于 DSRC 系统误码率有明显降低,并且在通信性能的可靠性和稳定性方面也有明显的优势。在持续演进和发展层面,C-V2X 向前包含了 Rel-14 LTE-V2X 和 Rel-15 LTE-V2X,向后包括了 NR-V2X,而 DSRC 的演进路线相对来说并不明确。在商用层面上,DSRC 发展时间较久,并且美国政府治理力度也很大。根据政策,美国希望 2022 年能够有 70% 的新生产车辆装备有 DSRC 系统,2023 年则达到覆盖率 100%。此外,DSRC 经过多年的测试验证,其可行性有保障。相对来说,C-V2X 潜力更高,更多的是后发优势。

目前,DSRC 产业链更为成熟,但 LTE-V 可能后来居上。在具体国家市场,要结合该国在本领域的基础与技术领导地位,如美国-DSRC 主导、中国-C-V2X 主导,未来可能形成两者共存的局面。

为发挥标准在车联网产业生态环境构建中的顶层设计和引领规范作用,工业和信息化部联合国家标准化委员会共同组织制定了《国家车联网产业标准体系建设指南》系列文件,包括总体要求、智能网联汽车、信息通信和电子产品与服务分册,并相继正式印发。其中信息通信和智能网联汽车分册分别从通信技术演进和智能网联汽车应用的角度明确了 LTE-V2X 和 5G-V2X 的技术选择标准。国内各标准组织的相关标准化工作已支持形成我国 C-V2X 标准体系,包括应用定义及需求、总体技术要求、关键技术、信息安全等多方面。但是,大部分标准是分散在不同的团体组织或行业标准化委员会内来开展研究制定的,仍然

需要相互之间的统筹协调,加快推进形成体系完整的统一国家标准。

## 3.2.2 DSRC 技术

### 1. DSRC 基础

DSRC 协议主要由 IEEE 802.11p 与 IEEE 1609 两个协议族的若干协议组成。其中 IEEE 802.11p 构成 DSRC 协议的底层,即物理层与介质访问控制层(media access control, MAC)层,IEEE 1609 族构成 DSRC 协议的中层。DSRC 的高层则是由应用层协议 SAE J2735 和 SAE J2945.1 组成。

DSRC 协议的最底层是物理层,物理层主要考虑如何在不同传输媒体上进行数据传输,确定传输媒体接口特性和物理参数,尽可能规避传输媒体或者通信手段之间的差异,使上层的数据链路层只需要完成本层的协议和服务。物理层还需要提供数据传输、同步、定时功能,以实现连接、数据传输以及传输方式转换。此外,物理层还提供了信道切换与选择功能,并定义了上/下层链路参数和 DSRC 协议物理层与数据链路层接口参数。

DSRC 协议的中间层为数据链路层,该层负责信息的可靠传输,提供差错控制和流量控制,使之对上层表现为一条无差错链路,定义通信帧的具体结构,提供实现相应功能的程序和程序单元。该层分为 LLC(逻辑链路控制)子层和 MAC(介质访问控制)子层。

DSRC 上层为应用层。应用层用于解决具体应用问题,通过不同主机、不同进程之间通信来协同完成,因此其最主要的功能为定义通信规则,并为这些 DSRC 应用提供通信接口,包括传送内核(T-KE)、广播内核(B-KE)、初始化内核(I-KE)。其中,T-KE 负责在两个用户之间传送信息;I-KE 主要根据规定提供有效服务,进行通信的初始化。

### 2. DSRC 关键技术

#### 1) 信道复用技术

信道复用是指利用一条信道同时传送多个不同数据流的方法,以提高传输线路的利用率。一般来说,RSU 对于数据的处理能力是要远大于 OBU 的,也就是说,RSU 可以同时处理多个 OBU 数据。要想实现上述功能,通信协议的物理层和数据链路层必须提供相应的多址访问控制机制,以支持多个 OBU 能够在同一时间段内访问 RSU。信道复用技术大体可分为频分复用和时分复用两种。

#### (1) 频分复用。

频分复用是让所有用户在同一段时间内占用不同的带宽资源的一种复用技术。由于公共信道所具有的频带要宽于传输信号,故可以将公共信道的频带划分为几个比较窄的频带,并单独构成一个信道给用户分配传输信息。当用户被分配到一定频带之后,会在通信过程中一直占用这个频带。经过上述处理,一条频带宽度固定的公共信道就可以传输由不同用户产生的互不相关的信号组成的复合信号。

根据中华人民共和国国家标准《电子收费 专用短程通信 第 1 部分:物理层》(GB/T 20851.1—2019)的定义和规范,物理层包括了上行链路和下行链路两个通信信道,物理层每条链路应包括 A 类和 B 类两类要求,分别对应 ETC 应用和在 ETC 应用基础之上的数据传



输应用。两类应用都规定了信道1的载波频率：上行5.790GHz、下行5.830GHz；信道2的载波频率：上行5.800GHz、下行5.840GHz。信道占用带宽 $\leq 5$ MHz，相同方向的相邻信道间相差大约5MHz，同一信道上行链路和下行链路载波频谱之间间隔大约40MHz。不同频率的信道往往用在不同的区域盖上，这里指不同的车道。在实际应用时，为实现频分复用，可以让相邻车道使用不同信道，并交替设置信道。这样在理想情况下，即使是相邻信道也不存在互相干扰，问题的重点就转移到了每条车道上的RSU如何处理其覆盖区域内的其他OBU。

### (2) 时分复用。

与频分复用类似，时分复用是让所有用户在不同时间占用同样频带宽度的一种复用技术。这种技术将若干用户设备接到一个公共信道上，将信道在时间上划分为一段一段等长的时间段，称之为时分复用帧。每个使用这条信道的用户在每个时分复用帧上占用固定序号的时隙。每个用户所占用的时隙周期性地出现，出现周期即为时分复用帧的长度。当某个设备开始工作时，这个设备就会和公共信道连接，并断开其他设备的使用，直到下一个时分复用帧的时隙起始时刻，将信道的使用权分配给下一个设备。等到分配完所有设备，即时间走过了一个时分复用帧，然后重新开始。这种复用技术保证了不同用户信号传输时可以互不干扰。

上述时分复用方法称为同步时分复用。但实际的信道传输过程往往并不是理想的，当某一个用户在某个时隙由于各种原因无法传输数据或者无数据传输时，只能让这段时隙处于空闲状态，其他用户无法使用这段资源，导致信道利用率不高。异步时分复用又称为统计时分复用，是一种能够明显提高信道利用率的时分复用技术，其核心思想在于动态地按需分配时隙。复用器无须为每个连接的用户设备分配专门的时隙，而是各个用户当需要传输数据时，将数据发往集中器的输入缓存上，集中器将会按顺序依次扫描输入缓存，将数据放入时分复用帧中，对于没有输入的缓存将直接跳过，直至一个时分复用帧数据已满，进行发送。因此，异步时分复用需要缓冲数据存储、寻址、控制等复杂技术，但是却可以大大提高信道的使用效率。

### 2) 协议配置

前面已经说过，根据《电子收费 专用短程通信 第1部分：物理层》(GB/T 20851.1—2019)的规范，物理层包括上层和下层两组信道，每个信道对应了A、B两类要求，其他参数都可以根据类别和信道得到。这样，将物理层配置分为A类信道1的配置、B类信道1的配置、A类信道2的配置、B类信道2的配置，配置都采用8位二进制编码。物理层载波频率、杂波发射等要求主要参考《关于使用5.8GHz频段频率事宜的通知》，并定义了频分双工通信方式。

数据链路层主要由《电子收费 专用短程通信 第2部分：数据链路层》(GB/T 20851.2—2019)所规范，主要规定了数据链路层的主要参数、信息帧以及MAC子层和LLC子层的要求。MAC子层主要提供下行链路、上行链路、公共上行链路、专用上行链路几个链路窗口的时间窗口的申请和分配。为了在实际应用中降低设备复杂度，减少或避免应用层干预，做出了如下规定。

(1) 设备应该支持确认和非确认命令，非确认命令请求可以由RSU和OBU主动发起，但确认命令请求只能由RSU主动发起。

(2) 专用链路时间窗口只能由不带链路协议数据单元(link protocol data unit,LPDU)的数据链路层帧申请,并且只能通过公用上行链路时间窗口申请。

DSRC 应用层协议配置由《电子收费 专用短程通信 第 3 部分:应用层》(GB/T 20851.3—2019)进行规范,主要规定了初始化内核、传送内核以及广播内核所提供的服务。每个服务包括 request、indication、response、confirm 四个原语,落实到具体服务通过“命令、原语”进行使用,对应规定如下。

(1) 设备应该支持确认和非确认的请求,但请求只能由 RSU 主动发起,响应只能由 OBU 完成。

(2) 不支持 Dsrc-EID、DSRC Application Entity ID、Profile、Action Type 和 FileID 等数据取值范围和列表长度的扩展。

(3) ST 中的 ApplicationList 应该包含一个 AID 序列,但不列出 EID 和其他相关参数;VST 中的 ApplicationList 应该包含 AID、EID 和其他参数序列。

### 3. DSRC 技术应用

一套 DSRC 系统由车上装备的 OBU、路侧部署的 RSU,以及专用短距离无线通信协议三部分组成。其中,OBU 类似移动终端,相比之下 OBU 的通信方式不同,且处理能力更强;RSU 指固定的路侧通信设备。DSRC 系统一般具有车-路通信和车-车通信两种通信模式,其中车-路通信常见采用基于单跳的 Ad Hoc 网络模型,车辆间通信常见采用的是基于多跳的 Ad Hoc 网络模型,车、路通信自治,无须集中协调。

#### 1) 车-路通信

现有基于 DSRC 的车-路通信主要面向车辆的非安全性应用。最具代表性的就是 ETC 系统,其主要利用了安装在车辆前挡风玻璃上的电子标签和收费处的微波天线进行专用短程通信,实现自动收费和调度功能。

除此之外,基于 DSRC 的车-路通信还广泛应用于电子地图下载以及交通调度等功能,OBU 通过与 RSU 建立通信关系,RSU 与交通网络或者互联网连接,从而获得电子地图信息或者局部交通信息。

#### 2) 车-车通信

车-车通信是车辆安全性应用的主要通信方式。将 DSRC 技术应用于交通安全领域能够提高交通的安全系数,减少交通事故的发生,降低直接和非直接的经济损失,以及减少地面交通网络的拥塞。主要包括了两类场景:第一种为环境感知,即传感器收集有关其他车辆和障碍物的位置信息、速度信息、朝向信息等,并将这些信息以定期的方式发送至邻近的车辆上;第二种为预警消息,如当自车发生故障或者前方发生意外而导致需要紧急制动时,需要将此次事故迅速地传播给后方跟随的车辆,提醒它们前方存在危险区域。此外,车-车通信还应用于弯道控制、队列管理和安全超车等功能。

## 3.2.3 C-V2X 通信技术

### 1. C-V2X 通信基础

车联网和智能网联汽车概念的提出和技术的推进,给汽车和交通行业引入了新的革命



性理念。2015年,3GPP在Rel.14版本中启动了基于LTE系统的V2X服务标准研究,即LTE-V2X,国内多家通信企业(华为、大唐、中兴)参与了LTE-V标准的制定和研发。2016年9月,首版涵盖了V2V和V2I的V2X标准发布;2017年6月,进一步增强型V2X操作方案发布。在Rel.14中,V2V通信基于D2D(device-to-device)通信,其为Rel.12和Rel.13版本中的Proximity Services(ProSe)近距离通信技术的一部分。新的D2D接口被命名为PC5接口,以实现可支持V2X要求的增强型功能,这些增强型功能包括:支持高达500km/h的相对车速、支持eNB覆盖范围内的同步操作、提升资源分配性能、拥塞控制和流量管理等。

在Rel.14中,LTE-V2X主要有两种操作模式:通过PC5接口点对点通信(V2V)和通过LTE-Uu与网络通信(V2N)。基于PC5接口的V2V通信也包括两种模式:管理模式(PC5 Mode 3)和非管理模式(PC5 Mode 4),当网络参与车辆调度时称为管理模式,当车辆独立于网络时称为非管理模式。在非管理模式下,基于车辆间的分布式算法来进行流量调度和干扰管理;在管理模式下,通过Uu(user to network interface/universal)接口的控制信令由基站(eNB)辅助进行流量调度和干扰管理。

C-V2X是LTE技术的进一步延伸,它是专门为车辆间通信而设计的一种V2X通信协议标准,相比之下,C-V2X容量更大,可靠性更强。C-V2X中,V2V网络主要包括演进分组核心网、演进地面无线接入网和车载设备三部分。其中,演进分组核心网和演进地面无线接入网仍然使用LTE的架构协议。对于演进地面无线接入网,eNode在网络层传输采用IP协议,通过X2接口进行连接,并可以实现越区切换数据包和信令。而车载设备则在LTE架构的基础上,将自身作为移动收发设备,通过空口和基站连接。由于V2V网络整体上仍然沿用LTE的网络结构设计,其实现只需要升级现有的网络设备和安全机制即可。eNode架构全称为Evolved Node,即演进型Node,顾名思义,它是Node在移动通信系统层面上的演进。相比于Node,eNode与多个移动管理实体和业务网关进行连接,增加了无线网络控制器的物理层、媒体接入控制子层等功能。此外,eNode还可以进行移动性管理和无线资源管理,换句话说,eNode可以实现接入网的所有功能。

目前基于LTE的V2N已经覆盖了很多车联网应用场景,如交通信息提示、地图更新、OTA固件更新。未来V2V和V2I将广泛应用于车联网的低时延、远距离通信场景,C-V2X依托成熟的蜂窝网络生态,将在未来自动驾驶领域发挥关键的作用。

## 2. C-V2X 关键技术

C-V2X具备清晰的5G演进路径,并支持前向及后向兼容。与4G技术相比,5G数据传输的速率和网络时延提高了10倍,而传输时间间隔(TTI)却从1ms降到了0.2ms,网络的容量和基站的密度也都大幅增加。5G需要具备比4G更高的性能,支持0.1~1Gb/s的用户体验速率,100万/km<sup>2</sup>的连接数密度,毫秒级的端到端时延,10Tb/s/km<sup>2</sup>的流量密度,500km/h以上的移动性和数十Gb/s的峰值速率。其中,用户体验速率、连接数密度和时延为5G最基本的三个性能指标。同时,5G还需要大幅提高网络部署和运营的效率,相比4G,频谱效率提升5~15倍,能效和成本效率提升百倍以上,性能需求和效率需求共同定义了5G的关键能力。

### 1) 基于正交频分复用优化的波形和多址接入

5G NR设计过程中最重要的一项决定,就是采用基于OFDM(orthogonal frequency

division multiplexing, 正交频分复用)优化的波形和多址接入(Optimized OFDM-based waveforms and multiple access)技术,因其可扩展至大带宽应用,而具有高频谱效率和较低的数据复杂性,因此能够很好地满足 5G 要求。OFDM 技术可实现多种增强功能,例如,通过加窗或滤波增强频率本地化、在不同用户与服务间提高多路传输效率,以及创建单载波 OFDM 波形,实现高效上行链路传输。

OFDM 系统把高速数据流分散到各并行的正交子载波上传输,使得每个用于去调制子载波的数据符号周期扩大为原始数据符号周期的  $N$  倍,因此时延扩展与符号周期的比值亦同样降低  $N$  倍。若从时域分析,OFDM 可以表示为式(3-1)所示。

$$x(t) = \operatorname{Re} \left\{ \sum_{i=0}^{N-1} d_i \operatorname{rect} \left( t - t_s - \frac{T}{2} \right) \exp \left[ j2\pi \left( f_c + \frac{i}{T} \right) (t - t_s) \right] \right\} \quad (3-1)$$

其中:  $N$  为子载波个数;  $T$  表示 OFDM 符号的时间周期;  $d_i$  是每个子载波的数据符号;  $f_c$  是第零个子载波的载波频率;  $t_s$  是 OFDM 符号的起始时间。若对其中一子载波数据进行解调,即在时间周期内进行积分,则根据指数函数的积分准则,所得的解调结果仍然是原数据符号。

## 2) 软件定义网络

软件定义网络(software defined network, SDN)是一种新型的网络架构,也是一种设计思想。SDN 并不是一种具体的技术,而是一种广泛的概念,即网络硬件可以通过集中控制器的软件平台进行软件管理、可编程化控制底层硬件、分离网络控制平面和数据转发平面的网络都可以称为 SDN。通过 SDN 可以实现对网络资源的按需分配。在 SDN 中,网络设备实现了与其他业务解耦,只负责数据转发。控制器中的操作系统为独立的网络操作系统,对不同业务、不同网络协议进行统一处理和适配。更重要的是,网络的操作系统和服务更加虚拟化,即可以通过编程实现。与传统网络相比,SDN 具有如下能力。

(1) 控制和转发分离。网络架构被拆分为控制和转发两个平面。转发平面由受控转发设备组成,转发方式和业务逻辑在控制平面上由集中控制器进行整体调度,并将信息发送回转发平面。

(2) 开放接口。可以通过控制平面所提供的开放接口进行编程,实现网络的集中管理,无须关注底层实现原理和细节。

因此,相比于传统网络,SDN 具有更好的扩展性和开放性,方便人们进行开发和维护。此外,由于其集中控制的特性,可以简化很多协议处理流程。

## 3) 多输入多输出

多输入多输出(multi input multi output, MIMO)是指在发射端和接收端分别使用多个发射天线和接收天线,使信号通过发射端与接收端的多个天线传送和接收,从而提高信道容量或改善通信质量。MIMO 利用了多径传播(multipath propagation),在不增加频谱资源和天线发射功率的情况下,可以成倍地提高系统信道容量,显示出明显的优势,被视为新一代移动通信的核心技术。MIMO 大致分为如下两类。

(1) 空间分集:空间分集是一种通过让发射端和接收端的天线位置不同,在发射端发送相同信息,利用场强随空间位置变化而随机变化的性质,当空间位置差距足够大时,不同电磁波信号衰落的相关性极小,可看作相互独立的信号,从而实现分集的一种技术。目前在多输入多输出系统中,时分组码技术和波束成形技术是最常见的两种技术。其中,时分组码



是一种在时域上进行编码,在频域上进行共轭运算的编码形式,最基本的就是二天线的 Alamouti 方案;波束成形则是一种对各路信号加权求和而形成期望信号的一种经典方法。

(2) 空分复用:面对日益增长的流量需求,无论是大量扩建、部署新设备还是修改终端规格都需要耗费大量成本,且无法快速实现,因此空分复用成为解决该问题的关键技术和有效途径。空分复用是一种能够在不增加频谱带宽的条件下,利用自适应天线阵列间距较大,不同方向阵列波束互不影响性质,让同一频段在不同的空间内提供不同的数据流,从而大大提升频谱利用率的一种技术。由于在信道维度上增加了空间域维度,相同时间内同一频率下的波束在不同空间上有足够的区分度,使得不同发射端传送的信号之间能够相互区别,在接收端无须增加额外的频率或者时间便能够加以区分。此外,空分复用技术能够提高信道容量,并且能够在发射端无法获得信道信息的条件下使用。

#### 4) 同时同频全双工

在空中接口方面,同时同频全双工(co-time co-frequency full duplexing, CCFD)技术能够在相同频率同时收发电磁波信号,相对于现在广泛应用的时分双工和频分双工,频谱效率有望提升一倍。采用 CCFD 的无线通信系统,首先要清楚发射机和接收机的节点对于有用信号都是干扰源,要求接收点在工作时还能消除干扰,并且能够抑制干扰,大大降低对有用信号的干扰,因此 CCFD 的关键技术在于干扰的最大幅度消除,一般来说,抵消方式有如下三种。

(1) 天线干扰消除。两根发射天线和一根接收天线就是目前天线干扰消除原理的设备。当满足发射天线发射信号的波长是两根发射天线距离的半波长的奇数倍时,通过两根天线的信号会以相反的相位到达天线的接收点,这样会使干扰信号自行干涉相消,这样,接收天线收到的自干扰信号就会大大减弱,再利用噪声信号消除电路和数字域消除技术,就可以达到预期目标。

(2) 射频干扰消除。射频干扰消除的重点是调节干扰参考信号的相位和幅度,从而达到精确干扰消除的目的,这样干扰消除的重点是自适应调节。MIMO 技术可以与射频干扰消除技术同时使用,然而在多天线情况下的高阶 MIMO 难以实现。

(3) 数字干扰消除。在同频全双工系统进行数字自干扰时,自干扰的信息为已知,因此相比传统的数字对消,省去了需先解出不期望的发射机信息的过程。对数字干扰消除进行清晰的了解,可以建立一个 ADC(analog-to-digital converter,模拟/数字转换器)量化进行数字通过对消实现同频全双工的 M-QAM(multiple quadrature amplitude modulation,多进制正交幅度调制)的系统模型。

### 3.2.4 C-V2X 技术应用

车-路-云协同技术可以有效提高出行安全性和出行效率。通过将所有交通参与者,包括出行者、车辆、道路基础设施和云端有机结合,对交通参与者状态信息进行数字化采集和计算,实现不同交通参与者间的信息交互和协同管理。在大规模智能交通系统中,C-V2X 作为信息交互基础,具有重要的地位。不同场景下对于 C-V2X 的时延、通信距离、精度等指标都有不同的要求,因此基础设施部署也有所不同。目前,C-V2X 主要应用于高速公路、城市道路和自动驾驶园区三大场景。

## 1. 高速公路

目前高速公路基础通信设施相对城市环境较少,且车辆行驶速度快,车辆行驶安全要求高,因此对通信时延、通信距离有相对较高的要求。对于一般情况,应基于 V2V 通信,进行车-车之间的协同驾驶,通过协同其他车辆速度,为车辆变道、超车提供足够空间。当发生紧急事件时,例如,车辆故障、交通事故、异常驾驶行为且不能及时实施纠正时,应基于 V2V 对下游车辆进行及时预警,并向路侧或者云端的交通管控平台发送警报,及时通知进行处理,再通过路侧通信进行交通疏导。这也要求 RSU 之间有快速信息交换机制,并可针对具体事件推送相应消息。

## 2. 城市道路

城市道路特点为车速相对较低、交通参与者多、交通参与者类别多、行为关系复杂。在这种复杂的交通环境下,要充分考虑多种情况,除了对通信时延、通信距离的要求外,还对通信质量有较高的要求。例如,对于无信号路口协同通行的情况,常见通过与 RSU 相连的感知系统识别车辆状态以及车辆驾驶意图,基于当前路口整体交通流信息进行统一调度,并将规划信息和控制信号下发给通行车辆。此外,城市道路中还有许多行驶盲区问题,如倒车出库等情况,此时由 OBU 设备将当前车辆状态和意图发送给附近的 RSU,基于动态环境感知信息判断周围车辆存在冲突的可能性,则将警告信号和调度结果发送给对应车辆,消解车辆冲突。

## 3. 自动驾驶园区

自动驾驶最先可能落地的场景就是相对封闭、有相对固定路线的物流园区、港口、办公场所等。在这些场景,可以投入成本建设相对完善的基础设施,这些基础设施可以为自动驾驶车辆提供超视距感知服务和大规模数据计算服务,极大地降低自动驾驶车辆自身的成本、算力要求以及功耗,让自动驾驶车辆更加轻量化、小型化。在园区内部的自动驾驶行驶路线上,可以部署路侧传感器,从上帝视角来对交通参与者的状态进行感知,降低单车传感器的成本。同时还可以在道路两侧部署计算单元,利用边缘计算技术协同规划车辆的行驶路径,将规划后的结果通过部署在道路两侧的路侧单元发送给车辆,这样既降低了车辆的计算成本,又实现了有效协同。此外,在交叉路口等复杂的交通场景下,可以使用 5G 技术进行远程人工接管。

此外,在园区运营早期,需要对园区内所有的道路基础设施和车辆进行全方位的监督与管理,这可以通过 C-V2X 进行实现。通过构建园区内的通信网络和数字网络,快速感知并处理普通交通事件和突发事件,提供远程动态控制和应急处置功能。

### 3.2.5 TCP/UDP 与 XML/JSON

TCP/IP(transmission control protocol/internet protocol,传输控制协议/网际协议)是指能够在多个不同网络间实现的协议簇,由它构成了目前最大的开放的互联网络系统——因特网。TCP/IP 模型包括四个独立的层次,自上而下分别为应用层、传输层、互联网络层、



网络接入层。TCP(transmission control protocol,传输控制协议)和 UDP(user datagram protocol,用户数据报协议)均属于传输层协议,是 TCP/IP 的核心,这两种协议有着不同的特点与不同的应用场景。

TCP 是一种面向连接的、可靠的、基于字节流的传输层通信协议,定义了两台计算机之间进行可靠的传输而交换的数据和确认信息的格式,以及计算机为了确保数据的正确到达而采取的措施。

TCP 连接的过程称为“握手”。每实现一次 TCP 连接,都需要在客户端和服务端进行如下三次握手。

(1) 在建立 TCP 连接时,客户端向服务器发送请求报文段,首部的同步位信号 SYN=1,客户端此时进入 SYN-SENT 状态。

(2) 服务器收到请求信号后,向客户端发送同意连接的确认信号,在确认报文段中令 SYN=1,ACK=1,服务器进入 SYN-RCVD 状态。

(3) 当客户端收到确认信号后,还会向服务器返回一个确认信号,ACK=1,此时客户端进入 ESTABLISHED 状态。

值得注意的是,服务器在收到最后一次“握手”之后,也会进入 ESTABLISHED 状态。类似地,把 TCP 的释放过程称为“挥手”。每次 TCP 连接的释放,都需要在客户端和服务端进行如下四次挥手。

(1) 在数据传输结束后,客户端发送连接释放报文段,同时停止数据发送,关闭 TCP 连接,令首部终止控制位 FIN=1,此时客户端进入 FIN-WAIT-1 状态。

(2) 服务器端收到信号后向客户端发送一个 ACK 确认信号,并进入 CLOSE-WAIT 状态。此时服务器仍可以向客户端传输数据。客户端进入 FIN-WAIT-2 状态。

(3) 当服务器无数据需要传输时,服务器会向客户端发送释放报文段,并使 FIN=1,此时服务器进入 LAST-ACK 状态。

(4) 客户端收到服务器的释放信号后,在确认报文段中令 ACK=1,进入 TIME-WAIT 状态,此时服务器进入 CLOSED 状态。在经过了一段时间后,客户端进入 CLOSED 状态。

UDP 是一种面向无连接的传输层通信协议,由于传输数据不建立连接,也就不需要维护连接状态,包括收发状态等,因此可以实现多对一、一对多和一对一的信息传递。UDP 不能对数据进行分组、组装,报文发送以后,无法得知报文是否安全完整地到达,即使数据丢失或数据出错时也不会要求重新传输,这也使得 UDP 建立的连接的延迟会比以 TCP 建立的连接的延迟更低。

对比两种协议可知,TCP 具有可靠、能保证数据顺序和正确性的优点,但是速度较慢,所以主要应用于对传输精确性要求较高,对传输速度敏感度较低的场景,如密码的登录。而 UDP 具有发送速度更快、延迟更低、实时性更好的优点,更加适用于数据量大而精确度要求不高的场景,如视频、直播等。

XML(extensible markup language),全称为可扩展标记语言,是一种标准通用标记语言,用于描述网络上的数据内容和结构,其本质是包含了数据以及数据说明的文本格式规范。它继承了标准通用标记语言的大部分功能,同时保证为了用户友好性,XML 通过重新定义标准通用标记语言的一些参数和内部值,去掉一些标准通用标记语言中使用率低的功能。同时,为了使网站设计者能够定义自己的文档类型,XML 保留了标准通用标记语

言的结构化功能,XML 同时也推出了一种新的文档类型,开发者可以省去定义文档类型的操作。

举个例子,假如需传输小明的信息,内容为 Xiao Ming, eighteen, Tsinghua University, 用于描述姓名、年龄以及学校,用 xml 来表示为:

```
< person >
  < name value = " Xiao Ming " />
  < age value = " eighteen " />
  < school value = " Tsinghua University " />
</person >
```

JSON(javascript object notation)是一种轻量级的数据交互格式,采用完全独立于编程语言的文本格式来存储和表示数据,数据类型主要包括: number、boolean、string、null、array 和 object 六种。相比于 XML 格式,JSON 没有结束标签,书写更短,读写的速度更快。简洁和清晰的层次结构使得 JSON 成为理想的数据交换语言,易于人们阅读和编写,同时也易于机器解析和生成,并能有效地提升网络传输效率。

如前面例子中小明的信息,若用 JSON 格式来书写,可表示为:

```
{
  " name " : " Xiao Ming ",
  " age " : " eighteen ",
  " school " : " Tsinghua University "
}
```

由这个例子也可以看出,相较于 XML 格式,JSON 格式在编码和传输时也更为方便。事实上,尽管 JSON 格式在 Web Service 中推广还属于初级阶段,但已经展现出广阔的发展前景,未来必将成为主流数据的交换格式。

### 3.3 车联网应用技术

在网联条件下,行驶在道路上的车辆不再是相互孤立的个体,而是通过无线通信网络连接组成了多车系统。在多车系统中,网联车辆可基于车-车通信和车-路通信获取通信范围内其他网联车辆和道路的信息,并利用该信息进行分布式的决策与控制,从而实现整个系统的协同控制。其中,通信网络会影响多车系统中车辆间的信息传输关系与质量,也会影响车辆间的通信距离与通信拓扑结构,进而影响多车系统的网络化控制性能。对于该问题,可以用图论的方法来对车-车之间的通信网络进行建模分析;通过纠错编码方案进行有效、可靠的数据传输;用博弈论的方法来预测其他交通参与者的行为。本节将重点介绍图论、信息论与博弈论三大车联网的基础理论。

#### 3.3.1 信息论与网联信道模型

##### 1. 信息论基础知识

Claude Shannon(克劳德·香农)在 1948 年发表的论文 *A Mathematical Theory of*



*Communication* 的引言中提出：“通信中的基本问题就是在某一点精确或近似地再生另一点选择的信息。”在这篇论文中，提出了“熵”的概念，用不确定性来度量信息，奠定了信息论的基础。

信息论是研究信息的基本性质、度量方法及信息的有效处理和可靠传输的一般规律的科学。通过寻找信息传输过程中的共同规律，提高信息传输的可靠性、有效性、保密性和认证性，使信息传输系统最优化。通信系统的组成如图 3-5 所示，主要包括了信源、信道和信宿三大部分。信源是信息的产生和发送者，发出的信息为消息(message)，包括无失真的文本消息、数据等以及允许失真的语音、图像、影视信息等。信息(信号)传输的通道为信道，实际信道包括电缆、光纤、微波、无线通信等。存储介质包括 U 盘、光盘等。信宿是信息的接收者，信源与信宿通过信道进行连接。而信道是有容量的，即信道的最大理论信息传输速率，信道只能用低于信道容量的速率来可靠地传输信息，否则就会出现错误。此外，信道中通常会有噪声干扰，使传输的消息失真。因此，如何使信宿尽可能准确地接收到信源发送的信息是需要解决的问题。

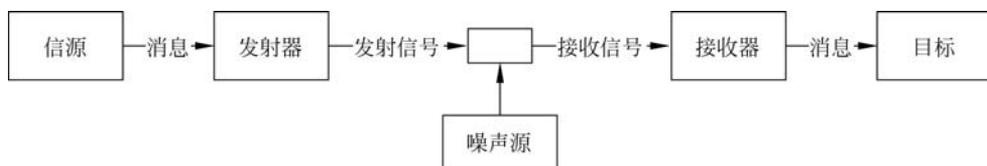


图 3-5 通信系统的组成

编码是解决上述问题的一个重要方法，主要可分为信源编码和信道编码。信源编码的作用之一是，设法减少码元数目和降低码元速率，即通常所说的数据压缩。码元速率将直接影响传输所占的带宽，而传输带宽又直接反映了通信的有效性。作用之二是，当信息源给出的是模拟信号时，信源编码器将其转换成数字信号，以实现模拟信号的数字化传输。模拟信号数字化传输有两种方式：脉冲编码调制(pulse code modulation, PCM)和增量调制( $\Delta M$ )。脉冲编码调制简称脉码调制，是一种用一组二进制数字代码来代替连续信号的抽样值，从而实现通信的方式。由于这种通信方式抗干扰能力强，它在光纤通信、数字微波通信、卫星通信中均获得了极为广泛的应用。增量调制是传输差值编码，同样可传输模拟信号所含的信息。此差值又称“增量”，其值可正可负。这种用差值编码进行通信的方式，称为增量调制，主要用于军方通信中。信源编码为了减少信源输出符号序列中的剩余度、提高符号的平均信息量，而对信源输出的符号序列施行变换。具体说，就是针对信源输出符号序列的统计特性来寻找某种方法，把信源输出符号序列变换为最短的码字序列，使后者的各码元所载荷的平均信息量最大，同时又能保证无失真地恢复原来的符号序列。

信道编码是为保证信息传输的可靠性、提高传输质量而设计的一种编码。信道编码的实质就是在信息码中增加一定数量的多余码元(称为监督码元)，使它们满足一定的约束关系，这样由信息码元和监督码元共同组成一个由信道传输的码字，使码字具有一定的抗干扰能力。

前面已经介绍，编码能够有效解决信道干扰等问题，提升信息传输的可靠性。但是如何找到这样一种编码，能够从编码定理的意义上来说性能好，同时比较容易实现，是一个需要解决的重要问题。线性码就是这样一种码，下面将详细介绍线性码的定义及性质。

定义有限域  $F_q$  和有限域上的  $n$  维线性空间  $V_n(F_q)$ , 若  $C$  为  $V_n(F_q)$  的一个子空间, 那么称  $C$  为一个  $q$  元线性码。线性码是非常重要的分组码, 是讨论各类码的基础。一般来说, 线性码的编码和译码都比较简单, 且更容易定义。线性码中有  $n$  个码元, 其中  $k$  个信息码元,  $(n-k)$  个监督码元, 可表示为  $(n, k)$  线性码。一个  $(n, k)$  线性码  $C$  完全可以由任意一组  $k$  个线性无关的码字来描述。如果将这一组码字排列成一个  $k \times n$  维矩阵  $\mathbf{G}$ , 则称  $\mathbf{G}$  为  $C$  的生成矩阵。更普遍的是: 令  $C$  是  $F_q$  上的一个  $(n, k)$  线性码, 一个行空间等于  $C$  的  $k \times n$  阶矩阵  $\mathbf{G}$  称为  $C$  的生成矩阵。相反地, 如果  $\mathbf{G}$  是元素取自  $F_q$  的一个矩阵, 则它的行空间称为由  $\mathbf{G}$  生成的码。

线性码的另一个优点是编码易于实现。一个  $(n, k)$  线性码有  $q^k$  个码字, 如果假设这些信息表示为  $q^k$  个  $k$  维  $\mathbf{u} = (u_1, u_2, \dots, u_k) \in V_k(F_q)$ , 并且  $\mathbf{G}$  中各行是线性无关的, 则将信息  $\mathbf{u}$  映射为码字  $x$  的一个映射规则是

$$\mathbf{u} \rightarrow \mathbf{uG} \quad (3-2)$$

例如, 一个  $(5, 3)$  线性码  $C$  具有生成矩阵:

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (3-3)$$

利用该生成矩阵, 将映射(3-2)变为:

$$(u_1, u_2, u_3) \rightarrow (u'_1, u'_2, u'_3, u'_4, u'_5) \quad (3-4)$$

其中:

$$\begin{cases} u'_1 = u_1 + u_3 \\ u'_2 = u_1 + u_3 \\ u'_3 = u_1 + u_2 + u_3 \\ u'_4 = u_2 + u_3 \\ u'_5 = u_3 \end{cases} \quad (3-5)$$

设  $C$  和  $C'$  是两个线性码,  $\mathbf{G}$  和  $\mathbf{G}'$  是两个生成矩阵, 若存在置换矩阵  $\mathbf{P}$  使  $\mathbf{G} = \mathbf{G}'\mathbf{P}$ , 则称  $C$  和  $C'$  是等价码。在等价意义下, 每个  $(n, k)$  线性码都有一个型为  $(\mathbf{I}_k : \star)$  的生成矩阵, 其中  $\mathbf{I}_k$  是  $k \times k$  阶单位矩阵, 称其为标准生成矩阵。在无记忆信道上, 将任意生成矩阵置换为标准生成矩阵, 其生成码的性能不发生改变。

此外, 对应每一个线性码, 都存在一个更有价值的矩阵, 称之为校验矩阵。在介绍校验矩阵之前, 首先引入形式内积的概念。

若在  $\bar{V}(n, q)$  上取一组基  $(\xi_1, \xi_2, \dots, \xi_n)$ , 则称  $\bar{V}(n, q)$  上的双线性函数:

$$(\alpha, \beta) = \left( \sum_{i=1}^n a_i \xi_i, \sum_{j=1}^n b_j \xi_j \right) = \sum_{i=1}^n a_i b_i \quad (3-6)$$

称为  $n$  维线性空间上的形式内积。对形式内积, 引入子空间  $C$  的正交补:

$$C^\perp = \{ \xi \in \bar{V}(n, q) \mid (\xi, C) = 0 \} \quad (3-7)$$

若  $C$  为  $\bar{V}(n, q)$  的一个  $k$  维子空间,  $(n, k)$  线性码  $C$  的正交补  $C^\perp$  定义了另一个线性码, 称为线性码  $C$  的对偶码,  $C^\perp$  的生成矩阵  $\mathbf{H}$  称为码  $C$  的校验矩阵。下面来求  $C^\perp$  的生成矩阵: 取定子空间  $C$  的一组基为  $\alpha_1, \alpha_2, \dots, \alpha_k$ , 由式(3-7)可知, 当且仅当  $(\xi, \alpha_i) = 0$  时,  $\xi =$



$\sum_{j=1}^n z_j \xi_j \in C^\perp$ , 即:

$$(\xi, \alpha_i) = \left( \sum_{j=1}^n z_j \xi_j, \sum_{k=1}^n a_{ik} \xi_k \right) = \sum_{j=1}^n a_{ij} z_j = 0 \quad (3-8)$$

记  $\mathbf{z} = (z_1, z_2, \dots, z_n)$ , 则式(3-8)可表示为  $\mathbf{Gz}^T = 0$ 。因此:

$$C^\perp = \left\{ \xi \in \bar{V}(n, q) \mid \xi = \sum_{j=1}^n z_j \xi_j, \mathbf{Gz}^T = 0 \right\} \quad (3-9)$$

由齐次线性方程组解的理论可知, 方程  $\mathbf{Gz}^T = 0$  的解空间是  $n-k$  维子空间, 取一组基础解系  $\eta_1, \eta_2, \dots, \eta_{n-k}$ , 定义秩为  $n-k$  的  $(n-k) \times n$  矩阵:

$$\mathbf{H} = \begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_{n-k} \end{bmatrix} \quad (3-10)$$

满足

$$\mathbf{GH}^T = 0 \quad (3-11)$$

因此, 对每个原数字信息  $\mathbf{u}$  得到的线性码字  $\mathbf{x} = \mathbf{uG}$  可知, 当且仅当  $\mathbf{Hx}^T = 0$  时,  $\mathbf{x} \in C$ , 故可以用矩阵  $\mathbf{H}$  来检验  $\mathbf{x}$  是否为  $C$  中的码字。

更直接的, 令  $C$  是  $F_q$  上的一个  $(n, k)$  线性码, 如果当且仅当  $\mathbf{x} \in C$  时, 矩阵  $\mathbf{H}$  具有性质  $\mathbf{Hx}^T = 0$ , 则称矩阵  $\mathbf{H}$  为码  $C$  的一致校验矩阵。

若  $(\mathbf{I}_k : \mathbf{A})$  为  $q$  元  $(n, k)$  线性码  $C$  的生成矩阵的标准型, 则其检验矩阵  $\mathbf{H} = (-\mathbf{A}^T : \mathbf{I}_{n-k})$ 。如果生成矩阵  $\mathbf{G}$  不具有这种形式, 则可先通过置换将  $\mathbf{G}$  转换为标准形式, 再对所得的  $(-\mathbf{A}^T : \mathbf{I}_{n-k})$  型矩阵进行逆置换得到校验矩阵  $\mathbf{H}$ 。

目前为止, 对于线性码的讨论还未涉及信道输出符号集  $A_y$  或者信道特性, 假定  $A_y = F_q$ , 即输入和输出符号集相同。因此, 若传输的是  $\mathbf{x} = (x_1, x_2, \dots, x_n) \in V_n(F_q)$ , 那么接收向量  $\mathbf{y} = (y_1, y_2, \dots, y_n) \in V_n(F_q)$ , 二者的差值  $\mathbf{z} = \mathbf{y} - \mathbf{x}$  称为错误图案, 若  $z_i \neq 0$ , 就称在第  $i$  个位置上出现了一个错误。此时, 前文所引入的校验矩阵  $\mathbf{H}$  被证明是译码的有效工具。假设传输  $\mathbf{x}$ , 则必然满足  $\mathbf{Hx}^T = 0$ , 因此当传输发生错误时, 即  $\mathbf{z} \neq 0$ , 那么很有可能  $\mathbf{Hy}^T \neq 0$ , 此时称向量  $\mathbf{s} = \mathbf{Hy}^T$  为伴随式, 它只依赖于错误图案  $\mathbf{z}$  而不依赖于所传输的码字, 即:

$$\mathbf{s} = \mathbf{Hy}^T = \mathbf{Hz}^T \quad (3-12)$$

当知道  $\mathbf{z}$  后, 可根据已知的接收向量  $\mathbf{y}$  求得传输信息  $\mathbf{x} = \mathbf{y} - \mathbf{z}$ 。

## 2. 网联信息编码

汉明码(Hamming code)和循环码是两种典型的线性码。

### 1) 汉明码

汉明码也称作海明码, 由理查德·卫斯里·汉明于1950年发明, 在通信领域有广泛的应用。汉明码是一种错误纠正码, 可以用来检测并且纠正数据从发送端发往接收端中发生的错误。

汉明码的发明者汉明在 20 世纪 40 年代晚期,运用贝尔模型 V(Bell Model V)计算机在贝尔实验室工作,由于输入端依靠打孔卡,难免会造成一些读取错误。在工作日,当机器检测到错误时将停止并闪灯,此时操作员可以解决这个错误。但在周末和下班期间,没有操作员的情况下,机器只会简单地转移到下一个工作。汉明在周末工作,他对于不可靠的读卡机发生错误后,总是不得不重新启动程序变得越来越沮丧。在接下来的几年中,他为了解决侦错的问题,开发了功能日益强大的侦错算法。时至今日,汉明码仍在 ECC(error-collecting code) memory 上显示其应用价值。

介绍汉明码之前,首先定义两个向量  $x$  和  $y$  的汉明距离如下,两个等长向量之间的汉明距离是两个向量对应位置的不同元素的个数,即:

$$d_H(x, y) = w_H(y - x) \quad (3-13)$$

式中,  $w_H(\cdot)$  表示汉明重量,即向量中元素不为 0 的个数。从另一方面看,汉明距离度量了通过替换元素的方式将向量  $x$  变成  $y$  所需要的最小的替换次数。汉明距离和汉明码的纠错能力之间有非常重要的关系。令  $C = \{x_1, x_2, \dots, x_M\}$  是码长为  $n$  的码,假设希望能够纠正汉明重量  $w_H(z) \leq e$  的所有错误图案,即当发送  $x_i$  时,接收到  $y = x_i + z$  且  $w_H(z) \leq e$ ,希望译码器的输出  $\hat{x} = x_i$ 。假设每个码字以等概率发送,那么接收方猜测发送码字的最佳策略是选出与  $y$  距离最近的码字。如图 3-6 所示,通过这种方法能够纠正所有  $w_H(z) \leq e$  的错误图案的充要条件是每一对码字的汉明距离都要大于或等于  $2e + 1$ ,换句话说,两个半径为  $e$  的汉明球体不相交。定义码  $C$  的最小距离:

$$d_{\min}(C) = \min\{d_H(x, x') \mid x, x' \in C, x \neq x'\} \quad (3-14)$$

并有定理:码  $C = \{x_1, x_2, \dots, x_M\}$  能够纠正  $w_H(z) \leq e$  的所有错误图案,当且仅当  $d_{\min}(C) \geq 2e + 1$ 。

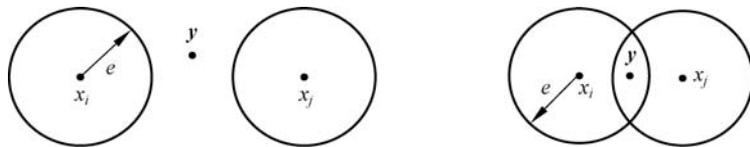


图 3-6 汉明球体示意图

当  $C$  为线性码时,对于任意  $x \neq x' \in C, x - x' \in C$  也一定是  $C$  中的非 0 码字。也就是说,线性码的最小距离和它的最小汉明重量是等价的,即  $d_{\min}(C) = w_{\min}(C)$ 。因此,要计算  $(n, k)$  线性码的  $d_{\min}$ ,只需要计算  $|C| - 1$  个汉明重量  $w_H(x)$  即可。按照前面定理计算,需要计算  $\lfloor |C|(|C| - 1) \rfloor / 2$  个码字间的汉明距离,相比之下,大大增加了计算量。

下面给出一个更为简单计算线性码  $d_{\min}$  的定理:如果  $C$  是  $F_q$  上的一个  $(n, k)$  线性码,具有校验矩阵  $H$ ,则  $d_{\min}(C)$  等于  $H$  中线性相关列的最小数目。因此如果  $H$  的任意  $2t$  及更少的列组成的子集都是线性无关的,则这个码能够纠正所有汉明重量小于或等于  $t$  的错误图案。

**证明:**  $C$  的码字是满足  $Hx^T = 0$  的所有向量  $x \in V_n(F_q)$ 。设  $H = (c_1, c_2, \dots, c_n)$ , 则  $Hx^T = x_1c_1 + x_2c_2 + \dots + x_nc_n = 0$ 。也就是说,一个汉明重量为  $w$  的码  $x$  产生了  $w$  个线性相关列,反之亦成立。



举例来说,假设一个(7,4)线性码  $C_0$  具有生成矩阵:

$$\mathbf{G}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (3-15)$$

可写出  $C_0$  的校验矩阵:

$$\mathbf{H}_0 = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (3-16)$$

现在利用上述定理确定其  $d_{\min}$ , 显然  $d_{\min} \neq 1$  或  $2$ , 因为  $\mathbf{H}_0$  各列不为  $0$  且互不相同。而当中存在许多三列组成的子集线性相关, 因此  $d_{\min} = 3$ ,  $C_0$  最多能够纠正单个错误, 即能够纠正所有汉明重量为  $0$  或  $1$  的错误图案。

$C_0$  就是一个二进制汉明码, 下面给出二进制汉明码的一般定义: 令  $\mathbf{H}$  是一个  $m \times (2^m - 1)$  阶的二进制矩阵,  $\mathbf{H}$  的列向量由  $V_m(F_2)$  以某种顺序排列的  $2^m - 1$  个非零向量组成。则在  $F_2$  上, 校验矩阵为  $\mathbf{H}$  的 ( $n = 2^m - 1, k = 2^m - 1 - m$ ) 线性码称为码长为  $2^m - 1$  的二进制汉明码, 记为  $\text{Ham}(m, 2)$ 。

显然, 对于给定的  $m$ , 由于  $\mathbf{H}$  中的列向量可以任意排列, 所以二进制汉明码  $\text{Ham}(m, 2)$  指的是一类等价码中的任意一个, 在等价意义下,  $\text{Ham}(m, 2)$  是唯一的。例如, 对于校验矩阵为  $\mathbf{H}_1$  的码长为  $3$  的汉明码, 其中:

$$\mathbf{H}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \quad (3-17)$$

在等价意义下可以将列向量重新排序, 得到标准型校验矩阵  $\mathbf{H}_0$ 。

汉明码主要有两大特点。首先非常容易实现伴随式编码, 如果错误图案  $z = 0$ , 则伴随式  $s = 0$ ; 若  $w_H(z) = 1$ , 假设  $z_t = 1$ , 则  $s = c_t$ , 即  $\mathbf{H}$  的第  $t$  列, 由此可直接确定错误的位置, 其算法如表 3-1 所示。其次, 码  $C_0$  能够纠正重量  $\leq 1$  的所有图案, 当且仅当围绕各码字的半径为  $1$  的汉明球体互不相交。但是  $V_n(F_2)$  中半径为  $1$  的汉明球体包含  $n + 1$  个向量, 因此一个能够纠正单个错误的码至多包含  $2^n / (n + 1)$  个码字。特别地, 当  $n = 2^m - 1$  时, 至多有  $2^{2^m - 1 - m}$  个码字, 恰好等于汉明码中码字的数目。因此汉明码围绕各码字的半径为  $1$  的球体恰好填满  $V_n(F_2)$  而没有重叠, 具有完美的几何性质。

表 3-1 汉明码的伴随式译码

- 
- |                                                          |
|----------------------------------------------------------|
| (1) 计算伴随式 $s = \mathbf{H}\mathbf{y}^T$                   |
| (2) 若 $s = \mathbf{0}$ , 输出 $\hat{x} = \mathbf{y}$       |
| (3) 若 $s = c_t$ , 在 $\mathbf{y}$ 的第 $t$ 个分量上加 $1$ 作为输出结果 |
- 

## 2) 循环码

循环码是另一种重要的线性分组码。

首先, 给出循环码的定义: 对于一个  $(n, k)$  线性码  $C = (C_0, C_1, \dots, C_{n-1})$ , 如果它的每

一个循环移位( $C_{n-1}, C_0, C_1, \dots, C_{n-2}$ )仍是一个码字,那么称  $C$  是一个循环码。

例如,一个(3,2)二进制码  $C$ ,具有生成矩阵:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad (3-18)$$

那么可以得到三个非零码字:

$$C^0 = 101 \quad (3-19)$$

$$C^1 = 011 \quad (3-20)$$

$$C^2 = C_0 + C_1 = 110 \quad (3-21)$$

显然,将上述每一个码字右循环移位,可以得到:

$$C^0 \rightarrow C^2 \quad (3-22)$$

$$C^1 \rightarrow C^0 \quad (3-23)$$

$$C^2 \rightarrow C^1 \quad (3-24)$$

即它的任意码字通过循环移位后仍然是码组中的码字,因此上述码字为循环码。

下面再介绍生成函数的概念。如果  $C = (C_0, C_1, \dots, C_{n-1})$  是一个码字,那么其生成函数可以定义为:

$$C(x) = C_0 + C_1x + \dots + C_{n-1}x^{n-1} \quad (3-25)$$

通过生成函数,可以给出码字右循环移位的代数描述。

在此之前,先给出 mod 运算符的定义:如果  $p$  和  $m$  是两个整数,且  $m > 0$ ,则  $p \bmod m$  表示  $p$  除以  $m$  得到的余数。类似地,若两个多项式  $P(x)$  和  $M(x)$ ,则  $P(x) \bmod M(x)$  表示  $P(x)$  除以  $M(x)$  的余式,即  $P(x) \bmod M(x)$  等于唯一能使  $P(x) - R(x)$  被  $M(x)$  整除,且  $\deg R(x) < \deg M(x)$  的多项式  $R(x)$ 。

有了上述定义,可以证明如下定理:设码字  $C = (C_0, C_1, \dots, C_{n-1})$  的生成函数  $C(x) = C_0 + C_1x + \dots + C_{n-1}x^{n-1}$ ,则右循环移位的码字  $C^R$  的生成函数为:

$$C^R(x) = xC(x) \bmod (x^n - 1) \quad (3-26)$$

因为  $C(x) = C_0 + C_1x + \dots + C_{n-1}x^{n-1}$ ,可以得到:

$$xC(x) = C_0x + C_1x^2 + \dots + C_{n-1}x^n \quad (3-27)$$

$$C^R(x) = C_{n-1} + C_0x + \dots + C_{n-2}x^{n-1} \quad (3-28)$$

则  $xC(x) - C^R(x) = C_{n-1}(x^n - 1)$ ,且  $\deg C^R(x) < n$ ,故可证明上述定理。

基于生成函数,引入生成多项式的概念:如果  $C$  是一个  $(n, k)$  循环码,那么  $C$  中的一个最低次的非零多项式被称为它的一个生成多项式,用符号  $g(x)$  表示,并且  $k = n - \deg g(x)$ 。如式(3-18)所示的生成矩阵的码字,其生成多项式为  $C^2$  对应的生成函数  $C^2(x) = 1 + x$  称为  $C$  的生成多项式,  $k = 3 - 1 = 2$  对于次数相同的多项式,通常选取最高次项系数为 1 的多项式作为码的生成多项式。与之同样重要的多项式是循环码的一致校验多项式,表示为:

$$h(x) = \frac{x^n - 1}{g(x)} \quad (3-29)$$

与汉明码一样,循环码的编码译码也可以用生成矩阵和一致校验矩阵来描述。如果  $C$  是一个  $(n, k)$  循环码,具有生成多项式  $g(x) = g_0 + g_1x + \dots + g_r x^r$ ,其中  $r = n - k$  以及一



致校验多项式  $h(x) = h_0 + h_1x + \dots + h_kx^k$ , 那么生成矩阵与一致校验矩阵就可写为:

$$\mathbf{G}_1 = \begin{bmatrix} g_0 & g_1 & \cdots & \cdots & g_r & 0 & \cdots & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & \cdots & g_r & 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & \cdots & 0 & g_0 & g_1 & \cdots & \cdots & g_r \end{bmatrix} \quad (3-30)$$

$$\mathbf{H}_1 = \begin{bmatrix} h_k & h_{k-1} & \cdots & \cdots & h_0 & 0 & \cdots & \cdots & 0 \\ 0 & h_k & h_{k-1} & \cdots & \cdots & h_0 & 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & \cdots & 0 & h_k & h_{k-1} & \cdots & \cdots & h_0 \end{bmatrix} \quad (3-31)$$

### 3. 网联信道模型

在云控系统与车联网架构中, 由于通信协议与机制, 以及网络带宽有限的原因, 当控制器、传感器和执行器通过网络交换数据时, 经常会出现多包传输、数据多路径传输、网络拥塞、数据碰撞等现象, 这使得网络控制系统不可避免地产生网络时延、数据包丢失等问题, 因此传统的控制理论已经不能适用, 需要将网联与控制的混合系统进行建模。其中影响比较大的就是时延、丢包两大问题。当多个节点通过有限带宽网络进行数据交换时, 通常会不可避免地出现数据碰撞、网络拥塞等现象, 从而导致信息交换延迟, 这种由网络引起的信息交换之间的延迟称为网络诱导时延。网络中由于不可避免地存在网络阻塞和连接中断等现象, 必然会导致数据包丢失, 也就是丢包现象。下面主要介绍时延和丢包在网络控制系统中的经典模型。

#### 1) 时延模型

网络时延主要由四部分组成: ①信息产生时延。发送端等待发送信息封装成数据包并进入队列所需的时间。②数据包等候时延。当网络处于忙碌状态或者发生数据包碰撞时, 节点等候网络空闲再次发送数据所用的时间。③传输时延。数据包在实际传输介质上传输数据所需的时间, 由数据包大小、网络带宽和传输距离决定。④数据处理、计算时延。节点在数据处理过程中所需的时间。目前, 针对网络时延这一问题, 提出了多种控制模型, 下面介绍几种典型网络时延条件下的控制系统。

#### (1) 增广状态离散域模型法。

考虑如下被控对象:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \quad (3-32)$$

以周期  $T$  对其进行采样后得到离散状态方程:

$$\begin{cases} x(k+1) = \Phi x(k) + \Gamma u(k) \\ y(k) = Cx(k) \end{cases} \quad (3-33)$$

先行离散控制器模型为:

$$\begin{cases} \boldsymbol{\eta}(k+1) = F\boldsymbol{\eta}(k) - Gz(k) \\ u(k) = H\boldsymbol{\eta}(k) - Jz(k) \end{cases} \quad (3-34)$$

$$z(k) = y(k - i), \quad i = \{1, 2, \dots, j\} \tag{3-35}$$

其中： $\eta$  为控制器的状态向量； $z$  为控制器得到的最新时刻的量测数据； $u$  为控制器输出的控制量，引入增广状态向量  $\mathbf{X}(k)$ ：

$$\mathbf{X}(k) = [\mathbf{x}^T(k), \mathbf{y}^T(k - 1), \dots, \mathbf{y}^T(k - j), \eta^T(k), \mathbf{u}^T(k - 1), \dots, \mathbf{u}^T(k - l)] \tag{3-36}$$

则整个系统的增广状态离散时间模型为：

$$\mathbf{X}(k + 1) = \Omega \mathbf{X}(k) \tag{3-37}$$

(2) 摄动法。

假设在没有观测噪声的情况下，把从传感器端到控制器端的网络时延  $\tau_k^{sc}$  的影响描述成连续时间系统的非线性摄动。该方法所使用的结构图如图 3-7 所示。

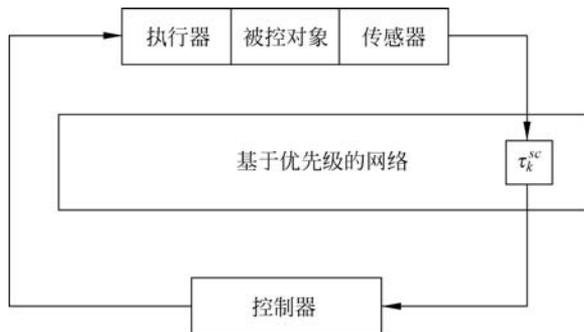


图 3-7 摄动法结构图

摄动法所采用的数学模型为：

$$\dot{\mathbf{x}}(t) = f(t, \mathbf{x}(t), \mathbf{e}(t)) \tag{3-38}$$

其中： $\mathbf{x}(t) = [\mathbf{x}_p^T(t), \mathbf{x}_c^T(t)]^T$  是系统的增广矩阵； $\mathbf{x}_p^T(t)$  是被控对象的状态； $\mathbf{x}_c^T(t)$  是控制器的状态； $\mathbf{e}(t)$  是误差函数； $f(\cdot)$  是非线性函数。定义误差函数为：

$$\mathbf{e}(t) = \mathbf{y}(t) - \hat{\mathbf{y}}(t) \tag{3-39}$$

摄动法是将误差函数  $\mathbf{e}(t)$  看作系统的一种摄动，通过推导出时延的上界来保证系统稳定性。这就要求系统的采样周期必须足够长才能将系统看作是连续时间系统并且仅适用于有优先级的网络。由于实际的被控对象往往是非线性的，因此该方法最大的优点是可以用于非线性被控对象。

(3) 随机最优控制法。

随机最优控制法针对传感器采用时间驱动方式，执行器和控制器均采用事件驱动方式，考虑的是在时延小于一个采样周期时，在离散域中用一个线性随机系统模型来描述带有随机时延的网络控制系统，并把随机时延的影响转化为 LQG 问题。LQG 问题是用途最广的且可以用分离原理设计全局最优控制系统的一类问题，是最优随机控制的主要研究问题之一。

随机最优控制法所用的线性随机离散被控对象模型为：

$$\begin{cases} \mathbf{x}(k + 1) = \Phi \mathbf{x}(k) + \Gamma_0(\tau_k) \mathbf{u}(k) + \Gamma_1(\tau_k) \mathbf{u}(k - 1) \\ \mathbf{y}(k) = C \mathbf{x}(k) + \mathbf{u}(k) \end{cases} \tag{3-40}$$



其目标是在系统具有完全状态信息的情况下,使最优随机控制性能指标  $J$  最小:

$$J = E \left\{ \mathbf{x}^T(N) \mathbf{Q}_0 \mathbf{x}(N) + \sum_{k=0}^{N-1} [\mathbf{x}^T(k) \mathbf{Q}_1 \mathbf{x}(k) + \mathbf{u}^T(k) \mathbf{Q}_2 \mathbf{u}(k)] \right\} \quad (3-41)$$

通过令  $J$  最小,其最优反馈控制为:

$$\mathbf{u}(k) = -\mathbf{L}(k, \tau_k) [\mathbf{x}^T(k), \mathbf{u}^T(k-1)]^T \quad (3-42)$$

其中:  $\mathbf{L}$  是最优增益矩阵。上述方法是假设时延的变化服从独立的概率分布,并且需要全部状态可测以及输出可测。随机最优控制方法虽然能获得较好的控制性能,但是需要大量的存储空间。

## 2) 丢包模型

由于受网络通信的影响,除了不可避免地带来时延以外,数据包的丢失也时常发生。丢包的原因主要包括: ①网络节点出现通信故障。当节点发生故障时,数据包所在的缓冲区被清空进而出现数据包丢失现象。②频繁的通信冲突。随机访问的网络中,本质上是无法避免冲突的发生的。发生冲突后,虽然多数网络通信协议可以实现重发机制,但信息重传都设置了超时,一旦超时则视为此次数据包丢失。③信道的干扰。在实际系统中外界的环境将不可避免地影响信道传输的质量。干扰造成物理信号的丢失、错位等,使实际数据到达目的节点后产生失真,无法通过算法恢复有效数据,数据包也会丢失。一般来说,网络控制系统的丢包模型可以分为多包传输和单包传输两种情况。

### (1) 单包传输时的丢包模型。

假设系统中的网络只存在于传感器和控制器之间,考虑采用状态反馈时的情况得到系统结构,如图 3-8 所示。

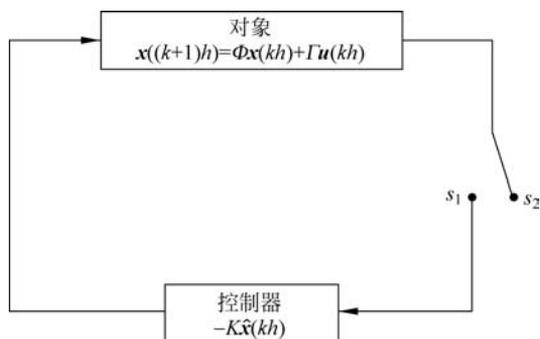


图 3-8 单包传输时的丢包模型

给定对象为:

$$\mathbf{x}((k+1)h) = \Phi \mathbf{x}(kh) + \Gamma \mathbf{u}(kh) \quad (3-43)$$

其中:  $\mathbf{x}(kh) \in R^n$ ,  $\Phi = e^{A_h}$ ,  $\Gamma = \int_0^h e^{As} B ds$ ,  $h$  为采样周期。当网络存在数据包丢失时,用一个切换模型来对系统进行简化。并且利用以下切换规则: 当开关在  $s_1$  时,  $\hat{\mathbf{x}}(kh) = \mathbf{x}(kh)$ ; 当开关在  $s_2$  时,系统的反馈信息丢失,  $\hat{\mathbf{x}}(kh) = \mathbf{x}((k-1)h)$ 。

令增广状态矢量为  $\mathbf{z}(kh) = [\mathbf{x}^T(kh), \hat{\mathbf{x}}^T(kh)]^T$ , 则当开关在  $s_1$  时有:

$$\begin{cases} \mathbf{x}((k+1)h) = \Phi\mathbf{x}(kh) + \Gamma\mathbf{u}(kh) \\ \hat{\mathbf{x}}((k+1)h) = \mathbf{x}((k+1)h) \\ \mathbf{u}(kh) = -K\hat{\mathbf{x}}(kh) \end{cases} \quad (3-44)$$

即

$$\begin{cases} \mathbf{x}((k+1)h) = \Phi\mathbf{x}(kh) - \Gamma K\hat{\mathbf{x}}(kh) \\ \hat{\mathbf{x}}((k+1)h) = \mathbf{x}((k+1)h) \end{cases} \quad (3-45)$$

由  $\mathbf{z}(kh) = [\mathbf{x}^T(kh), \hat{\mathbf{x}}^T(kh)]^T$ , 得:

$$\mathbf{z}((k+1)h) = \Phi_{s_1} \mathbf{z}(kh) \quad (3-46)$$

其中:

$$\Phi_{s_1} = \begin{bmatrix} \Phi & -\Gamma K \\ \Phi & -\Gamma K \end{bmatrix} \quad (3-47)$$

当开关在  $s_2$  时有:

$$\begin{cases} \mathbf{x}((k+1)h) = \Phi\mathbf{x}(kh) + \Gamma\mathbf{u}(kh) \\ \hat{\mathbf{x}}((k+1)h) = \hat{\mathbf{x}}(kh) \\ \mathbf{u}(kh) = -K\hat{\mathbf{x}}(kh) \end{cases} \quad (3-48)$$

即

$$\begin{cases} \mathbf{x}((k+1)h) = \Phi\mathbf{x}(kh) - \Gamma K\hat{\mathbf{x}}(kh) \\ \hat{\mathbf{x}}((k+1)h) = \hat{\mathbf{x}}(kh) \end{cases} \quad (3-49)$$

由  $\mathbf{z}(kh) = [\mathbf{x}^T(kh), \hat{\mathbf{x}}^T(kh)]^T$ , 得:

$$\mathbf{z}((k+1)h) = \Phi_{s_2} \mathbf{z}(kh) \quad (3-50)$$

其中:

$$\Phi_{s_2} = \begin{bmatrix} \Phi & -\Gamma K \\ \Phi & I \end{bmatrix} \quad (3-51)$$

因此可得单包传输状态反馈模型为:

$$\mathbf{z}((k+1)h) = \Phi_{s_i} \mathbf{z}(kh) \quad (3-52)$$

(2) 多包传输时的丢包模型。

当传感器采集的数据大于协议规定的最大传输范围后, 必须将控制信号分为多个包传输, 从而需对多包传输建模。下面以两个数据包传输的模型为例进行描述。

状态反馈情况下双包传输时丢包模型结构如图 3-9 所示。

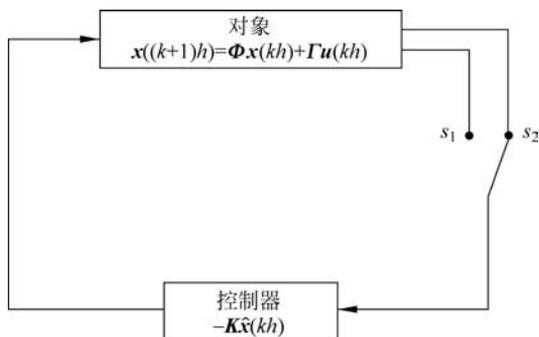


图 3-9 多包传输时的丢包模型



控制系统中各参数为:

$$\mathbf{x}(kh) = [\mathbf{x}_1^T(kh), \mathbf{x}_2^T(kh)]^T, \quad \hat{\mathbf{x}}(kh) = [\hat{\mathbf{x}}_1^T(kh), \hat{\mathbf{x}}_2^T(kh)]^T \quad (3-53)$$

$$\Phi = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix}, \quad \Gamma = \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \end{bmatrix}, \quad \mathbf{K} = [K_1, K_2] \quad (3-54)$$

被控对象的状态信号被分装在独立的两个数据包内依次传输至控制器,传输顺序为  $x_1 \rightarrow x_2 \rightarrow x_1 \rightarrow x_2 \dots$  这里用开关 S 来模拟两数据包的传输情况。

当开关切换到  $s_1$  时:

$$\begin{cases} \hat{\mathbf{x}}_1((k+1)h) = \mathbf{x}_1((k+1)h) \\ \hat{\mathbf{x}}_2((k+1)h) = \mathbf{x}_2(kh) \end{cases} \quad (3-55)$$

当开关切换到  $s_2$  时:

$$\begin{cases} \hat{\mathbf{x}}_1((k+1)h) = \mathbf{x}_1(kh) \\ \hat{\mathbf{x}}_2((k+1)h) = \mathbf{x}_2((k+1)h) \end{cases} \quad (3-56)$$

令增广状态矢量为  $\mathbf{z}(kh) = [\mathbf{x}_1^T(kh), \mathbf{x}_2^T(kh), \hat{\mathbf{x}}_1^T(kh), \hat{\mathbf{x}}_2^T(kh)]^T$ , 则当开关在  $s_1$  时有:

$$\begin{cases} \mathbf{x}_1((k+1)h) = \Phi_{11}\mathbf{x}_1(kh) + \Phi_{12}\mathbf{x}_2(kh) + \Gamma_1\mathbf{u}(kh) \\ \mathbf{x}_2((k+1)h) = \Phi_{21}\mathbf{x}_1(kh) + \Phi_{22}\mathbf{x}_2(kh) + \Gamma_2\mathbf{u}(kh) \\ \mathbf{u}(kh) = -\mathbf{K}\hat{\mathbf{x}}(kh) \end{cases} \quad (3-57)$$

即

$$\begin{cases} \mathbf{x}_1((k+1)h) = \Phi_{11}\mathbf{x}_1(kh) + \Phi_{12}\mathbf{x}_2(kh) - \Gamma_1K_1\mathbf{x}_1(kh) - \Gamma_1K_2\mathbf{x}_2(kh) \\ \mathbf{x}_2((k+1)h) = \Phi_{21}\mathbf{x}_1(kh) + \Phi_{22}\mathbf{x}_2(kh) - \Gamma_2K_1\mathbf{x}_1(kh) - \Gamma_2K_2\mathbf{x}_2(kh) \end{cases} \quad (3-58)$$

因此:

$$\mathbf{z}((k+1)h) = \Phi_{s_1}\mathbf{z}(kh) \quad (3-59)$$

其中:

$$\Phi_{s_1} = \begin{bmatrix} \Phi_{11} & \Phi_{12} & -\Gamma_1K_1 & -\Gamma_1K_2 \\ \Phi_{21} & \Phi_{22} & -\Gamma_2K_1 & -\Gamma_2K_2 \\ \Phi_{11} & \Phi_{12} & -\Gamma_1K_1 & -\Gamma_1K_2 \\ 0 & 0 & 0 & I \end{bmatrix} \quad (3-60)$$

同理可得,当开关切换到  $s_2$  时:

$$\mathbf{z}((k+1)h) = \Phi_{s_2}\mathbf{z}(kh) \quad (3-61)$$

其中:

$$\Phi_{s_2} = \begin{bmatrix} \Phi_{11} & \Phi_{12} & -\Gamma_1K_1 & -\Gamma_1K_2 \\ \Phi_{21} & \Phi_{22} & -\Gamma_2K_1 & -\Gamma_2K_2 \\ 0 & 0 & 0 & I \\ \Phi_{21} & \Phi_{22} & -\Gamma_2K_1 & -\Gamma_2K_1 \end{bmatrix} \quad (3-62)$$

因此可得单包传输状态反馈模型为:

$$\mathbf{z}((k+1)h) = \Phi_{s_i}\mathbf{z}(kh) \quad (3-63)$$

并且,若令  $z(kh)=[x_1^T(kh), \dots, x_n^T(kh), \hat{x}_1^T(kh), \dots, \hat{x}_n^T(kh)]^T$ ,则可推广到多包传输模型,其数学表达为:

$$z((k+1)h) = \Phi_{s_i} z(kh) \quad (3-64)$$

### 3.3.2 图论与网关节点模型

#### 1. 图论

在18世纪初的哥尼斯堡,有一条河流从城市中穿过,河中有A、B两座小岛,有七座桥将两座小岛与河岸相连(见图3-10),当地居民产生了一个疑惑:“能否每座桥恰好经过一次,并最终回到起点呢?”这就是著名的七桥问题。七桥问题一直困扰着人们,直到1736年欧拉发表论文《哥尼斯堡的七座桥》。欧拉从理论上证明了七桥问题无解,并在此基础上,开创了一个数学的新分支——图论。

图论发展到今天,在生物、化学、物理、计算机等领域都发挥着重要的作用,本节将和大家一起探讨图论的基础知识以及图论在车联网领域的应用。

##### 1) 图论基础知识

图论作为数学的一个分支,其研究对象是图,那么什么是图呢?通常,用  $G$  来表示图,前面所提到的七桥问题,就可以抽象为一个如图3-11所示的图  $G$ 。图  $G$  是关系的数学表示,具体由两个集合构成:非空的节点集  $\mathcal{V}$  和有限的边集  $\mathcal{E}$ ,其中边指节点组成的无序对。七桥问题中A、B小岛与C、D两岸即为图  $G$  的节点,七座桥即为图  $G$  的边。

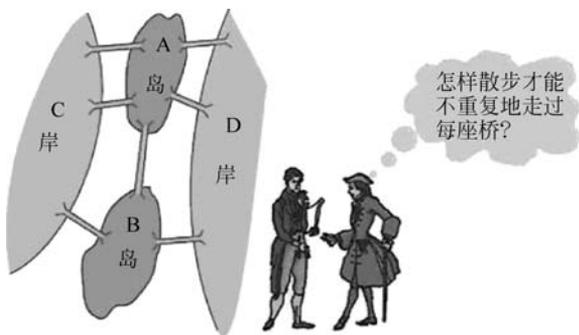


图 3-10 哥尼斯堡七桥

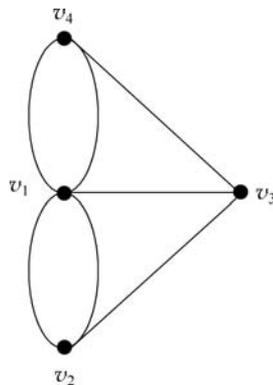


图 3-11 七桥问题的图论表述

用  $\mathcal{V}(G) = \{v_1, v_2, \dots, v_n\}$  来表示所有节点所组成的集合,集合  $\mathcal{V}$  的基数  $n$  称为图  $G$  的阶;用  $\mathcal{E}(G) = \{e_1, e_2, \dots, e_m\}$  表示所有边组成的集合,集合  $\mathcal{E}$  的基数  $m$  称为图  $G$  的规模。一个图的完整表示记为  $G = (\mathcal{V}, \mathcal{E})$ 。

如果一个图节点的数量与边的数量是有限的,则称图  $G$  为有限图,否则称为无限图;如果节点的数量为1,边的数量为0,则称图  $G$  为平凡图;如果图  $G$  的节点与节点之间由箭头连接,则称图  $G$  为有向图(见图3-12(a)),如果由直线与曲线连接,则为无向图(见图3-12(b))。

如不做说明,则默认图为无向图。

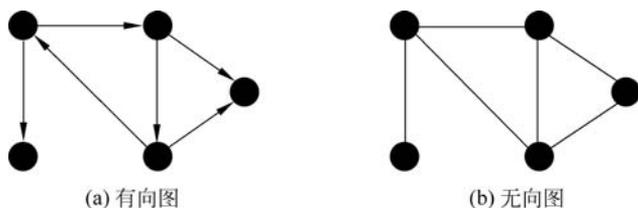


图 3-12 有向图与无向图

对于节点来说,如果  $v_1, v_2 \in \mathcal{V}(G), e = v_1 v_2 \in \mathcal{E}(G)$ , 则称  $v_1, v_2$  为  $e$  的端点, 并称  $v_1$  与  $v_2$  邻接, 否则称为非邻接。与节点  $v$  关联的边的数目, 称为节点  $v$  的度, 记为  $\text{deg}(v)$ , 一个环记两次。一个节点的度如果是奇数, 则称为奇点; 如果是偶数, 则称为偶点。对于前面所提到的七桥问题, 需要起点和终点的度为偶数, 中间节点的度也为偶数, 显然七桥问题中所有节点的度均为奇数, 不能满足这一要求, 所以七桥问题无解。

图论中还有如下一些概念。

(1) 完全图: 如果图  $G$  每个节点之间有且只有一条边相连, 则称图  $G$  为完全图。完全图是一个简单的无向图, 如图 3-13 所示。

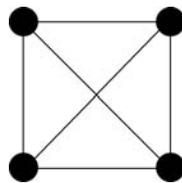


图 3-13 完全图

(2) 子图: 如果一个图  $H$  的节点集和边集分别是另一个图  $G$  的节点集的子集和边集的子集, 则称图  $H$  为图  $G$  的一个子图, 如图 3-14(a)、(b)所示, 图  $G$  称为图  $H$  的母图。

(3) 完全子图: 若  $H$  是  $G$  的子图, 且  $H$  为完全图, 则  $H$  是  $G$  的完全子图, 如图 3-14(c) 所示。

(4) 极大完全子图: 若  $H$  不能被  $G$  的任意完全子图包含, 且  $H$  为  $G$  的完全子图, 则  $H$  是  $G$  的极大完全子图, 如图 3-14(d) 所示。

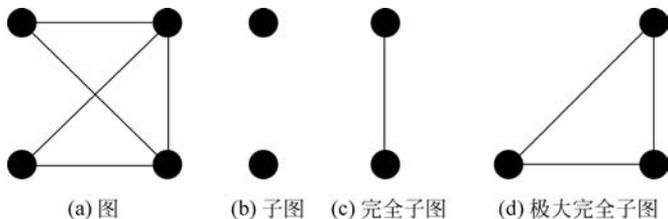


图 3-14 图、子图、完全子图与极大完全子图

(5) 补图: 如果  $\mathcal{V}(G_1) = \mathcal{V}(G_2), u, v \in \mathcal{E}(G_1)$ , 当且仅当  $u, v \notin \mathcal{E}(G_2)$  成立, 则称  $G_1$  与  $G_2$  互为补图, 记为  $G_1 = \bar{G}_2$ , 如图 3-15 所示。

(6) 同构图: 若两个图中对应节点的邻接关系相同, 则称为同构图, 如图 3-16 所示。

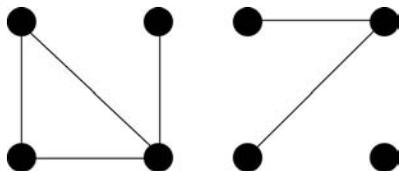


图 3-15 补图

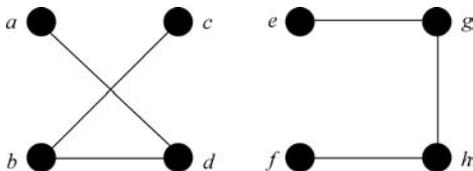


图 3-16 同构图

(7) 连通图：如果图  $G$  的每个节点都有路径到达其他任意节点，即不存在孤立节点，则称图  $G$  为连通图，如图 3-17 所示。

(8) 树：连通的无圈图称为树，如图 3-18 所示。

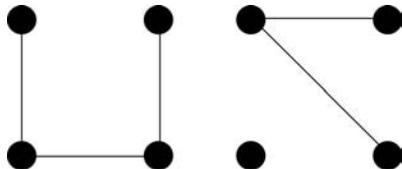


图 3-17 连通图

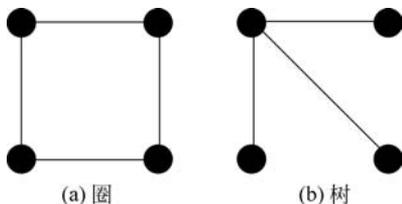


图 3-18 圈与树

## 2) 图论的直接应用

图是关系的数学表示，能将一个复杂的问题通过图简单地表示出来，所以图论最直接的应用是建模，例如，可以将旅行商问题和邮递员的最短路径问题用图论的方法来建模。

### (1) 旅行商问题。

一个商人需要在各个城市间往返，他要怎样选择路径，才能从一个城市出发，每个城市都到达一次，最终回到出发的城市呢？我们可以将公路网抽象为一个图（见图 3-19），用图的每个节点来表示城市，用节点之间的边来表示城市之间的道路，这样，就能用图的方式，表达出每个城市、每条道路之间的关系，再从中选出一个满足要求的子图，就能解决该问题。

### (2) 最短路径问题。

一个邮递员，每天需要给  $m$  个乡镇送信，他该怎么安排送信的顺序，来保证自己每天走的路程最小？这个问题可以看成旅行商问题的复杂版本，我们依然可以将每个乡镇看作图的节点，将乡镇之间的道路看作图的边，但是与旅行商问题不同的是，需要给每条边加上权重，权重表示了两个乡镇之间的距离。这样就获得了如图 3-20 所示的一个图  $G$ 。解决最短路径的最经典的算法是 Dijkstra 算法。

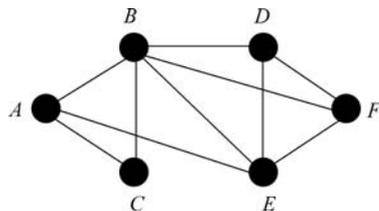


图 3-19 旅行商问题

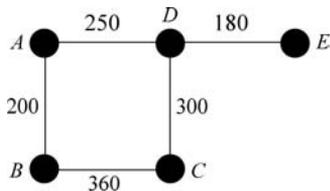


图 3-20 最短路径问题

## 3) 图论中算法的应用

图论中的算法也可以用来求解各类问题，其中包括深度优先搜索算法和广度优先搜索算法。

在一些问题当中，不是所有的问题都可以用一个确切的数学模型直接求解，如迷宫问题，对于这种问题，可以采用搜索的方法来解决，按照一定的顺序、规则，不断去试探，直到试探完所有情况后，返回问题的解（或者无解）。



### (1) 深度优先搜索。

深度优先搜索(depth first search,DFS),是一种经典的图搜索算法,为了更好地理解深度优先搜索,可以用一个迷宫问题加以说明。当身处一个迷宫中时,走出迷宫的方法是不断往前探索并记录走过的路径,当走到一个死胡同时,返回上一个岔路口,并探索该路口的其他路径,这样就能够保证一定能搜索到迷宫的出口。

深度优先搜索的基本思想也是如此,将迷宫问题扩展到一个图论问题:遍历如图 3-21 所示的图  $G$  中所有的节点。

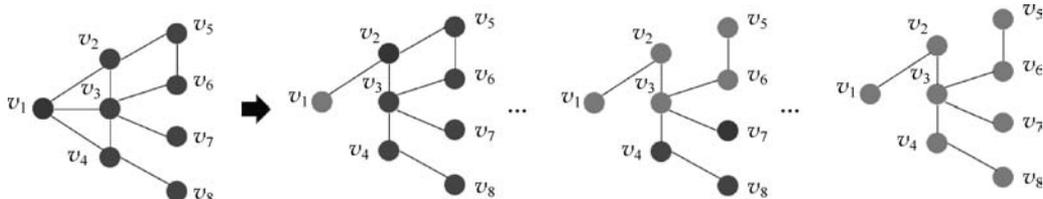


图 3-21 深度优先搜索

步骤 1: 选定一个节点  $v_0$ , 设为初始状态并将该点加入访问集。

步骤 2: 访问当前节点  $v_n$  的邻居节点。

a) 如果节点  $v_n$  存在未被访问的邻居节点  $v_{n+1}$ , 则将该邻居节点  $v_{n+1}$  放入访问集, 并将当前状态更新为  $v_{n+1}$ , 转至步骤 2。

b) 如果不存在未被访问的邻居节点, 转至步骤 3。

步骤 3: 返回当前节点的上一个节点  $v_{n-1}$ 。

a) 存在节点未被访问, 将节点  $v_{n-1}$  设为当前状态, 转至步骤 2。

b) 所有节点都被访问, 结束。

### (2) 广度优先搜索。

广度优先搜索(breadth first search,BFS),又称为宽度优先搜索或横向优先搜索,广度优先搜索与深度优先搜索相似又不相同,可以用来求解不同的问题。

广度优先搜索的基本思想是: 在图中选取任意一个节点作为初始节点, 然后将该点的所有邻居节点作为第一层的节点, 再访问第一层节点的邻居节点作为第二层节点, 经过有限步之后, 将图中的所有节点访问完成, 就产生了生成树, 如图 3-22 所示。

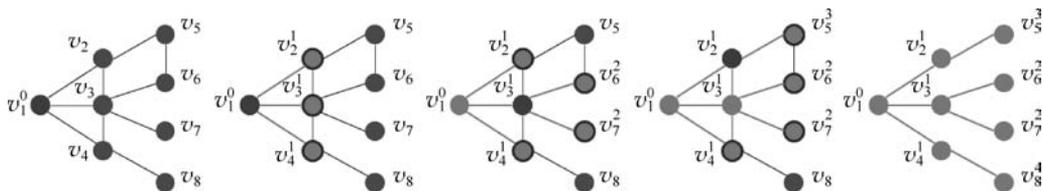


图 3-22 广度优先搜索

广度优先搜索算法的步骤如下。

步骤 1: 选取图中任意节点  $v$  作为初始节点, 令  $f=0$ , 将  $v$  标记为  $f$ 。

步骤 2:

a) 若所有标号为  $f$  的节点有邻居节点未被标记, 将所有未被标记的邻居节点标记为

$f+1$ , 令  $f=f+1$ , 转至步骤 2。

b) 若所有标号为  $f$  的节点不存在邻居节点未被标记, 结束。

## 2. 网关节点模型

### 1) 节点拓扑模型

了解完图论的基础知识, 接下来将学习图论在车联网领域的应用。随着车联网的发展, 在“网联化”条件下, 行驶在道路上的车辆不再是相互孤立的个体, 而是通过无线通信网络连接组成了多车系统。在多车系统中, 网联车辆可基于车-车通信和车-路通信获取通信范围内其他网联车辆和道路的信息, 并利用该信息进行分布式的决策与控制, 从而实现整个系统的协同控制。其中, 通信网络会影响多车系统中车辆间的信息传输关系与质量, 也会影响车辆间的通信距离与通信拓扑结构, 进而影响多车系统的网络化控制性能。可以用图论的方法来对车-车之间的通信网络进行建模分析, 接下来给出图论在车辆队列通信拓扑结构建模的实例。

车辆队列系统也被称为协同式自适应巡航(cooperative adaptive cruise control, CACC)系统, 其在巡航控制(cruise control, CC)和自适应巡航控制(adaptive cruise control, ACC)系统的基础上, 通过引入车-车通信以实现网联车辆间的信息传输与共享, 进而实现多个车辆的连续跟车控制, 保证车辆安全, 提升整个车辆队列的性能。

表征车辆之间信息传递的信息流拓扑结构, 可以直观地抽象为图的结构(见图 3-23), 进而用相应矩阵及其性质进行刻画。

将车辆队列中的通信拓扑记为图  $G=(V, E, A)$ , 其包括三个主要元素, 其中:

(1)  $V=\{V_1, V_2, \dots, V_N\}$  为节点集合,  $V_i$  为节点,  $N$  为节点个数。在多车系统中, 每个车辆即可视为一个节点。

(2)  $E \subseteq V \times V$  为边集合, 在多车系统中, 若车辆间存在通信, 即可将该通信连接视作车辆之间的一条有向边。

(3)  $A=a_{ij} \in \mathbb{R}^{N \times N}$  为邻接矩阵, 其中  $a_{ij}$  表明了节点  $i$  与节点  $j$  之间的连接关系, 如式(3-65)所示, 即当  $(j, i) \in E$  时,  $a_{ij}=1$ , 此时, 称  $j$  为  $i$  的邻居; 否则,  $a_{ij}=0$ , 称  $j$  不是  $i$  的邻居。

$$a_{ij} = \begin{cases} 1, & (j, i) \in E \\ 0, & (j, i) \notin E \end{cases} \quad i, j \in N \quad (3-65)$$

在多车系统中,  $a_{ij}=1$  表明车辆  $i$  可以获取  $j$  的信息, 反之, 有  $a_{ij}=0$ , 假设图中无自环现象, 即  $a_{ii}=0$ 。节点  $i$  的邻域集定义为:

$$N_i = \{j \mid a_{ij} = 1\} \quad (3-66)$$

集合  $N_i$  表示在车辆队列中, 车辆  $i$  能够获取信息的车辆集。

在邻接矩阵的基础上, 可以进一步定义如下矩阵。

(1) 入度矩阵  $D = \text{diag}\{\text{deg}_i\} \in \mathbb{R}^{N \times N}$ , 其中  $\text{deg}_i = \sum_{j=1}^N a_{ij}$  为指向节点  $i$  的边的个数, 即节点  $i$  的邻居的个数。

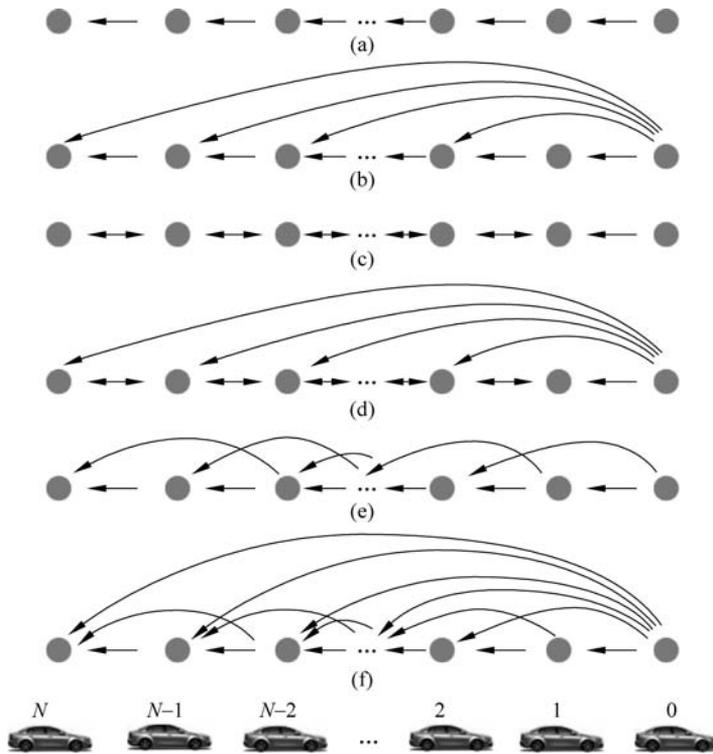


图 3-23 车辆队列通信拓扑

$$\mathcal{D} = \begin{bmatrix} \text{deg}_1 & & \\ & \ddots & \\ & & \text{deg}_N \end{bmatrix} \quad (3-67)$$

(2) 出度矩阵  $\mathcal{D}_o = \text{diag}\{\text{deg}_{oi}\} \in \mathbb{R}^{N \times N}$ , 其中  $\text{deg}_{oi} = \sum_{j=1}^N a_{ji}$  为由节点  $i$  出发的边的个数, 即以节点  $i$  为邻居的节点的个数。

$$\mathcal{D}_o = \begin{bmatrix} \text{deg}_{o1} & & \\ & \ddots & \\ & & \text{deg}_{oN} \end{bmatrix} \quad (3-68)$$

(3) 拉普拉斯矩阵  $\mathcal{L} = [l_{ij}] \in \mathbb{R}^{N \times N}$ , 其中当  $i \neq j$  时, 有  $l_{ij} = -a_{ij}$ ; 当  $i = j$  时, 有  $l_{ii} = \sum_{j=1}^N a_{ij} = \text{deg}_i$ 。易知:

$$\mathcal{L} = \mathcal{D} - \mathcal{A} \quad (3-69)$$

$\mathcal{L} \cdot \mathbf{1}_N = \mathbf{0}_{NN}$ , 其中  $\mathbf{1}_N$  是元素均为 1 的  $N$  维列向量。

(4) 牵引矩阵  $\mathcal{P} \in \mathbb{R}^{N \times N}$ , 用于描述跟随车辆获取领航车辆信息的情况, 定义为:

$$\mathcal{P} = \begin{bmatrix} \mathcal{P}_1 & & \\ & \ddots & \\ & & \mathcal{P}_N \end{bmatrix} \quad (3-70)$$

其中：若  $(0, i) \in \mathcal{E}$  时,  $\mathcal{P}_i = 1$ ; 否则  $\mathcal{P}_i = 0$ 。  $\mathcal{P}_i = 1$  表示车辆  $i$  能够获得领航车辆的状态信息, 此时, 车辆  $i$  也称为被领航车辆直接牵引。同时, 定义车辆  $i$  的领航车辆可达性集合为:

$$\mathcal{P}_i = \begin{cases} \{0\}, & \mathcal{P}_i = 1 \\ \emptyset, & \mathcal{P}_i = 0 \end{cases} \quad (3-71)$$

下面以双前车跟随式(TPF)和双前车-领航者跟随式(TPLF)拓扑结构为例, 给出上述定义矩阵的具体表达式。

如图 3-24 所示, TPF 拓扑结构下, 车辆能获取其前面两辆车的状态信息; 而 PLF 拓扑结构下, 每个跟随车辆还能获取领航车辆的信息。所以, 在这两种结构下, 节点  $i$  的邻域集均为:

$$(N_i)_{\text{TPF or TPLF}} = \begin{cases} \emptyset, & i = 1 \\ \{1\}, & i = 2 \\ \{i - 1, i - 2\}, & i = \mathcal{N} \setminus \{1, 2\} \end{cases} \quad (3-72)$$

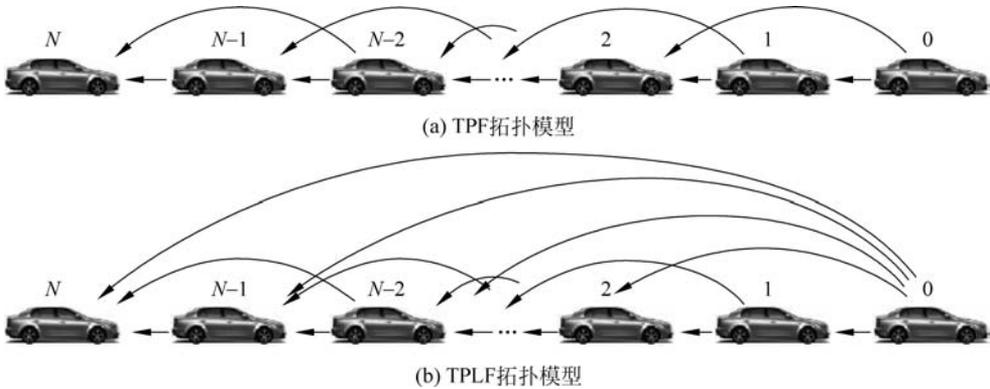


图 3-24 TPF 拓扑模型和 TPLF 拓扑模型

所以这两种拓扑结构的邻接矩阵与拉普拉斯矩阵均为:

$$(\mathcal{A})_{\text{TPF or TPLF}} = \begin{bmatrix} 0 & & & & & & \\ 1 & 0 & & & & & \\ 1 & 1 & 0 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & & 1 & 1 & 0 & \end{bmatrix} \quad (3-73)$$

$$(\mathcal{L})_{\text{TPF or TPLF}} = \begin{bmatrix} 0 & & & & & & \\ -1 & 1 & & & & & \\ -1 & -1 & 2 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & & -1 & -1 & 2 & \end{bmatrix} \quad (3-74)$$

在 BD 与 BDL 拓扑结构中, 节点  $i$  的领航车辆可达集分别为:

$$(\mathcal{P}_i)_{\text{TPF}} = \begin{cases} \{0\}, & i = \{1, 2\} \\ \emptyset, & i = \mathcal{N} \setminus \{1, 2\} \end{cases}$$



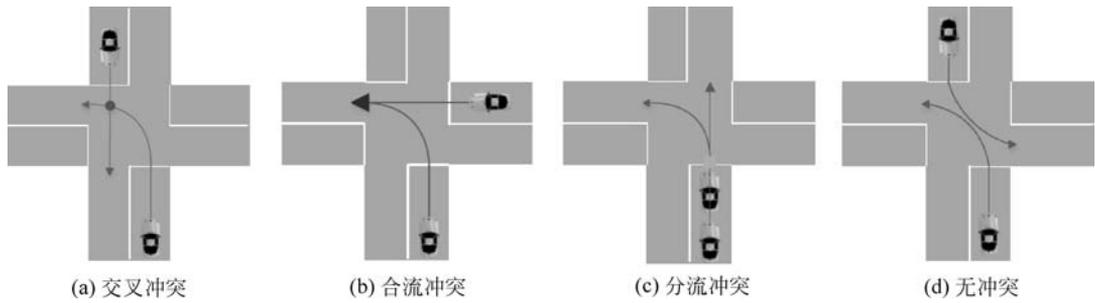


图 3-26 交叉路口冲突关系图

表 3-2 典型十字交叉路口交通流编号

$M_{A^{(1)}d^{(2)}}$	$M_{A^{(1)}c^{(2)}}$	$M_{A^{(2)}c^{(1)}}$	$M_{A^{(2)}b^{(1)}}$	$M_{B^{(1)}a^{(2)}}$	$M_{B^{(1)}d^{(2)}}$	$M_{B^{(2)}d^{(1)}}$	$M_{B^{(2)}c^{(1)}}$
1	2	3	4	5	6	7	8
$M_{C^{(1)}b^{(2)}}$	$M_{C^{(1)}a^{(2)}}$	$M_{C^{(2)}a^{(1)}}$	$M_{C^{(2)}d^{(1)}}$	$M_{D^{(1)}c^{(2)}}$	$M_{D^{(1)}b^{(2)}}$	$M_{D^{(2)}b^{(1)}}$	$M_{D^{(2)}a^{(1)}}$
9	10	11	12	13	14	15	16

交叉路口存在四种模式的交通流冲突关系,如图 3-26 所示,包括:交叉冲突关系、合流冲突关系、分流冲突关系和无冲突关系。

基于交通流冲突关系,可定义交通流冲突集。定义交通流  $i(i \in \{1, 2, \dots, N_M\})$  的交叉冲突集  $S_i$  包含与之存在交叉关系的所有交通流的集合,合流冲突集  $Q_i$  包含与之存在合流冲突关系的所有交通流的集合,分流冲突集  $R_i$  包含与之存在分流冲突关系的所有交通流的集合。进一步地,定义交通流  $i$  的总冲突集  $C_i$  为与之存在冲突关系的所有交通流的集合,也即  $S_i$ 、 $Q_i$  和  $R_i$  的并集。例如,图 3-26 中交通流 1 的交叉冲突集  $S_1 = \{5, 11, 13, 14, 15\}$ ,合流冲突集  $Q_1 = \{6\}$ ,分流冲突集  $R_1 = \{1, 2\}$ ,总冲突集  $C_1 = \{1, 2, 5, 6, 11, 13, 14, 15\}$ 。若车辆所处交通流位于另一车辆所处交通流的冲突集,则这两个车辆位移轨迹必然在路口入口、路口内或路口出口发生冲突。反过来,如若两个车辆在交叉路口位移轨迹在交叉路口发生冲突,则两个车辆所属交通流必处于彼此的冲突集中。

如图 3-27 所示,在交叉路口引入一与实际车道交于交叉路口中心的虚拟车道。对于交叉路口入口不同车道的所有车辆,以交叉路口中心为圆心,以车辆与交叉路口中心的距离为半径,旋转投影到虚拟车道上,最终形成虚拟车道上的一维车辆虚拟队列,从而在队列几何拓扑上将二维车群转化为一维队列。

虚拟队列中的车辆按照与交叉路口中心的距离远近依次编号为  $1 \sim N$ ,其中  $N$  为虚拟队列中车辆总数,同时将车辆  $i$  所属交通流定义为  $S_i$ 。此外,在虚拟队列编号为 1 的车辆前方虚拟生成一虚拟领航车,并编号为 0,其车速设定为匀速  $v_l$ 。此处,  $v_l$  设定为车辆安全高效通过交叉路口的最高车速,其根据交叉路口安全车速预先设定。虚拟队列与实际一维车辆队列的跟车模式不同。虚拟队列中车辆不一定跟随其相邻前车,而是可能跟随其前方另一车辆。车辆跟随关系在后文几何拓扑结构中描述。

对于属于交通流  $S_i(S_i \in \mathcal{N}^+, S_i \leq N_M)$  的车辆  $i(i \in \mathcal{N}^+, i \leq N)$ ,若属于交通流  $S_j(S_j \in \mathcal{N}^+, S_j \leq N_M)$  的车辆  $j(j \in \mathcal{N}^+, j \leq N)$  与其存在冲突关系,则车辆  $j$  所属交通流必然位于车辆  $i$  所属交通流的冲突集中,即满足关系  $S_j \in C_{S_i}$ 。定义车辆  $i$  的冲突车辆集

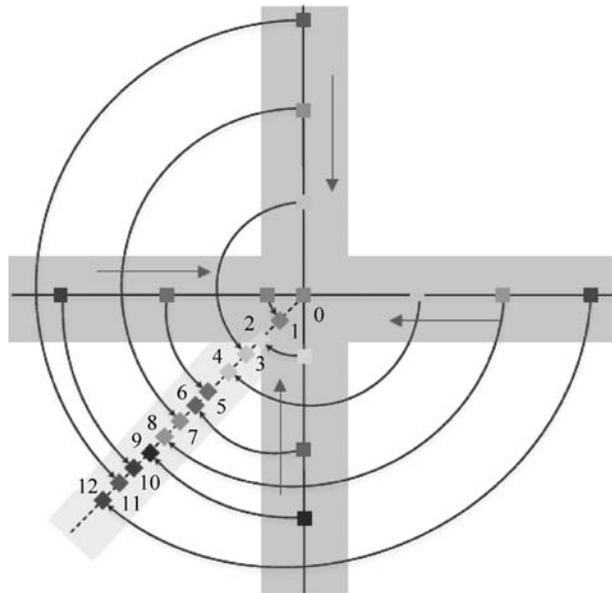


图 3-27 车辆旋转投影

$P_i$  为所有位于车辆  $i$  前方并与车辆  $i$  存在冲突关系的车辆的集合,也即:

$$P_i = \{j \mid j < i, S_j \in C_{S_i}, j \in \mathcal{N}^+\} \quad (3-77)$$

虚拟队列中某些车辆前方可能不存在冲突车辆,如图 3-28 所示的车 1,其前方没有真实车辆,无冲突前车,此时  $P_i = \emptyset$ 。在此情况下,将此车辆的冲突前车设置为虚拟领航车 0,也即:

$$P_i = \{0\} \quad (3-78)$$

因此,对于除虚拟领航车外的任意车辆  $i$ ,其冲突车辆集  $P_i$  均不为空集。图 3-28 为交

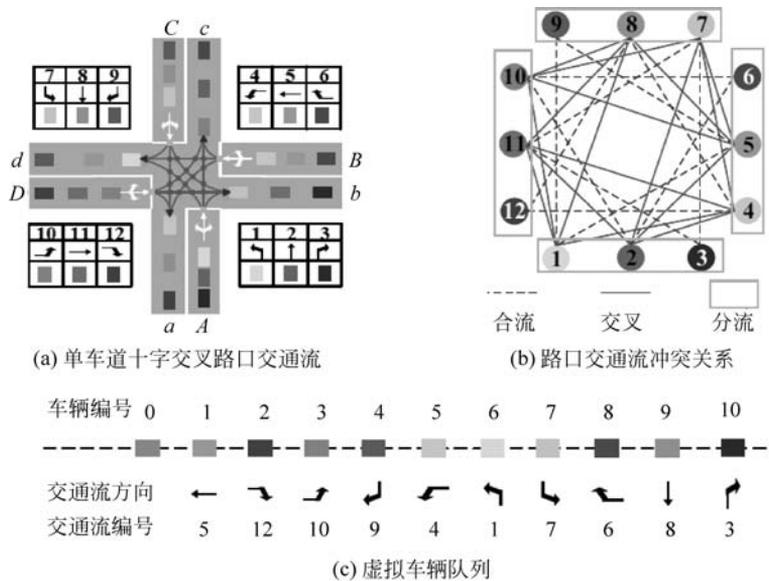


图 3-28 单车道十字交叉路口交通流及虚拟队列

叉路口渠化及交通流轨迹,图 3-28(b)为交叉路口交通流冲突关系。由定义可知,虚拟队列中车 1~10 分别属于 5 号、12 号、10 号、9 号、4 号、1 号、7 号、6 号、8 号和 3 号交通流。可知,车 1~车 10 的冲突车辆集分别为:  $P_1 = \{0\}$ ,  $P_2 = \{0\}$ ,  $P_3 = \{1, 2\}$ ,  $P_4 = \{1\}$ ,  $P_5 = \{1, 2\}$ ,  $P_6 = \{1, 3, 4, 5\}$ ,  $P_7 = \{3, 4, 5\}$ ,  $P_8 = \{1, 3, 5\}$ ,  $P_9 = \{1, 2, 3, 4, 5, 6, 7\}$ ,  $P_{10} = \{6, 7\}$ 。

基于以上定义的冲突集,采用冲突有向图  $G_{N+1} = \{\mathcal{V}_{N+1}, \mathcal{E}_{N+1}\}$  表征车辆虚拟队列中各车辆的冲突关系。在有向图中,  $G_{N+1}$  的节点集  $\mathcal{V}_{N+1} = \{0, 1, 2, \dots, N\}$  表示虚拟队列中的所有车辆(用车辆编号表示)的集合; 有向边集  $\mathcal{E}_{N+1} = \{(i, j) | i, j \in \mathcal{V}_{N+1}\}$  表示车辆两两之间存在冲突关系的集合。因此,给出冲突有向图的定义。

**定义:** 给定虚拟车辆队列  $\{0, 1, 2, \dots, N\}$ 、各车辆冲突集  $P_i$  以及有向图  $G_{N+1} = \{\mathcal{V}_{N+1}, \mathcal{E}_{N+1}\}$ , 若  $\mathcal{V}_{N+1} = \{0, 1, 2, \dots, N\}$ , 且  $\forall i \in \mathcal{V}_{N+1} \setminus \{0\}$ , 当且仅当  $j \in P_i$ , 有  $(j, i) \in \mathcal{E}_{N+1}$ , 则  $G_{N+1}$  为虚拟车辆队列的冲突有向图。

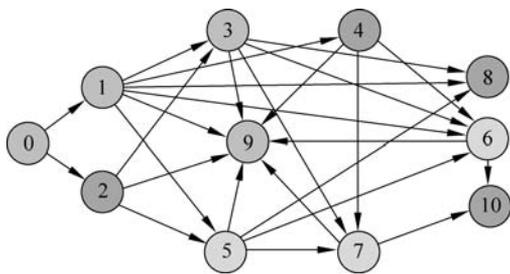


图 3-29 虚拟队列的冲突有向图

于是,图 3-28 中虚拟队列的冲突有向图如图 3-29 所示,图中 0~10 共 11 个顶点分别表示虚拟领航车 0 和其余车辆 1~10,图中的有向边分别表示不同车辆的冲突关系。

**定义:** 若冲突有向图中存在一个有向边序列  $\{(i_1, i_2), (i_2, i_3), \dots, (i_{n-1}, i_n)\}$ , 其中  $\forall k \in \{2, 3, \dots, n\}$  有  $(i_{k-1}, i_k) \in \mathcal{E}_{N+1}$ , 则该有向边序列集称为从节点  $i_1$  到节点  $i_n$  的一条有向路径。

**定义:** 若冲突有向图的一个子图中存在从某一节点到其余任意节点的有向路径,则称该子图为冲突有向图的有向生成树,该节点称为有向生成树的根节点。

### 3.3.3 博弈论与行为收益模型

博弈论是一门古老的学科,有着非常悠久的历史,春秋战国时期的田忌赛马就是一个例子,经过几千年的发展,博弈论在 20 世纪 40 年代开始逐步形成了一套完善的理论体系,并逐渐在多个领域发挥出巨大的作用。

博弈论又被称为对策论,作为数学的一个新的分支,它模拟参与博弈的参与人之间的合作与竞争,并且给出一种方法来预测参与人之间的决策结果,像围棋、象棋都是博弈论的一种自然的应用。

#### 1. 博弈论基础知识

在博弈论中,可以根据参与人的行动顺序分为动态博弈和静态博弈,也可以根据参与人对于其他参与人的了解分为完全信息博弈和非完全信息博弈。有两种描述博弈的方法:策略式和扩展式,策略式只能建模描述参与人同时行动且仅行动一次的博弈(静态博弈),而扩展式能够描述参与人决策的时间,即能描述参与人根据不同行动顺序做出决策的动态博弈。

本节将主要关注完全信息静态博弈:每个参与者都能获得其他参与者的信息,即每一



个参与者的收益函数在所有参与者之间是共同知识,并且在开始时每个参与者同时选择行动,然后根据所有参与者的选择,每个参与者得到各自的收益或支出。

策略式博弈是博弈的一种常用表达方法,策略式博弈包含三种元素:参与人、策略空间和效用函数,参与人集合  $i \in I$ , 设集合  $I$  为有限集合  $\{1, 2, \dots, I\}$ , 对于每一个参与人  $i$  都有策略空间  $S_i$ , 以及收益函数  $u_i$ , 这一函数对每种策略组合  $s = (s_1, s_2, \dots, s_I)$  给出参与人  $i$  的效用  $u_i(s)$ 。简单来说,策略式博弈需要满足以下三个条件: ①有两个或者两个以上的参与人,但是参与人数量是有限的; ②每个参与人都有可以单独选择的策略空间; ③每一个策略都对应一个收益函数。同时,常常把参与人  $i$  以外的其他所有参与人标记为  $-i$ 。

**定义:** 在一个有  $I$  个参与人博弈的标准式表述中,参与者的策略空间为  $S_1, S_2, \dots, S_I$ , 收益函数为  $u_1, u_2, \dots, u_I$ , 用  $G = \{S_1, S_2, \dots, S_I; u_1, u_2, \dots, u_I\}$  表示此博弈。

可以用“囚徒困境”这一个经典的问题来加以说明。囚徒困境的故事是,两个犯罪嫌疑人被警察以涉嫌偷窃而逮捕,警察知道二人有罪,但是又缺乏足够的证据。于是警察将两人分别关押,不能互相沟通情况。如果两人都不揭发对方,则由于证据不确定,每个人都判一年;若一人揭发,而另一人沉默,则揭发者因为立功而立即获释,沉默者因不合作而判十年;若互相揭发,则因证据确凿,二者都判八年。囚徒困境可以用表 3-3 的矩阵表示,在矩阵中第一列代表囚徒 A 的策略,第一行代表囚徒 B 的策略,第一个数字代表囚徒 A 的奖励(惩罚),第二个数字代表囚徒 B 的奖励(惩罚)。

表 3-3 囚徒博弈收益矩阵

A	B	
	坦白	抵赖
坦白	(-8, -8)	(0, -10)
抵赖	(-10, 0)	(-1, -1)

在如上博弈的例子中,每一个囚徒均有两种策略可以选择:坦白或者抵赖,当两人在互相不知道对方决策的条件下,选择一种策略执行(虽然不是同时),假设囚徒 A 选择坦白,囚徒 B 选择抵赖,则策略组合  $s = (\text{坦白}, \text{抵赖})$ , 这种情况下囚徒 A 的效用  $u_A(s) = 0$ , 囚徒 B 的效用  $u_B(s) = -10$ 。

建立了一个博弈之后,需要求解博弈中存在的均衡,下面将介绍占优均衡和博弈论中常用的纳什均衡。

## 2. 占优均衡

在囚徒困境中,如果一个囚徒选择坦白,那么对于另一个囚徒来说,选择坦白会被判八年,而选择抵赖则会被判十年;如果一个囚徒选择抵赖,另一个囚徒选择坦白则不会被判刑,选择抵赖会被判一年。这样,对于囚徒来说,不管另一个囚徒选择什么动作,自己选择坦白所获得的收益均会大于抵赖的收益,说明坦白是一个占优策略,即无论其他参与人选择什么策略,参与人  $i$  选择的这个策略,总比其他策略要好;与之相对的是劣势策略,即无论其他参与人选择什么策略,参与人  $i$  选择的这个策略,总存在一个策略比选择的这个策略要好。

**定义：**用 $(s_i, s_{-i})$ 表示组合 $(s_1, \dots, s_{i-1}, s_i, s_{i+1}, \dots, s_I)$ ，对于参与者 $i$ 的策略空间 $S_I$ ，存在策略 $s'_i$ ，使得 $u_i(s'_i, s_{-i}) \geq u_i(s_i, s_{-i}) \forall s_i \in S_I \setminus s'_i$ ，则称策略 $s'_i$ 为占优策略，若 $u_i(s'_i, s_{-i}) > u_i(s_i, s_{-i}) \forall s_i \in S_I \setminus s'_i$ ，则称策略 $s'_i$ 为严格占优策略。

一个理性的参与者不会选择劣势策略，我们可以用这一原则解决其他的博弈问题，思路如下：

- (1) 找到其中某个参与者的严格劣势策略，并将其剔除。
- (2) 重新构建一个不包含剔除策略的博弈。

(3) 重复步骤(1)和(2)，直到只剩下唯一的策略组合，该策略组合被称为“重复剔除的占优均衡”。

如上方法称为重复剔除劣势策略，需要注意的是，如果不存在严格劣势策略，则不能用这种方法来解决博弈问题；如果最终剔除后的策略组合不唯一，也不能用该方法求解。

此外，重复剔除劣势策略不仅要求参与人是理性的，还要求每个人知道其他人是理性的，每个人知道其他人知道自己是理性的，如此……即理性是一个“共同知识”。

考虑这样一个例子，参与人1与参与人2分别能执行策略 $\{a_1, a_2, a_3\}$ 和策略 $\{b_1, b_2, b_3\}$ ，两个参与人所获得的收益如表3-4所示。

对于参与人1来说，策略 $a_3$ 相比于其他策略，是一个严格劣势策略，当参与人选择策略 $a_3$ 时，不论参与人2选择什么动作，参与人1所获得的收益都要小于策略 $a_1$ 和 $a_2$ ，所以将策略 $a_3$ 剔除。剔除之后，策略 $b_3$ 是一个严格劣势策略，将策略 $b_3$ 剔除，接着剔除策略 $a_1$ 与策略 $b_1$ ，获得重复剔除的占优均衡策略组合 $(a_2, b_2)$ 。

表 3-4 收益矩阵

P1	P2		
	b1	b2	b3
a1	(-2, -2)	(0, 2)	(11, -3)
a2	(0, 0)	(1, 3)	(7, -1)
a3	(-5, 3)	(-3, 2)	(4, 2)

### 3. 纳什均衡

重复剔除劣势策略能够很简单地获得一个均衡的策略组合，但是在许多实际博弈中，严格劣势策略并不总是存在的，所以不能用这种方法求解。而纳什均衡则存在于广泛类型的博弈中。

在介绍纳什均衡之前，首先定义如下两个概念。

**纯策略：**参与人在给定信息的情况下，只选择一个特定的行动。

**混合策略：**在一个 $I$ 人博弈的策略式表述 $G = \{S_1, S_2, \dots, S_I; u_1, u_2, \dots, u_I\}$ 中，假定参与人 $i$ 有 $K$ 个纯策略： $S_i = \{S_{i1}, S_{i2}, \dots, S_{iK}\}$ ，如果参与人 $i$ 以一个概率分布 $P_i = \{p_{i1}, p_{i2}, \dots, p_{iK}\}$ 来选择对应的策略 $S_{ik}$ ，那么概率分布 $P_i$ 称为参与人 $i$ 的一个混合策略。对于所有的 $k = 1, \dots, K, 0 \leq p_{ik} \leq 1, \sum_1^K p_{ik} = 1$ 。



在纯策略情况下,参与者  $i$  的效用函数  $u_i = u_i \{s_1, s_2, \dots, s_I\}$ , 即对于一个任意的策略组合,每个参与者的收益都是一个确定的值。而在混合策略的情况下,参与者不能获得一个固定的收益,而只能计算得到每一个参与者的期望效用:

$$U_i(P_i, P_{-i}) = \sum_{s \in S} \left( \prod_{j=1}^I P_j(s_j) \right) u_i(s) \quad (3-79)$$

用一个“石头、剪刀、布”的游戏对这两种策略加以说明。在该博弈中,纯策略就是选择“石头、剪刀、布”中的一个,显然这样的选择不是最优的,因为无论参与人选择什么样的纯策略,另一个参与人总能选择一个策略来获得游戏的胜利。更好的办法是每一次选择以一个概率分布执行石头、剪刀或布,这样对手就很难猜出我们的策略。可以证明,当参与人双方都以每个策略按  $1/3$  的概率执行时,双方都不再愿意改变这种概率分布,这种平衡称为“混合策略纳什均衡”。

**定义:** 混合策略  $P^* = (P_1^*, P_2^*, \dots, P_I^*)$  是一个纳什均衡,如果对于所有参与人  $i$  的任意一个混合策略  $P_i$  来说:

$$u_i(P_i^*, P_{-i}^*) \geq u_i(P_i, P_{-i}^*) \quad (3-80)$$

纯策略纳什均衡是满足同样条件的纯策略组合。

纳什均衡表示参与人的最优策略组合,是对其他参与人策略的最佳反应。在一种纳什均衡中,如果每个参与人具有对对手策略的唯一最优反应,那么这种纳什均衡被称为是严格的,即当且仅当它是一种纳什均衡时,  $s^*$  是一种严格均衡,而且对于所有的  $i$  和所有的  $s_i \neq s_i^*$  有:

$$u_i(s_i^*, s_{i-1}^*) > u_i(s_i, s_{i-1}^*) \quad (3-81)$$

由定义可知,严格均衡必然是纯策略均衡。

求解纯策略的纳什均衡最简单的方法是划线法,给定收益矩阵如表 3-5 所示的博弈,首先固定参与人 1,参与人 1 选择策略 C,则参与人 2 的最好选择是策略 C,所获得的收益是 -5,在 -5 用圆圈圈出;如果参与人 1 选择策略 N,则参与人 2 的最优选择是策略 C,所获得的收益是 0,将 0 用圆圈圈出。接着固定参与人 2,参与人 2 选择策略 C,则参与人 1 的最好选择是策略 C,所获得的收益是 -5,将 -5 用方框框出;如果参与人 2 选择策略 N,则参与人 1 的最好选择是策略 C,所获得的收益是 0,将 0 用方框框出。最终,得到策略组合 {C,C} 为纳什均衡。

表 3-5 收益矩阵

1	2	
	C	N
C	( <span style="border: 1px solid black; padding: 2px;">-5</span> , <span style="border: 1px solid black; border-radius: 50%; padding: 2px;">-5</span> )	( <span style="border: 1px solid black; border-radius: 50%; padding: 2px;">0</span> , -15)
N	(-15, <span style="border: 1px solid black; border-radius: 50%; padding: 2px;">0</span> )	(-1, -1)

纳什均衡表示参与人的最优策略组合,但是纯策略的纳什均衡并不一定在每一个博弈中存在。给出这样一个博弈,老板和工人作为博弈的两个参与人,老板的策略为{监督,不监督},工人的策略为{偷懒,不偷懒},二者的收益如表 3-6 所示。

表 3-6 老板、工人博弈收益矩阵

老板	员 工	
	偷懒	不偷懒
监督	(1, -1)	(-1, 2)
不监督	(-2, 3)	(2, 2)

如果选择一个纯策略,假设工人偷懒,老板的最优选择是监督;假设老板监督,工人的最优选择是不偷懒;假设工人不偷懒,老板的最优选择是不监督;假设老板不监督,工人的最优选择是偷懒。在这个博弈中,每一个参与人都想参透对方的策略选择,又不想让对方猜透自己的策略选择,这样就会陷入循环,在这种情况下纯策略的纳什均衡不存在,但是混合策略的纳什均衡是存在的。

在上文提到的监督博弈中给定工人的混合策略为 $(p, 1-p)$ ,老板的混合策略为 $(q, 1-q)$ ,可以利用收益最大法来求解上述的混合策略,该方法通过给定其他参与人的混合策略,自己选择策略的概率分布来使自己的期望收益最大化。给定老板的混合策略,工人的期望收益函数为:

$$-pq + 2(1-p)q + 3p(1-q) + 2(1-p)(1-q) = p(1-4q) - 2q + 2 \quad (3-82)$$

由于工人的期望收益在 $1-4q > 0$ 时随着 $p$ 递增,在 $1-4q < 0$ 随 $p$ 递减,则当 $q < 1/4$ 时,工人的最优选择是 $p=1$ ,即选择偷懒;当 $q > 1/4$ 时,则工人的最优选择是 $p=0$ ,即选择不偷懒;当 $q=1/4$ ,工人无论如何选择都不会对收益产生影响。

同样的,老板的期望收益函数为:

$$pq - (1-q)p - 2q(1-p) + 2(1-p)(1-q) = 6pq - 3q - 4p + 2 \quad (3-83)$$

当 $2p-1 > 0$ 时,老板的期望效益随着 $q$ 递增;在 $2p-1 < 0$ 时,老板的期望效益随着 $q$ 递减。则当 $p > 1/2$ 时,老板的最优选择是 $q=1$ ,即选择监督;当 $p < 1/2$ 时,老板的最优选择是 $q=0$ ,即选择不监督;当 $p=1/2$ ,老板无论如何选择都不会对收益产生影响。

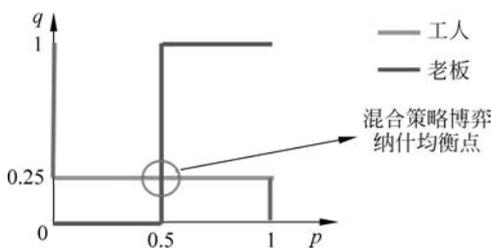


图 3-30 老板和工人的最佳反应函数

图 3-30 展示了参与者在面对对手以不同概率选择策略时的最佳反应决策,这些决策内容为选择概率分布,这些概率分布构成了反应函数。图 3-30 中两个函数的交点代表着纳什均衡点,上文所展示的监督博弈下混合策略下的纳什均衡点为 $(p, q) = (0.5, 0.25)$ ,即工人以 50% 的概率选择偷懒,以 50% 的概率选择不偷懒;而老板以 25% 的概率选择监督,以 75% 的概率选择不监督。

在每个有限的博弈中,至少存在一个纳什均衡,并且纳什均衡的数目是一个奇数。如果把纯策略看作混合策略的退化,则可以表述为在每个有限博弈中,混合策略纳什均衡一定存在。如果一个博弈存在两个纯策略的纳什均衡,那么一定存在第三个混合策略纳什均衡。

在博弈论中还存在着其他的均衡,如子博弈精练的纳什均衡、贝叶斯纳什均衡等,因为篇幅原因在此不做详细介绍,感兴趣的读者可以查阅相关资料。



#### 4. 行为收益模型

在多车协同系统中,因为环境中存在多个交通参与者,所以每个参与者的行为都有可能对环境中的其他车辆产生影响。为了更好地帮助智能网联汽车做出决策,可以建立车辆的收益模型,用博弈论的方法来预测其他交通参与者的行为,接下来将以变道场景和交叉路口场景来展示网联车辆的收益模型建立。

考虑一个两车变道合并的交通场景,变道车辆与正常行驶的车辆之间为了避免碰撞,两车需要互相配合,这可以看作一个博弈,为了求解这个博弈,首先需要定义博弈中的三个元素。变道车辆 A 和从后方接近的正常行驶车辆 B 是博弈的两个参与者,车辆 A 的策略空间为{变道,直行},车辆 B 的策略空间为{加速,减速},接下来还需要定义整个博弈的收益模型,车辆 A 和 B 的收益矩阵如表 3-7 和表 3-8 所示。

表 3-7 车辆 A 收益矩阵

A	加 速	减 速
变道	$a_{11}$	$a_{12}$
直行	$a_{21}$	$a_{21}$

表 3-8 车辆 B 收益矩阵

B	加 速	减 速
变道	$b_{11}$	$b_{12}$
直行	$b_{21}$	$b_{21}$

可以基于各种特征来定义 $\{a_{11}, a_{12}, \dots\}$ ,在这里,考虑基于车辆 A、B 之间以及 A、B 与其他周围车辆之间的间隙来定义。

如图 3-31 所示,车辆 1 为变道车辆,车辆 3 为后方接近车辆,定义如下参数。

$t_1$ : 变道车辆到车道终点的距离/合并车辆的速度。

$t_2$ : 变道车辆 1 与后方接近车辆 3 的碰撞时间。

$t_3$ : 变道车辆 1 与后方车辆 4 的碰撞时间。

$t_4$ : 车辆 2 与车辆 3 的车头时距。

$t_5$ : 车辆 3 与车辆 5 的碰撞时间。

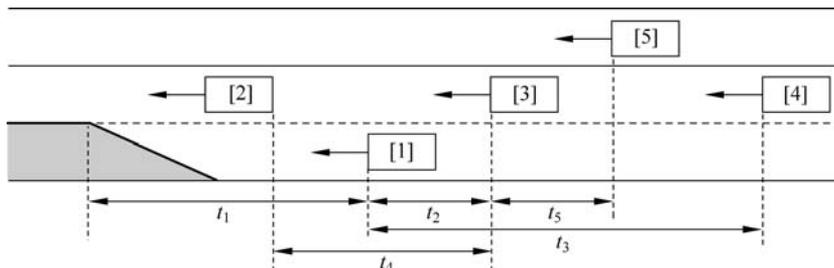


图 3-31 间隙示意图

$\{a_{11}, a_{12}, \dots\}$ 的各元素为  $a_{11} = \theta_1 t_3 + \theta_2, a_{12} = \theta_1 t_2 + \theta_2, a_{21} = a_{22} = \theta_3 t_1, b_{11} = b_{21} = \theta_4 t_5 + \theta_5, b_{12} = \theta_6 t_2, b_{22} = \theta_7 t_4$ 。其中  $\theta_1, \theta_2, \dots, \theta_7$  为需要设计的参数。

设计完一个完整博弈后,可以使用纳什均衡的方法来求解博弈,考虑一个如表 3-9 所示的博弈。

表 3-9 车辆 A、B 博弈收益矩阵

A	B	
	加速	减速
变道	(-1, -1)	(1, 3)
直行	(3, 1)	(0, 0)

在该博弈中,可以用划线法求出存在的两个纯策略的纳什均衡点:(变道,减速)和(直行,加速),也可以用收益最大法来求得所有的纳什均衡点。假设车辆 A 变道的概率为  $p$ ,直行的概率为  $(1-p)$ ; 车辆 B 加速的概率为  $q$ ,减速的概率为  $(1-q)$ ,根据收益函数最大化可以画出车辆 A 和 B 的反应函数,如图 3-32 所示。

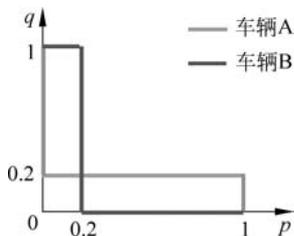


图 3-32 车辆 A 和 B 的最佳反应函数

得到三个纳什均衡点分别为  $(p_1, q_1) = (1, 0)$ 、 $(p_1, q_1) = (0, 1)$ 、 $(p_1, q_1) = (0.2, 0.2)$ ,其中前两个为纯策略纳什均衡,第三个为混合策略纳什均衡。

下面考虑一个交叉路口通过的交通场景,正如 3.3.2 节所示,交叉路口有着复杂的冲突关系,如果把所有的车辆都纳入考虑,将建立一个非常复杂的博弈,难以求解,下面介绍一个简化的如图 3-33 所示的交叉路口。

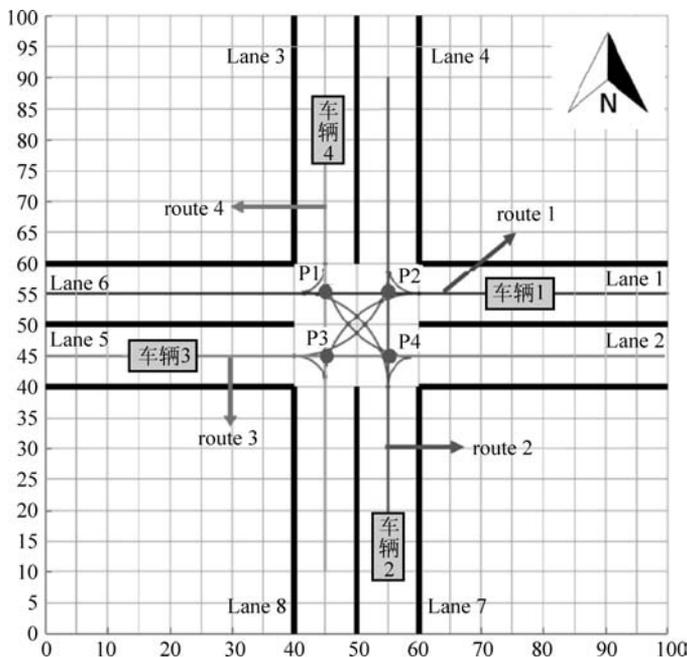


图 3-33 简化的交叉路口交通流关系图



在简化的场景下,只考虑车辆的直行,这样博弈中将最多包含三个参与人,以图 3-33 中的车辆 1 为例,考虑有冲突关系的车辆为 2 和 4,则博弈的参与人为{车辆 1, 车辆 2, 车辆 4},每一个参与人的策略为加速度大小,策略空间为 $\{-3, -2, -1, 0, 1, 2, 3\}$ 。

收益模型的确定有三个指标,分别为安全指标、速度指标、舒适指标。各个指标的具体定义如下所示。

(1) 安全指标。

$$GTH = TAC_{\text{rear}} - TDC_{\text{front}} \quad (3-84)$$

GTH 表示前车离开时间  $TDC_{\text{front}}$  与后车到达时间  $TAC_{\text{rear}}$  的时间差。

(2) 速度指标。

$$\Delta v = v^{t+1} - v^t \quad (3-85)$$

$$v^{t+1} = v^t + a^{\text{desired}} \Delta t \quad (3-86)$$

$$v_{\min} \leq v^{t+1} \leq v_{\max} \quad (3-87)$$

其中:  $v^t$  为目标车辆的当前速度;  $v^{t+1}$  为下一个时刻的速度,是所获得的期望加速度的估计值,目标车辆的速度被限制在 $[v_{\min}, v_{\max}]$ 。

(3) 舒适指标。

$$\Delta a = a^{\text{desired}} - a_{\text{veh}} \quad (3-88)$$

在通过路口时,频繁的加减速会使乘客感到不适,所以用加速度的变化来作为舒适度的指标。 $a^{\text{desired}}$  表示车辆选择的策略, $a_{\text{veh}}$  表示车辆目前的加速度。

收益函数的定义如下所示:

$$f = \alpha \cdot \text{std}_s(GTH) + \beta \cdot \text{std}_v(\Delta v) + \gamma \cdot \text{std}_c(\Delta a) \quad (3-89)$$

$$\text{std}_s(GTH) = \begin{cases} -100, & GTH < 1 \\ 0.25(GTH - 1), & 1 \leq GTH \leq 5 \\ 1, & GTH > 5 \end{cases} \quad (3-90)$$

$$\text{std}_v(\Delta v) = \frac{1}{6}(\Delta v + 3), \quad -3 < \Delta v < 3 \quad (3-91)$$

$$\text{std}_c(\Delta a) = \begin{cases} 1 - 0.5(\Delta a), & \Delta a \leq 3 \\ 0, & \Delta a > 3 \end{cases} \quad (3-92)$$

其中:  $\alpha, \beta, \gamma$  为权重系数,反映了对三种性能指标的偏好,  $\text{std}_s(GTH), \text{std}_v(\Delta v), \text{std}_c(\Delta a)$  三个函数是为了将 GTH、 $\Delta v$ 、 $\Delta a$  三者的影响标准化。

## 习题

1. 简述车联网的主要部分构成。
2. 简述 DSRC 和 C-V2X 的异同点。
3. 设三元线性码  $L$  的生成矩阵为  $\mathbf{G} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix}$ , 求码长  $L$  的最小距离。
4. 设  $C$  为定义在整数域  $F_3 = \{0, 1, 2\}$  上的线性码, 其生成矩阵为  $\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & 0 & 1 & 1 \end{bmatrix}$ ,

利用伴随式编码对矢量 2121 和 1201 进行译码。

5. 设五元线性码  $C$  的生成矩阵为  $G = \begin{bmatrix} 1 & 2 & 4 & 0 & 3 \\ 0 & 2 & 1 & 4 & 1 \\ 2 & 0 & 3 & 1 & 4 \end{bmatrix}$ , 请写出标准生成矩阵和一致

校验矩阵。

6. 根据多包传输推导过程, 写出式(3-64)中  $\Phi_{s_i}$  的具体表达式。

7. 完成前车跟随式、前车-领航者跟随式、双向跟随式、双向-领航者跟随式通信拓扑结构的矩阵描述。

8. 完成如图 3-25 所示的典型双车道丁字路口交通流冲突图和交通流冲突补图。

9. 试分析为什么每个有限策略式博弈均具有混合策略均衡。

## 参考文献

- [1] 《车联网网络安全白皮书(2017)》发布[J]. 中国信息安全, 2017(10): 31.
- [2] KENNEY J B. Dedicated short-range communications (DSRC) standards in the United States[J]. Proceedings of the IEEE, 2011, 99(7): 1162-1182.
- [3] 南洋, 董馨, 陈博, 等. C-V2X 技术在智能网联汽车上的应用场景研究[J]. 汽车文摘, 2019(9): 8-12.
- [4] XU Q, SENGUPTA R. Vehicle-to-vehicle safety messaging in DSRC[C]//InProc. of the 1st ACM Workshop on Vehicular Ad-hoc Networks. America: ACM Press, 2004: 19.
- [5] 吴冬升. 美国车联网(V2X)发展现状分析[J]. 智能网联汽车, 2019(5): 66-72.
- [6] 郭海陶. 智能交通专用短程通信(DSRC)关键技术与应用研究[D]. 广州: 华南理工大学, 2010.
- [7] 贾敏. OFDM 通信系统的信道估计及多址技术研究[D]. 哈尔滨: 哈尔滨工业大学, 2010.
- [8] 吴骏. 智能交通系统中的信息处理关键技术研究[D]. 天津: 天津大学, 2006.
- [9] REZGUI J, CHERKAoui S, CHAKROUN O. Deterministic access for DSRC/802. 11p vehicular safety communication[C]//Wireless Communications & Mobile Computing Conference, IEEE, 2011.
- [10] 张杰. C-V2X 与智能车路协同技术的深度融合[J]. 中兴通讯技术, 2020, 26(1): 19-24.
- [11] VUKADINOVIC V, BAKOWSKI K, MARSCH P, et al. 3GPP C-V2X and IEEE 802. 11p for vehicle-to-vehicle communications in highway platooning scenarios [J]. Ad Hoc Networks, 2018: S157087051830057X.
- [12] 赵军辉, 彭巍, 史家康. 应用于智能运输系统的专用短程通信[J]. 运输经理世界, 2011(9): 98-99.
- [13] 陈沛吉. C-V2X 技术在城市轨道交通协同驾驶中的设计与实现[D]. 北京: 北京邮电大学, 2019.
- [14] INC Q T. 5G Vision[J]. Telecommunications Network Technology, 2015.
- [15] CHEN S, HU J, SHI Y, et al. LTE-V: A TD-LTE-Based V2X solution for future vehicular network [J]. IEEE Internet of Things Journal, 2017, 3(6): 997-1005.
- [16] HALINGER G, HOHLFELD O. The gilbert-elliott model for packet loss in real time services on the internet[C]. Gi/itg Conference-measurement, VDE, 2011.
- [17] 麻磊. 带有时延和丢包的网络控制系统的切换控制算法研究[D]. 北京: 北京交通大学, 2012.
- [18] 刘娜. 丢包网络环境下的约束非线性系统的 MPC 研究[D]. 重庆: 重庆邮电大学, 2018.
- [19] NILSSON, JOHAN. Real-time control systems with delays [J]. D. Dissertation, Lund Institute of Technology, 1998.
- [20] MINERO P, COVIELLO L, FRANCESCHETTI M. Stabilization over Markov feedback channels: the general case[J]. IEEE Transactions on Automatic Control, 2013, 58(2): 349-362.
- [21] 郑洋. 基于四元素构架的车辆队列动力学建模与分布式控制[D]. 北京: 清华大学, 2015.



- [22] WEST, DOUGLAS B. Introduction to graph theory[M]. Hong Kong: Longman, 1985.
- [23] BÉIA B. Modern graph theory[J]. Graduate Texts in Mathematics, 1998: 184.
- [24] PETERSEN J. Die Theorie der regulären graphs[J]. Acta Mathematica, 1970, 15(1): 193.
- [25] TARJAN R. Depth-first search and linear graph algorithms[J]. Switching and Automata Theory, 1996. IEEE Conference Record of Seventh Annual Symposium on, 1971(2): 114-121.
- [26] CAYLEY A. On the Theory of the analytical forms called trees[J]. Philosophical Magazine, 1966, 18(4): 172-176.
- [27] CHANG L, CHUNG W. Distributed conflict resolution for connected autonomous vehicles[J]. IEEE Transactions on Intelligent Vehicles, 2017.
- [28] KHAYATIAN M, MEHRABIAN M, DEDINSKY R. A survey on intersection management of connected autonomous vehicles[J]. ACM Transactions on Cyber-Physical Systems, 2020.
- [29] MYERSON R B. Game theory: analysis of conflict[M]. Boston: Harvard University Press, 1997.
- [30] OSBORNE M J, RUBINSTEIN A. A course in game theory[J]. Economica, 1992, 63(249).
- [31] ELHENAWY M, ELBERY A A, HASSAN A A, et al. An intersection game-theory-based traffic control algorithm in a connected vehicle environment [C]. IEEE International Conference on Intelligent Transportation Systems. IEEE, 2015.
- [32] SCHWARTING W, PIERSON A, ALONSO M J, et al. Social behavior for autonomous vehicles[J]. Proceedings of the National Academy of Sciences, 2019, 116(50).
- [33] WEI H, MASHAYEKHY L, PAPINEAU J. Intersection management for connected autonomous vehicles: a game theoretic framework [C]. IEEE International Conference on Intelligent Transportation Systems. IEEE, 2018.
- [34] CHENG C, YAO D, ZHANG Y, et al. A vehicle passing model in non-signalized intersections based on non-cooperative game theory [C]. 2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE, 2019: 2286-2291.
- [35] LI N, KOLMANOVSKY I, GIRARD A, et al. Game theoretic modeling of vehicle interactions at unsignalized intersections and application to autonomous vehicle control [C]. Annual American Control Conference.
- [36] YANG Z, HUANG H, YAO D, et al. Cooperative driving model for non-signalized intersections based on reduplicate dynamic game [C]. 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2016.
- [37] KITA H. A merging-giveway interaction model of cars in a merging section: a game theoretic analysis[J]. Transportation Research Part A Policy & Practice, 1999, 33(3-4): 305-312.
- [38] ALI Y, ZHENG Z, HAQUE M M, et al. A game theory-based approach for modelling mandatory lane-changing behaviour in a connected environment[J]. Transportation Research Part C Emerging Technologies, 2019, 106: 220-242.