Ph.D. Dissertation

Context-Aware Navigation for Safe and Efficient Maneuver Control of Moving Objects in Cyber-Physical Systems

MUGABARIGIRA BIEN AIME

Department of Electrical and Computer Engineering

The Graduate School

Sungkyunkwan University

Ph.D. Dissertation

Context-Aware Navigation for Safe and Efficient Maneuver Control of Moving Objects in Cyber-Physical Systems

MUGABARIGIRA BIEN AIME

Department of Electrical and Computer Engineering

The Graduate School

Sungkyunkwan University

Context-Aware Navigation for Safe and Efficient Maneuver Control of Moving Objects in Cyber-Physical Systems

MUGABARIGIRA BIEN AIME

A Ph.D. Dissertation Submitted to
the Department of Electrical and Computer Engineering and
the Graduate School of Sungkyunkwan University
in partial fulfillment of the requirements
for the degree of Ph.D. in Engineering

April 2025

Supervised by

Jaehoon Jeong

Major Advisor

This certifies that the Ph.D. Dissertation
of MUGABARIGIRA BIEN AIME is approved.

Committee Chair:

Committee Member 1:

Committee Member 2:

Committee Member 3:

Major Advisor:

The Graduate School
Sungkyunkwan University
June 2025

Contents

List of Tables

List of Figures

Abstract

Context-Aware Navigation for Safe and Efficient Maneuver Control of Moving Objects in Cyber-Physical Systems

The growth of seamless moving nodes in cyber-physical systems such as vehicles and UAVs demands a safe framework to ensure the quality of their intended services. This dissertation studies the context-aware navigation mechanism for terrestrial vehicles and unmanned aerial vehicles (UAVs).

A connected network of automated vehicles on roads can increase the driving safety of driverless vehicles (i.e., autonomous vehicles). The critical level of dangerous situations on the road while driving can be increased by the speed, orientation, and traffic density of the vehicles involved. Therefore, there is a need for a maneuvering mechanism that handles both the current driving vehicle and the oncoming vehicles headed toward an emergency zone (e.g., road hazard and road accident spot). In this thesis, we present a context-aware navigation protocol (CNP) that enhances the safety of vehicles driving on urban roads. Firstly, CNP includes a collision avoidance module that analyzes the driving risks to determine the necessary maneuvers in dangerous situations. Secondly, CNP establishes a collision mitigation strategy that limits the severity of collision damages in hazardous roads during non-maneuverable scenarios. The performance results show that CNP can reduce communication overhead from a baseline scheme by up to 60% while the risk of road collisions is less than 5%.

Unmanned Aerial Vehicles (UAVs) have recently been attracting growing attention from researchers because of their properties, which are highly suited to mission-oriented applications. This thesis proposes a Collision-Avoidance Navigation Algorithm (CANA) that uses cooperative flight controls in heterogeneous heavy-traffic UAV networks. It has an autonomous swarm-based cooperation and coordination design that supports context sharing for a large amount of UAV traffic mobility. This UAV navigation's sensing and communication allow it to detect and avoid possible collisions by establishing detour paths. Our algorithm's simulation performance shows a reduction in communication control overhead that reaches 66.6% of the legacy scheme's overhead while keeping the lowest collision risk compared with the baselines.

The Graph Convolutional Networks (GCN) have emerged in Intelligent Transportation Systems (ITS) to enhance safety, efficiency, and traffic predictability, which benefits vehicular navigation services in road networks. This study proposes a Machine Learning-Assisted Self-Adaptive Interactive Navigation Tool (ML-SAINT) for efficient parcel delivery in road networks. It investigates the ability of GCN in predicting the best route that satisfies the requirements of parcel delivery. It proposes rapid and accurate spatial-temporal traffic information perception to predict optimal courier delivery schedules. The simulation results show that this proposed scheme reduces up to 44.19% of the travel delay of delivery vehicles compared to baseline schemes.

Keywords: Intelligent transportation systems, context-aware navigation, vehicular networks, collision avoidance, graph convolution networks

Chapter 1

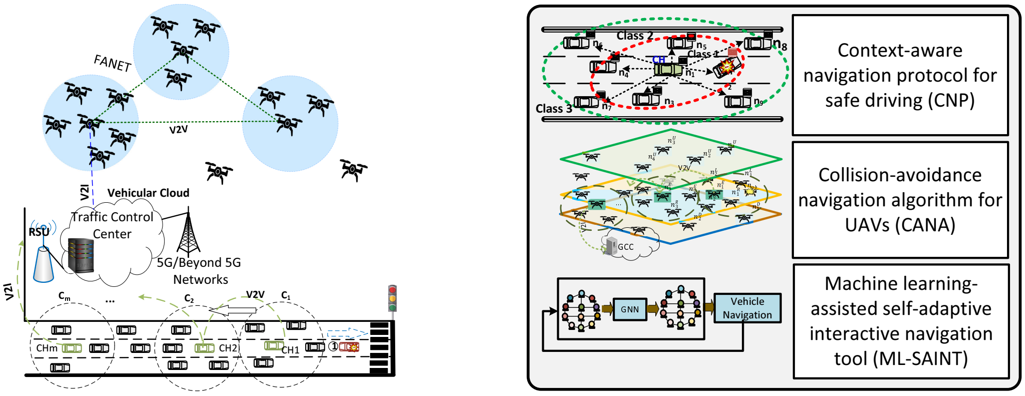
Introduction

Vehicular cyber-physical systems (VCPs) have been receiving increasing attention as they provide means for efficient and safe transportation systems [1]. The rapid development of sensing and perception facilities, such as networking technologies, paved the way for the VCPs. The increasing number of connected ground nodes (vehicles) and aerial nodes (unmanned aerial vehicles (UAVs)) is benefiting from the potential of VCPs. However, this increases the demand for safety-preserving mechanisms to ensure safe and robust services of these systems.

By moving objects in this thesis, we refer to intelligent autonomous vehicular entities like vehicles and UAVs. The moving objects in VCPs comprise ground and aerial vehicles, as illustrated in Fig. 1.1, forming an intelligent transportation system (ITS) where nodes can make informed navigation decisions. For their efficiency, a technological infrastructure that enabled their computing, communication, and control was put in place. This enhances their safety, efficiency, and convenience. Fig. 1.1a depicts a high-level illustration of communication in ground and aerial vehicles. The ground vehicles' autonomous driving enables advanced systems such as adaptive cruise control, lane departure warning, and cooperative driving to enhance road safety. For aerial vehicles (i.e., UAVs, future air taxis), ITS involves automated flight control, collision avoidance systems, and integration with air traffic management at the ground control centers (GCC). The core idea is to make these vehicles smarter and more connected to their environment and infrastructure.

The autonomous driving major anticipated benefit is to improve driving safety. However, they face a lack of public acceptance stemming from safety concerns [2, 3]. Several remedies to those concerns resulted in the development of road safety preserving systems such as driver warning systems (e.g., night vision, lane departure, and adaptive cruise control) and automated driver-system cooperation systems (e.g., low-speed automated driving and forward collision prevention) [4]). On the other hand, several use cases of UAVs have also been ongoing. Several recent studies resulted in a number UAV applications such as the mission-oriented UAVs (i.e., patrolling, recording, surveillance and inspections, parcel delivery, disaster management such as prediction, assessment and response, and the air taxis which in future will enhance the urban transportation) [5, 6, 7, 8, 9, 10, 11]. When this different mission will be operating in the same airspace, the risks of crashes and interference will grow higher, therefore necessitating a safe coordinated context-awareness mechanism to navigate them.

This thesis studies a context-aware navigation for safe maneuver control of moving objects in cyber-physical systems. It explores the possibilities of urban transportation, analyzes collision risks, and provides safety-preserving maneuvers as illustrated in Fig. 1.1b. Firstly, it presents a context-aware navigation protocol (CNP) with two critical features, namely, collision avoidance and collision mitigation. Assuming a fully connected environment supported by vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and the infrastructure-to-vehicle (I2V) communication, it reduces the communication overhead and the risk of crashing. It designs a communication protocol for connected and automated vehicles' driving, considering communication as a means of



(a) A network of moving objects

(b) Context-awareness of moving objects

Figure 1.1: High-level architecture of moving objects in vehicular cyber-physical systems.

safety-preserving schemes. Secondly, it models a collision-avoidance navigation algorithm (CANA) for UAV flying networks with heavy traffic, which proposes to avoid collisions in airspace autonomously. CANA proposes an autonomous aerial collision avoidance mechanism benefiting from air-to-air and air-to-ground communications. From air to ground communication data, the ground control center (GCC) predicts the flying settings that avoid collision among UAVs, and minimizes the detour cost required by the UAVs' maneuvers. CANA proposes heuristic solutions using acceptable approximations to remedy the complexity of the Unmanned Aerial Vehicles (UAVs) collision-avoidance algorithms optimization. Thirdly, this thesis explores the potential of machine learning to boost the application-specific vehicular navigation by designing a machine learning-assisted self-adaptive interactive navigation tool (ML-SAINT) for efficient parcel delivery in road networks. It designs a distributed traffic algorithm that integrates a graph attention-aware machine learning model for application-specific traffic forecasting and cost-efficient last-mile delivery route planning.

The rest of this dissertation is organized as follows:

- Chapter 2 studies a context-aware navigation protocol (CNP) for safe driving in vehicular cyber-physical systems.

- Chapter 3 proposes a collision-avoidance navigation algorithm (CANA) for UAV flying networks with heavy traffic.
- Chapter 4 proposes a machine learning-assisted self-adaptive interactive navigation tool (ML-SAINT) for efficient parcel delivery in road networks.
- Chapter 5 concludes this dissertation and suggests challenges and future issues for moving objects in cyber-physical systems.

Chapter 2

CNP: Context-Aware Navigation Protocol for Safe Driving in Vehicular Cyber-Physical Systems [1]

The benefits of advancements in vehicular cyber-physical systems have increased the possibility of connected vehicles. These vehicles require an intelligent ability to sense, perceive, and decide on the situations arising on the road. This chapter proposed a context-aware navigation protocol (CNP) for safe driving using risk assessment-based maneuver planning. Two features, namely, collision avoidance and collision mitigation, are proposed to enhance road safety. Assuming a fully connected environment supported by vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and infrastructure-to-vehicle (I2V) communication, the designed CNP reduces the risk of crashing on a road with obstacles.

It studies a strategy to guide vehicles that could be potentially affected by obstacles towards a safe lane to avoid out-of-sight obstacles using wireless communications. Firstly,

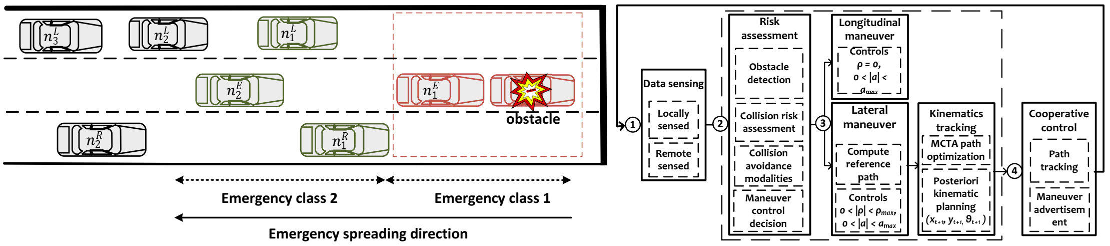
[1] The work presented in this chapter has been published in [12].

it proposes a collision avoidance based on the cost of lane quality to define possible vehicle maneuvers. Secondly, for non-avoidable obstacles, a collision mitigation is proposed based on the energy transfer and the number of emergency vehicles. The corresponding collision avoidance and maneuvers mitigation are generated and sent to related vehicles grouped in clusters via cluster heads (CH). It proves the proposed method's effectiveness by comparing it with sensor-based (i.e., LIDAR) and the networked collision avoidance system (NCAS).

2.1 Introduction

Vehicle networking enables interoperability among vehicles, drivers, pedestrians, and road network infrastructures, which directly impacts the transportation safety [13]. Driverless vehicle technology has attracted increasing research interest in both academia and industry due to its potential to substantially enhance vehicular transportation in terms of accessibility, safety, and convenience. One of the most anticipated benefits of self-driving vehicles is the lack of a driver's errors that lead to accidents. Human drivers are responsible for 94% of critical events that cause crashes, while the environment, vehicles, and unknown-related causes cumulatively account for the remaining 6% [14]. The expected increase in driving safety is one of the main motivations to invest in autonomous driving research.

Although autonomous vehicles promise to improve driving safety, they face a lack of public acceptance stemming from safety concerns [2, 3], and significant research initiatives have gone toward addressing those concerns. Over the past three decades, a vehicle has reached a new technological maturity with driver assistance systems (DAS) [4]. These include driver warning systems (e.g., night vision, lane departure, and adaptive cruise control) and automated driver-system cooperation systems (e.g., low-speed automated



(a) Complex Urban Driving.

(b) Solution Steps.

Figure 2.1: Illustration of the complexity of driving with the presence of an obstacle in the road.

driving and forward collision prevention). However, human driving errors can be fully eradicated with the introduction of autonomous driving. Despite recent research efforts, there is still a need for safety-enabling systems for vehicles to autonomously cooperate in assessing driving risk outside of the line of collision. These systems need to safely guide vehicles to avoid collisions in the road without jeopardizing their safety and with minimal or no effect on their overall trajectory. A system that integrates road infrastructures with vehicles to have computing, communication, and control abilities is commonly known as a vehicular cyber-physical system (VCPS) [1].

This chapter presents a context-aware navigation protocol (CNP) to increase driving safety in VCPS. The CNP is a protocol that solves a complex urban driving scenario as shown in Fig. 2.1a. When a vehicle abruptly stops in the middle of a lane, it creates an unexpected driving situation for any following vehicles. The reaction of e_1 may impact the driving of either e_2, l_1 or r_1, which in turn will need to react, thereby creating a more complex driving scenario. We name “emergency zone” the part of the road where the vehicles need to maneuver to bypass an obstacle, and those vehicles are “emergency vehicles”. This system distinguishes two classes of emergency vehicles. A directly concerned vehicle e_1 is emergency class 1, while indirectly affected ones are emergency class 2.

Through modular steps described in Fig. 2.1b, CNP provides a relevant driving solution for vehicles driving in hazardous situations. First, a data sensing which consists of sending and receiving the self-sensed and remote-sensed mobility information according to the CNP communication protocol described in Section 2.4.1. Second, a risk assessment that analyzes the received data to identify abnormal behavior in the driving environment and collision avoidance modalities. Third, a path maneuver that defines the appropriate trajectories changes either longitudinally or laterally across the road for the concerned vehicles. Lastly, a path tracking, which keeps the maneuver path until the maneuvered vehicle reaches the target without deviation. The path maneuvers are shared with the neighbors via communication for awareness purposes.

The major contributions of this work are summarized as follows:

- **The design of networked context awareness navigation protocol (CNP)**: It is a protocol that uses an IPv6 Neighbor Discovery (ND) option to convey vehicles' mobility information. This protocol enables cooperative sensing through light-weight message sharing. It renders ambient awareness by quickly analyzing sensed data to determine anomalous behaviors such as abrupt slowing and sudden lane changes. Upon the detection of emergency risks, the CNP takes further measures to quickly adjust a vehicle's trajectory (see Section 2.4).

- **Lane quality-based collision avoidance**: We develop a lane metric indicating the state of lane collision risk prior to emergency vehicle maneuver decision. It permits vehicles to maneuver towards the safest lane and lets their maneuver have the minimal impact on the overall road safety. Through the probabilistic collision risk assessment, we deduce the best maneuver for a vehicle in risk of collision with an obstacle in the road (see Section 2.4).

- **A simulation-based evaluation of the proposed model**: In order to demonstrate the efficiency and usefulness of the proposed model in this chapter, we implemented a vehicular simulation and evaluated it under various conditions. The evaluation results show that the CNP can reduce the risk of collision with a little communication overhead compared to the baselines (see Section 2.5).

Note that this is an improvement of the preliminary work presented in our previous paper [15].

The rest of this chapter is organized as follows. Section 2.2 summarizes the previous work related to our study. Section 2.3 makes the notation definitions and assumptions used in this chapter, and defines the problem that this work solves as well. Our network-based vehicle tracking mechanism is described in detail in Section 2.4. Section 2.5 validates our CNP mechanism by comparing it with other sensing and tracking mechanisms. Finally, we conclude this chapter along with future work in Section 2.6.

2.2 Related Work

This section summarizes the previous work focusing on driving safety for connected and automated vehicles.

Beginning with the DARPA urban challenge in 2007 [16], several studies have attempted to design autonomous vehicles that can be used in real driving. Autonomous vehicles can make meaningful assessments according to the significance of any perceived data, and they can also predict future events and make proactive decisions to avoid accidents [17]. Proper awareness and assessment mechanisms are needed to respond to the wide variety of situations that arise in real road traffic.

To avoid collisions in autonomous driving, several mechanisms have been proposed, including both sensor-based and non-sensor-based approaches. A typical sensor-based mechanism is the lane-level beacon-less, infrastructure-less, and GPS-less cooperative collision avoidance (BIG-CCA) framework proposed by Chen and Chou [18]. This mechanism uses V2V communication to warn vehicles in the same lane of any danger. Another model proposed by [19] uses a platooning paradigm to model accidents involving a platoon equipped with a warning notification system. A parallel autonomy framework uses a nonlinear model predictive control to compute a safe trajectory for an automated vehicle based on human input [20]. The cooperative collision avoidance (CCA) systems enable vehicles to cooperate to achieve driving safety.

CCA mechanisms include platooning [21], a networked collision avoidance system [22], an agent-based situational assessment [23], and a feature-based cooperative perception framework (F-Cooper) [24]. In the platooning, vehicles follow the same route and drive closely, enabling them to cooperate through low-latency data delivery. The networked collision avoidance system (NCAS) allows vehicles to work together by broadcasting driving information over a shared channel. A central controller collects the driving data and manages vehicles that behave according to an agent-based modeling [23]. F-Cooper is an object detection model that was proposed to remedy the limitations caused by network bandwidth and the constraints of autonomous applications. The cooperative adaptive cruise control proposed by [25] regulates the interdistance between vehicles to achieve string stability in a networked control system.

Junietz et al. studied an approach to define the criticality metrics that validate the safety of automated driving [26]. Using the model predictive control (MPC), they define the driving required in a specific situation for real-time trajectory planning and control. Feng et al. reduced the required testing miles required to validate the driving safety performance in autonomous vehicles by adversarial adjustments to the naturalistic driving environment [27]. The model predictive instantaneous safety proposed by [28] analyzes the closeness to collision to guarantee safety in terms of time to collision. Li et al. proposed a collision avoidance (CA)-based risk assessment where metrics such as time to

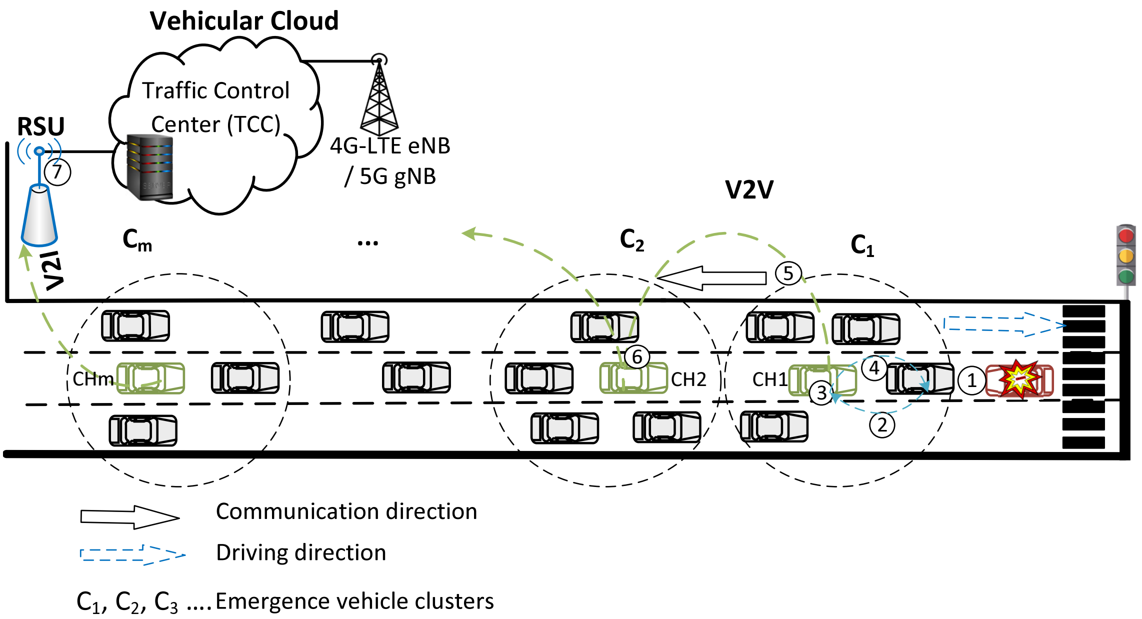


Figure 2.2: CNP networked emergency handle target model.

collision, time to stop, and time to react are combined to define the safety status of driving vehicles [29]. Unlike the prior models, our context-aware navigation protocol provides a possibility to preemptively avoid collision with out-of-sight obstacles through wireless communication by guiding them towards a safe lane. The CNP includes an emergency path planning and tracking strategy based on a minimal contour tracking algorithm proposed in [30] to define the path maneuver for vehicles both in line-of-sight and out-of-sight of the obstacle in the road.

2.3 Problem Formulation

We designed the CNP with two-fold safety oriented objective such as collision avoidance and collision mitigation. The collision avoidance model ensures that the autonomous vehicles drive safely when they are faced with obstacle(s) in a road segment. Meanwhile, the collision mitigation model, which is proposed by this chapter, guarantees a minimal collision impact for unavoidable accident scenarios. In this section, we describe in detail the notations, assumptions, and the problem targeted by our model.

2.3.1 Notation Definitions

Fig. 2.2 shows a networked emergency processing architecture. In this chapter, we adopt a communication protocol wherein vehicles are grouped into m clusters $C = \{C_1, C_2, ..., C_m\}$; the cluster head (CH) is the leader of the cluster. Our model uses the K-mean clustering approach [31] where CH is a cluster-head vehicle which makes the smallest intracluster Euclidean distance from its cluster members. CHs share driving information with each other via Vehicle-to-Vehicle (V2V) communication. Meanwhile, the Traffic Control Center (TCC) maintains traffic statistics and communicates with the vehicles via Vehicle-to-Infrastructure (V2I) communication.

Let an emergency event be any unusual behavior that happens on the road that degrades the safety of vehicles. The emergency driving process described in Fig. 2.2, which avoids collisions of vehicles, consists of the following steps:

Step 1: An emergency event suddenly occurs on the road, becoming a driving obstacle.

Step 2: A cluster member detects and broadcasts the obstacle information to its neighbors.

Step 3: The CH receives and evaluates this information to identify the obstacle collision risk. It calculates a required maneuver for each vulnerable member.

Step 4: The CH then informs the members in step 3 of the obstacles and the appropriate maneuvers.

Step 5: The safety information is shared with neighboring clusters via their CHs.

Step 6: Each CH will take proactive steps to address any safety issues.

Step 7: Through V2I communication, the TCC receives and maintains up-to-date global mobility statistics and calculates safe trajectories for vehicles in the road network.

Now we formulate equations and functions for our CNP. The nodes in vehicular ad hoc networks (VANET) follow a very well-structured path organized in roads and their sub-roads, or lanes. CNP considers a road segment e_{ij} with the number of lanes $l > 1$, where N vehicles in a set $\mathcal{V}$ are traversing the road segment from entrance i toward exit j intersections at a particular instant time t. At time t, a vehicle $n_i, i = \{1, ..., N\}$ state is defined as

$$s = \begin{bmatrix} x_i & y_i & v_i & \theta_i \end{bmatrix}^T, \tag{2.1}$$

where (x_i, y_i) is its position, v_i is its speed, θ_i is its moving direction, and its trajectory control input is defined as

$$u = \begin{bmatrix} \rho_i & a_i \end{bmatrix}^T, \tag{2.2}$$

where the steering angle is ρ_i and acceleration is a_i. For a vehicle with a distance L between two axles of a wheelbase [30], its θ_i can be derived by

$$\dot{\theta}_i = \frac{v_i}{L} \tan(\rho_i). \tag{2.3}$$

The dynamic that specifies n_i's future behavior is defined by the following nonlinear differential equation:

$$\dot{s} = f(s, u(t)). \tag{2.4}$$

A vehicle at risk of collision is an *emergency vehicle*, n_{em}, and the one it is about to collide with is an *obstacle*, n_{ob}. A lane where n_{em} is positioned is an *emergency lane*, L_E. The n_{ob} can either be moving (i.e., moving extremely slowly) or resting (i.e., completely stopped). A resting obstacle can only be defined by its position $\begin{bmatrix} x_{ob} & y_{ob} \end{bmatrix}^T$.

2.3.2 Assumptions

We take into consideration the following assumptions in our CNP, which investigates the traffic flow λ for a road segment in a defined time slot:

- Each vehicle is allowed to share both local and remote kinematics information.
- Periodically, each vehicle broadcasts its own mobility information, such as speed, direction, and position, which allows the cluster header to identify its kinematics in real time.
- CNP assumes that a non-sliding vehicle's wheels roll with non-holonomic constraints, thus allowing a vehicle's maneuvers to be defined without the impacts of friction.
- Vehicles in a road segment are assumed to be moving in the same direction from intersections $i \rightarrow j$ or at rest. A vehicle is said to be at rest when its speed $v = 0$.
- An acceleration of a vehicle a has a minimum and maximum bound, i.e., $a \in [a_{min}, a_{max}]$, where a_{min} and a_{max} are the lowest and highest accelerations attainable by a vehicle, respectively.
- A steering angle of a vehicle ρ has a minimal and maximum bound, i.e., $\rho \in [\rho_{min}, \rho_{max}]$, where ρ_{min} and ρ_{max} are the minimal and maximal angles, respectively, that a vehicle can steer its wheels.

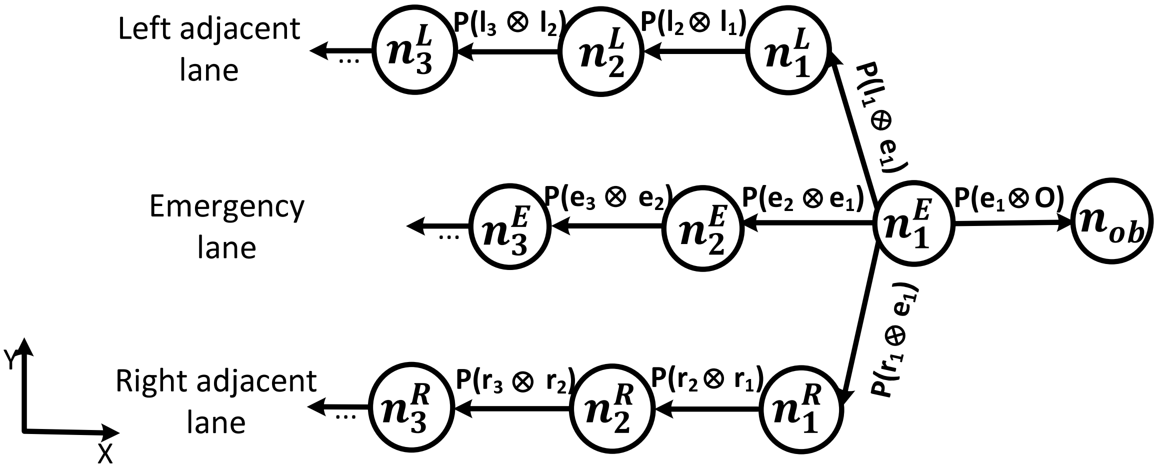


Figure 2.3: Emergency vehicles collision probabilities graph.

2.3.3 Problem Definition

Emergency Driving Handler

We illustrate a road network as a graph $G = (\mathcal{V}, E, c)$ constructed by a set of vehicles $\mathcal{V}$ that defines the vertices, and a set of edges E with a cost equivalent to the collision risk associated with a neighboring vehicle. The function $c : E \rightarrow \mathbb{W}$ assigns the cost to each edge as shown in Fig. 2.3. Given a graph $G = (\mathcal{V}, E, c)$, CNP computes a feasible lateral driving path for a vehicle when an obstacle is identified in its safe driving direction.

Problem 2.3.1. ***Emergency path planning problem:*** *It consists of determining a path P_{em} for a vehicle $n_i \in \mathcal{V}$ such that the collision probability P_{col} associated with the relative kinematics of adjacent vehicles shall always be low during the entire maneuver time Δ_t.*

Let $v_p(t)$ be the speed of the front vehicle n_p (called the parent vehicle) driving or resting ahead and $v_c(t)$ be the speed of the following vehicle n_c (called the child vehicle) at time t. Let Δ_t be the required maneuver change time of the following vehicle n_c. Let β be the Euclidean distance between vehicle n_p and n_c.

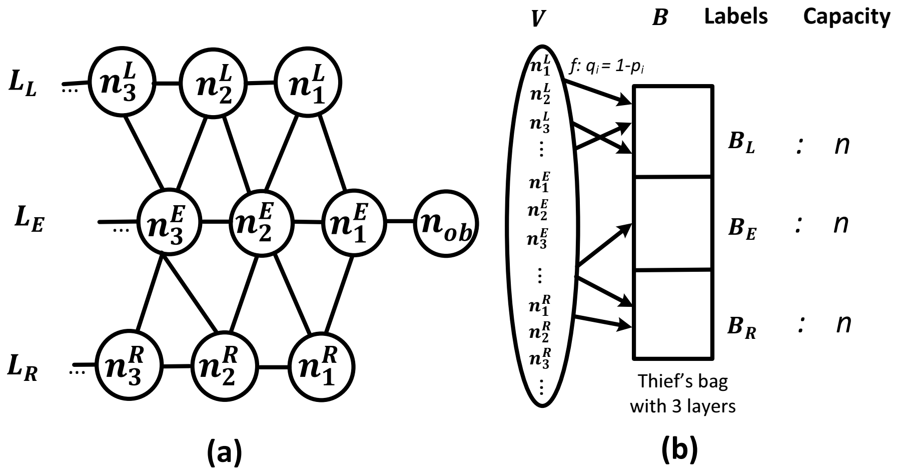


Figure 2.4: An illustration of NP-Hardness for the emergency driving problem.

The collision probability P_{col} for two vehicles n_p and n_c during a maneuver time Δ_t is calculated as

$$P_{col}\left[\int_{t}^{t+\Delta_t} v(t)\,\mathrm{d}t + \epsilon_v \geq \beta\right], \tag{2.5}$$

where $v(t) = |v_c(t) - v_p(t)|$ is the relative speed, and $\epsilon_v \sim \mathcal{N}(0,\, \sigma)$ is the speed measurement error, $\sigma = \{1, 2, 3, ..., 10\}\ km/h$. The trajectory distance $d_i(t)$ of an emergency vehicle n_i during the maneuver process is defined as

$$d_i(t) = \int_{t}^{t+\Delta_t} \left\{ v(t) + \int_{t}^{\tau} \big(a_i u + \epsilon_a\big)\mathrm{d}u \right\} \mathrm{d}t, \tag{2.6}$$

where $v(t)$ is the speed function, u is the maneuver control input according to (3.2), a is the acceleration, and ϵ_a is the acceleration error at time t.

NP-Hardness of Emergency Driving Problem

The solution of Problem 4.3.1 defines a safe path of n_1^E in a situation shown by Fig. 2.4 (a). The forward driving of n_1^E will collide with an obstacle n_{ob} and can collide with n_2^E if it abruptly slows its speed down. Moving to the left lane can cause a collision with n_1^L and its maneuvers to the right lane can cause a collision with n_1^R. Whatever the driving decision of n_1^E is, it can affect the maneuver of at least one among its adjacent vehicles. Suppose that n_1^E moves to the left lane by stimulating the neighbor n_1^L's maneuver decision. Any maneuver of n_1^L should also consider the kinematics of its adjacent vehicles n_2^E and n_2^L. Those decisions can also result in the involvement of exponentially large-space searching to find out a safest path for each vehicle in either a road segment or intersection.

Given a sequence of vehicles, each at the risk of colliding with neighbors, the decision-making problem to find vehicles' safe maneuvers is an NP-complete problem. Assume that the road e_{ij} is a thief's bag with three layers B_L, B_E, and B_R each with capacity n, respectively, representing the left lane of the emergency lane, the emergency lane and the right lane of the emergency lane, as shown in Fig 2.4 (b). We demonstrate that the knapsack problem [32] is a special case of the emergency path planning problem. The knapsack problem is a well-known NP-complete problem. It is infeasible to find a polynomial-time algorithm that can safely control the maneuvering of many vehicles on the road. By knowing the safe lane quality among adjacent vehicles (defined in Section 2.4.3) as a weight for knapsack, linear program (LP) relaxation can alleviate the difficulty of the problem of emergency maneuver decision. The emergency driving decisions set S^* that maximizes the safe driving profit, is defined as function

$$\begin{aligned} f: \quad & V \rightarrow B \\ & \text{such that } S^* \leftarrow \arg\max \sum_{i=1}^{N} S_i, \end{aligned} \tag{2.7}$$

where S_i is the safe lane quality of each vehicle as a safety metric in the maneuver lane computed according to (4.14) and (2.13) in Section 2.4.3.

Minimized Collision Scenario

Our approach minimizes the collision impact in two ways. First, we ensure that the colliding vehicles crash with as little energy transfer as possible. Second, we minimize the number of vehicles affected by this accident through collaborative maneuver control.

Problem 2.3.2. ***Collision impact minimization problem:*** *When a child vehicle n_c detects a dangerous situation that is too close to be avoided, the collision impact shall be minimized so that the involved vehicles may suffer minimal damage.*

Equivalent Energy Speed (EES) [33] is computed as follows when a child vehicle n_c collides with a front, parent vehicle n_p:

$$EES = \hat{v}_c - v_c = \frac{2m_p}{m_c + m_p}(v_p - v_c), \tag{2.8}$$

where m_c and m_p are the masses of n_c and n_p, respectively, v_c and v_p are the speeds of n_c and n_p, respectively, and $\hat{v}_c$ is the speed of n_c after collision, i.e., resultant speed.

In Section 2.4, we discuss how our scheme computes the road collision risks and deduces the proper maneuvers.

2.4 Sensing and Perception Module

2.4.1 CNP Communication Protocol

The proposed CNP enhances driving safety by offering a light-weight driving information sharing method. This model has two types of messages that serve as IPv6 neighbor discovery (ND) options for the CNP services: the cooperation context message (CCM) and the emergency context message (ECM). CCM enables cooperative driving through the exchange of a vehicle's mobility information (e.g., speed, position, and direction) and its driving actions (e.g., braking and accelerating) with its neighbors. ECM notifies a vehicle's neighbors of emergency situations (e.g., accidents). This protocol gives higher priority to the ECM than the CCM in the message delivery process of vehicles.

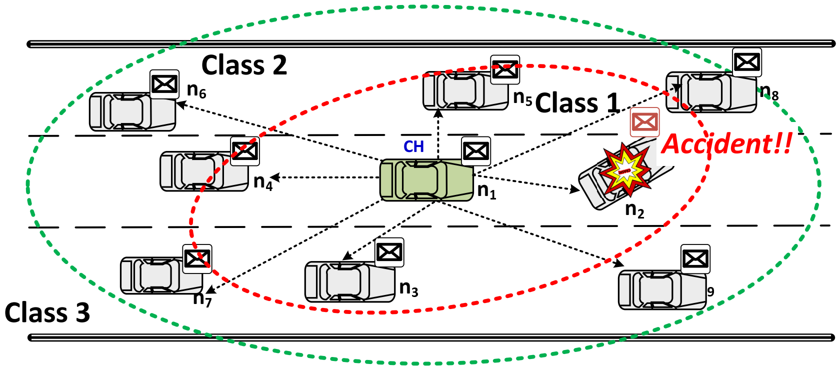


Figure 2.5: Cluster-head-coordinated dynamic path maneuver planning.

Due to ECM's higher priority than CCM, a vehicle can take immediate action in response to an emergency situation. As shown in Fig. 2.5, if there is an obstacle in a road, an ECM message is sent to adjacent vehicles via a channel for safety purposes. We advocate the use of multiple DSRC service channels that prioritize ECM safety messages. We employ vehicle clustering wherein the cluster head (CH) leads the cluster members' safety decisions. CH orchestrates the emergency maneuvers of its members to avoid collisions. Both CCM and ECM transmissions are performed with IPv6 packets in IEEE standard 802.11-OCB network mode [34]. The members' maneuvering plans are decided by a collision probability-based risk assessment, which will be defined in Section 2.4.2.

2.4.2 Probabilistic Risk Assessment

Dynamic road traffic participants drive within uncertain locations, directions, and speeds. The maneuvering decisions of a vehicle in such a road are computationally uncertain. The CNP determines the appropriate maneuverable driving lane depending on the collision probability of the emergency vehicle.

We define the "*collision probability*" as the risk that a moving vehicle will collide with an obstacle ahead in a road. The collision risk is generally calculated as time to collision T_c through the relative kinematics of adjacent vehicles on the road [33]. The minimal maneuverable time to collision T_{cmin} is reached when a vehicle maximally accelerates toward the obstacle, and the maximal maneuverable time to collision T_{cmax} is reached when it minimally accelerates (i.e., maximally decelerates) toward the obstacle. Considering that the collision risk of an n_{em} is uniformly distributed within $T_c \in [T_{cmin}, T_{cmax}]$. The collision probability that an n_{em} will collide with n_{ob} is computed as follows:

$$P(n_c \otimes n_p) = \begin{cases} 1, & \text{if } T_c \leq T_{cmin}, & (2.9a) \\ 0, & \text{if } T_c \geq T_{cmax}, & (2.9b) \\ P_{c,p}, & \text{otherwise}, & (2.9c) \end{cases}$$

where (2.9a) means that a child vehicle n_c will certainly collide with its parent vehicle n_p, (2.9b) suggests that n_c is safely driving away toward n_p, and (2.9c) defines a collision probability (denoted as $P_{c,p}$) that n_c is driving toward n_p which is computed as follows:

$$P_{c,p} = 1 - \left(\frac{T_c - T_{cmin}}{T_{cmax} - T_{cmin}} \right). \tag{2.10}$$

Vehicles in an emergency driving situation can collide with other vehicles not only in a line-of-sight unsafe range, but also in a non-line-of-sight but unsafe range [35]. Consider an emergency vehicle n_1^E driving towards n_{ob} as shown in Fig. 2.3. Let $\mathcal{V}_L = \{n_1^L, n_2^L, ..., n_u^L\}$, $\mathcal{V}_E = \{n_1^E, n_2^E, ..., n_v^E\}$, and $\mathcal{V}_R = \{n_1^R, n_2^R, ..., n_w^R\}$ be vehicles sets in the left L_L, emergency L_E, and right L_R lanes respectively. The emergency graph G depicted in Fig. 2.3 enables the CH to determine the lateral maneuver of n_1^E toward the least-risk lane, which is a lane with the minimum collision probability.

2.4.3 Link Quality-Based Maneuver Lane Selection

Considering the collision risk events $(n_1^L \otimes n_1^E)$, $(n_2^L \otimes n_1^L)$, ..., $(n_u^L \otimes n_{u-1}^L)$, respectively, for children $n_1^L, n_2^L, n_3^L, ..., n_u^L$ to collide with their parents $n_1^E, n_1^L, n_2^L, ..., n_{u-1}^L$, let their corresponding probabilities be $p_1, p_2, ..., p_u$. Let the events $(n_1^R \otimes n_1^E)$, $(n_2^R \otimes n_1^R)$, ...,

Algorithm 1 Determine Emergency Maneuver Lane

```
function DETERMINE_MANEUVER_LANE($G = \mathcal{V}, E$)   ▷ $G$ is the graph constructed by a set of vehicles $\mathcal{V}$ where each vehicle is identified by its position and speed
    $n_p \leftarrow n_1^E$   ▷ initialize the parent node $n_p$ to the vehicle $n_1^E$ most risky to collide with $n_{ob}$
    for each vertex $v$ in $\mathcal{V}_{np}$ do   ▷ $\mathcal{V}_{np}$ is the vertexes set of $n_p$'s children
        if $n_c \neq null$ then   ▷ Compute the edge cost when the current vertex has a predecessor
            $T_c \leftarrow Compute\_Time\_To\_Collision(n_c, n_p)$
            $P_{c,p} \leftarrow Compute\_Probability(T_c)$
            $n_{c.p} \leftarrow P_{c,p}$   ▷ Assign the collision probability as the metric of each graph edge cost
        end if
    end for
    $L_x \leftarrow 0$   ▷ Candidate maneuver lane index which varies from 0 to 2 for a three-lane road segment
    $Q_{lane} \leftarrow 0$   ▷ Maneuvers toward the lane which has the greatest value of lane quality $Q_{lane}$
    for each lane $l$ in $L$ do   ▷ $L$ is a set of road lanes and check the neighboring lanes of the defected lane that may be two elements (right and left lanes), or only one lane side
        $Q_l \leftarrow Calculate\_Lane\_Quality(\mathcal{V})$
        if $Q_{lane} < Q_l$ then
            $Q_{lane} \leftarrow Q_l$
            $L_x \leftarrow l$
        end if
    end for
    return $L_x$
end function
```

$(n_{w-1}^E \otimes n_w^R)$, respectively, for children $n_1^R, n_2^R, n_3^R, ..., n_w^R$ to collide with their parents be $n_1^E, n_1^R, n_2^R, ..., n_{w-1}^R$ and let their corresponding probabilities be $p'_1, p'_2, ..., p'_w$. The n_1^E maneuvers toward a lane with better quality. Let the safe probability q_i of a vehicle n_c driving toward another vehicle n_p be

$$q_i = 1 - p_i, \tag{2.11}$$

where p_i is the collision probability of event $n_p \otimes n_c$. Consequently, we deduce the **safe lane quality** of every lane of the road.

Definition 2.4.1 (Safe Lane Quality). *Let the **Safe Lane Quality** S_{Lx} of a given road lane L_x be a lane metric that indicates the state of lane safety during the emergency driving decision.*

The safe lane quality of the left lane will be computed as follows:

$$S_L = \prod_{i=1}^{u} q_i, \tag{2.12}$$

and the safe lane quality towards the right lane is computed as:

$$S_R = \prod_{i=1}^{w} q_i', \tag{2.13}$$

where q_i is the safe probability for n_p and n_c in the left lane, and q_i' is the safe probability for n_p and n_c in the right lane.

An n_1^E's collision avoidance maneuver follows the algorithm 1. Line 2 of Algorithm 1 initializes the parent node with n_1^E. The while loop in lines 3-9 assesses the risks of the graph G nodes with the parent-child relationship. Lines 4-8 calculate the collision risks among the adjacent vertices, and then assign a risk cost to the edge (n_c, n_p) in terms of collision probability according to (2.10). Lines 10-11 initialize the maneuver safe lane cost, and n_1^E should maneuver to the lane with maximal safety. Lines 2-18 determine and compare lane qualities to choose the best lane. n_1^E will steer toward the lane returned by line 19.

Given the safe lane qualities S_L and S_R in the left and right lanes, respectively, the decision of the lane (denoted as L_x^*) for maneuvering emergency vehicles in adjacent lanes is computed as follows:

$$L_x^* \leftarrow \max_{L_x \in \{L,E,R\}} (S_L, S_E, S_R). \tag{2.14}$$

Calculating an emergency vehicle's maneuvers is time-critical. The complexity of the CNP maneuver planning algorithm is $\mathcal{O}(Nl)$. This complexity increases with the number of vehicles N driving on the defective road and the number of lanes l. To remedy this, we make a trade-off between the optimal risk assessment and the assessment time.

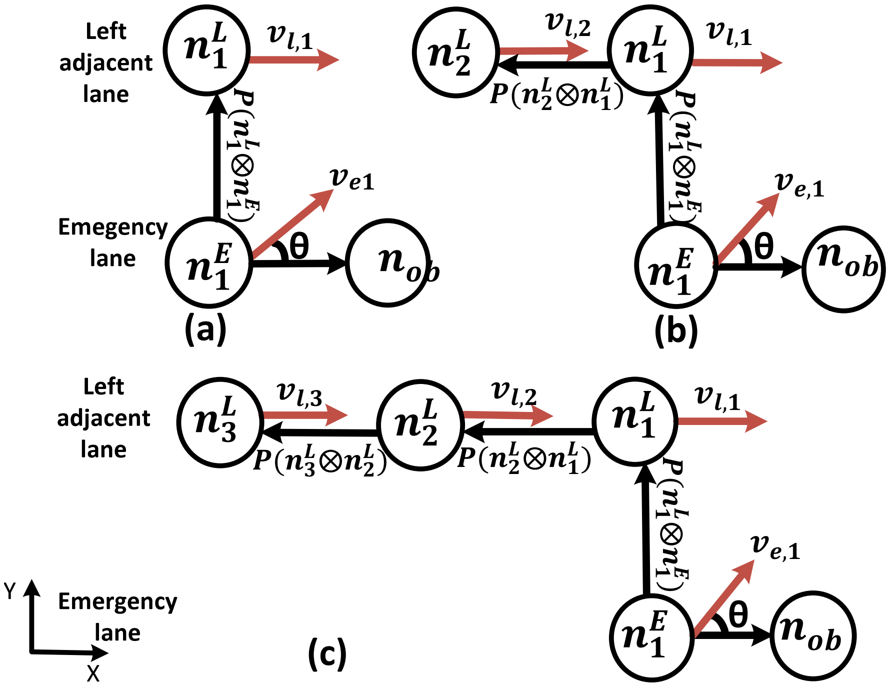


Figure 2.6: Expansions of the emergency vehicle e_1 maneuver effect on the maneuver lane L_l pre-maneuver driving vehicles.

Fig. 2.6 illustrates a scenario for lane change of an n_1^E avoiding collision with n_{ob}. Considering it maneuvers towards the left lane, Fig. 2.6 shows possible scenarios that may arise.

- In Fig. 2.6(a), n_1^E bypasses n_{ob} without affecting the trajectory planning of existing vehicles on the maneuver lane. This often happens on roads with low vehicle density.

- In Fig. 2.6(b), n_1^E bypasses n_{ob} by affecting only the trajectory of its adjacent vehicles. This typically happens when vehicle n_1^L needs to either accelerate, decelerate, or change lane in order to create enough space for n_1^E to avoid an accident.

- In Fig. 2.6(c), n_1^E affects the trajectories of multiple adjacent vehicles when bypassing n_{ob}.

The following section will discuss in detail how the CNP plans emergency paths for vehicles.

2.4.4 Emergency Maneuver

Emergency Path

Vehicles are sequentially maneuvered to avoid accidents in the road. Sequentially within the maneuver lane, CNP assesses the collision risk to make sure that an emergency vehicle n_{em} will safely access the lane without degrading the safety of the existing driving vehicles. CNP handles emergency driving, starting with the closest vehicle to the obstacle toward the rear vehicles. CNP defines the timely maneuver path of n_{em} from its contour area, in a way that guarantees the other vehicles' safety. Please refer to Appendix A for the detailed path maneuver. Vehicles are laterally maneuvered to avoid any possible collisions with an obstacle n_{ob}. Longitudinal changes are only made when lateral maneuvers are not feasible toward other lanes. In the case of a reckless cut-in maneuver, our CNP mechanism will enable a vehicle to identify such an obstacle and quickly broadcast it among the driving vehicles, thus creating a preparedness to react to it. We prioritize the maneuver by changing lanes to enable vehicles to keep on their trajectories despite the obstacle in the road. However, for non-maneuverable vehicles, we enable vehicles to break or collide with as little energy as possible to limit damage.

Collision strength minimization

For the unavoidable collision situation, a collision strength minimization mechanism is needed to minimize the energy transfer between the colliding vehicles, thus reducing damage. The severity of a collision is proportional to the masses of the two colliding vehicles and their corresponding speeds. Assuming that an emergency vehicle n_{em} with speed v_{em} and mass m_{em} collides with an obstacle n_{ob} with speed v_{ob} and mass m_{ob}, the collision strength calculation is made using their Equivalent Energy Speed (EES) [33] calculated according to (2.8). The derivation of the collision strength minimization can be found in Appendix A.1.2.

Up to now, we have described in detail the theoretical mechanism of the CNP. In the next section, we will explore the performance evaluation.

2.5 Performance Evaluation and Simulation Results

This section evaluates the performance of the CNP risk assessment mechanism. It compares the performance of CNP with other communication mechanisms in terms of communication overhead. It evaluates the CNP safety mechanism performance in terms of collision risk reductions. It assesses the collision mitigation by the equivalent energy speed and the number of colliding vehicles. We carried out a simulation implementation to evaluate CNP performance.

2.5.1 Simulation Setup

To be able to evaluate the correctness and efficiency of this scheme, we conducted a simulation with the simulation of urban mobility (SUMO) [36] and OMNeT++ simulation framework [37]. We made an urban mobility simulation where multiple vehicles are traversing a road and exchanging mobility information for safety purposes. With OMNeT++, we simulated CNP centralized network communication. Vehicles exchange the Emergency Context Messages (ECM) as WAVE Short Messages (WSM) within an IEEE.11-OCB enabled network simulation environment. Table 2.1 lists the simulation configuration parameters.

Table 2.1: Simulation Configuration

Parameters	Description
1. Road network	2 km road segment from entrance intersection i through exit intersection j.
2. Lambda λ	Vehicle injection rate is from $0.2 \sim 0.7$.
3. Driving Speed (v)	$v = 20 \sim 120$ km/h.
4. Acceleration	Vehicle acceleration a ranges $[a = -5 \sim 5] m/s^2$.
5. Acceleration Error	$\epsilon = -1 \sim 1$ meters per second square.
6. ECM Transmission Rate	Frequency of safety information transmission. The default is 10 packets per second.

To allow SUMO to meet the demands of the CNP mechanism, we modified the $SL2015$ [38] lane changing mechanism to comply with the coordinated maneuver mechanism. We extended Krauss' car following model and SUMO lane changing strategy that enables both the collisions to occur when not avoided. Through a communication strategy in our proposal, vehicles get informed of the existence of obstacles and can preemptively maneuver prior to reaching a high collision risk.

2.5.2 Performance Parameters and Metrics

The evaluation of this mechanism is based on the following evaluation settings.

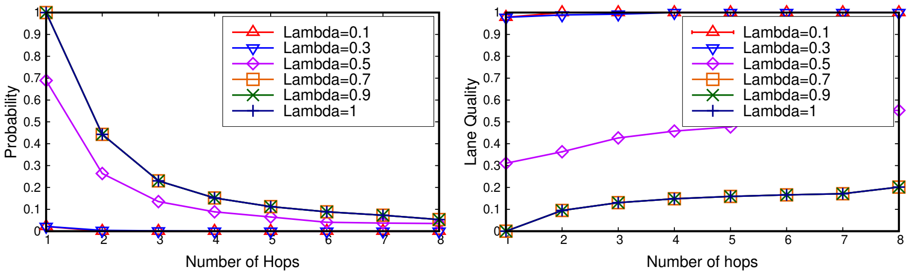
- **Parameters:** The parameters for evaluation are: the impact of *(i) vehicle injection rate (λ), (ii) maximum driving speed, and (iii) acceleration*.
- **Metrics:** The metrics for evaluation are: the communication overhead to measure the communication performance, the collision probability to measure collision risk, the number of collisions, and the collision equivalent energy speed as metrics for collision strengths.

- **Baselines:** We compare CNP with legacy situational awareness, which are sensor-based approaches (e.g., LIDAR), a collision avoidance (CA) based [29], and a networked approach (e.g., NCAS [22]). Unlike CNP and NCAS, the sensor detects the situation only in the Line-of-collision (LOC) of vehicles.

To test the performance of the CNP, we use a road segment with three lanes. Our simulation results use a 95% confidence interval.

2.5.3 Simulation Results

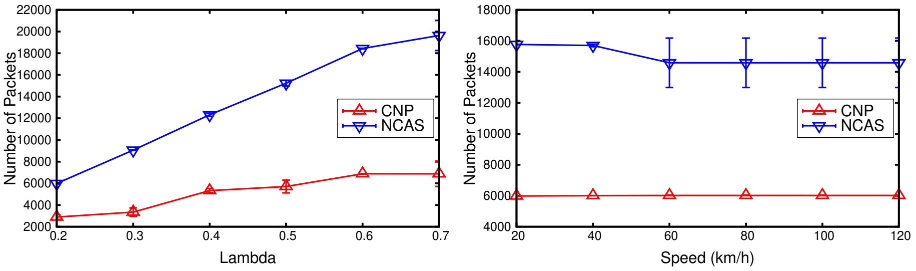
Lane Quality



(a) Lane collision probability

(b) Lane quality

Figure 2.7: Impact of the number of considered hops on the lane quality.



(a) Impact of λ on communication overhead.

(b) Impact of speed on communication overhead.

Figure 2.8: Impact of CNP on communication overhead.

This section investigates the safe lane quality (mentioned as lane quality) of the CNP, which is defined by Definition 2.4.1, by measuring the impact of the number of hops that were taken into account in the CNP's risk assessment calculation on the lane quality. To determine the quality of the lane driving, we evaluate the impact of the hop number on collision probability, considering the different number of hops in the lane. First, consider only vehicles within the LOC of n_1^E, and then consider two hops away from n_1^E, and then three hops away, and so on.

Fig. 2.7a shows that for both the vehicle injection rates, the risk of collision is high if the CNP assesses the risk only to its neighbors. The greater the number of vehicles in the lane that are considered by the risk assessment is, the lower the risk of collision is. Similarly, as shown in Fig. 2.7b, considering a small number of vehicles in the CNP risk assessment will result in poor safe lane quality. Higher injection rates (e.g., $\lambda = 1$) result in poorer safe lane quality as road congestion increases. Assessing up to 3 hops can ensure adequate safe lane quality and less complex calculations. This gives a collision probability equivalent to 0.15 for $\lambda = 1$ and one equivalent to 0.11 for $\lambda = 0.5$. Their safe lane qualities are 0.13 and 0.5, respectively, which is safe enough to limit the risk of a chain of collisions.

Communication Control Overhead

Both CNP and NCAS are communication-based mechanisms to avoid crashes in vehicular networks. Unlike NCAS, CNP uses a coordinated communication protocol, and CH is the orchestrator of the remaining CMs' maneuvers. Fig. 2.8a shows the evaluation of communication overhead in vehicular networks wherein vehicles drive at $80km/h$ and are injected at an injection rate $\lambda \in [0.2\, 0.7]$. The results show that CNP reduces the communication overhead caused by NCAS from 15% to 60%. Fig. 2.8b shows an overhead comparison figure where vehicles drive with maximum speeds varying from 20 to

$120km/h$ while maintaining an injection rate of 0.6. The results shows that CNP reduces the overhead caused by the NCAS communication control by up to 60%.

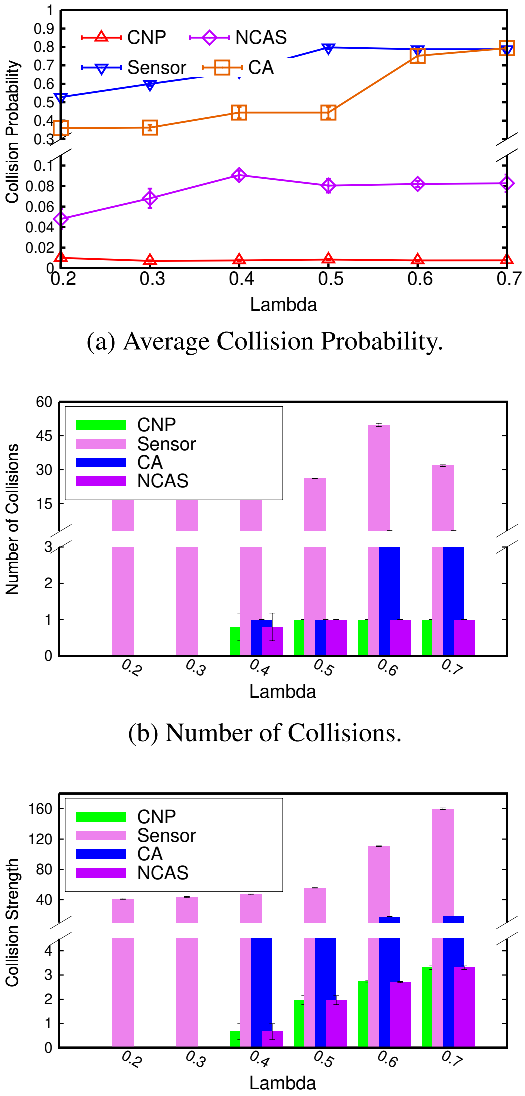
Impact of the Vehicles Injection Rate λ

This section investigates the impact of vehicle injection rate λ on the overall CNP performance compared with the baselines. It is seen that CNP reduces the risks of vehicles colliding with an obstacle. Fig. 2.9a shows that over time, on average, the risk of colliding either with obstacles or other vehicles in CNP is always the smallest. For unavoidable crashes, Fig. 2.9b shows that CNP has the smallest number of collisions at all injection rates. At higher injection rates, the number of sensor-based collisions in the presence of obstacles will be greater than the collisions found with both communication mechanisms (i.e., CNP and NCAS). Another factor that measures the nature of collisions is the energy transfer between collision nodes (including vehicles and obstacles), which is measured by EES.

Fig. 2.9c shows that in most cases, the cumulative energy when CNP is applied is lower than that found only by relying on the sensor-based approach (Sensor) and equivalent to the benefits of NCAS. The smaller the EES, the lighter the collision which reduces the accident casualties. As shown in Fig. 2.9, the overall results indicate that CNP outperforms the sensor-based approach on all levels of tested injection rates.

Impact of Speed

To investigate the impact of speed, we set different speed limits in the range from $20km/h$ to $120km/h$ and fixed the injection rate to 0.6, and tested the behavior of CNP in comparison with the baselines. Fig. 2.10 shows the evaluation of collision risks when there is an obstacle on the road. For all speeds, the CNP outperforms the sensor and NCAS approaches in terms of the three metrics, such as collision probability, the number of col-

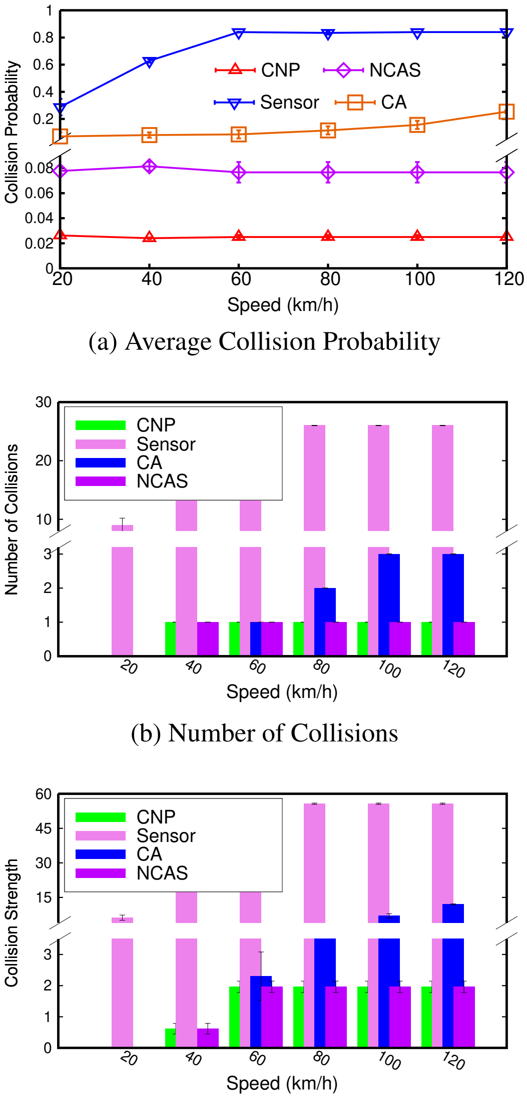


(a) Average Collision Probability.

(b) Number of Collisions.

(c) Collisions and their strengths.

Figure 2.9: Impact of injection rate (λ).



(a) Average Collision Probability

(b) Number of Collisions

(c) Collisions and their strengths

Figure 2.10: Impact of speed.

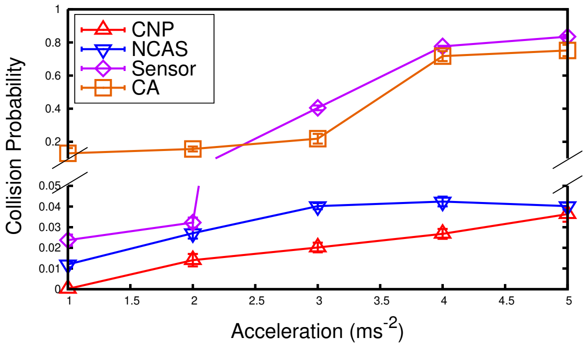


Figure 2.11: Impact of acceleration.

lisions, and collision strength. The collision probability with CNP is 5% less than NCAS, while it grows to 85% of the sensor-based approach as depicted by Fig. 2.10a. On average, CNP reduces 5% of the chances of collisions of NCAS, and 80% of those of the sensor-based risk assessment.

Fig. 2.10b shows that CNP outperforms the sensor-based approach in the reduction of the number of possible collisions. At all speeds, CNP has fewer or the same number of vehicle collisions compared with the NCAS, and much fewer than the sensor-based approach. For speeds greater than $60km/h$, the number of possible collisions when using the sensor risk assessment is much higher than those of the CNP and NCAS, showing that network-based risk assessments have better performance. CNP has the lightest collisions compared to the baseline as shown in Fig 2.10c.

Impact of Acceleration

We investigated the impact of acceleration by testing accelerations that vary from $1m/s^2$ to $5m/s^2$. The results in Fig. 2.11 show that the lower acceleration leads to higher safety. That is, the safety weakens with the increase of acceleration. For all the accelerations tested, CNP outperforms the other compared risk assessment methods.

2.5.4 Discussion

The communication protocol in CNP was simulated according to the 802.11p standard. This protocol needs the vehicles to respond to the driving environments in a timely manner and to handle complex driving functions. For this to be possible, a powerful computational system capacity that enables vehicles to collaborate with each other is required.

The standardization of 802.11p and 3GPP has substantially improved the robustness and reliability of vehicular networks, thus allowing them to communicate with each other without an infrastructure. However, a limited processing speed would lead to inaccurate decisions. Intel Corp. estimates that approximately 1 GB of data needs to be processed each second in the car for collaborative driving [39]. Vehicles with graphics processing units (GPUs) for computation acceleration can handle such cognitive processing loads efficiently.

2.6 Conclusion

This chapter introduced a context-aware navigation protocol (called CNP) to enhance the driving safety of vehicles moving in urban roads. The CNP's collision avoidance feature allows vehicles to drive safely in the presence of obstacles or accidents in the road by perceiving the situation and determining safe paths to follow. If a collision occurs, collision mitigation minimizes any damage. The simulation results have shown that CNP outperforms the sensor-based approach in reducing the risks of collisions, the number of collisions, and the strength of collisions.

As future work, we will enhance our collision probability computation, considering the reaction time of a vehicle. Also, we will implement and test this CNP protocol on real cars to improve its accuracy and usability for safe driving. We will also test the impact of the CNP on the overall trajectory performance of the driving vehicles as another way to test and improve the navigation efficiency.

Chapter 3

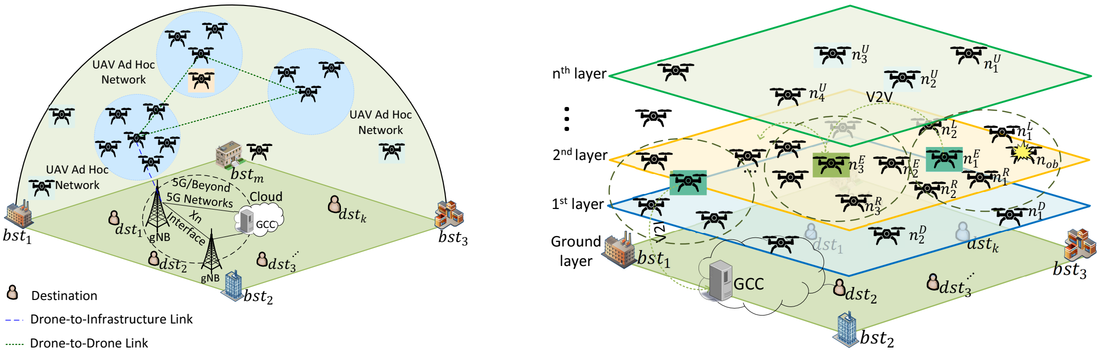
CANA: Collision-Avoidance Navigation Algorithm for UAV Flying Networks with Heavy Traffic

The advancements in vehicular cyber-physical systems (V-CPSs) have enabled several studies on Unmanned Aerial Vehicle (UAVs) applications. To list a few, UAVs have been proposed to provide services in the courier delivery industry, surveillance and monitoring applications, disaster management, and cinematography. Multiple UAVs flying in the same airspace necessitate a seamless coordinated navigation to avoid mid-air collisions. It is important to study the coordinated navigation approaches that enable planning, tracking, and updating the mission path in a flying ad hoc network. This chapter proposes a collision-avoidance navigation algorithm (CANA) for UAV flying networks with heavy-traffic. It proposes an autonomous swarm-based UAV trajectory planning that utilizes the sensing and network facilities of V-CPS to guarantee the safety of flying UAVs.

3.1 Introduction

Unmanned Aerial Vehicles (UAVs) have recently experienced rapid advancements, and they are increasingly popular for use in various applications. Many recent studies in this area have been primarily focused on mission-oriented UAVs for applications such as patrolling, recording, surveillance, and inspections; parcel delivery; disaster management for prediction, assessment, and response; and air taxi called Urban Air Mobility (UAM), which is expected to revolutionize urban transportation in the future [40]. The various benefits of UAVs, such as their speed, flexibility, and size, have led researchers to propose their use in various applications. The forthcoming expansion of UAV applications will lead to the commercialization of various UAV services. As a result, many UAVs with different purposes will be flying in the sky [41]. Multiple UAVs flying in the same airspace with the same mission are considered to be homogeneous UAVs. On the other hand, UAVs flying in the same airspace with different mission purposes are considered to be heterogeneous UAVs. Both homogeneous and heterogeneous UAVs need to cooperate for efficient navigation in shared airspace.

Recent studies have achieved technological progress in the Flying Ad Hoc Networks (FANET), which promote cooperation among UAVs [42]. UAV networks like FANET can enable effective coordination and communication in a multi-UAV environment [43]. Communications in UAV networks can be categorized into air-to-air communication and air-to-ground communication [44]. In air-to-air communication, UAVs communicate among themselves, which is a means for decentralized UAV coordination mechanisms [45]. Besides, air-to-ground communication allows a Ground Control Center (GCC) and UAVs to communicate, enabling the centralized UAV coordination mechanisms [44]. UAV sensors (e.g., cameras, Light Detection and Ranging (LIDAR)) and communication devices for UAVs are important for awareness in flying environments [46, 47]. Network-enabled



(a) A heavy-traffic UAV network architecture

(b) Multi-layered flights

Figure 3.1: Architecture of context-aware navigation model for a heavy traffic UAV network.

swarm-based collision avoidance mechanisms make it possible to avoid collision in congested airspace [48].

Collision avoidance methods include both the centralized mechanisms, where a central device (i.e., GCC) decides the paths of flying nodes [44, 49], and the decentralized mechanisms where all flying nodes are responsible for the computation of their safe paths [45, 48]. A typical centralized mechanism is a natural multi-modal communication method proposed [49], that enables a single system operator to supervise and orchestrate a set of UAVs involved in the rescue missions. The central device performs communication data fusion and bridges inter-UAV communication. The decentralized mechanisms include a multi-objective pigeon-inspired optimization [45] and distributed census control [48]. These methods present gaps in achieving realistic trajectory scheduling and optimization controls necessary to respond to a complex environment. Delays caused by the centralized inspection and monitoring of the airspace situations do hinder timely safety responses and the efficiency of the mission [44]. On the other hand, decentralized airspace awareness can be weakened by the limitation of UAV sensors' line-of-sight and computational capacity [50].

Thus, a system that leverages centralized and decentralized awareness will suitably address the safety challenges in a shared airspace. Cooperative controls are time-critical operations that necessitate using a cost-effective control system. Such controls and planning should be optimized to ensure successful mission completion even within a complex aerial mobility environment.

We propose a Collision-Avoidance Navigation Algorithm (CANA) that autonomously guarantees collision-free flights in a heavy-traffic UAV network illustrated in Fig. 3.1a. It uses a 5G-equipped UAV network architecture that facilitates lightweight sensing information shared among airspace nodes. CANA design involves multiple-layered flights in the airspace in which UAVs fly at different heights as depicted in Fig. 3.1b. Unlike a 3D map proposed in [51] where its multiple-layered approach allows UAVs only to cross from one layer to another in fixed path segments, CANA provides a scalable and flexible approach for UAV layer crossings while preserving the safety of the UAV network. To achieve safe aerial traffic, we modeled layers with virtual aerial roadways made of virtual lanes and virtual intersections in which UAVs fly toward the destination of their mission as depicted in Fig. 3.2a. When a UAV reaches a mission destination, it lands by crossing layers. Fig. 3.2b depicts a UAV landing process in which CANA determines a sequence of safe voxels that cannot be used by any other UAV simultaneously. The system calculates each UAV's flight kinematics (i.e., speed and direction) by ensuring collision-free air mobility with minimal cost through trajectory optimization to make knowledge-based collision-free flight decisions.

In this chapter, we formulate a collision avoidance algorithm for heavy-traffic UAV networks as an optimization problem. Our CANA uses heuristic solutions with acceptable approximations to reduce the complexity of the collision-avoidance optimization for UAVs. Specifically, the major contributions of this work are as follows:

- We design a coordinated UAV swarming system to plan safe UAV flight trajectories. It is an autonomous control system that models coordinated collision avoidance of

UAVs via a 5G V2X communication in a UAV network (see Section 3.4).

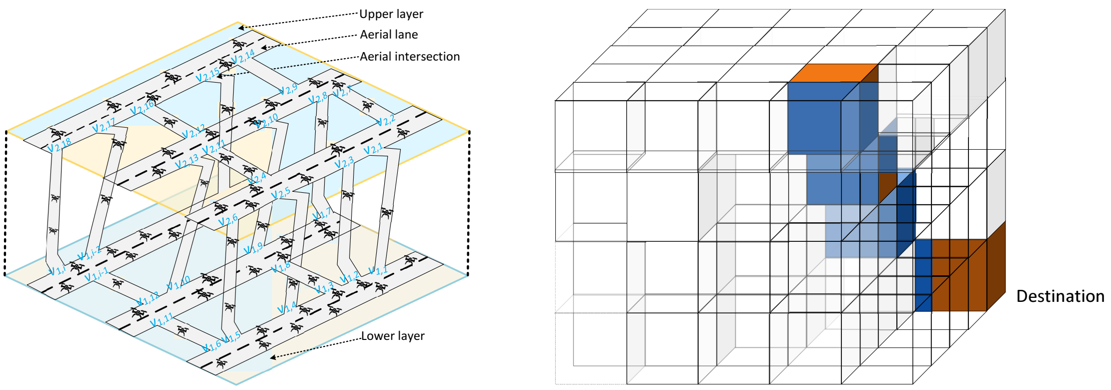
- We design a collision-free knowledge-based heterogeneous UAV navigation through 3D real time collision risk assessment. We develop coordinated flights in ground-road-like virtual aerial lanes and intersections in which UAVs can switch directions at the same heights or move toward higher or lower altitudes to avoid collisions (see Section 3.4).
- We formulate the collision avoidance process as an optimization problem, and solve it using heuristic solutions that achieve improved performances (see Section 3.4).
- We evaluate our CANA through simulation to demonstrate its efficiency in detecting and assessing obstacles appearing in the airspace with the smallest risk of collision compared to the baselines [50, 47]. The results show that the CANA reduces the control overhead by up to 66.6% compared with state-of-the-art collision avoidance schemes [47] while satisfying the safety constraints (see Section 3.5).

Note that this chapter extends the preliminary work presented in our previous paper [15].

The rest of this chapter is organized as follows. Section 3.2 explains related work. We specify a problem formulation for heavy-traffic UAV networks and centralized UAV systems in Section 3.3. A description of the autonomous UAV navigation system design and the collision avoidance algorithms is detailed in Section 3.4. Section 3.5 presents the performance evaluation of our collision avoidance algorithm and baselines. Section 3.6 concludes this chapter while providing future work directions.

3.2 Related Work

This section summarizes the recent work that has been conducted aiming at achieving safe flights in UAV networks.



(a) Layout of UAV virtual roadways

(b) UAV landing modeling

Figure 3.2: Sky road architecture for an aerial UAV navigation.

Recent studies have proposed several safety-preserving methods that can be used to ensure safe UAV mission planning and controls. They rely on standard flight planning algorithms such as A^* [52, 53], Bat Algorithm (BA) [54], and spline search [55], which are used to obtain safer routes. Two categories can be identified among these methods: (1) centralized UAV controls [53, 56, 46] and (2) decentralized UAV controls [52, 54, 57]. In centralized UAV control, a central device such as the Ground Control Center (GCC) defines the flight controls followed by flying UAVs. On the other hand, for decentralized UAV controls, all nodes in the UAV network participate in the definition of their safe flight paths. A centralized strategy for moving a swarm of autonomous UAVs was proposed to accomplish a specific task by considering trajectory calculation and collision avoidance [46]. A decentralized prioritized algorithm that combines the A^* algorithm with coordination has been proposed to enable collision-free mission planning [52]. One study suggested a method that involves improving the bat algorithm to adapt three-dimensional (3D) flight path planning for UAVs [54]. To obtain safe motion, boundary-based risk assessments are used [57]. Further cooperation among UAVs is possible via FANET to advance the airspace perception and response [58].

A swarming strategy using a distributed collaborative search of multiple UAVs has also been introduced for scalable and efficient cooperation in a complex UAV flying environment [56]. Cooperative tracking of formed UAV groups has been studied to ensure awareness of the dynamic environment [59]. Another study has proposed a communication-aware path planning method [53] that also involves A^* variants to improve the quality of UAVs' path planning. A collaborative multi-UAV-assisted system proposed in [60] uses the joint optimization of UAVs' trajectory planning to improve the task offloading in a multiple-UAV network. Aside from these environmental perception systems, a couple of other methods for avoiding collisions in a UAV network have also been proposed [61, 62, 63].

For successful UAV control, reliable Conflict Detection and Resolution (CDR) should be implemented. Avoiding middle-air collisions can be achieved through both pre-flight and in-flight collision detection and avoidance [61, 62]. Four control approaches are commonly used: a rule-based control approach, a deterministic optimal control approach, a stochastic optimal control approach, and a protocol-based control approach [63]. The rule-based approach is a decentralized approach, where each UAV decides its maneuver following pre-described rule-based policies. The deterministic optimal control approach determines maneuvers by minimizing an objective function of an optimal control problem. The stochastic approach considers a CDR as a stochastic problem and considers the trajectory results from problem optimizations. The protocol-based approach defines a protocol that all UAVs in the airspace must follow to guarantee the safety of the multi-UAV system. All these methods have their drawbacks. While the rule-based and protocol-based approaches are commonly known to be unsuitable for avoiding collisions when unexpected events occur, deterministic and stochastic optimization also suffer from heavy computational load. CANA defines a stochastic approach with heuristic approximation to relax its computational cost.

Airspace awareness information sharing among UAV network nodes significantly enhances the vehicular safety mechanisms[42, 51]. A Context-aware Navigation Protocol (CNP) in [12] proposes a guiding strategy for vehicles in road networks for collision risks. Based on the intuitive metrics of lane quality, CNP successfully guides the maneuvering of vehicles in and out of the line-of-sight of obstacles on the road. However, this methodology is not able to handle 3D UAV movement scenarios when an obstacle appears in the aerial environment. Carspeak has designed a communication system that enables cars to access 3D stream data from other vehicles in the manner in which they access their onboard sensor data [64]. Using an octree data structure, a vehicle can query and access sensor information for a specific region. Meanwhile, the Mixed Integer Linear Programming (MILP) approach proposed in [51] proposes a 3D map wherein UAVs can navigate while avoiding the risk of collisions. Although this method involves a multi-layered approach, UAVs can only cross from one layer to another in fixed path segments. CANA provides a more scalable and flexible approach in a 3D airspace that allows UAVs to cross from one layer to another without worsening the safety of UAVs that were already in that layer. The CANA scheme successfully handles the complexity of air mobility by using appropriate risk assessment metrics and a navigation strategy, and preserves the mission efficiency and the safety of UAVs.

3.3 Problem Formulation

In this work, we intend to design an aerial navigation mechanism that accommodates collision-free UAV heavy traffic, suitable for shared airspace. A UAV departs from its base station while aiming at a specific objective called a mission, such as search, rescue, and parcel delivery. We combine UAV mission and path planning strategies for a safe UAV swarm operation by airspace awareness and collision avoidance decision methods. This section describes a heavy-traffic UAV network, a navigation framework, a problem targeted by our design, and the assumptions we considered.

3.3.1 Heavy-Traffic UAV Network Navigation Description

A heavy-traffic UAV network consists of n UAVs that can be dispatched to m mission bases while operating in the same airspace toward k mission destinations, as illustrated in Fig. 3.1a. UAVs depart from a set of mission bases $B = \{bst_1, bst_2, ..., bst_m\}$ to fly toward a set of destinations $Dst = \{dst_1, dst_2, ..., dst_k\}$. An aerial navigation system defines a connected graph $G = (\mathcal{V}, E)$ that is constructed by a vertex set $\mathcal{V}$ of m mission bases and k mission destinations such that $|\mathcal{V}| = m + k$. An edge set E is made of all pairs of mission bases and mission destinations, where the edge cost in E is determined by considering the link cost (e.g., traversing delay, collision probability, and congestion level).

A framework of environment awareness for safe navigation in UAV networks is described in Fig. 3.3. It uses UAV network context awareness through communications over 5G to define a safe UAV trajectory in the steps described below. First, a perception step deals with information acquisition by UAV sensors in their sensing range. Second, a decision-making step assesses the sensed information to identify possible collision risks. Third, a control step reflects the decision and updates their flight plans for safer trajectories. Lastly, the awareness message exchange step disseminates UAV mobility information (e.g., speed, position, and direction) and flight control information (e.g., acceleration) through lightweight information sharing over UAV networks.

A UAV departs from a mission base and traverses through its assigned destinations. During its flight, several other UAVs are concurrently flying in the same airspace, thereby forming a FANET as shown in Fig. 3.1a. The flying information is shared among UAVs via air-to-air communication and to GCC through air-to-ground communication, enabling it to keep updated on the UAVs' mobility changes. Once a UAV has completed its mission,

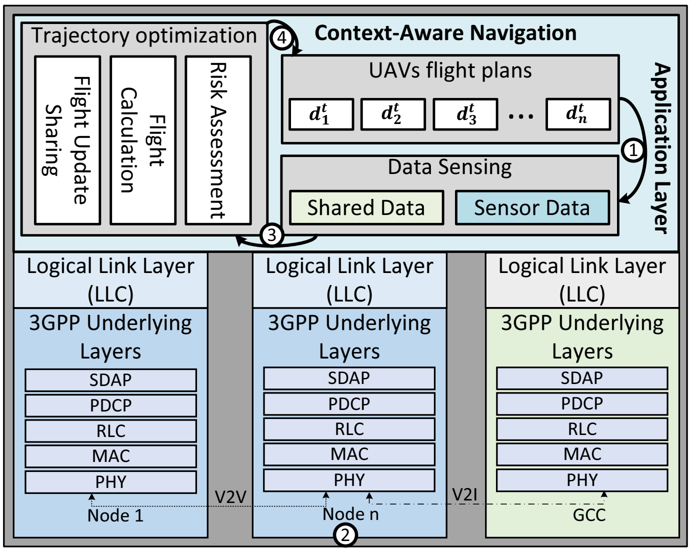


Figure 3.3: An aerial environment awareness framework architecture.

it returns to its base and waits for its next mission.

3.3.2 UAV Navigation Problem Formulation

In this section, we describe the UAV navigation problem we intend to address by formulating a set of corresponding equations and functions, as follows:

Problem Definition

Let a set $\mathcal{D} = \{n_1, n_2, ..., n_n\}$ be a set of nodes that are currently flying in the airspace toward their intended destinations at a particular time t. The state s_t of a given UAV n_i at that particular time t can be defined as

$$s_t = \begin{bmatrix} x_t & y_t & z_t & \phi_t & \theta_t & \psi_t \end{bmatrix}^T, \tag{3.1}$$

where (x_t, y_t, z_t) define its position in 3D space, and where $(\phi_t, \theta_t, \psi_t)$ be the roll, pitch, and yaw angles, respectively. The trajectory control input for its next movement is defined as

$$u_t = \begin{bmatrix} u_t & v_t & w_t & p_t & q_t & r_t \end{bmatrix}^T, \tag{3.2}$$

where (u_t, v_t, w_t) are the UAV's forward, side-way, and vertical velocities, respectively, and (p_t, q_t, r_t) are its roll, pitch, and yaw change rates, respectively. Within a continuous time, the UAV control dynamics system can be expressed as

$$\begin{gathered} \dot{s}_t = f(s_t, u_t) + w_t, \\ w_t \sim N(0, \sigma_t^2), \end{gathered} \tag{3.3}$$

where $f(s_t, u_t)$ is the dynamics equation of the UAV system for time t, and w_t is a Gaussian noise w_t, with a variance σ_t^2 for time t [65]. The w_t expresses the UAV motion control disturbances such as the actuation delay. Kalman filter is a predictive model that can accurately estimate the state of the UAV. Given a UAV state estimate at time $t-1$, $\hat{s}_{t-1}$ this model predicts its a priori state estimate at time t as

$$\hat{s}_t = F_{t-1}\hat{s}_{t-1} + B_{t-1}u_{t-1}, \tag{3.4}$$

where F_{t-1} expresses the state transition matrix that propagates a UAV state from $t-1$ to t and B_{t-1} is the control input matrix. The predicted error covariance matrix is

$$P_{t-1} = F_{t-1}P_{t-1}F_{t-1}^\top + Q_{t-1}, \tag{3.5}$$

where P_{t-1} is a priori error covariance matrix and Q_{t-1} is the error covariance matrix. A posteriori estimate covariance matrix P at time t is expressed as

$$P_t = (I - K_t H)P_{t-1}, \tag{3.6}$$

where I is the identity matrix, K is the Kalman gain at time t and H is the measurement matrix. K is derived from

$$K = \hat{P}_t H^\top (H\hat{P}_t * H^\top + R)^{-1}, \tag{3.7}$$

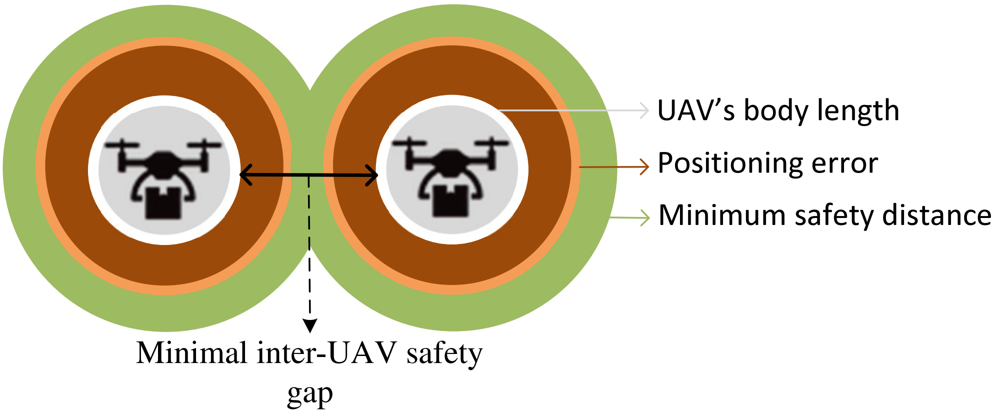


Figure 3.4: Illustration of safety distance for adjacent UAVs.

where $\hat{P}_t$ is a priori estimate covariance matrix at time t, $H^\top$ is the transpose of the measurement matrix, and R is the covariance matrix of observation noise. Having the measurement state matrix m_t, the next UAV state s_{t+1} can be expressed as

$$s_{t+1} = s_t + K(m_t - Hs_t). \tag{3.8}$$

Collisions occur when two or more UAVs attempt to use the same space position concurrently. For two flying UAVs, as shown in Fig. 3.4, there is a minimum safe gap that must be satisfied to ensure the safest flights. This gap is made of the UAV's body length l_i, the navigation error (i.e., GPS error) l_ϵ, and the minimum safety distance l_{min}, ultimately expressing the safe distance s_{safe} defined as

$$s_{safe} = l_i + l_{min} + l_\epsilon. \tag{3.9}$$

Note that a navigation error is an uncertainty margin that is fixed to limit the possibility of physical collision. A UAV collision is likely to happen when the distance between adjacent UAVs is less than s_{safe}.

Fig. 3.5 demonstrates a 3D graph structure that enables a determination of a collision-free shortest path in aerial space. A control system can be used to coordinate UAV control

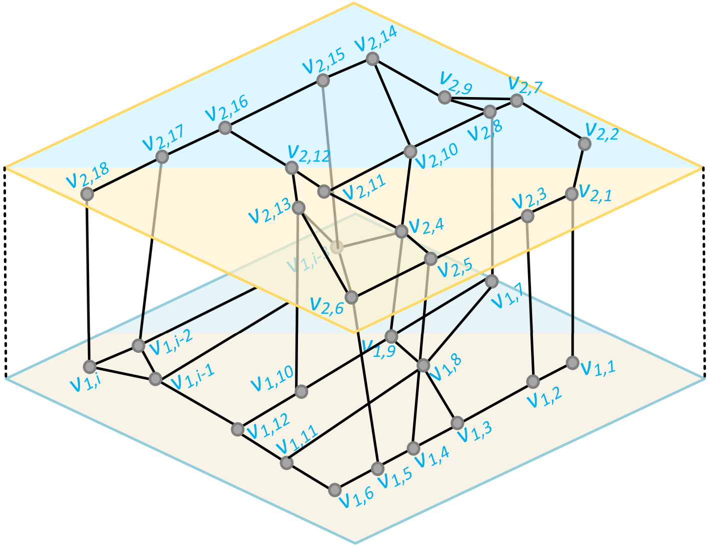


Figure 3.5: A multi-dimensional structure for the End-to-End navigation path determination.

inputs in (3.2) by satisfying safety conditions to ensure collision-free flights by regulating UAV flights with a 3D graph-based short path calculation.

Let $\mathcal{D}'$ be a set of UAVs ready to enter the airspace. For new request $n_j \in \mathcal{D}'$, a safe path $\mathcal{P}_{n_j}$ from $bst_j = p_1^j$ to $dst_j = p_t^j$ needs to be computed.

Problem 3.3.1. ***A safe path planning problem:*** *This consists of determining a path $\mathcal{P}_{n_j} = \{p_1^j, p_2^j, ..., p_t^j\}$ for an entrant UAV n_j in the airspace such that it will maintain a safe gap with all UAVs flying in the same time during the entire path traversing time samples vector $\Delta_t = \{t_1, t_2, ..., t_t\}$.*

For each UAV $n_j \in \mathcal{D}'$ with $j < |\mathcal{D}'|$, the algorithm in this chapter will solve Problem 4.3.1 by defining its mission path plan$\mathcal{P}_{n_j} = \{p_1^j, p_2^j, ..., p_t^j\}$ such that

$$||p_j^k - p_i^k|| \geq s_{safe}, \text{ for } \exists n_j \in \mathcal{D}', \ \forall n_i \in \mathcal{D}, \text{ and } k \leq t. \tag{3.10}$$

Given a UAV $n_j \in \mathcal{D}'$, an ideal solution to Problem 4.3.1 defines its mission path $\mathcal{M}^*_{path}$ that maximizes the spatial inter-distance with each of the flying UAV $n_i \in \mathcal{D}$. It is expressed as

$$\mathcal{M}^{*}_{path} \leftarrow \max_{\forall n_i \in \mathcal{D}} \min_{n_j,\, k} ||p^k_j - p^k_i||. \tag{3.11}$$

where $||p^k_j - p^k_i||$ is the estimated spatial distance between UAVs n_i and n_j at a time $k \leq t$. The navigation goal is to solve Problem 4.3.1 while minimizing the cost and limiting the complexity of the mechanism. This is expected to derive a safe solution to the cost estimation problem.

Problem 3.3.2. ***Cost of a safe way:*** *For safe ways $\mathcal{M}_i$ of UAVs that have been computed to avoid collisions with an obstacle in the aerial space (i.e., an uncontrollable UAV and an unexpected object like a bird in UAV's trajectory) in* (3.10)*, the cost of each safe way expresses the flight delay and/or increase in a travel distance compared to the original path.*

Let $g(.)$ be a function that calculates a UAV's temporal control inputs cost towards a UAV's trajectory path. The the path cost function $C_{\mathcal{M}}$ is expresses as

$$C_{\mathcal{M}} = \min_{u} \int_{t}^{t+\Delta_t} g(s(t), u(t)) \, \mathrm{d}t, \tag{3.12}$$

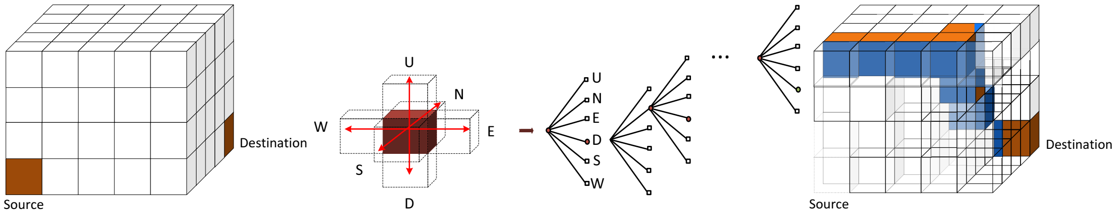
where $C_{\mathcal{M}}$ is dependent on the UAV's temporal state $s(t)$ in (3.1) and the control input $u(t)$ in (3.2).

Considering that the motion update of a UAV can affect the other UAVs' mobility, the cost which includes the cooperative motion of neighboring UAVs is illustrated as

$$C_{\mathcal{M}_{coop}} = \min_{n_i, n_j} \int_{t}^{t+\Delta_t} g(s_1, s_2, ..., s_m) \, \mathrm{d}t, \tag{3.13}$$

where $g(s_1, s_2, ..., s_m)$ is a function that combines the interactions caused by the update of a UAV n_i to each state s_j of another UAV n_j. Therefore, the UAV n_j mission is a set of way-points $\mathcal{M} = \{m^j_1, m^j_2, ..., m^j_k\}$ that the UAV will follow to accomplish its mission such that

$$\forall\, m^j_t \in \mathcal{M} : d_{i,j} \geq s_{safe}. \tag{3.14}$$



(a) Airspace representation (b) Path finding (c) Landing process

Figure 3.6: Volumetric path planning method.

Assumptions

The following assumptions were considered in our design of a heavy-traffic UAV network.

- Communication between UAVs and the GCC is based on a 5G or beyond 5G system.
- Time synchronization of UAVs and the GCC is based on GPS time.
- UAVs in a heavy-traffic UAV network must be equipped with a GPS transceiver. They must be able to fly freely in different directions, vertically and horizontally, and hover in space.
- A UAV network is a 3D airspace in which multiple two-dimensional planes are sliced spatially along the z-axis as shown in Fig. 3.2a.
- The arrival of UAVs in the heavy-traffic UAV network follows a Poisson distribution.
- To enable air-to-air and air-to-gound communication, each UAV operates in a half-duplex mode, transmitting and receiving airspace environmental sensing data simultaneously.

- For UAV flight safety efficiency, UAVs are assigned the flight layers that they must use during the flight and which can be changed while landing or avoiding an emergency event that might result in collisions.

In the next section, we will discuss how our CANA scheme assesses the flight collision risks and deduces the proper flight trajectories for UAVs in the airspace.

3.4 System Design and Algorithms

The CANA design aims to ensure that UAVs may fly safely in shared airspace where they have different mission purposes, which is called heterogeneous airspace. We achieve this safety goal by detecting potentially unsafe flying incidents and quickly responding to avoid collisions. This is achieved by a cooperative flight path calculation among UAVs during the execution of their missions. This section discusses in detail the UAV navigation algorithm and safety provisioning for use in a heavy-traffic UAV network environment.

3.4.1 Communication in UAV Flying Networks

The flying UAV network shown in Fig. 3.1a depicts an ultra-reliable cellular Vehicle-to-Everything (C-V2X)-like flying network. It involves FANET and 5G cellular infrastructure. We advocate the use of a sensing mechanism that exploits the mobility information obtained from the Vehicle-to-Infrastructure (V2I) (i.e., UAV-to-GCC) and Vehicle-to-Vehicle (V2V) (i.e., UAV-to-UAV) communications over 5G. Our design follows a User Equipment (UE)'s 5G V2X protocol stack (in terms of data plane in 5G) as shown in Fig. 3.3. FANETs enable the UAV-to-UAV communication, allowing UAVs to interact and exchange sensor information directly. The GCC interacts with UAVs in real time via vehicle-to-infrastructure (V2I) (i.e., UAV-to-GCC) communication to deliver information about the airspace environment and traffic conditions to achieve safe flights. We adopt the lightweight message exchange of the Cooperative Context Message (CCM) and the Emergency Context Message (ECM) in [12] to achieve safety goals.

The cooperation of UAVs and GCC via communication enables periodic dissemination of aerial environment information via CCMs and control messages involving reactions via ECMs. 5G provides a stable backbone for communication with the GCC, overcoming the limitations of purely ad-hoc networks by providing long-range communication to UAVs. The high bandwidth and low latency of the 5G network allow real time control and data exchange, which is crucial for complex cooperative maneuvers. Furthermore, 5G can facilitate communication between different FANETs operating in the same area, enabling larger-scale coordination of multiple UAV clusters to realize the full potential of cooperative UAV flights in diverse applications. The received awareness information serves to plan safe UAV missions. The details of communicating emergencies in a UAV network are included in B.1.

3.4.2 UAV Mission Planning

CANA planning module assesses the trajectories states of a set $\mathcal{D}$ of UAVs currently flying in the airspace to plan missions of a set $\mathcal{D}'$ of UAVs ready to begin their flights. The Algorithm 2 plans their collision-free trajectory paths to their assigned destinations. This algorithm defines each UAV's mission and calculates its way-point trajectory path.

This algorithm defines a function that computes the trajectories of each of a set of awaiting queued UAVs $\acute{\mathcal{D}}$. Lines 2-19 iterate through the UAVs and define each UAV's mission path P_j^t. In defining the mission path, lines 3 and 4 initialize the mission path by defining targeted destinations. Lines 5-19 define an addictive mission path planning method using Dijkstra's algorithm. For the number of UAVs $|\acute{\mathcal{D}}|$ planned for flights, the total destinations involved in their missions $M = \sum_{i=1}^{|\acute{\mathcal{D}}|} |\mathcal{M}_i|$. The computational

Algorithm 2 Addictive Mission Path Planning Algorithm

```
function CALCULATE_SAFE_MISSION_PATH($\mathcal{D}'$, $\mathcal{D}$)    ▷ $\mathcal{D}'$ is a set of UAVs that are currently waiting to fly in the controlled airspace, and $\mathcal{D}$ is a set of planned UAVs.
    for each $n_i \in \mathcal{D}'$ do
        $curNode \leftarrow getMissionBase(n_i)$    ▷ Initialize the current node with $n_i$'s the mission base.
        $M_i \leftarrow Extract\_Next\_UAV\_Mission(n_i, \mathcal{D})$    ▷ Define the destinations involved in the next UAV mission.
        while $M_i \neq \emptyset$ do
            $P_i \leftarrow curNode$    ▷ Initialize the UAV $n_i$'s mission path.
            for each neighbor $v_k$ of curNode do
                while $P_i \neq \emptyset$ do
                    $cost_k \leftarrow getEdgeCost()$    ▷ The edge cost is calculated according to (3.13).
                    if $cost_k <$ neighbors_cost then
                        $v_k \leftarrow min(v_k, v_{min} + cost_k)$    ▷ Update the cost to node $v_k$.
                        if $v_k \notin P_i$ then
                            $P_i.ENQUEUE(v_k)$
                        end if
                    end if
                end while
            end for
        end while
    end for
end function
```

complexity of this algorithm to plan $\acute{\mathcal{D}}$ UAVs is $\mathcal{O}(|\acute{\mathcal{D}}|M)log|\mathcal{V}|$. Note $|\mathcal{V}|$ indicates the total number of mission bases and mission destinations. This complexity increases with the increase in the number of UAVs whose flights are to be planned in the airspace. An appropriate 3D volumetric planning approach that provides a collision-free path to a UAV in heterogeneous airspace is proposed in CANA in the next section.

3.4.3 Volumetric Path Planning

The UAV carries out its mission by traversing through a sequence of waypoints ranging from the mission base to the mission destination. Within a 3D environment, the UAV path planning of heterogeneous missions is revealed to be complex. Our mechanism considers airspace in a cubic form, as shown in Fig. 3.6.

The atomic cube that is large enough to be safely occupied by the largest UAV flying in the airspace is called a voxel (i.e., cube). The size of a voxel is equivalent to s_{safe} in (3.9). The airspace is subdivided into multiple equal-sized voxels, and each voxel can only accommodate one UAV at an instant of time t_i. A UAV path is defined as a sequence of voxels, where the center of a voxel characterizes a UAV's position in space, as shown in Fig. 3.6a. We formulate a 3D voxel tree structure that the system will use to define the UAV's safe path. From a single voxel, the UAV moves to one of the horizontally or vertically adjacent voxels, or hovers in the same voxel. For simplicity, the maximum number of adjacent voxels is six. Note that a diagonal movement is not considered. Therefore, a tree structure representation $h : u \to \mathcal{R}^6$ defines the UAV's next displacement, as shown in Fig. 3.6b. During the landing (or takeoff) process or while avoiding an obstacle, a UAV can fly from a higher to a lower layer or vice versa. Fig. 3.6c shows the motion that was previously highlighted in Fig. 3.1b, thus demonstrating that UAVs are allowed to fly from a given layer to another layer while ensuring the safety of each flight. The decision to cross between two adjacent layers in this design follows knowledge-based decision-making informed by risk assessments, which is discussed in the next subsection.

3.4.4 Risk Assessment Metrics in a Heavy-traffic UAV Network

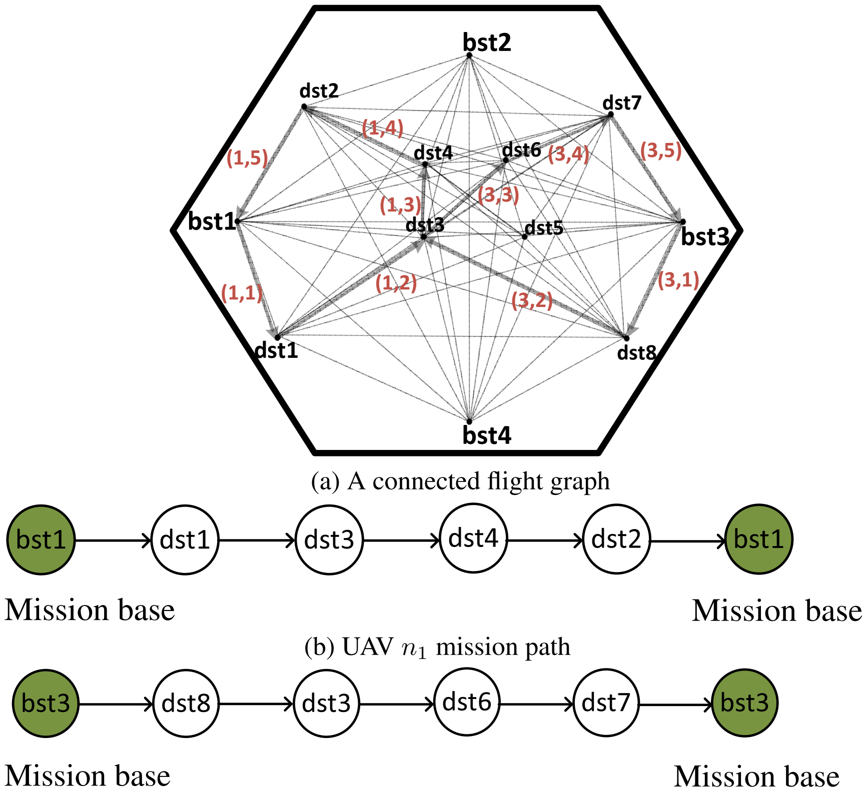
Collision risks in a heterogeneous UAV network are measured using three combined metrics: (a) UAV link delay, which measures the traversing delay, (b) collision probability, and (c) UAV congestion metric, which indicates the congestion level of flight link. As UAVs carry out their missions, the herein model assesses whether their trajectories can overlap spatio-temporally or not. Collision probability examines the risk of collisions occurring between the marginal space of neighboring UAVs. A metric that indicates the state of lane safety in an aerial environment is the virtual lane quality decision. Therefore, the congestion level assesses the future risk of collisions in the overall mission graph through virtual lane quality. Fig. 3.7 shows the flying path of a UAV for its missions, consisting of its start, intermediate destinations, and its end.

Fig. 3.7a illustrates a topology formed by mission bases, mission destinations, and transit nodes (e.g., charging stations). The flexibility of the air mobility enables that from any source (i.e., bst_m and dst_k), a UAV can reach any targeted mission base or mission destination, thus resulting in a mesh-type graph. Given the flying UAVs missions, Figs. 3.7b and 3.7c portray the end-to-end (E2E) paths from the mission bases bst_1 and bst_2 until UAVs n_1 and n_2 accomplish their missions. The risk assessment ensures that the UAV can safely achieve its assigned mission.

UAV Mission Delay

The GCC defines the mission waypoints of a given UAV n_i according to its capacity (i.e., battery and weight capacity). Each mission, $m_k \in \mathcal{M}$, heads toward a set of destinations that make the travel vertices set V, and its end-to-end (E2E) traversal path is made of $|V|$ destinations. For simplicity we assume that the travel delay d_i from a destination $dst_i \in V$ to $dst_{i+1} \in V$ is independent of the travel delay d_{i-1} from $dst_{i-1} \in V$ to $dst_i \in V$. Then, the mission E2E traversing delay is given as

$$E[D_{m_k}] = \sum_{i=1}^{k-1} E[d_i] = \sum_{i=1}^{k-1} \mu_i, \tag{3.15}$$



(a) A connected flight graph

(b) UAV n_1 mission path

(c) UAV n_2 mission path

Figure 3.7: The flying path of UAVs for their missions.

$$Var[D_{m_k}] = \sum_{i=1}^{k-1} Var[d_i] = \sum_{i=1}^{k-1} \sigma_i^2, \tag{3.16}$$

where $E[d_i]$ is the expected travel delay, which is a Gamma distribution $d_i \sim \Gamma(\kappa_i, \theta_i)$ [66], parameter κ_i is the mean travel delay and θ_i is proportional to the $Var[d_i]$ and inversely proportional $E[d_i]$. After the mission waypoints are defined, our algorithm defines the mission path such that the UAV will take the smallest delay to travel while completing its assigned task.

Collision Probability

For a UAV to safely fly in a heavy-traffic UAV network, its travel path must be a solution to Problem 4.3.1. The safe inter-UAV distance $\acute{s}_{safe}$ distance for the adjacent UAVs with the same length is

$$\acute{s}_{safe} = \{2(l_i + l_\epsilon)\} + l_{min}, \tag{3.17}$$

where l_i is the UAV's body length, l_ϵ is the navigation error (i.e., GPS error), and l_{min} is the minimum safety distance.

The system proposed herein ensures that the inter-UAV distance $d_{i,j} \geq \acute{s}_{safe}$ during the entire mission. However, due to flight conditions, such as a need to fly to the transit node (e.g., an energy charging station) or the existence of multiple arrivals at a mission destination, the inter-distance of UAVs may get smaller than $\acute{s}_{safe}$. We model a probabilistic risk assessment in terms of the time-to-collision (T_c) of the adjacent UAVs. Note that $T_c \in [T_{cmin}, T_{cmax}]$ where T_{cmin} is the minimal maneuverable time T_c and T_{cmax} is the maximal safety ensuring time T_c for two adjacent UAVs. Given two UAVs n_i and n_j, their probability of a collision event $P(n_i \otimes n_j)$ is computed as

$$P(n_i \otimes n_j) = \begin{cases} 1, & \text{if } T_c \leq T_{cmin}, & (3.18a) \\ 0, & \text{if } T_c \geq T_{cmax}, & (3.18b) \\ P_{n_i,n_j}, & \text{otherwise}, & (3.18c) \end{cases}$$

where (3.18a) means that the UAV n_i will certainly collide with the neighbor UAV n_j, (3.18b) implies that the UAV n_i is safely flying away of UAV n_j, and (3.18c) defines the relative collision risk of the adjacent UAVs n_i and n_j, which is computed as

$$P_{n_i,n_j} = 1 - \frac{T_c - T_{cmin}}{T_{cmax} - T_{cmin}}. \tag{3.19}$$

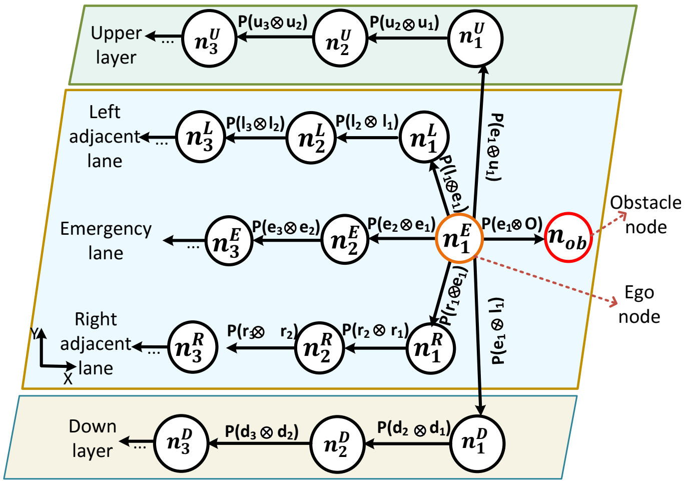


Figure 3.8: Probabilistic risk assessment in a 3D environment.

Aerial Virtual Lane

Even though UAVs have the freedom of multi-directional movement and flexible displacements, they can result in disastrous collisions if they are not controlled and coordinated. The GCC guides UAVs with their aerial routes, namely, virtual lanes in this chapter, that enable flexible control of UAVs. Fig. 3.1 describes the capability of aerial motion by flying in different horizontal layers, where each layer allows flights in different virtual lanes. Fig. 3.8 depicts a typical probabilistic risk assessment graph for aerial virtual lanes. UAVs from mission bases fly toward mission destinations in a versatile and efficient fashion that enables controllability through our collision risk detection and collision avoidance algorithm.

Let an obstacle node n_{ob} be detected in the airspace as illustrated in Fig. 3.8. A directly concerned UAV n_1^E collision risk expressed in collision probability $P(e_1 \otimes O)$ is calculated using (4.1b), defining the risk of n_1^E to collide with an obstacle O when it keeps the current flying kinematics. When e_1 is determined to be at collision risk, a decision that defines its control input for its motion update is considered. This design considers n_1^E to maneuver toward one of five virtual lanes or to hover (i.e., keeping the current voxel), depending on their safety metric. Those lanes first include an upper lane (U) representing the movement toward the upper layer. Secondly, in the emergency layer, an emergency UAV can maneuver toward the left lane (L), adjust (i.e., reduce) its speed in the current emergency lane (E), or change direction to the right lane (R). Lastly, it can move to the down layer, representing the down lane (D). The maneuvers include hovering, changing the flying layer, or kinematics updates such as acceleration, deceleration, and direction change according to a decision logic that meets both the requirements of safety and flight performance.

Definition 3.4.1. ***Virtual-lane quality****: Virtual-lane quality is a metric that indicates the safety level of the route that a UAV will follow while avoiding collisions.*

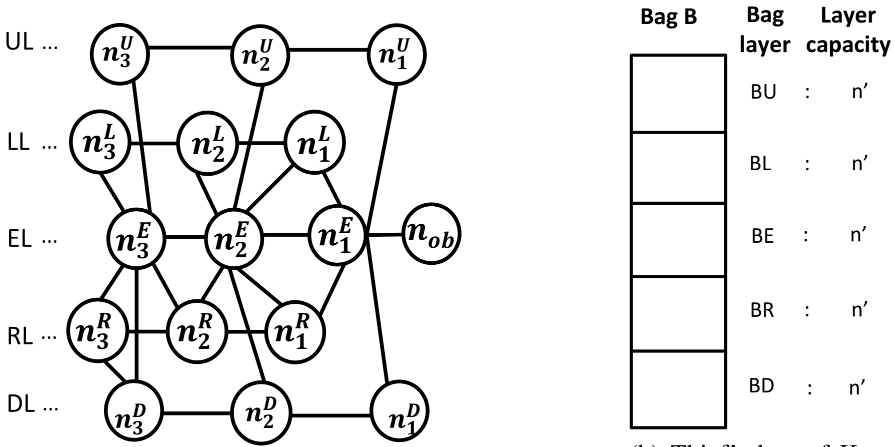
This metric extends the safe lane quality definition in [12] by accommodating $3D$ motion freedom for UAVs. Considering a UAV's next movement toward its neighboring UAVs and collision probabilities $p_1, p_2, ...p_u$, a safety probability denoted as q_i given as

$$q_i = 1 - p_i, \tag{3.20}$$

where p_i is the probability of a collision event $(n_i \otimes n_{i-1})$. Considering a flying period Δ_t, p_i expresses the risk that a UAV n_i will collide with UAV n_{i-1} at flight time $t \leq \Delta_t$. Given a list of UAVs in the same direction with n_i and their corresponding safety probabilities $\{q_1, q_2, ..., q_u\}$, we can deduce the safety quality. A virtual-lane quality S_L is a lane metric that indicates the safety measure of a specific virtual lane during the emergency flight decision. Please refer to Appendix B.2 for safe lane quality computations details. The n_i path maneuver is expressed as

$$S_L = \prod_{i=1}^{u} q_i. \tag{3.21}$$

After evaluating the safe lane qualities toward all possible maneuvers (i.e., five move-



(a) Collision avoidance graph

(b) Thief's bag of Knapsack problem

Figure 3.9: Collision avoidance problem hardness.

ments) in the aerial environment, a target UAV flies toward the safest direction, denoted as D_x^*, and given as

$$D_x^* \leftarrow \max_{D_x \in \{L,E,R,U,D\}} (S_L, S_E, S_R, S_U, S_D). \tag{3.22}$$

Section 3.4.5 details how a UAV minimizes collisions in high-risk airspace.

3.4.5 Collision Avoidance Algorithms

An optimal UAV flight is one that reduces the probability of position interference, thereby maximizing the inter-UAV distances by facilitating the real-time flights of UAVs in the shared airspace. This section illustrates the algorithms and optimizations that contribute to optimal safe flights of UAVs.

UAV Collision Avoidance Problem Hardness

Given the multilayered airspace as illustrated in Fig. 3.1b, in which there is a potential collision risk with an obstacle, the maneuver decision is obtained by solving Problem 4.3.1.

Fig. 3.9 shows the maneuver definition hardness. Fig. 3.9a shows the complexity of UAV n_1^E to define collision-free maneuvers while avoiding n_{ob}. We consider the airspace lanes to be a thief's bag made of five layers $\{BU, BL, BE, BR, BD\}$ as shown in fig. 3.9b, where BU equals the upper lane UL, BL equals the left lane LL, BE equals the emergency lane EL, BR equals the right lane RL, and BD equals the down lane DL, respectively. A knapsack problem [32], which consists of collecting different items in a thief's bag while maximizing the total collected items without exceeding the bag capacity, is a well-known NP-complete problem. We demonstrate that the knapsack problem is a special case of Problem 4.3.1. As the knapsack problem is an NP-complete problem, this means that Problem 4.3.1 is also an NP-complete problem, indicating that it is therefore infeasible to find its solution in polynomial time. The relaxation of Problem 4.3.1 through heuristic approximation and optimization alleviates its computational load. The trajectory planning and execution of many UAVs require extensive decision-making. Given the flexibility of aerial motion planning, each UAV's movement in the airspace represents an obstacle that is to be avoided by the rest of the UAVs to ensure safe flights. A safe flight path-finding decision is an NP-complete problem.

The procedure and mechanisms for solving the Problem 4.3.1 and algorithms for minimizing collision avoidance costs and the computation of flight speed and direction adjustments toward solutions for Problem 3.3.2 are described in Section 3.4.5.

Optimization Problem in Collision Avoidance

UAV collisions may occur when two or more different UAVs attempt to visit the same space voxel simultaneously. By taking into account the UAV size l_i, navigation error l_ϵ (i.e., GPS positioning error), and safety minimal gap (l_{min}), it can be determined that the UAV n_i will be in a collision when the following equation is satisfied.

$$\begin{aligned} &\exists\, n_j \in \mathcal{D},\ j \leq n \ \ \&\ \ i \neq j, \\ &\text{s.t. } d_{n_i,n_j}(p_{n_i}, p_{n_j}, \dot{p}_{n_i}, \dot{p}_{n_j}, \theta_{ij}) \leq s'_{safe}. \end{aligned} \tag{3.23}$$

where $\mathcal{D}$ is a set of UAVs in a heavy-traffic UAV network, $d_{n_i,n_j}(.)$ define the inter-distance of a pair of UAVs, $p_{n_i} = (x_i, y_i, z_i)$ is the airspace position of UAV n_i, $\dot{p}_{n_i}$ is its flight speed, and θ_{ij} is the angle between the flight directions of two UAVs at time t.

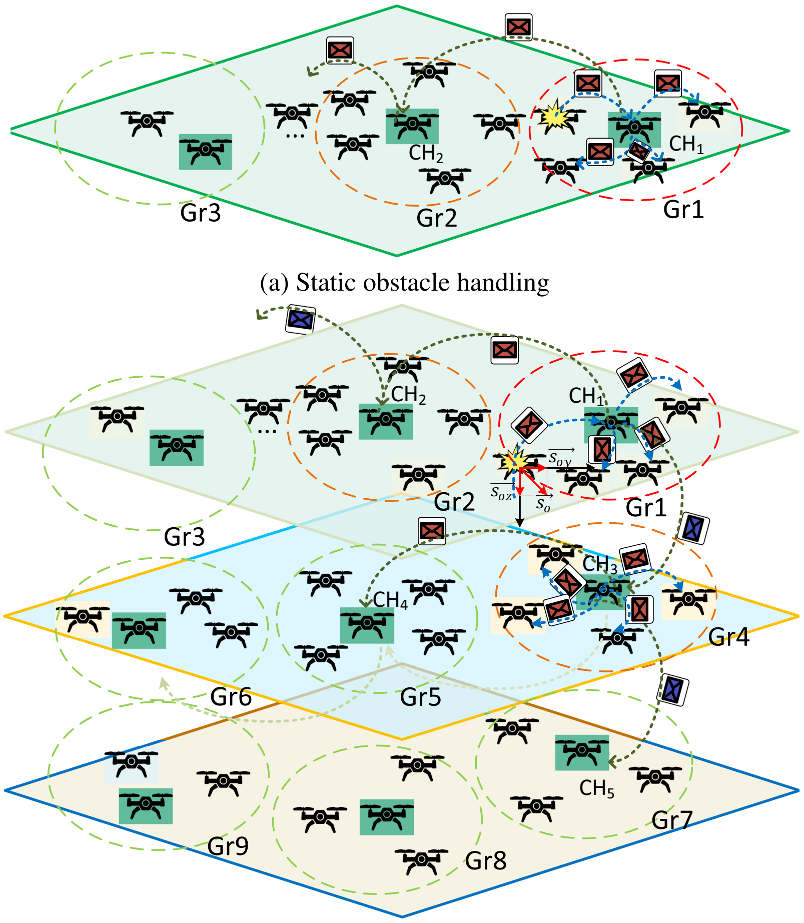
The distances between UAVs must be greater than the safety distance to avoid collisions between UAVs. We avoid collisions by defining the UAVs' speeds and direction angles to satisfy the following equation.

$$\begin{aligned} &d_{n_i,n_j}(p_{n_i}, p_{n_j}, \dot{p}_{n_i}, \dot{p}_{n_j}, \theta'_{n_i,n_j}) > s'_{safe}, \\ &\text{s.t. } \forall n_i, n_j \in \mathcal{D},\ i \neq j. \end{aligned} \tag{3.24}$$

where $\dot{p}_{n_i}$ is the flight speed of UAV n_i, and θ'_{n_i,n_j} is the changed angle between the flight directions of UAV n_i and n_j. The CANA updates the flight speed and direction of UAVs such that UAVs do not collide with each other while also minimizing the maneuver change cost for collision avoidance. That means that, by knowing the set of UAVs' positions P, the set of their speeds V, and the set of all angles between all pairs of UAVs Θ, the GCC can solve the following optimization problem:

$$\begin{aligned} &\text{Maximize } (V, \Theta) \\ &\qquad \text{s.t. } \forall\, n_i, n_j \in \mathcal{D},\ n_i \neq n_j, \\ &\qquad\quad p_i, p_j \in P,\ \ \dot{p}_{n_i}, \dot{p}_{n_j} \in V, \\ &\qquad\quad d_{n_i,n_j}(p_i, p_j, v_i, v_j, \theta_{i,j}) \geq s'_{safe}, \end{aligned} \tag{3.25}$$

where $\dot{p}_{n_i}$ and $\dot{p}_{n_j}$ are the speeds of UAVs n_i and n_j, respectively. The above optimization problem is non-linear, non-convex, and has too many variables. In particular, there are



(a) Static obstacle handling

(b) Dynamic obstacle handling

Figure 3.10: Cluster-based emergency handling in aerial environment.

too many unnecessary operations taking into account speeds, directions, inter-distances, and the maneuver costs of all maneuvered UAVs in UAV set $\mathcal{D}$. Therefore, it is necessary to appropriately divide the set of UAV $\mathcal{D}$ to reduce computational operations.

Collision Group

To reduce the unnecessary computation load of the optimization problem, this mechanism handles the entire set of UAV $\mathcal{D}$ into multiple collision groups following their cluster

Algorithm 3 UAV Safe Motion Planning

```
function CALCULATE_SAFE_NEXT_MOVE($\mathcal{G}, \mathcal{M}, \mathcal{D}$)    ▷ $\mathcal{G}$ is a connected graph constructed by mission-based and mission destinations, and $\mathcal{M}$ is a UAV's planned mission destinations.
    for $each\ n_i \in \mathcal{D}$ do ▷ Grouping the flying nodes based on their safety conditions.
        if $n_c \neq null$ then    ▷ Compute the edge cost when the current vertex has a predecessor.
            $P_{c,p} \leftarrow Compute\_Probability(T_c)$
            $C \leftarrow Group\_UAV(P_{c,p})$    ▷ Group UAVs based on their collision probability.
        end if
    end for
    $C_{di.next} \leftarrow Compute\_Destination\_Cost(d_i.next)$    ▷ Compute the flight cost of the UAV toward the planned next destination.
    for $each\ vertex\ u \in \mathcal{M}$ do    ▷ Evaluate the safety cost of moving to every destination among the planned destinations based on their virtual lane quality according to (4.14).
        $C_u \leftarrow Compute\_Destination\_Cost(u)$
        if $C_u < C_{di.next}$ then
            $di.next \leftarrow u$
        end if
        $d_{i+1} \leftarrow di.next$
    end for
    $C \leftarrow Update\_Collision\_Group(\mathcal{D})$
    $\mathcal{M} \leftarrow DEQUEUE(d_i)$
    $d_i \leftarrow d_{i+1}$
     return $d_i$    ▷ Return the next destination the UAV will fly to.
end function
```

formation of nodes in Ad Hoc networks [31]. Each collision group satisfies the following:

$$\begin{gathered}\bigcup_{C_n \in \mathcal{C}} \mathcal{C}_n = \mathcal{D}, \quad \bigcap_{C_n \in \mathcal{C}} \mathcal{C}_n = \phi, \\ \mathcal{C}_n = \{(n_c^i, n_c^j) |\ n_c^i, n_c^j \in \mathcal{D}, \\ \exists t' \in [0, \Delta_t] : d_{n_c^i, n_c^j} < s'_{safe}\},\end{gathered} \tag{3.26}$$

where Δ_t is the maneuver time of an emergency UAV.

Definition 3.4.2. ***Collision group:*** *A set of UAVs $\mathcal{C} \subset \mathcal{D}$, $\mathcal{C} = \{n_c^1, n_c^2, ..., n_c^m\}$ is a collision group if $\forall n_c^k, n_c^l \in \mathcal{C}$, such that $D(n_c^k, n_c^l) \leq s'_{safe}$.*

A collision group is defined as a set of pairs of UAVs where the inter-distance is less than or equal to the safety distance. To avoid the potential of collisions, UAVs are organized into clusters where either the cluster Head (CH) or GCC solves the optimization problem, and each cluster is treated as a collision group, thus reducing the computational load. Fig. 3.10 shows an illustration of collision groups. An emergency event is shared across the UAV network via lightweight message sharing.

The safety awareness is shared being a static obstacle (i.e., hovering obstacle) as shown in Fig. 3.10a or a dynamic obstacle (i.e., dropping obstacle) as shown in Fig. 3.10b. Static obstacle is handled within a single layer without affecting adjacent layers unless needed for maneuvers. Besides, a dynamic obstacle affects the safety of all layers through which they pass. Through UAV groups, the maneuver can be dealt with by a smaller number of UAVs, thus keeping the remaining UAV trajectories safe.

We classify UAV collision groups into three main categories, depending on how they might be affected by an obstacle. Those categories are class1 UAVs with a risk of colliding with obstacles, class2 UAVs likely to be affected by the maneuvers of class1 UAVs, and class3 safe UAVs. The class1 category, which is shown by group Gr1 in Fig. 3.10, consists of UAVs at the line of collision of the obstacle. This category is a zone of no crossing, meaning that all flying UAVs with a trajectory crossing that spot have to detour for collision risk avoidance. UAVs in that zone need to immediately evacuate toward safer zones using various maneuvers. The class2 category consists of UAVs depicted by groups Gr2 and Gr3 in Fig. 3.10, which are likely to be affected by the maneuvers of Gr1 UAVs. This group needs to keep their trajectories while carefully observing the motion updates of other UAVs. Lastly, the class3 category, as depicted in groups Gr4 and Gr5 of Fig. 3.10, consists of UAVs that are distant enough from the emergency and can therefore

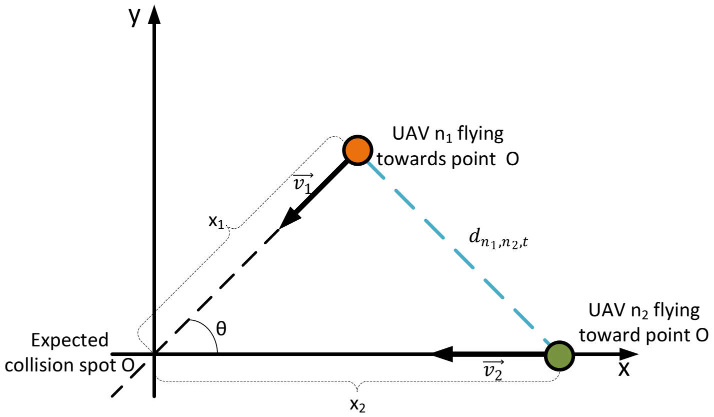

y
UAV n_1 flying towards point O
$\vec{v_1}$
x_1
$d_{n_1,n_2,t}$
UAV n_2 flying toward point O
θ
Expected collision spot O
$\vec{v_2}$
x
x_2

Figure 3.11: Collision avoidance by UAVs' velocity change.

safely continue along their pre-planned motion paths. All groups need to be updated, reflecting UAV maneuvers in real time.

3.4.6 Path Maneuver Through Heuristic Solutions

Algorithm 3 describes the heuristic way in which the planned path updates are computed to avoid possible collisions. The proposed heuristic way to solve the UAVs' collision problem reduces the size of the collision group by hovering specific UAVs.

Velocity Change Maneuver Method

The deceleration/acceleration collision avoidance method is applicable when UAVs risk colliding along the same virtual lane. Thus, these UAVs differ in speed but head in the same direction.

Fig. 3.11 represents a 2D scenario in a layer in the airspace (as shown in Fig. 3.1b), where two UAVs n_1 and n_2, with respective velocities v_1 and v_2, are expected to collide at the collision spot O. These UAVs are distant from the point O with distances x_1 and x_2 and $d_{n_1,n_2,t}$ is their temporal inter-distance. At the time n_1 and n_2 reach O, their inter-distance $d_{n_1,n_2} = 0$. Therefore, decelerating one of the UAVs, which causes it to arrive

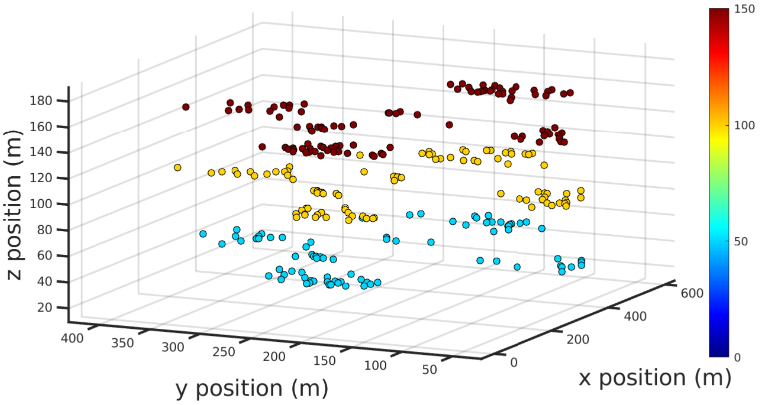


Figure 3.12: UAVs positioning before the accident.

later at the point of impact, is efficient enough as a maneuver. That is done along with optimization of the problem to limit the maneuver change cost.

Direction Change Maneuver Method

In a scenario where two or more UAVs' inter-distances are reducing over time, the GCC keeps track of each UAV's direction while adjusting it to increase the safety of flights. The change of the direction of the UAV's flight when two or more UAVs are about to collide follows five operations of the voxel implementation. The following section will assess the performance of this scheme in terms of safety.

3.5 Performance Evaluation

This section describes our simulation environment, the evaluation conditions, and the performance results of our safe UAV navigation algorithm in a UAV network with heavy traffic.

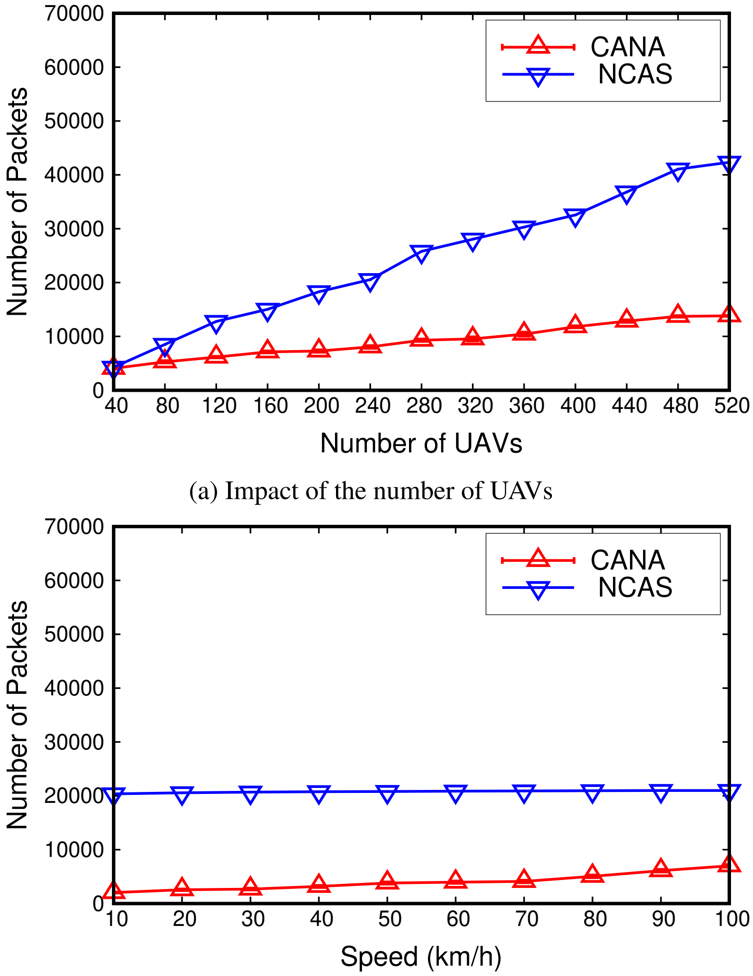
Table 3.1: Simulation Configuration

Parameters	Description
1. Airspace	A space of both length and width of 2km long, and a height of 200m.
2. Number of UAVs	Number of UAVs varies from $40 \sim 520$.
3. Flying speed	UAVs fly at speeds ranging from $10 \sim 100km/h$.
4. Speed deviation	UAV speed deviation varies from $1 \sim 10km/h$.
5. Position error	A position (i.e., GPS) error $0 \sim 20m$.
6. ECM transmission rate	10 packets per second.

3.5.1 Simulation Setup

We conducted a simulation of a UAV network in the OMNeT++ simulation framework [37]. We adopted a 5G simulation of the 5G New Radio user plane simulation model (Simu5G) for INET & OMNeT++ [67] and adapted it for use with UAVs specifically. Conceptually, multiple layers are possible in this work. However, to simplify our performance evaluation, this evaluation considers airspace in which UAVs fly in three layers: 50m, 100m, and 150m high, respectively. Table 4.2 summarizes the configuration of this simulation. Note that the source code of our CANA simulation is available at `https://github.com/jaehoonpauljeong/CANA` in GitHub and its demonstration video clip is available at `https://youtu.be/a_i83WxcuwM` in YouTube.

We validated our UAV navigation scheme in an airspace in the form of a regular cuboid that is 2km in length, 2km in width, and 200m in height. We deployed five different mission destinations, and UAVs were introduced in the airspace from four mission bases. Fig. 3.12 shows the UAVs' spatial position before the occurrence of an emergency in the airspace. The color bar at the right shows the heights of the airspace UAVs flying in.



(a) Impact of the number of UAVs

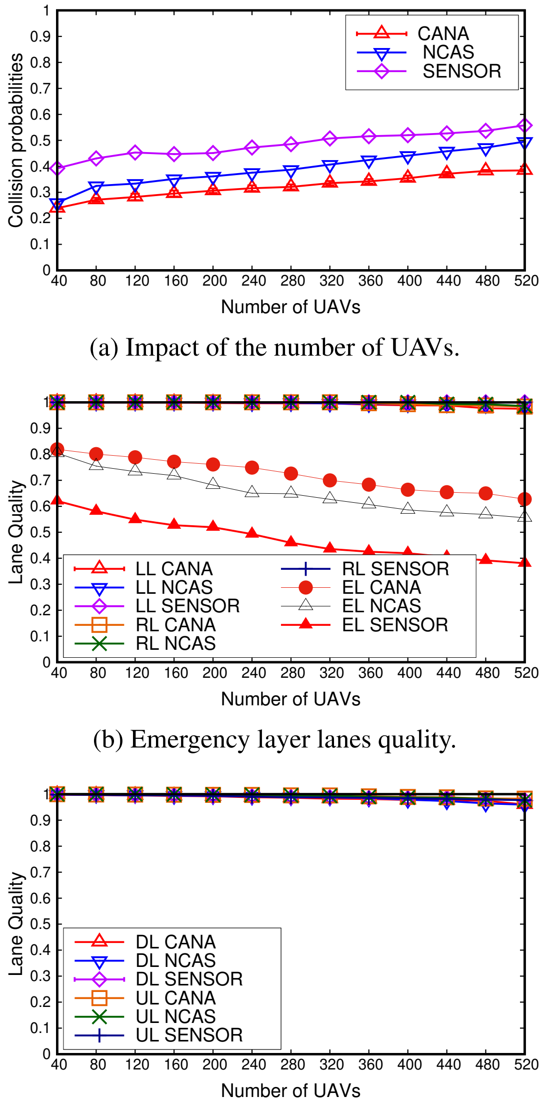
(b) Impact of speed

Figure 3.13: Impact of CANA on communication overhead.

3.5.2 Evaluation Parameters and Metrics

The evaluation of our UAV's safe navigation considered the following evaluation settings:

- **Evaluation Parameters:** The parameters for evaluation include the impact of *(i)* the number of UAVs, *(ii)* the maximum flying speed, *(iii)* the speed deviation, and *(iv)* the position error.

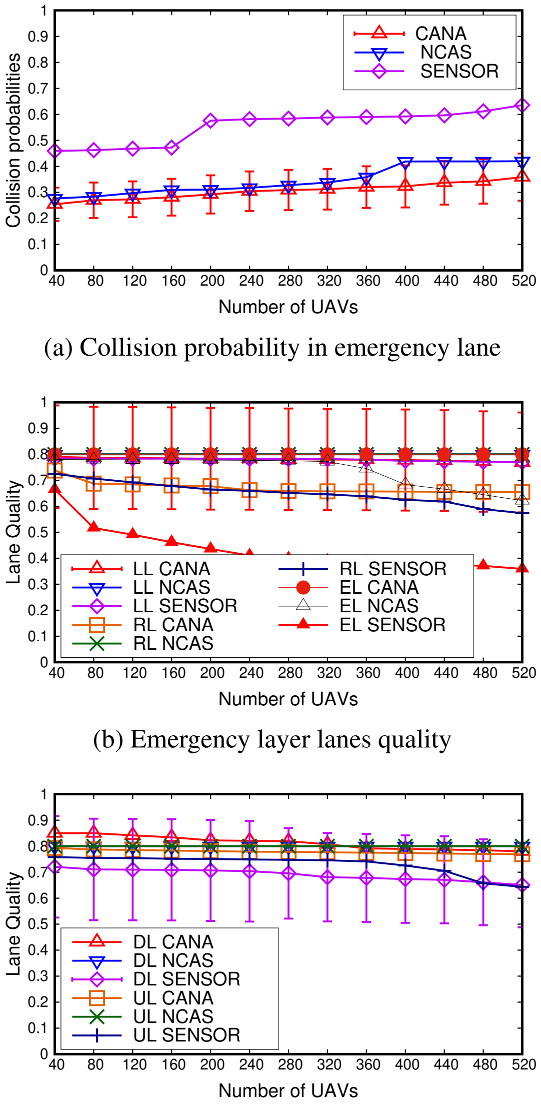


(a) Impact of the number of UAVs.

(b) Emergency layer lanes quality.

(c) Adjacent layers lane quality.

Figure 3.14: Impact of the number of UAVs, a case of a single static obstacle.



(a) Collision probability in emergency lane

(b) Emergency layer lanes quality

(c) Adjacent layers lane quality

Figure 3.15: Impact of the number of UAVs in the case of multiple obstacles.

- **Performance Metrics:** The metrics for evaluation are the communication overhead to measure the communication performance, the collision probability which measures collision risk when a UAV maneuvers toward an aerial virtual lane, and the safety lane quality to assess how safe the airspace is during the execution obstacle avoidance maneuvers.
- **Baselines:** We compare CANA with legacy situational awareness methods, including a sensor-based approach (i.e., LIDAR) [50] and a networked approach (i.e., NCAS [47]). Unlike CANA and NCAS, sensor-based schemes detect objects that are only within their line-of-collision (LOC).

A confidence interval of 95% is also used in this simulation to test CANA performance evaluation results.

3.5.3 Simulation Results

We investigate in this section the safety performance by assessing the communication overhead required by networked approaches, assessing the collision risk of each scheme when avoiding obstacles through collision probabilities, and evaluating the safety quality of airspace for the needed UAV maneuvers through its virtual lane quality.

Impact of CANA on Control Overhead

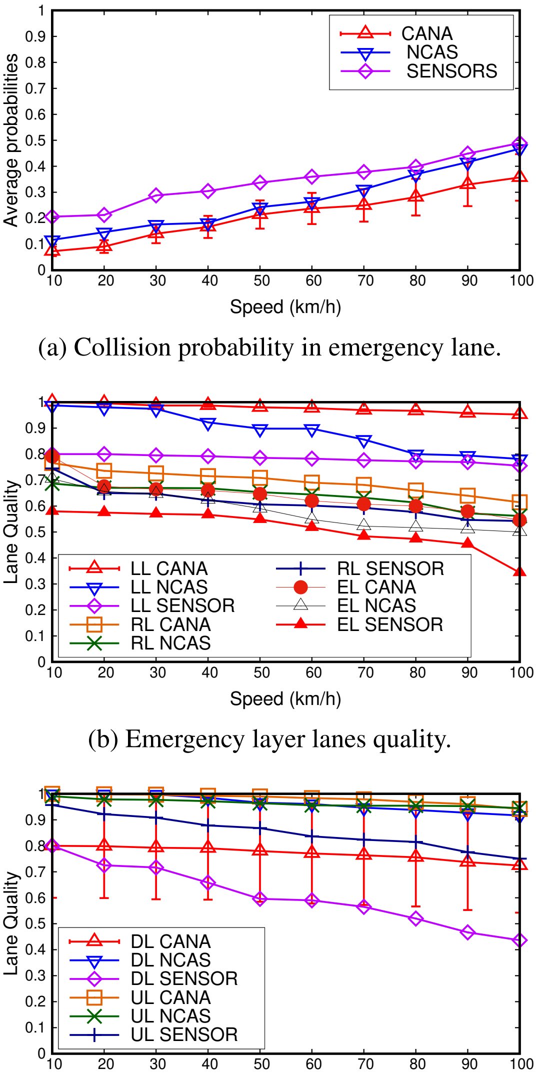
CANA, like NCAS, uses a context awareness model that relies on information sharing over a network. Such a model achieves its purpose by transmitting awareness packets, thus creating significant overhead. Therefore, a model that will reach the awareness goal by sending fewer packets performs better within a computation-constrained environment. We compared CANA and NCAS with different numbers of UAVs and different speeds to identify and disseminate collision risk in a UAV network.

For the evaluated number of aerial vehicles, which varied from $40 \sim 520$ UAVs, flying with a maximal speed of $80km/h$, CANA control overhead was found to be more stable compared to NCAS, the overhead of which continued increasing with an increasing number of UAVs. CANA reduces up to 63.3% of the control overhead of NCAS, as shown in Fig. 3.13a. We also evaluated the control overhead required by 320 UAVs flying with speeds $10km/h \sim 100km/h$, as shown in Fig. 3.13b. In this situation, CANA reduced up to 66.6% of the control overhead required by NCAS, thus outperforming NCAS for all the assessed parameters.

Impact of the Number of UAVs

This section assesses the impact of the number of UAVs on the overall safety of a UAV network. We evaluate the safety of a situation in which an obstacle is introduced in the aerial by computing the collision probabilities and the quality of virtual lanes of the obstacle's current layer and neighboring layers. Upon identifying an obstacle in the airspace, the concerned UAVs detour toward the spatial areas in the airspace without risks of colliding, called safe zones. Virtual lanes enable the maneuvering decision toward the safest zone while considering the paths of neighboring UAVs.

Fig. 3.14 assesses the impact of the number of UAVs flying with a maximal speed of $80km/h$, on the overall safety of a UAV network when a static obstacle (i.e., a hovering UAV due to some breakdown) is in the airspace. Within the emergency layer, we assess the safety condition within an Emergency Lane (EL), a Right Lane (RL), and a Left Lane (LL). We also assess the safety condition within the adjacent layers. The upper layer is considered to be an Upper Lane (UL) and the down layer is considered to be a Down Lane (DL). We assess situational safety by the collision probabilities obtained from simulation results and the quality of virtual lanes of the obstacle's current layer and neighboring layers. A lower collision probability indicates a safer airspace. Fig. 3.14a

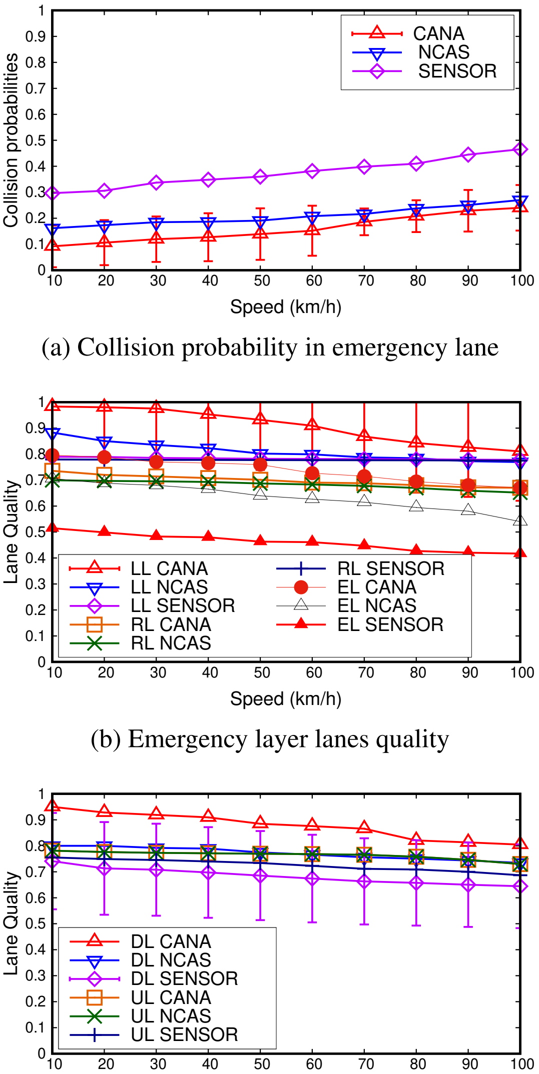


(a) Collision probability in emergency lane.

(b) Emergency layer lanes quality.

(c) Adjacent layers lane quality.

Figure 3.16: Impact of UAV speed in the case of a static obstacle.



(a) Collision probability in emergency lane

(b) Emergency layer lanes quality

(c) Adjacent layers lane quality

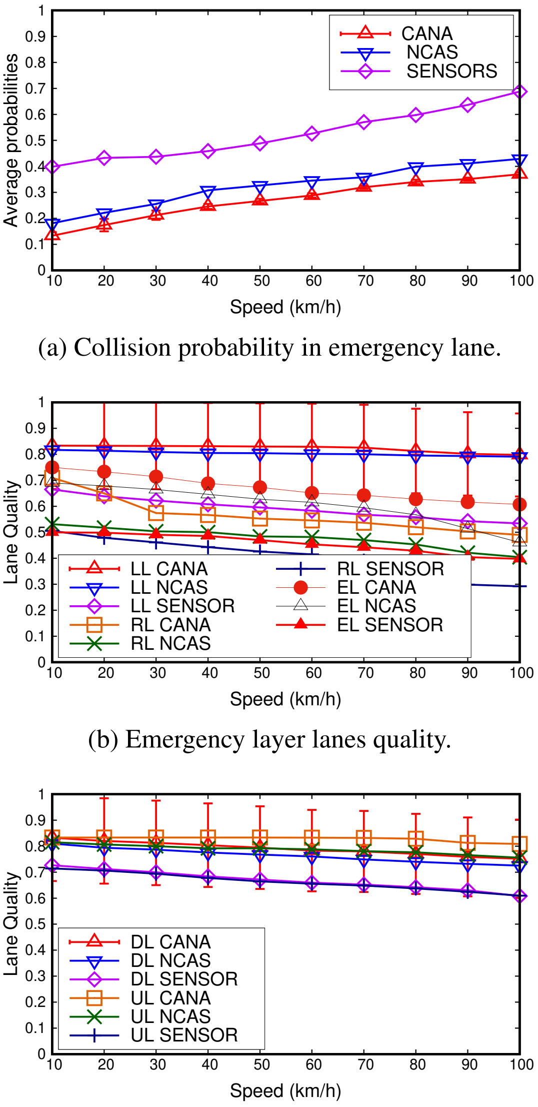
Figure 3.17: Impact of UAV speed in the case of multiple static obstacles.

shows that for all numbers of UAVs assessed herein, CANA was found to be safer than the baseline schemes. CANA showed a lower risk of collision than any of the tested baselines. As shown in Fig. 3.14b, which shows the safety quality in the emergency layer, and in Fig. 3.14c, which shows the safety quality of the upper and lower layers, the networked approaches (i.e., CANA and NCAS) were found to be capable of providing maneuver possibilities to different layers with better lane quality compared to the sensor-based approach.

Fig. 3.15 shows the impact of the number of UAVs on overall safety when multiple static obstacles are introduced in the UAV network. For this evaluation, three simultaneous obstacles were introduced while creating multiple collision spots in the aerial environment. These are UAVs that hover in the airspace by blocking the direction of other UAVs. The results in Fig. 3.15a show that CANA achieved the lowest collision risk with all numbers of UAVs tested. It also had the highest safety quality, as shown in Figs. 3.15b and 3.15c, thus indicating that it provides better maneuver options.

Impact of Speed

We evaluated the impact of the maximal speed on the safety of UAVs given the existence of static and dynamic obstacles in the UAV network. As a dynamic obstacle, we considered a dropping UAV falling unexpectedly towards the ground in the airspace. The tested speeds vary from $10km/h \sim 100km/h$. Fig. 3.16 shows the impact of speed on the different schemes when a static obstacle is identified in the airspace. Fig. 3.16a shows that CANA had the lowest collision probability, thus being safer than the compared schemes in assessing the overall safety of UAVs. CANA shows the lowest risk of collision compared to the baseline methods. While defining the maneuver dynamics of a UAV when an obstacle is identified in the airspace, Figs. 3.16b and 3.16c, respectively, show that the networked approaches (i.e., CANA and NCAS) provide higher quality maneuver

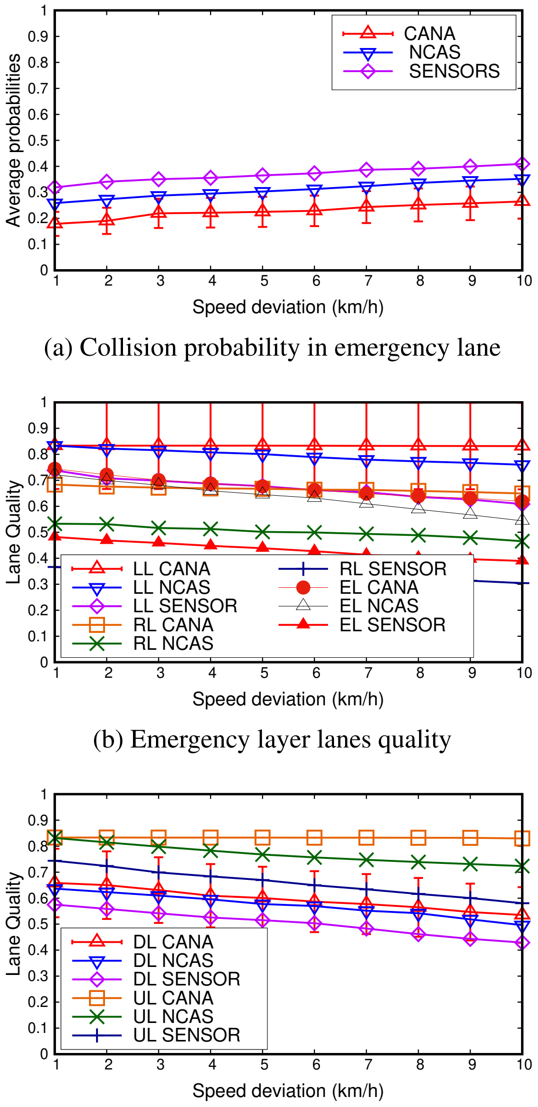


(a) Collision probability in emergency lane.

(b) Emergency layer lanes quality.

(c) Adjacent layers lane quality.

Figure 3.18: Impact of speed in the case of a dynamic obstacle.

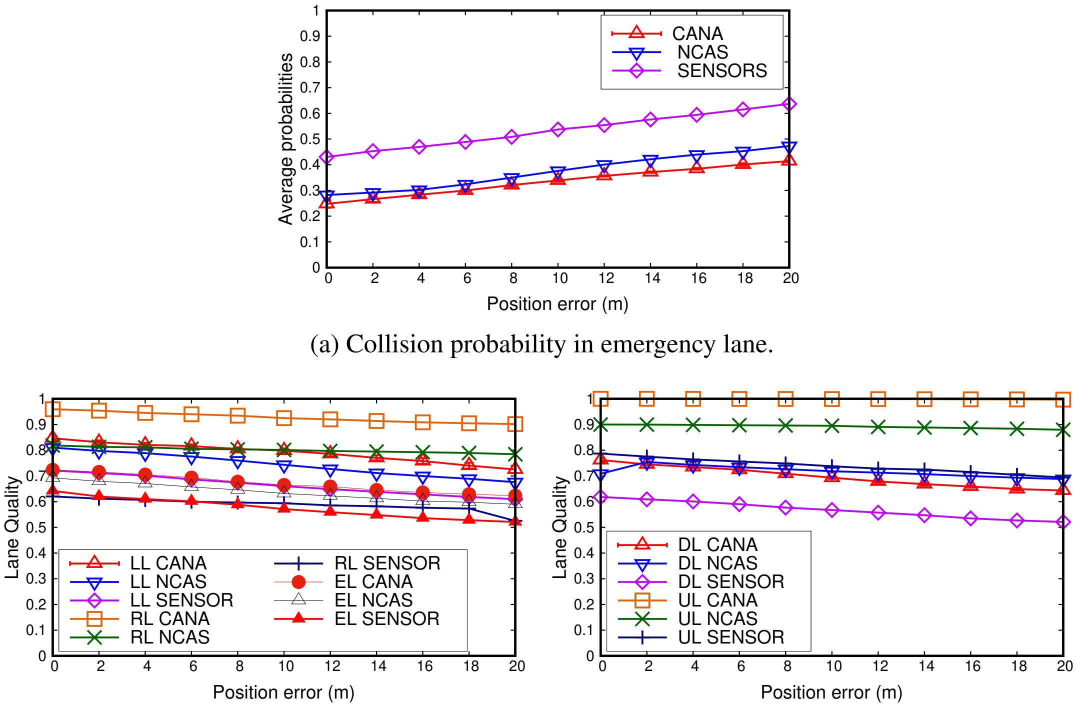


(a) Collision probability in emergency lane

(b) Emergency layer lanes quality

(c) Adjacent layers lane quality

Figure 3.19: Impact of speed deviation in the case of a dynamic obstacle.



(a) Collision probability in emergency lane.

(b) Emergency layer lanes quality.

(c) Adjacent layers lane quality.

Figure 3.20: Impact of position error in the case of a dynamic obstacle.

possibilities in different layers and virtual lanes than the sensor-based approach.

We also evaluated the impact of speed on UAVs' safety when multiple static obstacles were identified in the airspace. To this end, we introduced three obstacles, which are UAVs hovering in the airspace by causing the UAVs' direction blocking, and we tested the overall airspace safety at various speeds, as shown in Fig. 3.17. The results in Fig. 3.17a showed that networked awareness mechanisms outperformed the sensor-based mechanism. It also shows that CANA was found to be much safer than all the other mechanisms. On the other hand, Figs. 3.17b and 3.17c, respectively, show the state of the lane quality in the emergency layer and other layers. The networked awareness schemes provide safer maneuver options than the sensor-based scheme. Thus, CANA is the safest.

Fig. 3.18 shows a case testing the impact of speed when a dynamic obstacle (i.e., a vertically dropping UAV) is identified in the airspace. The results in Fig. 3.18a show that CANA has the lowest collision risk compared with the baselines. Figs. 3.18b and 3.18c show that the networked awareness schemes have higher lane quality than the sensor-based scheme, and CANA outperforms them all.

Impact of Speed Deviation

We evaluated the impact of speed deviation on the safety of UAVs given the existence of a dynamic obstacle in the UAV network. For UAVs flying with a maximal speed of $80km/h$, we tested the speed deviations varying from $1km/h \sim 10km/h$. Fig. 3.19 shows the impact of speed deviation on compared schemes.

Fig. 3.19a shows that CANA shows the lowest collision risk compared to the baseline schemes. Figs. 3.19b and 3.19c show the impact of speed deviation on lane quality in the emergency layer and adjacent layers, respectively. These results show that CANA performs better by providing maneuver options with higher lane quality both in the emergency layer and adjacent layers.

Impact of Position Error

We assessed the impact of position error when the safety assessment schemes identified a dynamic obstacle, by configuring a position error varying from $0m \sim 20m$ [68]. Fig. 3.20 shows the performance results of the compared schemes with position error.

Fig. 3.20a shows the collision risks in terms of collision probabilities and CANA is safer than the compared schemes. The communication-based schemes show better performance than the sensor-based schemes. Figs. 3.20b and 3.20c show the lane quality in the emergency layer and its adjacent layers, respectively. DL is more crowded than UL due to several UAVs landing at the destination, while others take off to the base station.

As a result, UL has higher quality than DL, as shown in Fig. 3.20c. Therefore, it can be concluded that CANA has higher lane quality in all the tested layers, thus outperforming the compared baselines.

3.5.4 Discussion

The performance results demonstrate that CANA excels in maintaining safety and assessing collision risks within UAV traffic, even under challenging conditions such as high UAV density, varying speeds, and the presence of obstacles. Its superior performance in reducing collision probabilities and maintaining high lane quality suggests that a similar cooperative, context-aware system could significantly benefit UAVs. More transparent and predictable airspace can be achieved by incorporating technologies that allow manned aircraft to share awareness and intentions, much like UAVs in the CANA system. This shared awareness can enable pilots and automated systems to anticipate potential conflicts better and proactively adjust flight paths, thereby minimizing the risk of mid-air collisions.

Moreover, the adaptability of CANA to different operational conditions indicates that a comparable system in shared airspace could effectively handle the complexities of mixed air traffic, including the diverse characteristics of different aircraft types and the unpredictable nature of airspace emergencies. The ability of CANA to maintain safety and efficiency under varying conditions suggests that a similar approach can enhance the overall airspace safety and efficiency. Our research aims at a more integrated, cooperative air traffic management system, inspired by the principles demonstrated in our CANA, which represents a significant step towards reducing mid-air collision risks and improving air travel's overall safety and efficiency.

3.6 Conclusion

This chapter proposes a Collision-Avoidance Navigation Algorithm (CANA) that autonomously prevents collisions in a heterogeneous heavy-traffic UAV network. It designs multi-dimensional airspace and virtual aerial roads to model UAV mobility with collision avoidance. A communication-based air mobility information sharing is proposed to sense and assess the flight conditions. Near-to-happening collisions are detected and avoided through probabilistic risk assessment and maneuvering UAV trajectories. It also explores collision avoidance algorithms based on an optimization and heuristic solution in a shared airspace. The simulation evaluation shows that UAV controls with our CANA scheme outperform the two compared baseline methods. As future work, we will apply AI schemes such as graph attention networks to improve our approach further. Aside from this simulation-based performance, we will use our CANA scheme to commercial off-the-shelf UAVs and conduct field experiments in the future.

Chapter 4

ML-SAINT: Machine Learning-Assisted Self-Adaptive Interactive Navigation Tool for Parcel Delivery Scheduling in Road Networks

The vehicular route planning in urban environments optimizes traffic and improves commute efficiency, which is important for enhanced vehicular navigation. The navigation task becomes more complex when it is required to satisfy time-critical driving applications such as parcel delivery. The Graph neural network approach is a typical solution to provide safe and efficient routes in urban mobility [69, 70, 71]. Through the pattern's relationship of the road graph represented as edges of a graph, it can capture mobility information such as speed, positions, acceleration, and orientation. Using classification models, it can predict the existence of road congestion and thereafter deduce a less congested vehicle route that guarantees fast destination arrivals. This chapter presents a Machine Learning-Assisted Self-Adaptive Interactive Navigation Tool for Parcel Delivery Scheduling in Road Networks. It investigates the performance of GCN in predicting the best route, which exploits the rapid and accurate spatial-temporal traffic information to predict optimal courier delivery schedules for connected vehicles in a road graph.

4.1 Introduction

Recently, the evolution of Intelligent Transportation Systems (ITS) has led to the availability of large traffic databases, which make it possible to automate vehicular traffic and efficiently navigate vehicles in road networks [72]. It enables the prediction of road congestion and the accurate determination of efficient vehicle routes, thus significantly improving the effectiveness of road use and reducing driving delays. Modern cars can perform fundamental driving route checks by independently selecting paths, avoiding obstacles, recognizing the surroundings, and following determined paths in cooperation with other vehicles for their successful driving [73, 74]. The efficient route planning task of navigation is the most fundamental requirement for successful intelligent driving.

Vehicle navigation systems currently utilize real-time road network information to offer vehicles the quickest paths to their destination [75]. Using Machine Learning (ML), traffic forecasting techniques capitalize on past driving information to identify patterns and predict outcomes in driving scenarios [76, 77, 69]. Traffic predictability enables ITS to optimize driving flow, control congestion, improve driving safety, provide better driving routes, and accurately estimate arrivals. However, traffic prediction is revealed to be a complex task. It relies on road topology and high-quality data from sensors and the cloud. It is also influenced by driving patterns such as weather and time (i.e., rush hour). To reach a specific driving service on time, such as parcel delivery, real-time prediction is needed to meet customers' satisfaction. This route prediction enables the definition of traffic flow, regulation of traffic speed, and response to traffic demands.

Recent research on efficient road traffic forecasting can be categorized into statistical methods [75] and machine learning methods [77]. The statistical method includes the time series [75] and regression models [76]. A typical time series of traffic data is proposed in [75], which can navigate vehicles in congested cities. Such a system lacks the correlation of traffic data among nodes, which can significantly impact traffic predictions. The regression method comes as a solution to these issues by estimating the relationship among several traffic variables. A regression approach that predicts traffic flow using machine learning algorithms was studied in [76]. An ensemble learning that combines multiple machine learning models improves the traffic prediction of traffic flow [77]. The statistical methods rely on historical traffic data from sensors (e.g., loop detectors, cameras, etc) to analyze statistical properties for traffic predictions. These methods have a limited view of road features due to their dependence on temporal patterns. Advanced methods such as machine-learning techniques significantly improve the traffic forecasting due to their capacity to explore multi-source data, and through feature engineering, capture complex relationships that impact road traffic [69]. Given the graph nature of road traffic, Graph Neural Networks (GNNs) are a well-suited machine learning approach for traffic prediction [69].

GNNs are predictive graph learning models that identify the representation patterns and the interconnection relationships of graph-structured data [70, 71]. Their capability to adapt to network structure and traffic evolution over time through spatial-temporal dependencies enables them to provide accurate traffic condition predictions. GNNs have been explored in the studies of delivery schemes, leading to the prediction of parcels' Estimated Time of Arrival (ETA) [70]. This work proposed a design of an inductive graph transformer that combines retailers and road information for ETA determination. Toward an accurate ETA estimation task, a heterogeneous information graph that jointly considers vehicle trajectory and temporal traffic information was proposed [71]. Despite these efforts, there is still a long way to be undertaken to meet the ETA satisfaction on the road network condition-dependent service, such as parcel delivery. Road congestion resulting from unprecedented events such as accidents can highly affect the cargo driver's route, thus affecting the entire service delivery.

In this chapter, we propose a distributed road network traffic that combines an ML model and a statistical model, which is a traffic distribution algorithm proposed by the Self-Adaptive Interactive Navigation Tool for Cloud-Based Autonomous Traffic Optimization (SAINT) [66]. We propose a navigation architecture in a road network depicted in Fig. 4.1. A data exchange between the network infrastructure (i.e., vehicular cloud) and vehicles allows the specific traffic data to be obtained for navigation decisions. We aim to determine a cost-efficient and fast route for delivery service targeting multiple destinations while minimizing the delivery cost in terms of ETA for the last-mile parcel delivery. We define the prediction horizon in timely intervals to respond smoothly to application-specific constraints and requirements such as deadline time and priorities. Through graph representations, the traffic states are defined. We define this representation in two categories: spatial correlation representation and temporal correlation. This last helps to communicate the relationships and interactions between nodes to enrich route planning and estimate time of arrival (ETA) prediction.

In specific terms, our key contributions to this work are as follows:

- *A cloud-empowered vehicular navigation architecture:* We propose a design that, through communication between the vehicular cloud and vehicles, allows the specific traffic data to be obtained for navigation decisions (see Section 4.3).

- *A design of a self-adaptive prediction mechanism:* We construct a graph attention-aware mechanism for application-specific traffic forecasting that explores the temporal and spatial traffic flow in road graph nodes to calculate a congestion-free and cost-efficient vehicle trajectory (see Section 4.4).

- *A simulation-based graph convolutional network (GCN):* We developed a road map training that benefits the realistic application-specific traffic forecasting in the Simulation of Urban Mobility (SUMO)[78] (see Section 4.4).
- *The design validation and performance evaluation:* We evaluate the ML-SAINT simulation for the intelligent travel path determination of cargo vehicles towards various assigned parcel destinations. The results show that ML-SAINT outperforms the compared schemes (legacy SAINT) [66] and multiple linear regression (LRSAINT) [79] as shown in the performance section (see Section 4.5).

The remainder of this chapter is structured as described below. Section 4.2 reviews the related work relevant to traffic forecasting algorithms. Section 4.3 describes the problem this chapter addresses. Section 4.4 describes the architecture of our proposed navigation system model, and Section 4.5 evaluates its performance. Finally, Section 4.6 wraps up this chapter and discusses future work.

4.2 Related Work

This section summarizes the recent advancements in road network path-finding methods and their applications for specific road data processing.

The legacy pathfinding mechanisms are based on statistics[80, 81, 82], regression [83, 84, 85, 86], or hybrid by combining statistics and regression [87] methods. The common statistical methods used in Pathfinding include the traveling salesman problem (TSP) [80], A* algorithm [81], and Branch and Bound (BnB) [82] optimization problems. A comparison of the route searching methods that use the greedy, Dijkstra, and BnB shortest path

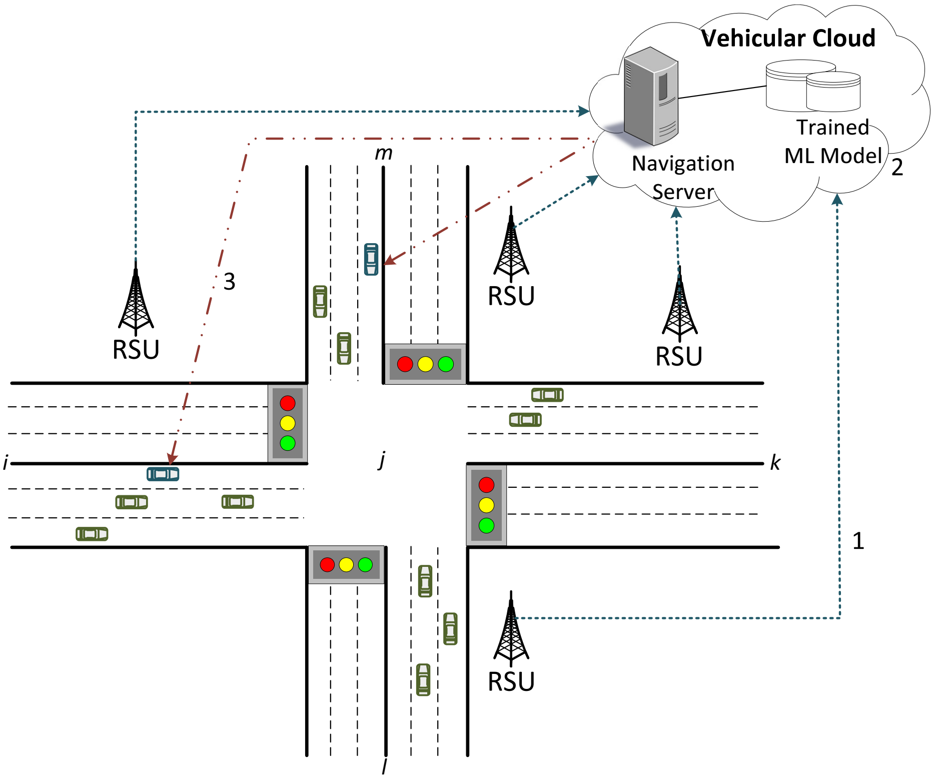


Figure 4.1: A system architecture of adaptive navigation in road networks.

algorithms was explored in [88]. An A* algorithm is a goal-directed search method that heuristically guides to a targeted destination without exploring all directions [81]. The Branch and Bound (BnB) algorithm retains the lowest value when assessing cases and contrasts it with the presently computed value [82]. The statistical methods cannot reflect the external factors influencing their predictions, thus, regression mechanisms were studied to address this issue.

Regression mechanisms include methods such as Polynomial regression [83], machine learning (ML) [84], deep learning [85], and graph neural network [86]. A polynomial regression model is a machine learning method that models the relationship between independent and dependent variables using a polynomial function [83]. It describes the deviation from a linear pattern, introducing curves, arcs, or bends in the movement or form. A network-wide approach that combines machine learning with fundamental statistical time series proposed in [87] improved the accuracy of road traffic prediction by significantly reducing the MSE (mean squared error) of predictions. A common deep learning mechanism gaining popularity in traffic forecasting is a graph neural network (GNN) [86]. GNN can capture spatial and temporal dependencies, thus improving the traffic prediction performance. Several studies explored the spatial-temporal road traffic forecasting for traffic prediction [89]

The recently proposed GNN mechanisms for road traffic forecasting include the attention-based graph convolutional networks [90] and multitask representation learning [91]. The attention-based [90] GNN exploits the spatial-temporal convolutions to capture the dependencies in the graph network. It results in spatial and temporal correlations in graph nodes, thus enhancing the accuracy of vehicle driving decisions. The temporal graph convolution demonstrated the capacity to capture spatial-temporal correlation useful for long-term traffic prediction tasks [92, 93]. A multitask representation learning [91] utilizes the path information to learn meaningful information about road graphs, thus estimating a suitable driving path that responds to network structure and traffic demands. These methods are also used to build systems that respond to the application-specific demands in road traffic [94].

GNN enhances the predictability of parcel delivery delays in road networks. The inductive graph transformers [94] were introduced to estimate the origin-destination (OD) travel estimation of parcel delivery. The multi-task representation learning model for arrival time estimation by preserving trip information to deduce the OD time estimates [91]. These methods have significantly improved the predictability of parcel arrival time, thus boosting customer satisfaction. The study of road parcel delivery applications that leverage trip duration and space is hard to realize on real roads, given the complexity of driving patterns. The complexity of city roadway machine learning is solved by integrating the

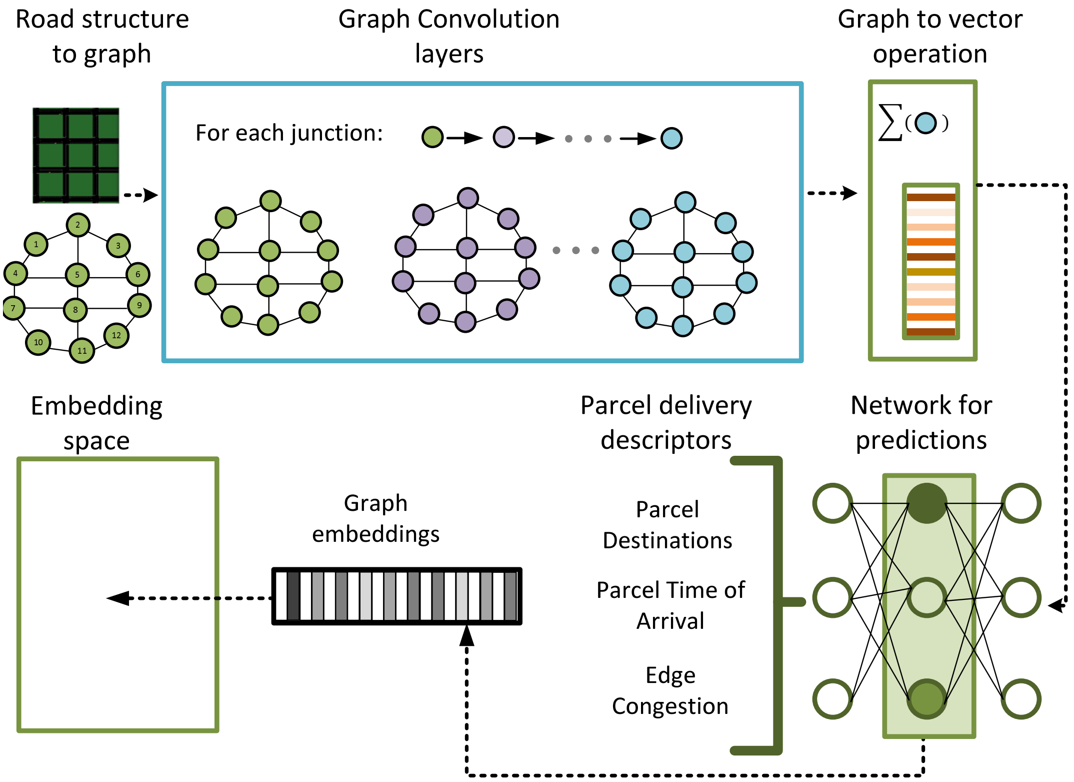


Figure 4.2: Road network graph embedding generation for graph feature representation learning.

routing and navigation through the SUMO simulator, rendering an integrated intelligent navigation research for dynamic urban network possible [85].

The state-of-the-art schemes described above have explored OD and ETA in terms of geographical shortest paths. However, a model that predicts road congestion to estimate its effect on overall navigation performance is required for efficient navigation. In this work, we design an appropriate, scalable graph neural network-based navigation scheme to optimize road parcel delivery services by capitalizing on congestion metrics to define the fastest routes. The next section discusses the problem targeted by this work.

Table 4.1: Main Notations

Notations	**Description**
$\mathcal{V}$	A set of vertices.
v_i	the ith vertex in a set of vertices $\mathcal{V}$.
v_i^F	The feature vector of vertex v_i.
$\mathcal{E}$	A set of edges.
e_{ij}	The edge between the ith vertex and the jth vertex in a set of edges $\mathcal{E}$.
e_{ij}^F	The feature vector of edge e_{ij}.
$\mathcal{N}_{vi}$	The set of vertices which are direct neighbors of vertex v_i.
$\mathcal{A}$	The adjacency matrix.
$\mathcal{D}$	The degree matrix.

4.3 Problem Formulation

In this section, we describe the architecture, the problem, and the assumptions of a Graph Convolution Network (GCN)-enhanced self-adaptive navigation system for efficient parcel delivery in a road network. Our goal in this work is to develop a traffic forecasting enriched parcel delivery system that efficiently maximizes the predictability of service delivery.

4.3.1 Road Network Description

We consider a road graph structure where the delivery system operates to be a graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{V}$ is a set of N road network intersections, $\mathcal{E}$ is a set of road segments with their cost c indicating the edge travel delay. $\mathcal{A} \in \mathbb{R}^{N\times N}$ represents an adjacency matrix in the road network. The node adjacency is expressed as:

$$a_{i,j} = \begin{cases} 1, & \text{if node } i \text{ is adjacent to node } j, \quad (4.1a) \\ 0, & \text{otherwise.} \quad (4.1b) \end{cases}$$

Note that in a road network, i nonadjacency to j does not necessarily mean that j is not adjacent to i. The traffic sampling frequency at each node is similar, thus allowing the system to generate the same vector length of node features. We consider a route of cargo n_i transporting parcels to be a subgraph $\mathcal{G}_{sbi}$ such that $\mathcal{G}_{sbi} \subset \mathcal{G}$. In this work, we design a mechanism that defines cargo routes to transport depot parcels while maximizing profits by adjusting to the ever-changing delivery demands and complying with customer satisfaction through enhanced time-precise service delivery.

The main components of the delivery system in a road network are described in Fig. 4.1. Its topological information comprises of intersections which represent the graph nodes $v_i \in \mathcal{V}$ and roads representing edges $e_{ij} \in \mathcal{E}$ linking intersections v_i and $v_j \in \mathcal{N}_{vi}$. Note, $\mathcal{N}_{vi}$ is a set of nodes directly adjacent to node v_i, and $\mathcal{N}_{vi} \subset \mathcal{V}$. Step 1 of Fig. 4.1 shows how statistical data observed at the vehicle level and roadside unit (RSU) level are shared via communication to the vehicular cloud. In step 2, using a trained machine learning model (i.e., STGCN and LR), a delay-constrained shortest path is calculated, and it is shared and followed by the ego vehicle in step 3. This enables vehicles to be equipped with knowledge-based traffic information from the vehicular cloud, which informs their navigation decisions. This information is transmitted via vehicle-to-infrastructure (V2I), which is transmitted by infrastructure such as Road Side Units (RSUs) to vehicles, or Vehicle-to-Vehicle (V2V), which allows vehicles to communicate with other vehicles within their communication range.

4.3.2 Assumptions

We formulate ML-SAINT under the following assumptions:

- A cargo vehicle can communicate with the vehicular cloud through cellular communication such as 5G [95], thus enabling it to exchange information with the cloud

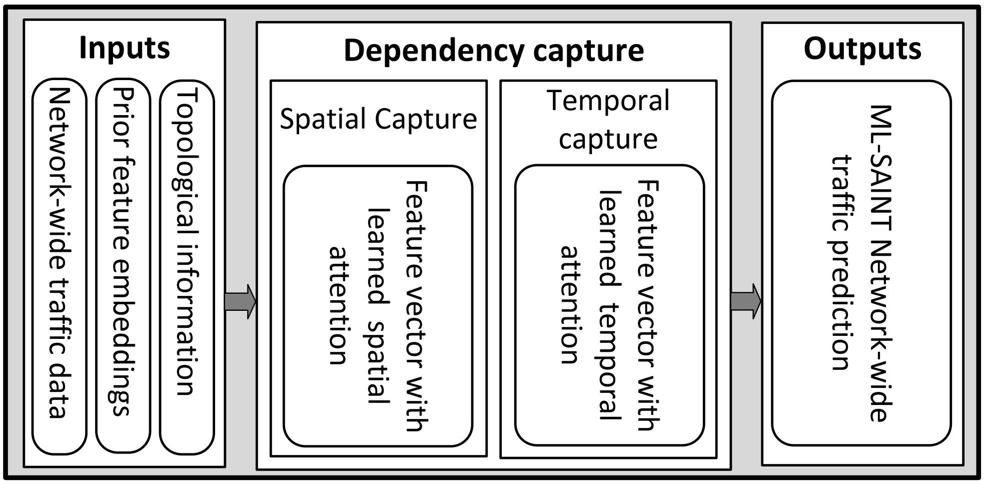

Inputs
Network-wide traffic data
Prior feature embeddings
Topological information
Dependency capture
Spatial Capture
Feature vector with learned spatial attention
Temporal capture
Feature vector with learned temporal attention
Outputs
ML-SAINT Network-wide traffic prediction

Figure 4.3: A spatio-temporal self-adaptive traffic prediction framework.

server.

- Cloud mobility updates are accepted by the vehicles' navigation whenever necessary.
- The communication delay between vehicular cloud and driving vehicles is too small compared to the trajectory delay, thus considered negligible in the ML-SAINT computations [96].

4.3.3 Driving State Collection

The traffic information observed at each intersection $v_i \in \mathcal{V}$ is represented in feature matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times \mathcal{F}_v}$. Note that feature $\mathcal{F}_v$ denotes the number of features explored at each direct neighboring intersection to intersection v_i. Features are the observable node states that include the average speed, traffic flow rate (λ_i), and occupancy. At a given time t, the graph features are extracted following this equation:

$$\mathcal{V}_t^{\mathcal{F}} = \left[v_1^{F_t}, v_2^{F_t}, ..., v_N^{F_t} \right] \tag{4.2}$$

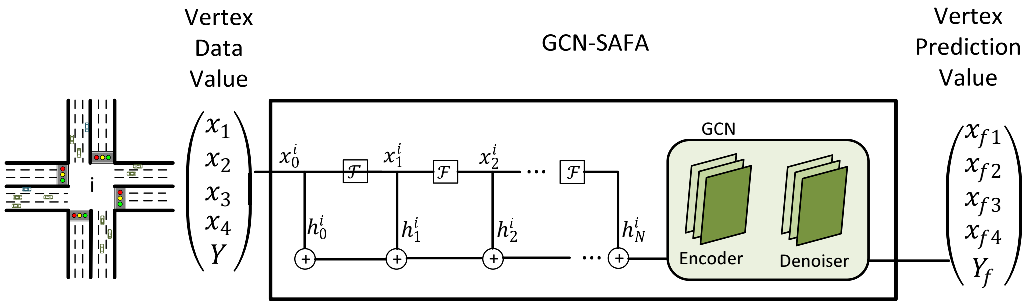


Figure 4.4: A node-level self-adaptive feature aggregation (SAFA) process for graph convolution networks.

where $v_i^{F_t}$ is a series of the collected information during a period t at a particular graph node v_i. A node v_i historical data observations is represented as:

$$v_i^{F_t} = \left[x_t^i, x_{t-1}^i, x_{t-2}^i, ..., x_{t0}^i\right], \tag{4.3}$$

where x_t^i is the feature observation at node v_i at a instant of time t. Spatially, the entire graph features recorded at an instant time t in N nodes are represented as:

$$\mathcal{G}^{F_t} = \left[x_t^1, x_t^2, ..., x_t^N\right], \tag{4.4}$$

where x_t^i incorporates the traffic data such as pending vehicles, traffic flow rate, and average speed observed at node v_i at a specific observation time t.

Graphical embeddings are computed to more appropriately identify the structural and relational representations of the road traffic data for each node v_i. Fig. 4.2 depicts the graph features mapping into learnable embeddings through multiple layers of neural network training. The embeddings allow the exploration of hidden patterns from other graph nodes that influence the traffic at a particular node.

A delivery path prediction takes advantage of the road graph topological information, prior embeddings, and network-wide traffic observation to learn both temporal and spatial dependencies, thus enabling the vehicle's self-adaptation navigation in a road network as depicted by Fig. 4.3. During the training process, features are weighted to extract the relationships to be adapted for the efficiency of a task at hand.

4.3.4 Delivery Problem Definition

Let a cargo route be a subgraph $\mathcal{G}_{sbi} \subset \mathcal{G}$ made of a set of nodes $\mathcal{V}_{sbi}$ that build a sequence of roads leading to parcels' destinations. A cargo departs from the depot carrying a set of parcels $\mathcal{P}_{sbi}$, where each parcel $p(o, d, \tau)$ is defined by its origin o, destination d, and deadline τ. A delivery problem is to predict an efficient cargo trip using the historical traffic states of the road network.

Problem 4.3.1. ***A parcel delivery problem:*** *This consists of determining a path $\mathcal{P}_{n_i} = \{p^i_1, p^i_2, ..., p^i_t\}$ for a cargo-carrying a set $\mathcal{P}$ of parcels such as that it satisfy the parcel delivery performance in terms of deadline satisfaction and via an optimized cargo trajectory by maximizing the overall trajectory delay.*

We define an optimal cargo traveling path as an objective-centric function that models a navigation solution by reflecting the historical road network state to define future states as:

$$\mathcal{O}^k_n = f_o\Big(\mathcal{G}^{F_t}, \mathcal{D}^n_t\Big), \tag{4.5}$$

where f_o is the objective function, $\mathcal{G}^{F_t}$ is the road features, and $\mathcal{D}^n_t$ represents parcel features.

4.3.5 Delivery Problem NP-Hardness

The shortest vehicle path determination in a road network is revealed to be complex. Considering that it requires exploring all nodes' possibilities, it resembles the Traveling Salesman Problem (TSP)[97]. TSP gets an efficient route while visiting all the assigned destinations [80]. It results in an infinity of possible routes, which cannot be exhaustively calculated within polynomial time. Thus, TSP is an NP-hard problem, and the path is found through heuristic approximations. As an NP-hard problem[98], determining all potential combinations is theoretically possible but practically unachievable within a reasonable timeframe. Thus, we can conclude that the definition of a cargo path is also an NP-hard problem, which requires solving through approximations. This study introduces a combination of Machine learning and SAINT methods to compute the shortest vehicle that minimizes the overall road congestion. In Section 4.4, we describe the framework of a proposed graph convolution-driven self-adaptive navigation tool for road traffic delivery efficiency.

4.4 ML-SAINT System Design

This section describes the system design of a Graph Convolution Network (GCN)-assisted self-adaptive navigation tool for the efficiency of parcel delivery service applications. It represents a prediction mechanism to explore the spatial-temporal traffic flow of graph nodes and a congestion-free cargo vehicle's trajectory decision strategy.

4.4.1 System Architecture

The system design we propose to optimize the vehicular navigation for parcel delivery optimization is depicted in Fig. 4.4. Road information such as pending vehicles, waiting time, last step vehicle number, and previous step mean speed is kept as each node's properties. It is passed to the model to extract learnable embeddings for navigation prediction in the next step. This process aims to train and adapt the road graph mobility information by extracting relationships among nodes that enhance the anticipation of navigation behavior affecting the vehicle's trajectory, adapting it for its route efficiency.

It is a feedback model with Graph Convolutional Networks (GCNs) for vehicle route planning in SUMO. It uses a feedback loop to enhance route planning by capturing spatial dependencies and predicting traffic conditions within the road map. When provided with the graph's environment, vehicle dynamics, and traffic flow data representation, GCNs predict the future traffic conditions that the route planning algorithm will use to calculate optimal and cost-efficient vehicle routes. The system compares the desired traffic conditions to actual conditions and adjusts route planning parameters and GCN predictions. This design exploits the self-adaptive interactive navigation tool (SAINT), a mechanism proposed in [66] which uses the interaction between vehicles and cloud to globally optimize the vehicle navigation route.

4.4.2 SAINT Navigation Description

This section presents the SAINT [66] navigation, a global traffic optimization approach through its tailored Congestion Contribution model, and its use for the shortest path algorithm.

Vehicle Trajectory Congestion Contribution

A vehicle's trajectory from origin to destination comprises a sequence of road segments. For an ego vehicle n_i which traverses through p road segments to its destination, its trajectory delay, also called the end-to-end delay (E2E) D_n^i is

$$D_n^i = \sum_{i=1}^{k} d_{(v_i, v_{i+1})}, \tag{4.6}$$

where $d_{(v_i, v_{i+1})}$ is the link delay of a road $e_{i,i+1}$ from intersection i to intersection $i+1$.

Shortest Path Definition in SAINT

Knowing that by simply computing the geographical shortest path for navigation, the congestion will propagate in the road network over time, thus causing the currently non-congested roads to be congested shortly. SAINT modeled a congestion contribution c_i as a congestion metric of the current edge e_i to the overall road trajectory as follows:

$$c_i = 1 - \frac{D_i}{D}, \tag{4.7}$$

where D is the end-to-end travel delay of a vehicle travel path and D_i is the current sub-route delay. A sub-route is a route towards the current parcel destination. Fig. 4.5 shows the SAINT's step functions of two vehicles whose routes are mapped. SAINT defines a Delay Constrained Shortest Path (DSP) algorithm, which enables the navigator to define a vehicle's shortest path with the smallest congestion increase to reach its intended destination. Through α-percent delay increase, it defines a time-wise path with the minimum congestion contribution. In this design, we model the cargo traffic dynamics prediction, which, knowing a congestion prediction, defines a commendable trajectory of a cargo vehicle.

Given a detour factor α and a set $X = \{x_1, x_2, ..., x_n\}$ of road features, the delay-constrained path for a cargo satisfies:

$$\begin{aligned} &D \leq \alpha d, \\ &\text{where } d = \alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_n x_n. \end{aligned} \tag{4.8}$$

4.4.3 Road Network Traffic Collection

The road network traffic sampling process consists of message passing and data exploration steps. In message passing, using an aggregation function, a local sampling of road

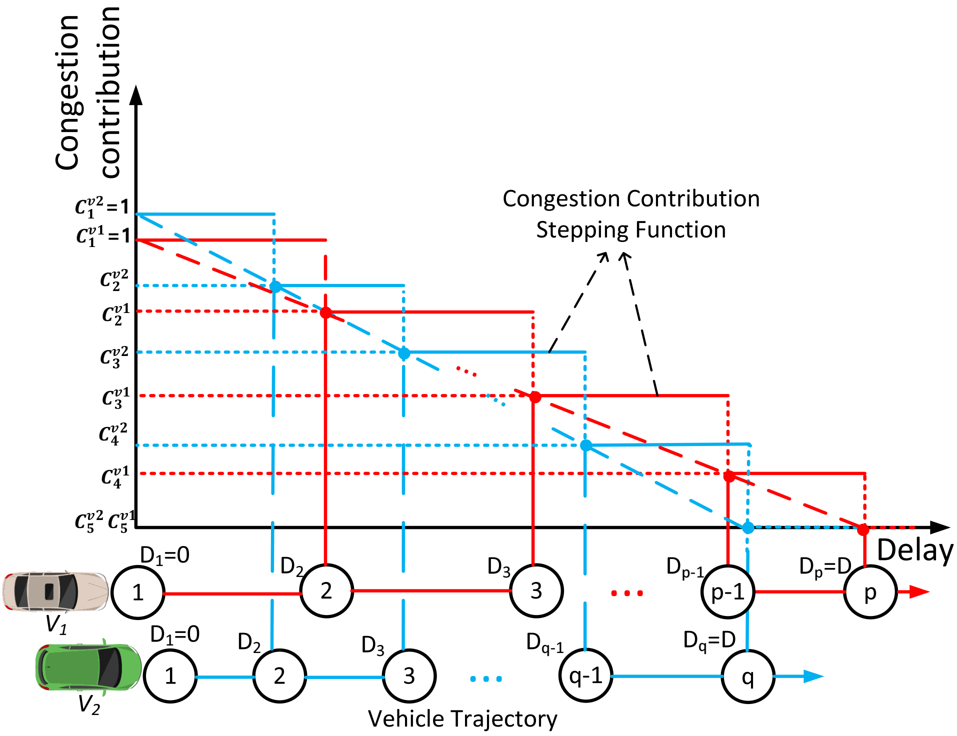


Figure 4.5: Edge congestion contribution with the step function.

information at the node level is performed in consideration with its neighbors N and updates node self-information via an update function. In data exploration, we calculate a feature vector based on the hidden layer state and predict the navigation task, such as path planning.

4.4.4 Self-Adaptive Feature Aggregation

This design proposed a graph convolution network (GCN) with self-adaptive feature aggregation (SAFA). It uses the learnable embeddings encoded from the collected node features $\mathcal{V}_t^{\mathcal{F}}$ to capture the spatial-temporal dependency in road traffic. It is a traffic prediction framework that captures road network-wide graph traffic to define an efficient trajectory of an ego vehicle (i.e., a cargo vehicle). This system models the capture of spatial dependence in a road network through graph convolution layers. The goal is to build an aggregation module that combines each node's gathered features to provide a graph with learnable embeddings for optimized task-enhanced predictions.

A Polynomial, such as the Chebyshev polynomial, is used to learn the graph embedding in GCN [99]. For a graph with an adjacency matrix $\mathcal{A} \in \mathbb{R}^{N \times N}$ and its degree matrix $\mathcal{D} \in \mathbb{R}^{N \times N}$, its initial Laplacian is expressed as $\mathcal{L} = \mathcal{D} - \mathcal{A}$. Its normalized Laplacian matrix is a symmetric matrix expressed as $\mathcal{L} = \mathcal{I} - \mathcal{D}^{\frac{-1}{2}} \mathcal{A}^{\frac{-1}{2}}$. A K-order Chebyshev polynomial approximation of a spectral filter is expressed as follows:

$$T_k(L) = \sum_{k=0}^{k} \theta_k T_k(\tilde{\mathcal{L}}), \tag{4.9}$$

where θ_k is the k-order learnable weight matrix, $\tilde{\mathcal{L}} = \frac{2\mathcal{L}}{\lambda_{max}} - I$ is the k-order Chebyshev polynomial of the scaled Laplacian matrix, λ_{max} being the biggest eigenvalue of $\mathcal{L}$.

A GCN node's embedding update rule for GNN from layer l is expressed as:

$$h_i^{l+1} = \sigma(\sum_{k=0}^{k} \theta_k^{(l)} T_k(\tilde{\mathcal{L}}) h_i^l, \tag{4.10}$$

Where h_i^l represents the node v_i embedding at the l obtained via an activation function σ. The encoder iteratively processes the node input features and produces the conceptualized node representation. On the other hand, the decoder uses this node representation to process the next navigation token. This process is calculated as follows:

$$g(\mathcal{F}, \mathcal{A}) = \sigma(\hat{\mathcal{A}} \mathcal{F} \mathcal{W}_t \mathcal{W}_{t-1}), \tag{4.11}$$

where $\hat{\mathcal{A}}$ is the estimate normalized adjacent matrix, $\mathcal{F}$ is the graph feature representation matrix, $\mathcal{W}_t$ is the learnable weight matrix of the current layer, and $\mathcal{W}_{t-1}$ is the previous layer learnable weight matrix.

The Algorithm 4 shows how these features are extracted and processed in the GCN-SAFA. It samples all nodes and assigns the feature information initialization. Line $2 - 10$

Algorithm 4 Graph Feature Extraction Algorithm

```
1:  function EXTRACT_GRAPH_FEATURE($\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{A})$)    ▷ For a graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{A})$, compute the node embedding using an interactive model training with Chebyshev aggregation.
2:      for each $v_i \in \mathcal{V}$ do
3:          $h_0^i = x_0^i, \forall v_i \in V$    ▷ Set initial embedding as the node's features vector.
4:          for each $l \in \mathcal{L}$ do    ▷ Each $l$ represents a message passing neural network layer.
5:              for each $v_j \in \mathcal{N}_{vj}$ do Iterate through a set of node $v_j$'s neighbor nodes $\mathcal{N}_{vj}$.
6:                  $h_l^j = AGGREGATE\{(x_{l-1}^i)\}, \forall j \leq L$    ▷ Aggregation of node features to adapt the navigation relationship of neighbor nodes with Chebyshev polynomial filter.
7:                  $h_l^j = \sigma(W^l.CONCAT(h_{l-1}^j, h_{l-1}^j)), \forall j \leq L$
8:              end for
9:          end for
10:     end for
11: end function
```

combines the node sampling stage and the aggregation stage of the node representation calculation stage. Line 6 aggregates the current node features with the neighboring nodes $v_j \in \mathcal{N}_{vi}$. Using features of the Laplacian matrix derived from the adjacency matrix, it captures the graph's connectivity, which is used for spectral graph theory-based GNNs. Line 7 concatenates the aggregated neighbor features vector and is fed to the activation function σ to obtain the actual current node representation, which is necessary for road traffic prediction. ML-SAINT design handles the road traffic through a sequence of functional steps described in Fig. 4.6.

4.4.5 Modeling Road Traffic

ML-SAINT design models the road traffic through a sequence of functional steps. Fig. 4.6 describes the steps involved in traffic forecasting. These steps are described below:

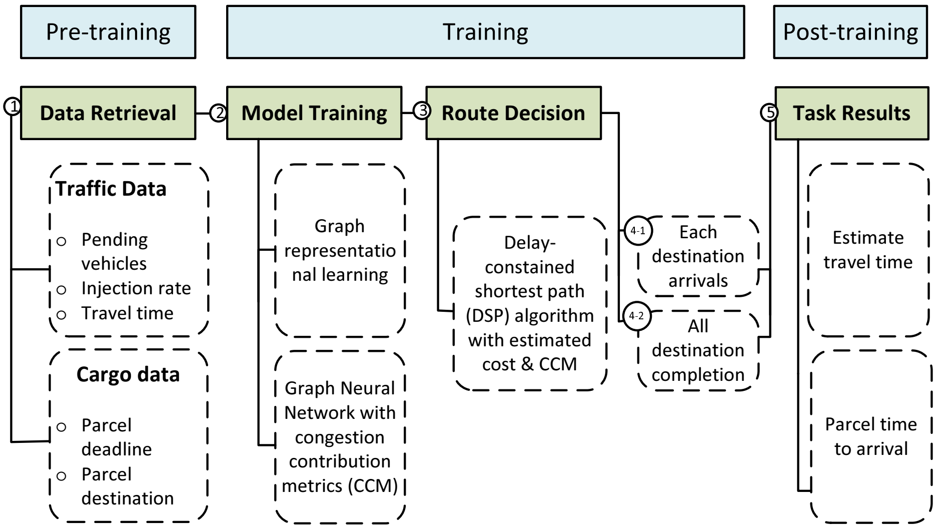


Figure 4.6: Road graph data processing description.

Step 1: Feature extraction step: It is a data collection of descriptive topological and spatial-temporal graph information.

Step 2: Model training: It consists of processing the graph features to produce a learnable embedding, including the graph representation and the relationships.

Step 3: Graph Convolution with optimization: Adam optimizer is a widely used machine learning optimization. Using the loss function, the optimizer keeps trying to minimize the loss between the predicted value and the real value.

Step 4: Vector parameter update: it involves computing a weighted average of past and squared gradients, adjusting the learning rate (η) accordingly, and updating the model parameters (θ). Following the training phase, the model's outcomes are preserved for subsequent learning iterations.

Step 5: The system selects an optimal choice at each parcel destination and constitutes an ego vehicle's overall navigation route. The ML uses the trained model and the available edge data to predict the travel time.

The Mean Squared Error (MSE) loss function is used to evaluate the proposed model's performance cost as:

$$c_{gcn} = \frac{1}{N}\sum_{i=1}^{N} y_i - \tilde{y}_i, \tag{4.12}$$

where y_i is the desired route information and $\tilde{y}_i$ is the actual GCN route prediction. MSE quantifies the average squared difference between predicted values and actual outcomes, evaluating the model's precision. The system minimizes a loss value, which measures the difference between the predicted and actual traffic states. Following the training phase, the model's outcomes are preserved for subsequent learning iterations.

4.4.6 An Application-Specific Path Determination Module

Our model designs a graph neural network convolution to enhance the performance of applications, whose effectiveness highly depends on graph information representations. We designed and evaluated the performance of parcel delivery by calculating its path to satisfy both customers and benefits.

Travel Time Estimation

The crucial step is to estimate the travel time $T_{ij}(t)$ on each edge e_{ij} at a given departure time t. We can model this relationship based on the predicted traffic conditions. Using Predicted Flow and Capacity (e.g., BPR-like function): If the GNN predicts flow $f_{ij}(t)$ and we know the capacity cij and free-flow speed $s_{free,ij}$, we can estimate the travel time as:

$$T_{ij}(t) = \frac{l_{ij}}{s_{free,ij}}\left[1 - \alpha\left(\frac{f_{ij}(t)}{c_{ij}}\right)^{\beta}\right], \tag{4.13}$$

where α and β are the parameters that define the congestion sensitivity.

Shortest Path Prediction

Assume a cargo c_v is assigned to deliver a number n_p of parcels, where each parcel is defined by its destination p_d and deadline p_{ft}. The goal is to find a path $P_{cargo} = \{v_1, v_2, ..., v_k\}$ in the graphs $\mathcal{G}$, where the total travel delay from v_1 to v_k is minimized.

The mathematical computation of the future cargo path mobility is defined as:

$$h_t^n = h_{t-1}^n \bigcup_{\forall nj:n \to nl} f_t(h_{t-1}^n, k, h_t^{nj}), \tag{4.14}$$

which represents the permutation operation for the mobility information combination operation.

$$C_{goal} = \mathbf{g}(\mathcal{G}, \mathcal{C}) = \{v_1, v_2, ..., v_{nd}\}. \tag{4.15}$$

We consider a cargo that drives starting from node v_i at time t_i, whose traversing delay from v_i to v_{i+1} is $t_{i+j} = t_i + T_{v_i,v_{i+1}}(t_i)$. The total predicted travel time, which is a route cost metric, is the sum of travel times of the path P.

$$T(P) = \sum_{i=1}^{k-1} T_{v_i,v_{i+1}}(t_i). \tag{4.16}$$

Definition 4.4.1. ***Cargo navigation path**: It is a subgraph $sub_{\mathcal{G}} \subset \mathcal{G}$ such that a cargo vehicle n_{ego} will deliver n_p by satisfying the minimal driving time according to* (4.17).

The objective path P^* that minimizes the total travel time is

$$P^* = \arg\min_P T(P). \tag{4.17}$$

The algorithm 5 describes the steps involved in computing a cargo trajectory path that meets the couriers' estimated time of arrival (ETA). The purpose is to shorten each road link delay, therefore shortening the overall End-to-End (E2E) path delay.

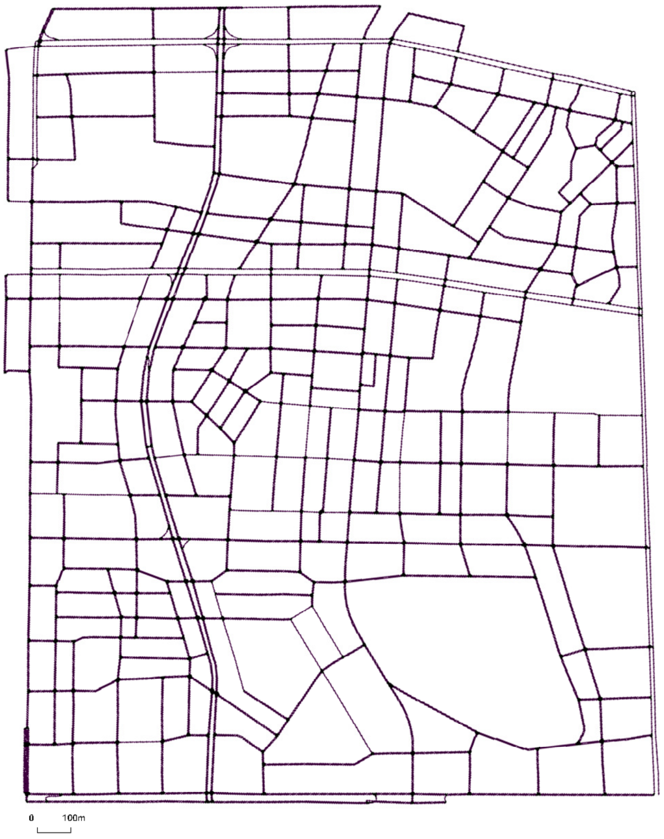


Figure 4.7: Road network in the Gangnam area of Seoul for simulation.

Our proposed self-adaptive navigation approach that determines a cargo navigation path is presented in Algorithm 5. Given a set $\mathcal{C}$ of couriers, destined to n_p number of destinations; compute a cargo trajectory $sub_{\mathcal{G}} \subset \mathcal{G}$ made of a sequence of graph edges such that each courier destination is reachable. Lines 2-9 define the targeted destination nodes and prepare them to be sorted based on their aggregated features. Lines 10-21 assess the trajectory of a cargo according to the node's reachability and congestion constraints. After getting the node's features in line 12, line 17 calculates a proper route through a graph convolution function $g(.)$.

The vehicle navigation control adjusts the cargo route planning parameters and GCN predictions. It adjusts the link weights (i.e., road congestion) in the cost function and the

Algorithm 5 An Enhanced Cargo Path Decision Algorithm

```
function COMPUTER_CARGO_PATH($\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{A}), \mathcal{C}$)    ▷ Knowing a graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{A})$, compute a cargo trajectory that satisfies a set $\mathcal{C}$ of $n_p$ couriers to be distributed along the roadway destinations.
    $\mathcal{D} \leftarrow \emptyset$
    $nd \leftarrow 0$
    for each $p_i \in \mathcal{C}$ do
        while $p_{di} \notin D$ do
            $\mathcal{D}[nd] \leftarrow p_{di}$    ▷ Assign destination $p_{di}$ the destination set $\mathcal{D}$
            $nd \leftarrow nd + 1$
        end while
    end for
    for $i = 1$ to $epochs$ do
        for each $v_i \in \mathcal{D}$ do
            $v_i = h_0^i, \forall v_i \in V$    ▷ Get node's features vector.
            for each $v_i$ to $v_0$ do    ▷ Learning appropriate road traffic through back propagation.
                for  $k = 1$ to $N$ do
                    $p_l^j = (p_{l-1}^i), \forall j \leq L$
                    $h_l^j = concat(p_{l-1}^i, h_{l-1}^i)$
                    $n_k = GCN(h_l^j)$
                    $Err = Get\_MSE(n_k)$
                end for
            end for
        end for
    end for
end function
```

GCN prediction horizon (Δt). The GCN model is periodically retrained with new traffic data to improve prediction accuracy. The error term can be used as a signal to fine-tune the GCN model. For instance, if the GCN consistently underpredicts congestion in a specific area, the error can be used to adjust the GCN's weights to increase its sensitivity to congestion in that area.

Navigation Route Assignment

Several vehicles are distributed in the road network, causing congestion and driving delays. The CCM measures how vehicles on the road network add to the congestion of each road network edge. The CCM concept includes proportional data for every graph edge, providing insight into potential future congestion along specific routes. Utilizing cloud-based navigation makes it feasible to predict future traffic congestion on roads currently free of congestion, enabling the determination of the optimal driving routes.

The design the road traffic predictions considering a road network's spatial and temporal dependencies. The performance evaluation compares two methods: the multiple linear regression (MLR) model and the Graph Neural Network (GNN). The MLR model predicts a single output with various inputs, while the GNN enables several outputs, thus providing more informed predictions.

4.4.7 A Description of Road Graph Training Scenario in SUMO

Using the Gangnam map extracted from OpenStreetMap, we converted it using the SUMO netconvert tool and refined the network representation for simulation accuracy. This constructs a trainable graph, where intersections represent graph nodes and edges link adjacent nodes. We then generate realistic vehicle routes for the Gangnam area using a demand modeling approach that implements a demand model to simulate the commuting patterns of cargo cars and the general background traffic influencing traffic flow in Gangnam. Relevant data is periodically extracted as simulation output, including node features (i.e., average vehicle speed and vehicle count) and edge features (i.e., average vehicle speed, vehicle flow, occupancy, and travel time), which is a time series of feature vectors for each node learnt by the spatio-temporal graph convolution networks (STGCN).

We trained our model to be able to predict traffic, detect congestion, and analyze traffic patterns. In the training process, the GCN model takes a sequence of historical traffic observations (e.g., traffic speed, flow, or density) for all nodes in the Gangnam road network as input. It is represented as a tensor of dimensions of the number of time steps, the number of nodes, the number of features, and the adjacency matrix of the road network fed into the GCN. The model's output predicts the traffic conditions for a future time horizon (e.g., next 10 seconds) for all nodes. We use the Adam optimizer, which uses a Mean Squared Error (MSE) loss function for version validation of the hyperparameter tuning to avoid overfitting in GCN layer configurations. We test the trained GCN model's performance on different traffic conditions in the Gangnam area to assess how effectively it can predict vehicle traffic within the complex urban environment of Gangnam, Seoul, using a realistic SUMO simulation. In section 4.5, we describe our proposed model validation implementation and its performance evaluation.

4.5 Performance Evaluation

This section describes our system validation environment, the evaluation conditions, and the performance results of this proposed vehicular navigation for the efficiency of parcel delivery services.

4.5.1 Simulation Setup

To validate this proposed framework, we integrated the Simulation of Urban Mobility (SUMO)[78], a microscopic traffic simulator for road networks, with Keras deep learning library [100]. Using this simulator, we collect the traffic data obtained while running the vehicles within a target road network based on the SUMO and train it for vehicle navigation predictions. The feedback control logic and GCN inference in an external Python script communicate with SUMO via TraCI.

Table 4.2: Simulation Configuration

	Parameters	Description
1.	Number of vehicles (N)	Total Number of vehicles $N = 1500$. The testing vehicles range from $20 \sim 100$ of the vehicles driving in the road network, and the rest are the traffic background.
2.	Speed limit (v_{max})	The maximum vehicle speed for road segments. The default value is $80km/h$.
3.	Acceleration a	The vehicle acceleration in time. The default value is $3m/s^2$.
4.	Number of destinations n_d	The number of destinations a cargo needs to pass through in a delivery path. The default value is 6.
5.	Number of charging stations n_c	The number of available charging stations in a road network where vehicles can be recharged. The default value is 10.

In this evaluation, we demonstrate the vehicle traffic simulation in a real map of the Gangnam area of Seoul for simulation obtained from OpenStreetMap [101] as depicted in Fig. 4.7. It is a $2km$ long width and $2.5km$ long height road network made of 582 junctions and 1415 edges. We simulated the electric vehicles in SUMO with the simulation settings for the vehicle length to $3m$ and the minimum gap between vehicles to $1.5m$.

4.5.2 Evaluation Parameters and Metrics

The evaluation of our machine learning-assisted self-adaptive navigation tool considered the following evaluation settings:

- **Evaluation Parameters:** The parameters for evaluation include the impact of *(i)* the number of delivery vehicles, *(ii)* the number of destinations, *(iii)* the maximum speed, *(iv)* the maximum acceleration, and *(v)* number of background vehicles.

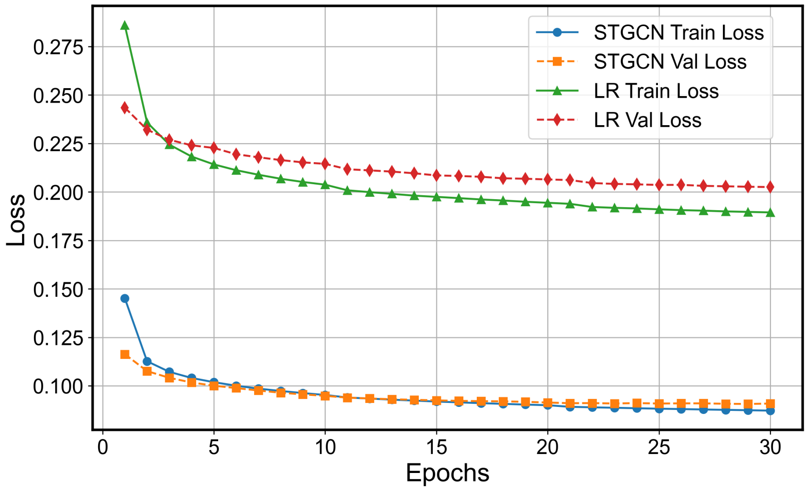


Figure 4.8: Training and validation loss of STGCN and LR Models

- **Performance Metrics:** The metrics for evaluation are the average link delay and the delivery vehicles' end-to-end delay (E2E).

- **Baselines:** We compare the ML-SAINT with the legacy SAINT [66] and the linear regression (LR-SAINT) [79] methods in this performance evaluation.

A confidence interval of 95% is used in this simulation to test the parcel delivery performance evaluation results.

4.5.3 Testing Results

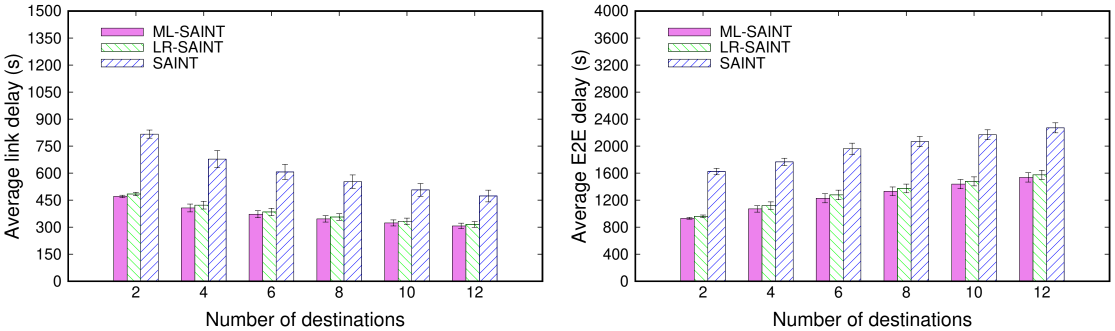
This section summarizes the testing results for our proposed model's performance. We first compare the spatial-temporal graph convolutional network (STGCN) training performance in contrast with the multivariate linear regression model and evaluate our model for various test cases for its performance evaluation.

Model Training Performance

We applied a mean square error (MSE) as a performance evaluation criterion of the training methods. For MSE evaluation, we compare the navigation task completion time against the epoch graph shown in Fig. 4.8. This figure shows that as the number of iterations increases, the data set training loss value continually decreases, tending to nearly zero. This demonstrates that the STCGN training can be correctly completed, improving navigation results. This validation figure shows that as the number of tested epochs increases, the STGCN training gives smaller errors for model training and validation than the LR-based training model.

Impact of Number of Delivery Destinations

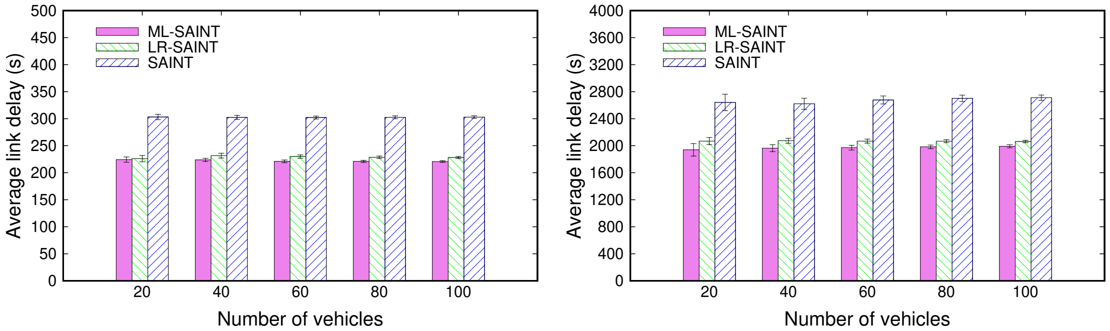
While evaluating this model, we simulated 1500 vehicles traversing the road network. Among these vehicles, 1400 background vehicles drive on the road in random routes, and 100 test vehicles (i.e., cargo vehicles) drive to their dedicated delivery destinations. To assess the impact of the number of delivery destinations, we measured the metrics by varying each test vehicle's destinations from 2 to 12 for all 100 test vehicles. In this test, vehicles drive with a maximum speed of 60 km/h and an acceleration of 3 m/s^2. We compared the two regression models, the multivariate linear regression (LRSAINT) and the machine learning approaches (ML-SAINT)-based on spatio-temporal graph convolution networks, to the statistically based traditional SAINT. The performance of the compared models is depicted in Fig 4.9. It compared the navigation performance of these models in terms of the average link delay depicted in Fig. 4.9a and the average end-to-end delay depicted in Fig. 4.9b. The results show that the regression models (i.e., LRSAINT and ML-SAINT) outperform the statistical-based SAINT regarding average link delay and E2E delay. ML-SAINT reduces up to 42.2% of the SAINT's navigation link delay



(a) Average link delay

(b) Average End to End delay

Figure 4.9: Impact of the number of delivery destinations on navigation.



(a) Average link delay.

(b) Average End to End delay.

Figure 4.10: Impact of the number of delivery vehicles on navigation.

and 42.6% of SAINT's navigation E2E delay. ML-SAINT gives the lowest values of average link delay and E2E, thus outperforming the baseline mechanism for the test vehicles' navigation route for all the compared numbers of destinations.

Impact of number of Delivery Vehicles

We tested the impact of the number of delivery vehicles by measuring the mean link delay and mean E2E delay for 20 to 100 delivery vehicles. It is to be noted that the total vehicles in the road network is 1500 in all the tested cases. When we evaluate the performance of 20 test vehicles, the background vehicles are 1480. When we test the performance for 40

test vehicles, the number of background vehicles is 1460, and so on. In this test, vehicles drive with a maximum speed of 60 km/h and an acceleration of 3 m/s^2. Fig. 4.10 depicts the performances of ML-SAINT on different numbers of vehicles against the compared baselines.

The results in Fig. 4.10 show the superiority of linear regression models (LRSAINT and ML-SAINT) over the statistical SAINT to provide the best routes for the cargo vehicles. Fig. 4.10a shows that on average, ML-SAINT provides the lowest value of link delay, thus ensuring that it provides the fastest traversal of a road compared to both LR-SAINT and SAINT. Fig. 4.10b shows that on average, ML-SAINT provides the fastest route from source to destination for all the tested test vehicles, compared with the linear regression-based SAINT and the statistics-based SAINT. ML-SAINT reduces up to 25.9% of SAINT's average link delay and up to 26.56% of SAINT's average E2E delay. Thus, ML-SAINT is the best navigation option for delivery route finding.

Impact of Speed

To assess the impact of vehicle speed on the navigation performance, we measured the metrics by varying the vehicle's speed limits from 20 km/h to 100 km/h. In this testing, we kept the vehicle's acceleration at 3 m/s^2 and tested 100 test vehicles (i.e., cargo vehicles) driving toward 6 delivery destinations. Fig. 4.11 shows the navigation performance results of ML-SAINT against LRSAINT and SAINT for the tested speeds.

The results in Fig. 4.11 show that ML-SAINT outperforms both the LRSAINT and the statistical SAINT for all the tested speeds, both in terms of link delays, Fig. 4.11a, and E2E delays, Fig. 4.11b. ML-SAINT reduces up to 22.2% of SAINT's average link delay and up to 19.64% of SAINT's average E2E delay. It also shows that the regression models (ML-SAINT and LRSAINT) perform much better than the statistical SAINT in terms of navigation performance.

Impact of the Acceleration

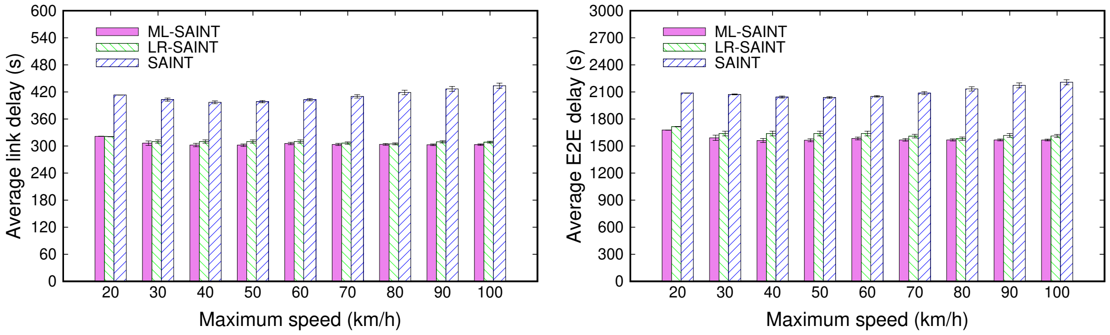
We also examined the impact of acceleration on the navigation performance by comparing ML-SAINT, LRSAINT, and statistical SAINT for the tested accelerations varying from $1\ m/s^2$ to $6\ m/s^2$. This testing is done while keeping the vehicle's speed at $60\ km/h$ and testing 100 test vehicles (i.e., cargo vehicles) driving toward 6 delivery destinations. Fig 4.12 shows the performance of these compared three navigation mechanisms.

Figs. 4.12a and 4.12b show the impact of speeds on average link delays and E2E delays, respectively. This result clearly shows that the regression mechanism outperforms the statistical method both in terms of estimating the edge traversal delay and the route traversal delay to the destinations. ML-SAINT reduces up to 45.91% of SAINT's average link delay and up to 44.19% of SAINT's average E2E delay. It also stipulates the superiority of ML-SAINT over both LRSAINT and SAINT, thus proving that ML-SAINT is the best navigation option to satisfy the fastest parcel delivery.

Impact of the Density

This section assesses the navigation performance of ML-SAINT, LR-SAINT, and SAINT schemes under different densities in the road network. We tested these schemes' performance for background traffic of vehicles varying from $200 \sim 1400$ vehicles, while 100 delivery vehicles are driving in the same road. The car drives with a maximum speed of $60 km/h$, an acceleration of $3m/s^2$. Fig. 4.13 illustrates the navigation performance under different densities.

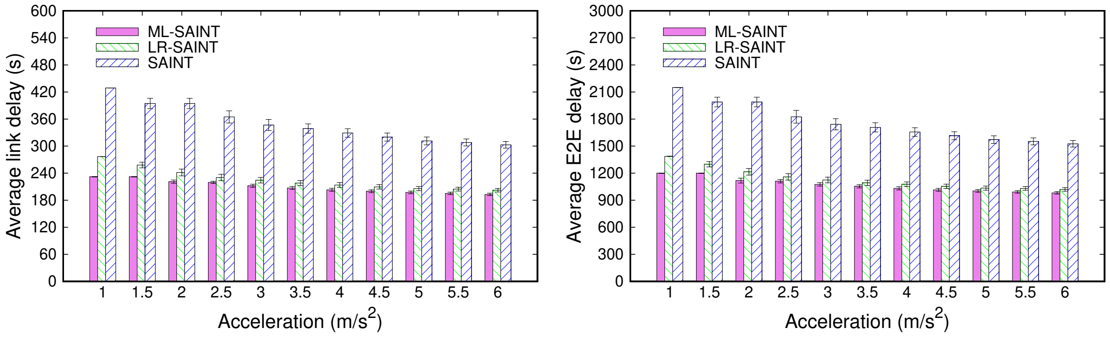
The results in Fig. 4.13 show the superiority of machine learning models (LR-SAINT and ML-SAINT) over the statistical SAINT to provide the best routes for the cargo vehicles when the traffic density grows larger. Fig. 4.13a shows that when the background vehicle is 600 or more, ML-SAINT provides the lowest value of link delay, thus ensuring



(a) Average link delay

(b) Average End to End delay

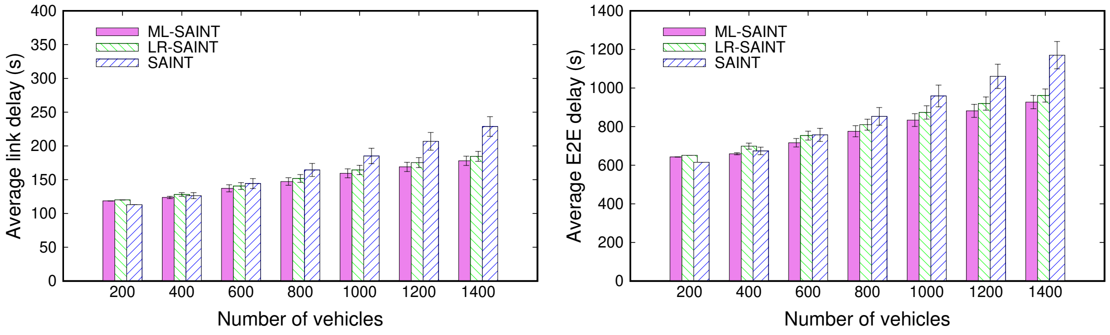
Figure 4.11: Impact of the maximum speed on navigation.



(a) Average link delay.

(b) Average End to End delay.

Figure 4.12: Impact of acceleration on navigation.



(a) Average link delay.

(b) Average End to End delay.

Figure 4.13: Impact of vehicle density on navigation.

that it provides the fastest traversal of a road compared to both LR-SAINT and SAINT in congested roads. Fig. 4.13b shows that ML-SAINT provides the fastest route from source to destination for the high-density road traffic (≥ 600 vehicles), compared with the linear regression-based SAINT and the statistics-based SAINT. ML-SAINT reduces up to 22.1% of SAINT's average link delay and up to 20.7% of SAINT's average E2E delay when the background traffic reaches 1400 vehicles. Thus, ML-SAINT is the best navigation option for delivery route finding.

4.5.4 Discussion

GCNs can be computationally intensive, especially for large graphs. It requires optimizing the GCN architecture and training process for real-time performance. It is to be noted that the graph structure influences the performance of the GCN. To exploit the full GCN potential, one must choose a graph representation that effectively captures the spatial dependencies in the traffic network. It also demands carefully tuning the GCN hyperparameters (e.g., number of layers, learning rate) and control parameters (e.g., control gains) for optimal performance. Optimizing the GCN training inference process makes it possible to run the navigation applications in real time. The system design of this work enables handling large-scale traffic networks, increases traffic volumes for scalability purposes, and ensures the system's robustness to noise and uncertainties in the traffic data. This spatio-temporal procedural mechanism adapts the graph to dynamic changes in the network (e.g., road closures, new constructions), thus proving its superiority for navigation systems.

4.6 Conclusion

This chapter explored the SAINT's performance when regression models were applied to spread traffic through road networks. It also assessed its impact on courier service delivery by enhancing its efficiency. ML-SAINT proved to be more effective when dealing with more destinations than baselines. In upcoming research, we will explore additional machine learning models or alternative methods to distribute traffic across road networks.

Chapter 5

Conclusion

This thesis studied the safety- and efficiency-enforcing mechanisms for terrestrial and aerial vehicular cyber-physical systems. It presents the mechanisms that, through the network-enabled opportunities, bridge the interoperability among nodes to achieve navigation safety and efficiency purposes. To summarize, the contributions and future perspectives of this work are as follows:

In the first place, it presented a context-aware navigation protocol (CNP) for safe driving in vehicular cyber-physical systems. It safely guides vehicles to avoid out-of-sight obstacles using wireless communications. It provides safe routes for vehicles at risk of being potentially affected by obstacles towards a safe lane through intuitive metrics defined to identify the safest lane for vehicle maneuvering. As out-of-sight obstacles cause many severe accidents, the vehicles' connectivity and communication play an important role in the proposed safety method of vehicular transportation.

In the second place, it proposed an autonomous collision-avoidance navigation algorithm for heavy-traffic UAV networks. This mechanism provides safe control suitable for maneuvering large numbers of UAVs in the growing urban air mobility. Through explicit decisions based on robust risk assessment proposed in this work, UAVs safely navigate the heterogeneous airspace, enabling the efficient achievement of their mission.

Finally, it advocated an advanced machine learning-assisted self-adaptive navigation approach that uses a graph neural network to enhance vehicle path planning and navigation in an application-specific environment. The vehicle selection and classification from continuous learning of how vehicular networks safely relate strengthens vehicles' capability to decide efficient paths autonomously.

For future work, in our first contribution, we will enhance our collision probability computation, considering a vehicle's reaction time. We will also implement and test this CNP protocol on real cars to improve its accuracy and usability for safe driving. We will also test the impact of the CNP on the overall trajectory performance of the driving vehicles as another way to test and improve navigation efficiency. In our second work, we will study the impact of the safety-enabling mechanism on the overall UAV service performance. For our third work, we will generalize our model for aerial vehicles to enable the modeling of aerial graph structures, study the aerial vehicles' trajectories and relationships to guarantee the aerial traffic predictability, and enhance the safety of service delivery in aerial networks.

References

[1] Hongyan Guo, Chen Shen, Hui Zhang, Hong Chen, and Rui Jia. Simultaneous trajectory planning and tracking using an mpc method for cyber-physical systems: A case study of obstacle avoidance for an intelligent vehicle. *IEEE Transactions on Industrial Informatics*, 14(9):4273–4283, 2018.

[2] Brandon Schoettle and Michael Sivak. A survey of public opinion about autonomous and self-driving vehicles in the us, the uk, and australia. 2014.

[3] Daniel Howard and Danielle Dai. Public perceptions of self-driving cars: The case of berkeley, california. In *Transportation Research Board 93rd Annual Meeting*, volume 14, 2014.

[4] Klaus Bengler, Klaus Dietmayer, Berthold Farber, Markus Maurer, Christoph Stiller, and Hermann Winner. Three decades of driver assistance systems: Review and future perspectives. *IEEE Intelligent Transportation Systems Magazine*, 6(4):6–22, 2014.

[5] Kian Seng Lee, Mark Ovinis, T Nagarajan, Ralph Seulin, and Olivier Morel. Autonomous patrol and surveillance system using unmanned aerial vehicles. In *2015 IEEE 15th International Conference on Environment and Electrical Engineering (EEEIC)*, pages 1291–1297. IEEE, 2015.

[6] Seon Jin Kim and Gino J Lim. Drone-aided border surveillance with an electrification line battery charging system. *Journal of Intelligent & Robotic Systems*, 92(3-4):657–670, 2018.

[7] Astrid Gynnild. The robot eye witness: Extending visual journalism through drone surveillance. *Digital journalism*, 2(3):334–343, 2014.

[8] Carmelo Donato Melita, Domenico Longo, Giovanni Muscato, and Gaetano Giudice. Measurement and exploration in volcanic environments. *Handbook of Unmanned Aerial Vehicles*, pages 2667–2692, 2015.

[9] Luis Merino, José Ramiro Martínez-de Dios, and Aníbal Ollero. Cooperative unmanned aerial systems for fire detection, monitoring, and extinguishing. *Handbook of Unmanned Aerial Vehicles*, pages 2693–2722, 2015.

[10] Konstantin Kondak, Aníbal Ollero, Ivan Maza, Kai Krieger, Alin Albu-Schaeffer, Marc Schwarzbach, and Maximilian Laiacker. Unmanned aerial systems physically interacting with the environment: Load transportation, deployment, and aerial manipulation. *Handbook of Unmanned Aerial Vehicles*, pages 2755–2785, 2015.

[11] Robin Kellermann, Tobias Biehle, and Liliann Fischer. Drones for Parcel and Passenger Transportation: A Literature Review. *Transportation Research Interdisciplinary Perspectives*, 4:100088, 2020.

[12] Bien Aime Mugabarigira, Yiwen Shen, Jaehoon Jeong, Tae Oh, and Han-You Jeong. Context-Aware Navigation Protocol for Safe Driving in Vehicular Cyber-Physical Systems. *IEEE Transactions on Intelligent Transportation Systems*, 24(1):128–138, 2023.

[13] Jaehoon Jeong, Yiwen Shen, Tae Oh, Sandra Céspedes, Nabil Benamar, Michelle Wetterwald, and Jérôme Härri. A comprehensive survey on vehicular networks for smart roads: A focus on ip-based approaches. *Vehicular Communications*, 29:100334, 2021.

[14] Santokh Singh. Critical reasons for crashes investigated in the national motor vehicle crash causation survey. Technical report, 2015.

[15] Bien Aime Mugabarigira and Jaehoon Paul Jeong. Context-Aware Navigation Protocol for Safe Flying of Unmanned Aerial Vehicles. 2024.

[16] Alberto Broggi, Andrea Cappalunga, Claudio Caraffi, Stefano Cattani, Stefano Ghidoni, Paolo Grisleri, Pier Paolo Porta, Matteo Posterli, and Paolo Zani. Terramax vision at the urban challenge 2007. *IEEE Transactions on Intelligent Transportation Systems*, 11(1):194–205, 2010.

[17] Mica R Endsley. Autonomous driving systems: a preliminary naturalistic study of the tesla model s. *Journal of Cognitive Engineering and Decision Making*, 11(3):225–238, 2017.

[18] Lien-Wu Chen and Po-Chun Chou. Big-cca: Beacon-less, infrastructure-less, and gps-less cooperative collision avoidance based on vehicular sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(11):1518–1528, 2015.

[19] Carolina Garcia-Costa, Esteban Egea-Lopez, Juan Bautista Tomas-Gabarron, Joan Garcia-Haro, and Zygmunt J Haas. A stochastic model for chain collisions of vehicles equipped with vehicular communications. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):503–518, 2011.

[20] Wilko Schwarting, Javier Alonso-Mora, Liam Paull, Sertac Karaman, and Daniela Rus. Safe nonlinear trajectory generation for parallel autonomy with a dynamic vehicle model. *IEEE Transactions on Intelligent Transportation Systems*, 19(9):2994–3008, 2017.

[21] Claudia Campolo, Antonella Molinaro, Giuseppe Araniti, and Antoine O Berthet. Better platooning control toward autonomous driving: An lte device-to-device communications strategy that meets ultralow latency requirements. *IEEE Vehicular Technology Magazine*, 12(1):30–38, 2017.

[22] Ehsan Moradi-Pari, Amin Tahmasbi-Sarvestani, and Yaser P Fallah. A hybrid systems approach to modeling real-time situation-awareness component of networked crash avoidance systems. *IEEE Systems Journal*, 10(1):169–178, 2014.

[23] Danilo Cavaliere and Sabrina Senatore. Towards an agent-driven scenario awareness in remote sensing environments. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1982–1989. IEEE, 2018.

[24] Qi Chen, Xu Ma, Sihai Tang, Jingda Guo, Qing Yang, and Song Fu. F-cooper: feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pages 88–100, 2019.

[25] Sinan Öncü, Jeroen Ploeg, Nathan Van de Wouw, and Henk Nijmeijer. Cooperative adaptive cruise control: Network-aware analysis of string stability. *IEEE Transactions on Intelligent Transportation Systems*, 15(4):1527–1537, 2014.

[26] Philipp Junietz, Farid Bonakdar, Björn Klamann, and Hermann Winner. Criticality metric for the safety validation of automated driving using model predictive tra-

jectory optimization. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 60–65. IEEE, 2018.

[27] Shuo Feng, Xintao Yan, Haowei Sun, Yiheng Feng, and Henry X Liu. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nature communications*, 12(1):1–14, 2021.

[28] Bowen Weng, Sughosh J Rao, Eeshan Deosthale, Scott Schnelle, and Frank Barickman. Model predictive instantaneous safety metric for evaluation of automated driving systems. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1899–1906. IEEE, 2020.

[29] Guofa Li, Yifan Yang, Tingru Zhang, Xingda Qu, Dongpu Cao, Bo Cheng, and Keqiang Li. Risk assessment based collision avoidance decision-making for autonomous vehicles in multi-scenarios. *Transportation research part C: emerging technologies*, 122:102820, 2021.

[30] Jaehoon Jeong, Taehyun Hwang, Tian He, and David Du. Mcta: Target tracking algorithm based on minimal contour in wireless sensor networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 2371–2375. IEEE, 2007.

[31] Iftikhar Hussain and Chen Bingcai. Cluster formation and cluster head selection approach for vehicle ad-hoc network (vanets) using k-means and floyd-warshall technique. *International Journal of Advanced Computer Science and Applications*, 8(12):11–15, 2017.

[32] Hans Kellerer, Ulrich Pferschy, David Pisinger, Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Multidimensional Knapsack Problems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[33] Sébastien Glaser, Benoit Vanholme, Saïd Mammar, Dominique Gruyer, and Lydie Nouveliere. Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction. *IEEE Transactions on Intelligent Transportation Systems*, 11(3):589–606, 2010.

[34] Nabil Benamar, Jerome Haerri, Jong-Hyouk Lee, Thierry Ernst, and Thierry Ernst. Basic Support for IPv6 Networks Operating Outside the Context of a Basic Service Set over IEEE Std 802.11. RFC 8691, December 2019.

[35] Yiwen Shen, Jaehoon Jeong, Tae Oh, and Sang Hyuk Son. CASD: a framework of context-awareness safety driving in vehicular networks. In *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pages 252–257. IEEE, 2016.

[36] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2575–2582. IEEE, 2018.

[37] OMNeT++. Network Simulation Framework. accessed 2024.

[38] Jakob Erdmann. Lane-changing model in sumo. *Proceedings of the SUMO2014 modeling mobility with open data*, 24:77–88, 2014.

[39] Intel. Technology and Computing Requirements for Self-Driving Cars.

[40] Mohammad Ghamari, Pablo Rangel, Mehrube Mehrubeoglu, Girma S Tewolde, and R Simon Sherratt. Unmanned Aerial Vehicle Communications for Civil Applications: A Review. *IEEE Access*, 10:102492–102531, 2022.

[41] Adam P. Cohen, Susan A. Shaheen, and Emily M. Farrar. Urban Air Mobility: History, Ecosystem, Market Potential, and Challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(9):6074–6087, 2021.

[42] J Sanchez-Garcia, JM Garcia-Campos, SL Toral, DG Reina, and F Barrero. A Self Organising Aerial Ad Hoc Network Mobility Model for Disaster Scenarios. In *2015 International Conference on Developments of E-Systems Engineering (DeSE)*, pages 35–40. IEEE, 2015.

[43] Evşen Yanmaz, Saeed Yahyanejad, Bernhard Rinner, Hermann Hellwagner, and Christian Bettstetter. Drone Networks: Communications, Coordination, and Sensing. *Ad Hoc Networks*, 68:1–15, 2018.

[44] Dong Yin, Xuan Yang, Huangchao Yu, Siyuan Chen, and Chang Wang. An Air-to-Ground Relay Communication Planning Method for UAVs Swarm Applications. *IEEE Transactions on Intelligent Vehicles*, 8(4):2983–2997, 2023.

[45] Huaxin Qiu and Haibin Duan. A Multi-Objective Pigeon-Inspired Optimization Approach to UAV Distributed Flocking Among Obstacles. *Information Sciences*, 509:515–529, 2020.

[46] Kleber Loayza, Pedro Lucas, and Enrique Peláez. A Centralized Control of Movements Using a Collision Avoidance Algorithm for a Swarm of Autonomous Agents. In *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, pages 1–6, 2017.

[47] Ehsan Moradi-Pari, Amin Tahmasbi-Sarvestani, and Yaser P. Fallah. A Hybrid Systems Approach to Modeling Real-Time Situation-Awareness Component of Networked Crash Avoidance Systems. *IEEE Systems Journal*, 10(1):169–178, 2016.

[48] Xuekuan Yang, Wei Wang, and Ping Huang. Distributed Optimal Consensus with Obstacle Avoidance Algorithm of Mixed-Order UAVs-USVs-UUVs Systems. *ISA transactions*, 107:270–286, 2020.

[49] Giuseppe Bevacqua, Jonathan Cacace, Alberto Finzi, and Vincenzo Lippiello. Mixed-Initiative Planning and Execution for Multiple Drones in Search and Rescue Missions. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 25, pages 315–323, 2015.

[50] Björn Lindqvist, Sina Sharif Mansouri, Jakub Haluška, and George Nikolakopoulos. Reactive Navigation of an Unmanned Aerial Vehicle With Perception-Based Obstacle Avoidance Constraints. *IEEE Transactions on Control Systems Technology*, 30(5):1847–1862, 2022.

[51] Xiangpeng Wan, Hakim Ghazzai, Yehia Massoud, and Hamid Menouar. Optimal Collision-Free Navigation for Multi-Rotor UAV Swarms in Urban Areas. In *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pages 1–5, 2019.

[52] Xiaobai Ma, Ziyuan Jiao, Zhenkai Wang, and Dimitra Panagou. 3-D Decentralized Prioritized Motion Planning and Coordination for High-Density Operations of Micro Aerial Vehicles. *IEEE Transactions on Control Systems Technology*, 26(3):939–953, 2017.

[53] Afshin Mardani, Marcello Chiaberge, and Paolo Giaccone. Communication-Aware UAV Path Planning. *IEEE Access*, 7:52609–52621, 2019.

[54] Xianjin Zhou, Fei Gao, Xi Fang, and Zehong Lan. Improved Bat Algorithm for UAV Path Planning in Three-Dimensional Space. *IEEE Access*, 9:20100–20116, 2021.

[55] James Keller, Dinesh Thakur, Maxim Likhachev, Jean Gallier, and Vijay Kumar. Coordinated Path Planning for Fixed-Wing UAS Conducting Persistent Surveillance Missions. *IEEE Transactions on Automation Science and Engineering*, 14(1):17–24, 2016.

[56] Yukai Hou, Jin Zhao, Rongqing Zhang, Xiang Cheng, and Liuqing Yang. UAV Swarm Cooperative Target Search: A Multi-Agent Reinforcement Learning Approach. *IEEE Transactions on Intelligent Vehicles*, 9(1):568–578, 2024.

[57] Xinyu Jiao, Junjie Chen, Kun Jiang, Zhong Cao, and Diange Yang. Autonomous Driving Risk Assessment With Boundary-Based Environment Model. *IEEE Transactions on Intelligent Vehicles*, 9(1):642–655, 2024.

[58] Muhammad Asghar Khan, Ijaz Mansoor Qureshi, and Fahimullah Khanzada. A Hybrid Communication Scheme for Efficient and Low-Cost Deployment of Future Flying Ad-Hoc Network (FANET). *Drones*, 3(1):16, 2019.

[59] Zhongjun Hu and Xu Jin. Formation Control for an UAV Team With Environment-Aware Dynamic Constraints. *IEEE Transactions on Intelligent Vehicles*, 9(1):1465–1480, 2024.

[60] Hao Hao, Changqiao Xu, Wei Zhang, Shujie Yang, and Gabriel-Miro Muntean. Joint Task Offloading, Resource Allocation, and Trajectory Design for Multi-UAV Cooperative Edge Computing With Task Priority. *IEEE Transactions on Mobile Computing*, 23(9):8649–8663, 2024.

[61] Florence Ho, Ruben Geraldes, Artur Gonçalves, Marc Cavazza, and Helmut Prendinger. Improved Conflict Detection and Resolution for Service UAVs in Shared Airspace. *IEEE Transactions on Vehicular Technology*, 68(2):1231–1242, 2019.

[62] Kaya Kuru, John Michael Pinder, Benjamin Jon Watkinson, Darren Ansell, Keith Vinning, Lee Moore, Chris Gilbert, Aadithya Sujit, and David Jones. Toward Mid-Air Collision-Free Trajectory for Autonomous and Pilot-Controlled Unmanned Aerial Vehicles. *IEEE Access*, 11:100323–100342, 2023.

[63] Sunan Huang, Rodney Swee Huat Teo, and Kok Kiong Tan. Collision Avoidance of Multi Unmanned Aerial Vehicles: A review. *Annual Reviews in Control*, 48:147–164, 2019.

[64] Swarun Kumar, Lixin Shi, Nabeel Ahmed, Stephanie Gil, Dina Katabi, and Daniela Rus. CarSpeak: A Content-Centric Network for Autonomous Driving. 42(4):259–270, aug 2012.

[65] Belkacem Kada and Y Ghazzawi. Robust PID controller design for an UAV flight control system. In *Proceedings of the World Congress on Engineering and Computer Science*, volume 2, pages 1–6, 2011.

[66] Jaehoon Jeong, Hohyeon Jeong, Eunseok Lee, Tae Oh, and David HC Du. SAINT: Self-Adaptive Interactive Navigation Tool for Cloud-Based Vehicular Traffic Optimization. *IEEE Transactions on Vehicular Technology*, 65(6):4053–4067, 2015.

[67] Giovanni Nardini, Dario Sabella, Giovanni Stea, Purvi Thakkar, and Antonio Virdis. Simu5G–An OMNeT++ Library for End-to-End Performance Evaluation of 5G Networks. *IEEE Access*, 8:181176–181191, 2020.

[68] Zhaoyao Bao, Ziyu Zhang, and Chi Xie. Landmark Selection and Path Planning for Unmanned Vehicles with Position Error Corrections. *Transportation research part C: emerging technologies*, 153:104186, 2023.

[69] Khac-Hoai Nam Bui, Jiho Cho, and Hongsuk Yi. Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues. *Applied Intelligence*, 52(3):2763–2774, 2022.

[70] Xin Zhou, Jinglong Wang, Yong Liu, Xingyu Wu, Zhiqi Shen, and Cyril Leung. Inductive graph transformer for delivery time estimation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, WSDM '23, page 679–687, New York, NY, USA, 2023. Association for Computing Machinery.

[71] Huiting Hong, Yucheng Lin, Xiaoqing Yang, Zang Li, Kung Fu, Zheng Wang, Xiaohu Qie, and Jieping Ye. Heteta: Heterogeneous information network embedding for estimating time of arrival. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 2444–2454, New York, NY, USA, 2020. Association for Computing Machinery.

[72] David Alexander Tedjopurnomo, Zhifeng Bao, Baihua Zheng, Farhana Murtaza Choudhury, and A. K. Qin. A Survey on Modern Deep Neural Network for Traffic Prediction: Trends, Methods and Challenges. *IEEE Transactions on Knowledge and Data Engineering*, 34(4):1544–1561, 2022.

[73] Joel Janai, Fatma Güney, Aseem Behl, Andreas Geiger, et al. Computer vision for autonomous vehicles: Problems, datasets and state of the art. *Foundations and Trends® in Computer Graphics and Vision*, 12(1–3):1–308, 2020.

[74] Eda Okur, Shachi H Kumar, Saurav Sahay, Asli Arslan Esme, and Lama Nachman. Natural language interactions in autonomous vehicles: Intent detection and slot filling from passenger utterances. In *International Conference on Computational Linguistics and Intelligent Text Processing*, pages 334–350. Springer, 2019.

[75] Mohammmad Asifur Rahman Shuvo, Muhtadi Zubair, Afsara Tahsin Purnota, Sarowar Hossain, and Muhammad Iqbal Hossain. Traffic forecasting using time-series analysis. In *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, pages 269–274, 2021.

[76] Ishteaque Alam, Dewan Md Farid, and Rosaldo JF Rossetti. The prediction of traffic flow with regression analysis. In *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2018, Volume 2*, pages 661–671. Springer, 2019.

[77] He Yan, Liyong Fu, Yong Qi, Dong-Jun Yu, and Qiaolin Ye. Robust ensemble method for short-term traffic flow prediction. *Future Generation Computer Systems*, 133:395–410, 2022.

[78] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of sumo-simulation of urban mobility. *International journal on advances in systems and measurements*, 5(3&4), 2012.

[79] Walid Fahs, Fadlallah Chbib, Abbas Rammal, Rida Khatoun, Ali El Attar, Issam Zaytoun, and Joel Hachem. Traffic Congestion Prediction Based on Multivariate Modelling and Neural Networks Regressions. *Procedia Computer Science*, 220:202–209, 2023. The 14th International Conference on Ambient Systems, Networks and Technologies Networks (ANT) and The 6th International Conference on Emerging Data and Industry 4.0 (EDI40).

[80] Georges A Croes. A method for solving traveling-salesman problems. *Operations research*, 6(6):791–812, 1958.

[81] Elham Ghaffari, Amir Masoud Rahmani, Morteza Saberikamarposhti, and Amir Sahafi. An optimal path-finding algorithm in smart cities by considering traffic congestion and air pollution. *IEEE Access*, 10:55126–55135, 2022.

[82] Marc E Pfetsch. Branch-and-cut for the maximum feasible subsystem problem. *SIAM Journal on Optimization*, 19(1):21–38, 2008.

[83] Eva Ostertagová. Modelling using polynomial regression. *Procedia Engineering*, 48:500–506, 2012. Modelling of Mechanical and Mechatronics Systems.

[84] Shridevi Jeevan Kamble and Manjunath R Kounte. Machine learning approach on traffic congestion monitoring system in internet of vehicles. *Procedia Computer Science*, 171:2235–2241, 2020. Third International Conference on Computing and Network Communications (CoCoNet'19).

[85] Songsang Koh, Bo Zhou, Hui Fang, Po Yang, Zaili Yang, Qiang Yang, Lin Guan, and Zhigang Ji. Real-time deep reinforcement learning based vehicle navigation. *Applied Soft Computing*, 96:106694, 2020.

[86] Weiwei Jiang and Jiayun Luo. Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications*, 207:117921, 2022.

[87] Tao Ma, Constantinos Antoniou, and Tomer Toledo. Hybrid machine learning algorithm and statistical time series model for network-wide traffic forecast. *Transportation Research Part C: Emerging Technologies*, 111:352–372, 2020.

[88] Junhee Kwon and Jaehoon Jeong. A parcel delivery scheduling scheme in road networks. In *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1750–1755. IEEE, 2022.

[89] Tao Liu, Aimin Jiang, Jia Zhou, Min Li, and Hon Keung Kwan. Graphsage-based dynamic spatial–temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 24(10):11210–11224, 2023.

[90] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 922–929, 2019.

[91] Yaguang Li, Kun Fu, Zheng Wang, Cyrus Shahabi, Jieping Ye, and Yan Liu. Multi-task representation learning for travel time estimation. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, KDD '18, pages 1695–1704, New York, NY, USA, 2018. Association for Computing Machinery.

[92] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3634–3640. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[93] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848–3858, 2020.

[94] Xin Zhou, Jinglong Wang, Yong Liu, Xingyu Wu, Zhiqi Shen, and Cyril Leung. Inductive graph transformer for delivery time estimation. In *Proceedings of the*

Sixteenth ACM International Conference on Web Search and Data Mining, pages 679–687, 2023.

[95] Miao Yu. Construction of regional intelligent transportation system in smart city road network via 5g network. *IEEE Transactions on Intelligent Transportation Systems*, 24(2):2208–2216, 2023.

[96] Miguel Sepulcre, Manuel Gonzalez-Martin, Javier Gozalvez, Rafael Molina-Masegosa, and Baldomero Coll-Perales. Analytical Models of the Performance of IEEE 802.11p Vehicle to Vehicle Communications. *IEEE Transactions on Vehicular Technology*, 71(1):713–724, 2022.

[97] Karla L Hoffman, Manfred Padberg, Giovanni Rinaldi, et al. Traveling salesman problem. *Encyclopedia of operations research and management science*, 1:1573–1578, 2013.

[98] Gerhard J Woeginger. Exact algorithms for np-hard problems: A survey. In *Combinatorial Optimization—Eureka, You Shrink! Papers Dedicated to Jack Edmonds 5th International Workshop Aussois, France, March 5–9, 2001 Revised Papers*, pages 185–207. Springer, 2003.

[99] Tianqi Zhang, Qitian Wu, and Junchi Yan. Learning high-order graph convolutional networks via adaptive layerwise aggregation combination. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):5144–5155, 2023.

[100] Taylor B Arnold. kerasr: R interface to the keras deep learning library. *J. Open Source Softw.*, 2(14):296, 2017.

[101] OpenStreetMap. OpenStreetMap. accessed 2025.

Appendix A

CNP Appendix

This section describes the supplement content of CNP.

A.1 Emergency Maneuver Planning

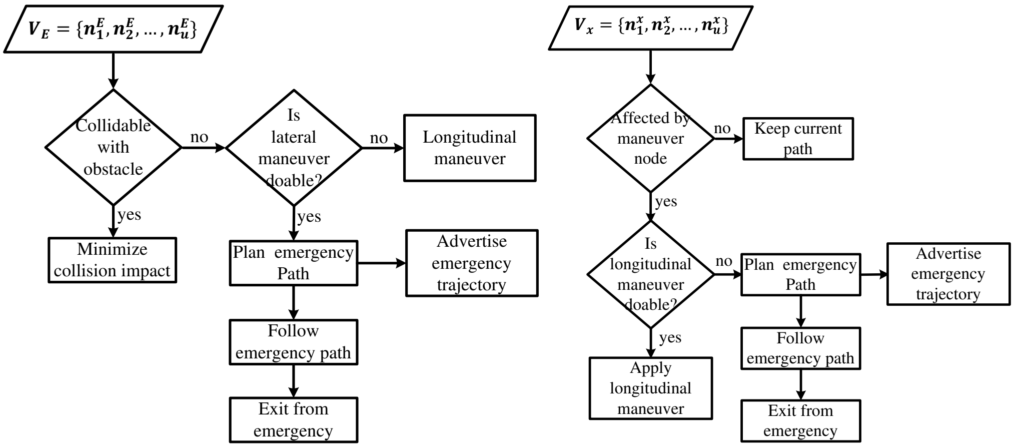
In this section, we expound on the vehicles' emergency maneuvers based on the criticality of the CNP risk assessment.

The CH orchestrates the maneuvers of the CMs from closest to farthest from the obstacle. A goal position that an emergency vehicle $n_{em} \in \mathcal{V}_E$ drives toward is called the target position t_{pos}, in the maneuver lane. The maneuver reference path (p_{ref}) from the contour area of a vehicle n_{em} has t_{pos} as its ending point.

A.1.1 CNP Emergency Vehicle Maneuver

We advocate for the minimal contour-based path planning to define both the control path and its tracking.

During the maneuver, the first task is to define the maneuver inputs $u = \begin{bmatrix} \rho & a \end{bmatrix}^T$


$V_E = \{n_1^E, n_2^E, \ldots, n_u^E\}$
Collidable with obstacle
no
Is lateral maneuver doable?
no
Longitudinal maneuver
yes
Minimize collision impact
yes
Plan emergency Path
Advertise emergency trajectory
Follow emergency path
Exit from emergency
$V_x = \{n_1^x, n_2^x, \ldots, n_u^x\}$
Affected by maneuver node
no
Keep current path
yes
Is longitudinal maneuver doable?
no
Plan emergency Path
Advertise emergency trajectory
yes
Apply longitudinal maneuver
Follow emergency path
Exit from emergency

(a) Emergency vehicle maneuver procedure

(b) Procedure for adjacent vehicles of the the emergency vehicle

Figure A.1: Cluster members' maneuver processing flow toward obstacle avoidance.

that guarantee an n_{em} to safely bypass the obstacle. Those control inputs are the smallest steering angle that will point n_{em} to t_{pos} from the n_{em} contour area definition steering angle inputs.

$$u = \min\left(\begin{bmatrix} \rho^1 \\ a^1 \end{bmatrix}, \begin{bmatrix} \rho^2 \\ a^2 \end{bmatrix}, \ldots, \begin{bmatrix} \rho^k \\ a^k \end{bmatrix} \right), \tag{A.1}$$

where

ρ = steering angle input,

a = acceleration input,

$\begin{bmatrix} \rho^j \\ a^j \end{bmatrix}, j \leq k$ = contour definition inputs.

We obtain an emergency control input from the emergency vehicle's minimal contour area.

Definition A.1.1 (Minimal Contour Polygon Area of a Vehicle)**.** *Let the **Minimal Contour Polygon Area of a Vehicle** be a range formed by a series of n positions ζ that a vehicle can reach within the interval time Δ_t, when driving at steering angles $[\rho_{min}, \rho_{max}]$.*

For $\rho \in [\rho_{min}, \rho_{max}]$, with $\zeta_p = \{Path_{\rho min}...Path_{\rho max}\}$ paths, each of the path time step positions are $Path_\rho = \{P_{to}, P_{t1}, ...P_{tk}\}$ for $k = \frac{\Delta_t}{\delta}$.

Given the current position P_t of a path, the next position $P_{t+\delta}$ is computed as follows:

$$\begin{aligned} P_{t+\delta} &= P_t + f(a_t, \rho_t) \\ &= (x_t, y_t) + v_t\delta(\cos(\theta_{t+\delta}), \sin(\theta_{t+\delta})), \end{aligned} \tag{A.2}$$

where $\theta_{t+\delta} = \theta_t + \frac{v_{t+\delta}}{L\tan(\rho_t)}$ and $v_{t+\delta} = v_t + a_t\delta$. Our algorithm decides $P_{ref} \in \zeta_p$ such that it heads to a destination that is as close to the t_{pos} as possible.

$$P_{ref} \in \zeta_p \text{ for} \arg\min[dist(t_{pos}, P_{tk})]. \tag{A.3}$$

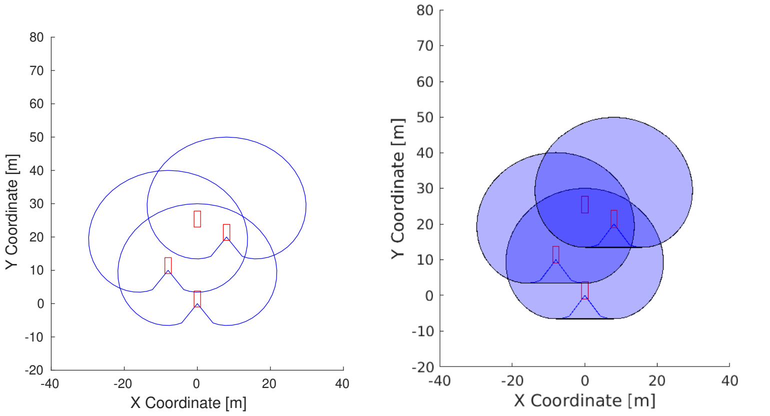
A vehicle n_{em} follows P_{ref} to avoid collisions with n_{ob} or other nodes throughout the entire Δ_t. For each step time δ, the control input (ρ, a) that defines P_{t+1} next to n_{em}'s position will always be as close to P_{ref} as possible. The maneuver process also ensures the non-degradation of the remaining vehicles' safety that might result from n_{em}'s movement. The emergency trajectories not only rely on the kinematics of the currently driving vehicles, but also flexibly adapt to the arrivals of new vehicles on the emergency road.

A.1.2 Sequential Probabilistic Control Definition

The probability of n_1^E colliding with n_1^L while maneuvering toward L_L to avoid a collision with n_{ob} is computed as:

$$P(n_1^E \otimes n_1^L) = \frac{\mathcal{A}_1^L \cap \mathcal{A}_1^E}{\mathcal{A}_1^E}, \tag{A.4}$$

where $\mathcal{A}_1^L$ and $\mathcal{A}_1^E$ are the minimal contour polygons areas respectively created by vehicles n_1^L and n_1^E during the maneuver time Δ_t. Fig. A.2 illustrates the contours of three adjacent vehicles driving in a three-lane road which has an obstacle in the middle lane.



(a) Vehicles' minimal contours (b) Minimal contour polygons area

Figure A.2: Contours of adjacent vehicles in an emergency situation.

Fig. A.2b shows the contour intersection area that has been taken into account while calculating the appropriate n_1^E's maneuver to avoid future collision risks.

We maneuver n_1^E within Δ_t in consideration of the neighbors' contour areas. The control inputs $\begin{bmatrix}\rho_i & a_i\end{bmatrix}^T$ shall guarantee the safe conditional collision probability toward the neighboring vehicles. The probability that n_1^E collides with two vehicles in the maneuver lane will be calculated by the independent and identically distributed (IID) as follows:

$$\begin{aligned} P(n_2^L \otimes n_1^L \otimes n_1^E) &= P(n_2^L \otimes n_1^L | n_1^L \otimes n_1^E) \\ &= P(n_2^L \otimes n_1^L) P(n_1^L \otimes n_1^E). \end{aligned} \tag{A.5}$$

Considering a maneuver lane wherein three vehicles are currently driving, the maneuver effects are computed in the same way as follows:

$$\begin{aligned} P(n_3^L \otimes n_2^L \otimes n_1^L \otimes n_1^E) &= P(n_2^L \otimes n_3^L | n_2^L \otimes n_1^E) \\ &= P(n_3^L \otimes n_2^L) P(n_2^L \otimes n_1^L) P(n_1^L \otimes n_1^E). \end{aligned} \tag{A.6}$$

By generalizing to the number of vehicles (u) driving in L, the collision probabilities

Algorithm 6 Compute Control Inputs

```
$LSD \leftarrow (x_0, y_0, v_0, \theta_0, \rho_0)$ ▷ Local Sensor Data (LSD) are data sensed by a vehicle's On-board Unity (OBU): position $(x_0, y_0)$, speed $v_0$, direction $\theta_0$, and current steering angle $\rho_0$, respectively
$RSD \leftarrow (x_i, y_i, v_i, \theta_i)$ ▷ Remote Sensor Data (RSD) are neighbors' data received through communication. Each neighbor $i$ is represented by its position $(x_i, y_i)$, speed $v_i$ and direction $\theta_i$
function COMPUTE_EMERGENCY_CONTROL_INPUTS($n_1^E$)
    $t_{pos} \leftarrow Define\_Target\_Position(x_{nob}, y_{nob})$ ▷ $(x_{nob}, y_{nob})$ is obstacle's position from $RSD$
    $\zeta_{n_1^E} \leftarrow Compute\_Contour\_Polygon\_Area()$
    $A_{inter} \leftarrow \emptyset$
    for each neighbor vehicle in $L_x$ do ▷ $L_x$ is the maneuver lane resulting from Algorithm 1
        $\zeta_{n_1^L} \leftarrow Compute\_Contour\_Polygon\_Area()$
        $A_v \leftarrow \zeta_{n_1^E} \cap \zeta_{n_1^L}$
        if $A_v \neq \emptyset$ then
            $A_{inter} \leftarrow A_{inter} \cup A_v$
        end if
    end for
    $n_{em}.a \leftarrow 0$ ▷ Initialize the acceleration input $a$ of $n_{em}$
    $n_{em}.\rho \leftarrow 0$ ▷ Initialize the steering angle input $\rho$ of $n_{em}$
    if $A_{inter} = \emptyset$ then ▷ No risk of colliding with neighbors. The algorithm defines the steering angle and maintains its speed while changing lanes
        $n_{em}.\rho \leftarrow Compute\_Best\_Steering\_Angle(t_{pos})$
    else ▷ Compute both the acceleration and steering needed to avoid the collision
        $n_{em} \leftarrow Compute\_Maneuver\_Inputs(t_{pos}, A_{inter})$
    end if
    return $n_{em}$
end function
```

become a chain of conditional probabilities.

$$\begin{aligned} & P(n_u^L \otimes n_{u-1}^L \otimes ... \otimes n_1^L \otimes n_1^E) \\ & = P(n_u^L \otimes n_{u-1}^L | n_{u-1}^L \otimes n_{u-2}^L) \\ = & \, P(n_1^L \otimes n_1^E) \prod_{i=u}^{2} P(n_i^L \otimes n_{i-1}^L). \end{aligned} \tag{A.7}$$

A.1.3 Control Input Computational Algorithm

To define n_1^E's maneuver inputs, the CNP considers the risks of colliding with the maneuvering vehicles in a chosen lane. Algorithm 6 shows the process of computing the control inputs (ρ_t, a_t) toward the target position t_{pos}. Lines 1-2 provide sensed data as input to the algorithm. Line 4 selects the target position in the maneuver lane defined by Algorithm 1. Line 5 calculates the emergency contour area. The next lines examine the proper control inputs of n_1^E while considering the other driving vehicles.

The contour-based risk analysis is computed in terms of intersection area accordingly to (A.4). Line 6 initializes the intersection area of the neighbor vehicles' contour areas. In lines 7-13, a comparison of the n_1^E's contour polygon area with those of the neighboring vehicles is made to examine the collision risks during any maneuvers. When the vehicle contours have no intersection, a vehicle maneuver will follow the definition of the steering angle ρ_t resulting from (A.3). Lines 14-15 initialize the control inputs. Lines 16-20 define the control inputs according to the neighboring vehicles' risks. After defining the needed inputs, the maneuvers of n_{em} are returned in line 21.

A.2 Collision Strength Minimization

If the collision probability of n_{em} is obtained from (8a), then the vehicle is in an inevitable collision state (ICS), and it will crash. For this unavoidable situation, a collision strength minimization mechanism is needed to minimize the energy transfer between the colliding vehicles and limit the number of vehicles involved in the collision to the possible extent.

The severity of a collision is proportional to the masses of the two colliding vehicles and their corresponding speeds. Assuming that an emergency vehicle n_{em} with speed v_{em} and mass m_{em} collides with an obstacle n_{ob} with speed v_{ob} and mass m_{ob}, the collision strength calculation is made using their Equivalent Energy Speed (EES). Knowing the approximate resultant speed $\hat{v}_{em}$ of the emergency vehicle n_{em} after collision, the EES can be calculated as:

$$EES = \hat{v}_{em} - v_{em} = \frac{2m_{ob}}{m_{em} + m_{ob}}(v_{ob} - v_{em}). \tag{A.8}$$

The EES computation for vehicles driving in different directions, such as collisions from changing or overstepping the front vehicle, will result in a more general calculation form that involves driving direction θ_{em}. The speed that n_{em} applies on n_{ob} during a collision is:

$$v'_{em} = v_{em} \cos(\theta_{em}), \tag{A.9}$$

then, the EES will be computed as

$$EES = \hat{v'_{em}} - v'_{em} = \frac{2m_{ob}}{m_{em} + m_{ob}}(v_{ob} - v'_{em}). \tag{A.10}$$

The greater the difference in vehicle speeds is during a collision, the more severe the collision is, and the more fatal it is. Autonomous emergency braking, when the vehicle is in the ICS, reduces the collision severity. Decelerating the n_{em} just before colliding with n_{ob} can reduce the energy transfer between them. It is clear that if the speed difference is closer to null during a collision, the EES will only depend on the mass difference. The deceleration of the vehicle just before the collision is:

$$Dec = \frac{\hat{v}_{em} - v_{em}}{\Delta_t}. \tag{A.11}$$

Thus, if $\hat{v}_{em} = 0$ km/h during a collision, the EES is nullified.

This means that the damages from a collision are negligible. The lower the speed is, the lower the EES is. Another aspect of minimizing the collision strength is to reduce the number of involved vehicles. CNP has communication-based situational awareness to notify the driving vehicle of a collision in advance. Prior decisions are required to steer the next vehicle before arriving at the point of collision. This reduces the number of vehicles involved in an accident and avoids a chain of collisions.

Appendix B

CANA Appendix

This section describes the supplement content of CANA.

B.1 Emergency Forwarding in a UAV Network

Fig. B.1 shows an aerial emergency processing structure. The emergency is handled through a communication protocol wherein UAVs are flying into m clusters $C = \{C_1, C_2, ..., C_m\}$ and the cluster head (CH) leads the cluster communication and maneuver calculations. Our model uses the K-means clustering approach [31] where CH is a cluster-head UAV, which makes the smallest intracluster Euclidean distance from its cluster members. CHs share driving information with each other via Vehicle-to-Vehicle (V2V) communication.

Let's consider an emergency event such as a UAV suddenly hovering in the direction of other UAVs' trajectories. The emergency handling process described in Fig. B.1 handles UAVs' flights in a manner that avoids collisions in the airspace, and follows these steps:

Step 1: An emergency event suddenly occurs in the airspace and blocks the UAV's trajectory, therefore becoming an obstacle.

Step 2: A cluster member detects and broadcasts the obstacle information to its neighbors.

Step 3: The CH receives and evaluates this information to identify the obstacle collision risk. It calculates maneuvers for its CMs.

Step 4: The CH then broadcasts the maneuvers changes of its members.

Step 5: The safety information is shared with neighboring clusters via their CHs.

Step 6: Each CH will take proactive steps to address any safety issues and reflect the trajectory changes from the neighboring cluster.

B.2 Emergency Risk Assessment in Aerial Virtual Lanes

In this section, we expound on the UAV's probabilistic sequential risk assessment and the computation of virtual lane quality.

The emergency UAV n_1^E maneuvers toward a lane with better quality. Let the safe probability q_i of a child UAV n_c whose flight heads to parent UAV n_p be

$$q_i = 1 - p_i, \tag{B.1}$$

where p_i is the collision probability of event $n_p \otimes n_c$. The safe lane quality of the left lane of the virtual aerial lanes will be computed as follows:

$$S_L = \prod_{i=1}^{u} q_i. \tag{B.2}$$

The safe lane quality towards the right lane of the virtual aerial lanes is computed as:

$$S_R = \prod_{k=1}^{v} q_k. \tag{B.3}$$

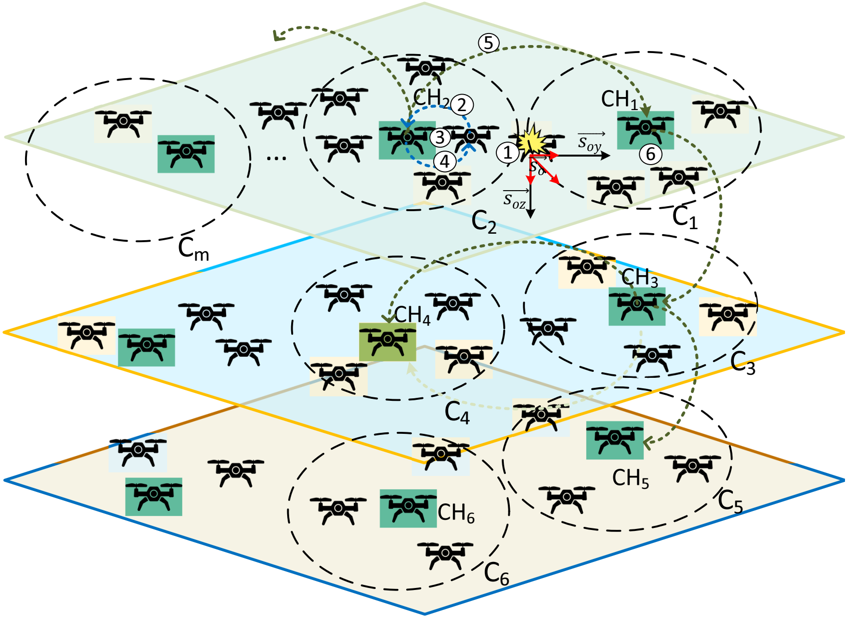


Figure B.1: Forwarding the emergency and the maneuver handling in clusters.

The safe lane quality in the emergency lane of the virtual aerial lanes is computed as:

$$S_E = \prod_{l=1}^{w} q_l. \tag{B.4}$$

The safe lane quality in the upper lane of the virtual aerial lanes is computed as:

$$S_U = \prod_{m=1}^{y} q_m. \tag{B.5}$$

And lastly, the safe lane quality in the down lane of the virtual aerial lanes is computed as:

$$S_D = \prod_{n=1}^{z} q_n. \tag{B.6}$$

In the above equations, q_i is the safe probability for n_p and n_c in the left lane, q_k is the safe probability for n_p and n_c in the right lane, q_l is the safe probability for n_p and n_c in the emergency lane, q_m is the safe probability for n_p and n_c in the upper lane, and q_n is the safe probability for n_p and n_c in the down lane. Given the safe lane qualities S_L,

S_R, S_E, S_U, and S_D in the left, right, emergency, upper, and down lanes, respectively, the decision of the lane where UAV n_1^E maneuvers toward is calculated according to (3.22).

논문요약

사이버 물리 시스템에서 이동 객체의 안전하고 효율적인 기동 제어를 위한 컨텍스트 인식 내비게이션

MUGABARIGIRA BIEN AIME

전자전기컴퓨터공학과

성균관대학교

차량 및 무인 항공기(UAV)와 같은 사이버-물리 시스템에서 끊김 없이 이동하는 노드의 증가는 의도된 서비스의 품질을 보장하기 위한 안전한 프레임워크를 요구합니다. 본 논문은 지상 차량 및 무인 항공기(UAV)(일반적으로 UAV로 알려짐)를 위한 컨텍스트 인식 내비게이션 메커니즘을 연구합니다.

도로상의 자율주행 차량 네트워크는 무인 차량(즉, 자율주행차)의 주행 안전성을 향상시킬 수 있습니다. 주행 중 도로상의 위험 상황은 관련 차량의 속도, 방향 및 교통 밀도에 따라 심각도가 높아질 수 있습니다. 주행 중 도로상의 위험 상황은 차량의 속도, 방향 및 교통 밀도에 따라 심각도가 높아질 수 있습니다.

따라서 현재 주행 중인 차량과 비상 구역(예: 도로 위험 및 교통사고 현장)으로 마주오는 차량을 모두 처리하는 조종 메커니즘이 필요합니다. 본 논문에서는 도시 도로에서 주행하는 차량의 안전성을 향상시키는 컨텍스트 인식 내비게이션 프로토콜(CNP)을 제시합니다. 첫째, CNP는 차량 네트워크와 차량 내 센서를 기반으로 차량의 동작을 추적하는 충돌 회피 모듈을 포함하고 있으며, 이 모듈은 주행 위험을 분석하여 위험 상황에서 필요한 기동을 결정합니다. 둘째, CNP는 차량의 기동이 불가능한 상황에서 위험한 도로의 충돌 피해 심각도를 줄이기 위한 충돌 완화 전략을 수립합니다. CNP의 효과를 입증하고 평가하기 위해 이론적 분석과 광범위한 시뮬레이션을 수행했습니다. 그 결과, CNP는 도로 충돌 위험을 5% 미만으로 유지하면서도 통신 오버헤드를 기준 체계 대비

최대 60%까지 줄일 수 있음을 보여주었습니다.

무인 항공기(UAV)는 임무 중심 응용 분야에 매우 적합한 특성으로 인해 최근 연구자들의 관심을 끌고 있습니다. 여러 대의 UAV가 동일 지역에서 함께 비행할 경우, 밀도 높은 UAV 네트워크가 생성되므로, UAV 간 충돌을 방지하기 위한 정밀한 제어가 필요합니다. 본 논문은 교통량이 많은 이종 무인기 네트워크에서 협력 비행 제어를 사용하는 충돌 회피 항법 알고리즘(CANA)을 제안합니다. 자율 군집 기반 협력 및 조정 설계를 통해 대규모 무인기 교통 이동 상황 공유를 지원합니다. 제안된 무인기 항법 시스템은 감지 및 통신을 활용하여 주변 무인기의 특정 상태를 감지하고, 이를 통해 우회 경로를 설정하여 충돌 가능성을 감지하고 회피한다. 시뮬레이션 기반 평가를 통해 제안된 CANA의 성능을 검증합니다. 본 알고리즘은 기존 방식 대비 통신 제어 오버헤드를 66.6%까지 절감하는 동시에, 기준선 대비 충돌 위험은 가장 낮게 유지합니다.

그래프 합성곱 네트워크(GCN)는 지능형 교통 시스템(ITS)에서 안전성, 효율성 및 교통 예측성을 향상시키기 위해 등장했으며, 이는 도로망의 차량 항법 서비스에 이점을 제공합니다. 도로망 교통량의 예측 가능성은 배송 서비스 요구 사항을 충족하는 데 핵심적입니다. 그러나 최적의 교통량 예측은 시간적, 공간적으로 복잡하며 끊임없이 변하는 환경에 의존하기 때문에 달성하기 어렵습니다. 본 연구는 도로망에서 효율적인 소포 배송을 위한 머신러닝 기반 자가 적응형 상호작용 내비게이션 도구(ML-SAINT)를 제안합니다. GCN이 소포 배송 요건을 충족하는 최적 경로 예측 능력을 분석합니다. ML-SAINT가 도로 그래프에서 연결된 차량의 최적 택배 배송 일정을 예측할 수 있도록, 빠르고 정확한 시공간 교통 정보 인식을 제안합니다. 시뮬레이션 결과, 제안된 기법은 배송 차량의 이동 지연 시간을 최대 44.19%까지 줄여 소포 목적지까지의 차량 이동 시간을 고려했을 때 기준 기법보다 우수한 성능을 보였습니다.

주제어: 지능형 교통 시스템, 컨텍스트 인식 탐색, 차량 네트워크, 충돌 회피, 그래프 합성 네트워크