Czech Technical University in Prague
Faculty of Transportation Sciences
Department of Applied Mathematics

Multi-Agent Systems for Highway Management

Dissertation thesis

Ing. Jana Kuklová

Ph.D. programme: Engineering Informatics
Branch of study: Engineering Informatics of Transportation and Communication
Supervisor: Prof. Ing. Ondřej Přibyl, Ph.D.

Prague, May 2022

Thesis Supervisor:
Prof. Ing. Ondřej Přibyl, Ph.D.
Department of Applied Mathematics
Faculty of Transportation Sciences
Czech Technical University in Prague
Na Florenci 25
110 00 Prague 1
Czech Republic

Declaration

I hereby submit for evaluation and defence the dissertation thesis elaborated at the CTU in Prague, Faculty of Transportation Sciences. I have no relevant reason against using this schoolwork in the sense of § 60 of Act No. 121/2000 Coll. on Copyright and Rights Related to Copyright and on Amendment to Certain Acts (the Copyright Act).

I declare I have accomplished my final thesis by myself and I have named all the sources used in accordance with the Guideline on ethical preparation of university final theses.

In Prague, May 2022

....

Ing. Jana Kuklová

Dedication

I would like to dedicate my dissertation thesis to my beloved father Jan Kukla who had always supported me until his untimely death.

Abstract

The use of a multi-agent approach for highway management is the principal topic of this dissertation. The work is divided into three main parts. The first part is purely theoretical and provides a general overview of the theory of Multi-Agent Systems. The overview is supplemented by a literature review focused on the use of multi-agent systems for highway management. Based on the theoretical background, the proposed architecture for agent-based highway management is presented in the end of this part. The second part describes the development of a simulation model in Anylogic. This model stems from the proposed architecture and is designed to test different approaches to the harmonization of traffic flow using variable speed limits. A total of three approaches were addressed in this work, namely manual centralized control, automated centralized control using decision trees, and automated decentralized agent-based control. For automated centralized control, a replication of the algorithm previously implemented on Czech highways within the INEP project was created. The control logic from this model was used to design the original agent-based algorithm, whose main goal was decentralization. Two extensions were further proposed for an agent-based control, benefiting from system decentralization. The goal of the first extension was to reduce the time aggregation of input data. The second extension generally aimed at implementing floating cars. All the mentioned control approaches were tested on the modeled section of the Prague Ring Road. The results of these simulations are presented in the third part of this work. The manual centralized control provided data for control validation, while the results for the automated centralized control were used as a reference for the newly designed agent-based algorithms. The results generally show that the extended agent-based algorithms result in positive effects, especially in terms of traffic safety.

Keywords: Multi-Agent Systems, Highway Management, Harmonization of Traffic Flow, Variable Speed Limits, Traffic Simulation.

Abstrakt

Hlavním tématem této dizertace je použití multiagentních systémů pro řízení dopravy na dálnicích. Práce je rozdělena do tří hlavních částí. První část je ryze teoretická a poskytuje obecný přehled teorie multiagentních systémů. Tento přehled je doplněn o rešerši zaměřenou na využití multiagentních systémů pro řízení dopravy na dálnicích. Na základě teoretických podkladů je v závěru této části představena navržená architektura pro agentní řízení dopravy na dálnicích. Ve druhé části je popsán vývoj simulačního modelu v prostředí Anylogic. Tento model vychází z navržené architektury a je určen pro testování různých přístupů k harmonizaci dopravního proudu pomocí proměnných rychlostních limitů. Celkem tři přístupy byly řešeny v rámci této práce, a to manuální centralizované řízení, automatické centralizované řízení pomocí rozhodovacích stromů a automatické decentralizované agentní řízení. Pro automatické centralizované řízení byl replikován algoritmus, který byl dříve implementován na českých dálnicích v rámci projektu INEP. Logika řízení z tohoto modelu byla využita pro návrh původního agentního algoritmu, jehož hlavním cílem byla decentralizace. Pro agentní řízení byla dále navržena dvě rozšíření využívající výhody decentralizace systému. Cílem prvního rozšíření bylo snížení časové agregace vstupních dat. Druhé rozšíření mělo obecně za cíl implementaci plovoucích vozidel. Všechny zmíněné způsoby řízení byly testovány na vymodelovaném úseku Silničního okruhu kolem Prahy. Výsledky těchto simulací jsou prezentovány ve třetí části této práce. Manuální centralizované řízení poskytlo data k validaci řízení, zatímco výsledky pro automatické centralizované řízení byly použity jako reference pro nově navržené agentní algoritmy. Výsledky obecně ukazují, že rozšířené agentní algoritmy přinášejí pozitivní účinky, zejména pak z hlediska bezpečnosti dopravního proudu.

Klíčová slova: Multiagentní systémy, Řízení dopravy na dálnicích, Harmonizace dopravního proudu, Proměnné rychlostní limity, Simulace dopravy.

Acknowledgements

I would like to acknowledge all the people who supported me during my doctoral studies, especially in the difficult time periods when I was facing health or family problems. I would like to express my gratitude to my dissertation thesis supervisor, prof. Ing. Ondřej Přibyl, Ph.D. who has guided me since 2014 and provided me with many useful consultations. I am no less grateful for the additional consultations provided by my colleagues from the Department of Applied Mathematics, namely Ing. Tomáš Třasák, Ph.D., Ing. Pavla Pecherková, Ph.D., Dr. Ing. Jan Přikryl, and Ing. André Maia Pereira.

I would like to express special thanks to B.A. Megan Pirunčíková, a native English speaker, who helped me revise the spelling and grammar in the text and provided useful language remarks. My greatest thanks then go to my family members for their infinite patience and care.

Finally, I also would like to acknowledge the Grant Agency of the Czech Technical University in Prague which partially funded my research within the project *Development of Advanced Algorithms for Harmonization of Highway Traffic Flow and Their Analysis*, grant No. SGS15/169/OHK2/2T/16.

List of Tables

List of Figures

List of Acronyms

ACL Agent Communication Language. 7
AID Automated Incident Detection. 30, 58
AOP Agent-Oriented Programming. 8, 9

BDI Belief-Desire-Intention. 7

CR Czech Republic. 1, 2, 10, 11, 17, 57

DAI Distributed Artificial Intelligence. 4

FIPA Foundation for Intelligent Physical Agents. 7, 9

GPS Global Positioning System. 30

HGV Heavy Goods Vehicle. 55, 57
HMS Highway Management System. 1, 2

JADE Java Agent DEvelopment framework. 9

KQML Knowledge Query and Manipulation Language. 7

MAS Multi-Agent System. 2–4, 7–10, 56, 57

OOP Object-Oriented Programming. 8, 9

RDS-TMC Radio Data System - Traffic Message Channel. 10

TCC Traffic Control Center. 17, 57

VMS Variable Message Sign. 1, 2, 10, 11, 13, 14, 16–18, 53–55, 57
VSL Variable Speed Limit. 1, 2, 17, 43, 48, 57

Nomenclature

Theory of Multi-Agent Systems (Chapter 2)

π	agent's policy
θ	agent's observation
a	action
i	index of action
j	index of state
s	state
t	time step
u	utility function
A	set of actions
S	set of states

Parameters of traffic flow for given time interval (Chapter 3)

LD	local density
Q_u	unit intensity of traffic flow
$\overline{V}$	mean speed

Contents

CHAPTER 1

Introduction

The highway network in the Czech Republic (CR) was originally designed for lower traffic flows than can be observed nowadays. The existing infrastructure is becoming insufficient and it is increasingly necessary to face the problem of traffic congestion which regularly develops in peak periods. There are two general approaches to this issue: the physical highway extension and widening represent the first approach, while the introduction of Highway Management Systems (HMSs) represents the second approach. In general, it can be said that the widening projects are less feasible for different reasons, such as the overall cost or environmental conditions.

HMSs consist of a set of strategies, system components, and technologies combined to monitor, control, and manage highway traffic more effectively [1, 2]. This work mainly concerns the strategy of speed harmonization, which is used to achieve a more uniform and stable traffic flow and to facilitate recovery from congestion by signs so-called Variable Speed Limits (VSLs). The strategy is thoroughly described in the deployment guidelines [3] published as a result of the European project EasyWay. It is realized through the infrastructure of Variable Message Signs (VMSs) which are mounted overhead every few kilometers on highways. VMSs allow, in addition to other signs, a dynamic display of speed limits.

Besides the mentioned deployment guidelines, the usage of VSLs for speed harmonization is documented in scientific publications. Whereas Khondaker et al. [4] provides a general overview of different approaches to VSLs in last two decades, Papageorgiou, Kosmatopoulos and Papamichail [5], and Weikl, Bogenberger and Bertini [6] focus more on the effects achieved by VSLs. Furthermore, Gu et al. [7] deal with the calibration of VSLs arising from the three-phase traffic theory proposed by Kerner [8]. A considerable number of case studies were performed in order to map out the benefits of VSLs from the safety point of view, their environmental benefits, drivers' compliance, or the consistency of displayed VSLs, e.g. [9, 10, 11, 12, 13, 14, 15, 16, 17].

Speed harmonization also forms a part of the HMS which has been developed in the CR. The first record of HMSs in the CR dates to the beginning of this millennium when new projects for highway modernization started to be discussed. Since then, three generations of speed harmonization have been determined [18, 19].

The first generation was developed by the company Eltodo EG a.s. in cooperation with the Faculty of Transportation Sciences, CTU Prague within the project INEP [20]. This project, occurring during the years 2008 – 2010, was financially supported by the Ministry of Transport of the CR. A modular architecture for HMS was designed (see Figure 1.1) and a rule-based algorithm for speed harmonization was devised. It is solved by a set of decision trees using bivalent logic. The knowledge and experience of traffic engineers has been embedded into the algorithm by means of the thresholds for decisions. The algorithm was evaluated with microsimulation models. The HMS defined in the project INEP is considered to be fully developed and it has been implemented on the Prague Ring Road and on a segment of the D1 highway which is equipped with control devices (detectors, VMSs etc.).

The second generation was also developed by the company Eltodo EG a.s. in cooperation with the Faculty of Transportation Sciences, CTU Prague. The common project SIRID [21], occurring during the years 2012 – 2015, was carried out as a project of TA CR (Program α). One of the objectives of this project was the changeover from bivalent logic to a fuzzy logic approach. Several fuzzy algorithms were developed as a natural extension of previously developed decision trees. They were tested and evaluated with microsimulation models. The changeover between the first two generations was documented in the article [22]. It describes the process where the mature algorithm from INEP was transferred to a prime fuzzy logic algorithm.

The third generation was initiated in 2015 within the project *Development of Advanced Algorithms for Harmonization of Highway Traffic Flow and Their Analysis*. One of the objectives of the project was the introduction of Multi-Agent Systems (MASs) to highway management in the CR. It was identified as a new research trend [18] and this dissertation builds upon the finished project.

The main aim of this dissertation was to develop an agent-based algorithm for traffic flow harmonization and to demonstrate that the agent approach has the potential to substitute and even improve the traditional rule-based approach. In order to achieve this, several partial goals were set: 1) To propose an architecture for the agent-based control for

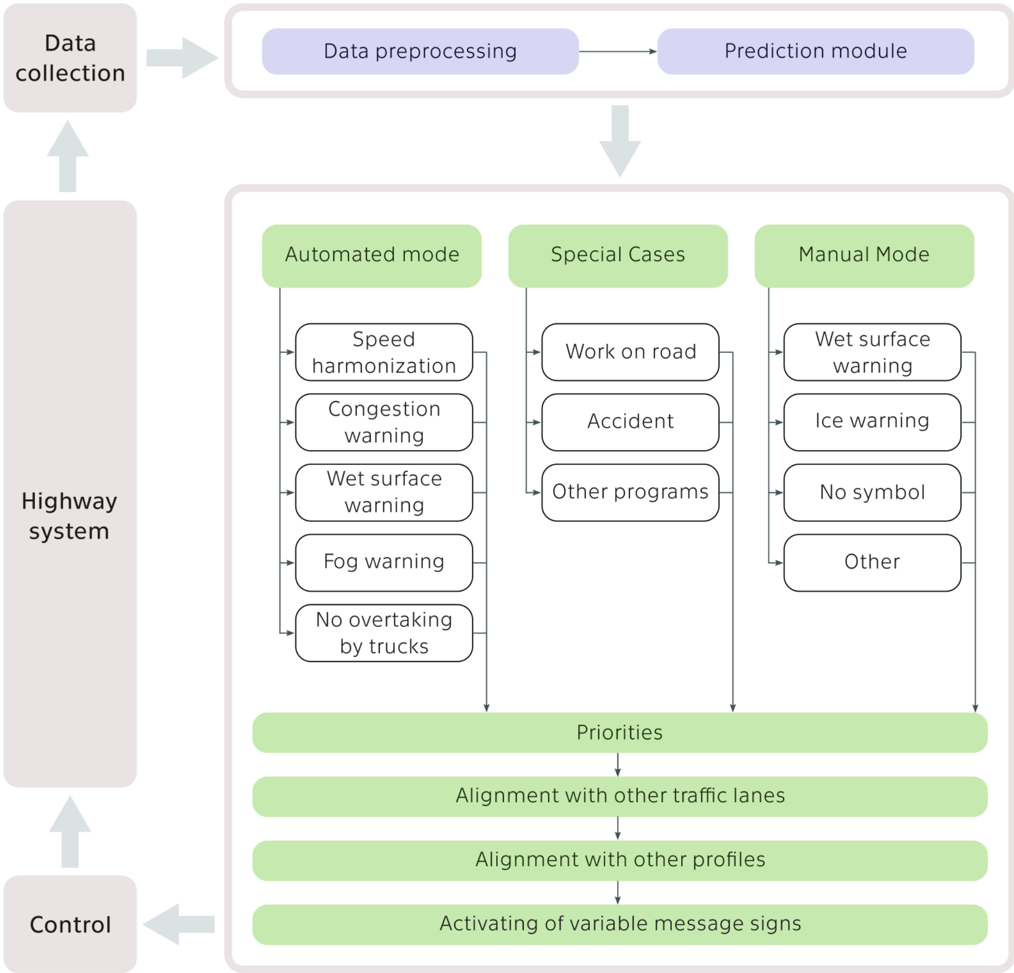

Figure 1.1: Modular architecture of INEP [19].

highway management; 2) To create a simulation environment for the testing of different approaches to highway management; 3) To replicate the rule-based algorithm from INEP and to develop a new agent-based algorithm; 4) To design, perform, and analyze a series of simulations in order to compare both approaches to speed harmonization.

This dissertation is divided into chapters according to the set goals. Chapter 2 provides a general theoretical background of MASs and a literature review on the usage of MASs for highway management. As a result, the last part of this chapter introduces the proposed architecture for the agent-based control for highway management. Chapter 3 presents Anylogic as a suitable software tool for the testing of different control approaches to highway management. This chapter then addresses the goals 2 and 3. Chapter 4 eventually deals with the simulations from design to evaluation. The obtained results are further discussed in Chapter 5.

CHAPTER 2

Multi-Agent Systems

The idea of MASs became topical in the eighties when the rapid growth of computer technology and computer networks enabled researchers to develop the field of Distributed Artificial Intelligence (DAI). It is a sub-field of Artificial Intelligence concerned with co-ordinated, concurrent action and problem solving [23]. Based on the historical context of its development, DAI has been divided into three more specific fields:

- Distributed Problem Solving;
- Multi-Agent Systems;
- Parallel Artificial Intelligence.

Since the eighties, MASs have been widely documented in many more or less rigorous publications. For the purpose of this work, references [24, 25, 26, 27, 28] were studied and a brief theoretical survey of MASs is provided.

2.1 Basic Concepts

A MAS is a system composed of multiple individual entities, intelligent agents, capable of independent actions. The agents are situated in an environment in which they interact together to coordinate their behavior and cooperate in order to achieve some collective goal. As a result, they solve problems that are beyond the individual capabilities or knowledge of each individual [24].

A block diagram of an agent and its environment is shown in Figure 2.1. The agent may perceive the environment (or its certain part) by means of sensors and it may act upon the environment by means of actuators. Between these two actions (perception and action), a decision has to be made by the agent. Usually, an agent has a pre-defined set of available actions and, thus, its key task is to select the action that should be performed in order to satisfy its delegated objectives. In general, there are several factors influencing the decision-making of an agent: the state of the environment, the way of perceiving the environment and its properties, the agent's architecture, interaction and cooperation with other agents, and the global goal. The mentioned factors are treated in this section.

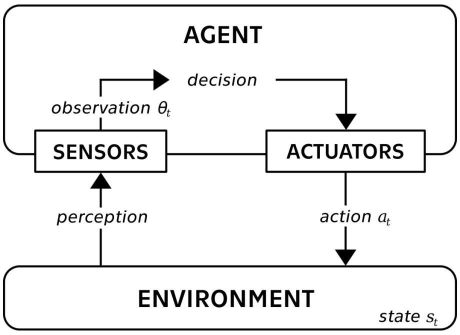

Figure 2.1: Agent and its environment.

In most applications, so-called computational agents are required. Such agents are explicitly designed for solving a particular task and are implemented on some computing device [28]. For this reason, a mathematical formulation should be introduced. A brief summary of the mathematical formulation treated in many publications, e.g. [24, 28], is provided in the following five steps:

Step 1: Determine global objectives and goals.

Step 2: Determine the agents and their roles in the global system.

Step 3: Determine the set of states of the world (environment) $S = \{s_j\}$.

Step 4: Define the agents' utility function $u : S \to \Re$.

Step 5: Define a set of actions $A = \{a_i\}$ arising from the agent observations θ_t.

The utility function from step 4 provides a map from the states of the world (environment) to a real number. Thereafter, each agent seeks to maximize this real number and, in that way, the utility function represents the agents' behavior.

Once all described steps are completed, the agent selects an action at the time step t, based on the history of observations θ_t and its previous actions

$$a_t = \pi(\theta_0, a_0, \theta_1, a_1, \cdots, \theta_t), \tag{2.1}$$

where the function π represents the policy of the agent. Such approach requires the complete history of observation-action pairs, which is, in most applications, impossible to store due to a very large number of observations in the past. For this reason, only the recent history of observation-action pairs is considered. In the simplest case, only the last observation θ_t is considered and Equation (2.1) takes the form $a_t = \pi(\theta_t)$.

If the problem is moved to the stochastic world, the expected utility $u(s_{t+1})$ is maximized and the optimal action a_t^* of the agent at state s_t can be expressed as

$$a_t^* = \arg\max_{a_t \in A} \sum_{s_{t+1}} p\left(s_{t+1} \mid s_t, a_t\right) u(s_{t+1}). \tag{2.2}$$

2.2 Environment

The properties of an environment can affect the selection of an agent action. There are several viewpoints to classify the environmental properties [24]:

Accessible/inaccessible: the agent can/cannot obtain complete and accurate information about the environmental state;

Deterministic/non-deterministic: the same task performed twice produces a single guaranteed result/may produce different results;

Episodic/non-episodic: the agent's performance is the result of a series of independent tasks/the agent has to consider the effect it may have on future tasks;

Static/dynamic: only the actions of an agent modify it/other processes are operating on it and it changes beyond the agent's control;

Discrete/continuous: there are/are not a fixed, finite number of actions.

It is obvious that the most complex environment will be inaccessible, non-deterministic, non-episodic, dynamic, and continuous. Such environment best describes most real environments.

2.3 Intelligent Agents and their Architectures

An intelligent agent is characterized by several features: it is autonomous, social, reactive, and proactive [24, 27]. Autonomy means that the agent is capable of operating without the direct intervention of humans or others and has control over its actions and its internal state. Social ability expresses the ability of the agent to cooperate with other agents or humans in order to achieve its goal. A reactive agent is able to perceive its environment and respond in a timely fashion to changes that occur in the environment. Finally, proactiveness means that the agent is able to exhibit goal-directed behavior by taking initiative, not only by acting in response to its environment.

An architecture proposes a particular methodology for building an autonomous agent. The architectures for intelligent agents can be divided into four groups:

- Logic-based (symbolic) architecture;
- Reactive architecture;
- Belief-Desire-Intention (BDI) architecture; and
- Layered (hybrid) architecture.

The main difference between the listed architectures lies in their method of decision-making. A logic-based agent performs an action based on logical deduction. It arises from traditional knowledge-based system techniques in which an environment is symbolically represented and manipulated using reasoning mechanisms. The decision making of a reactive agent is represented by a direct mapping from situation to action. It is based on a stimulus-response mechanism triggered by sensor data. The idea of a BDI agent comes from philosophy. The action of such an agent stems from the mental attitudes of belief, desire and intention. Whereas these three mentioned architectures represent either reactive or deliberative (logic-based and BDI) architecture, the layered architecture allows for both reactive and deliberative agent behavior. The decision-making of agents with such architecture is realized via various layers, and these layers process the environment at different levels of abstraction.

2.4 Communication

As the agents have to interact with each other, communication is the key component of MASs. Communication can be used either for coordination among cooperative agents or for negotiation among self-interested agents. For this purpose, an agent communication language is required. The first agent communication language was Knowledge Query and Manipulation Language (KQML), a language and protocol for exchanging information and knowledge, developed in the early nineties. The development of KQML was led by Finin and Weber [29, 30].

KQML has been superseded by Agent Communication Language (ACL) proposed by the Foundation for Intelligent Physical Agents (FIPA)[1]. This language is known under the abbreviation FIPA-ACL and its specification is available in [32]. It incorporates many aspects of KQML and, currently, is the most used and studied agent communication language [27].

[1]Foundation for Intelligent Physical Agents is an IEEE Computer Society standards organization that promotes agent-based technology and the interoperability of its standards with other technologies [31].

As mentioned above, multiple agents in MAS need to be coordinated. There are several reasons for it, among others [27]:

1. The agents' goals may cause conflicts among agents' actions;
2. The agents' goals may be interdependent;
3. The agents may have different capabilities and different knowledge; and
4. The agents' goals may be more rapidly achieved if different agents work on each of them.

There are several approaches to agent's coordination and communication. The coordination techniques, which are subsequently divided into four groups, can also be understood as the architectures of MASs:

Organizational structuring: based on traditional client-server architecture where a master agent controls and manages the slave agents in order to ensure global coherence;

Contracting: based on a decentralized market structure where agents can take on both the manager and contractor role;

Multi-agent planning: the coordination of the agents is treated as a planning problem, where it can be either centralized or distributed multi-agent planning;

Negotiation: the communication process of a group of agents leads to a mutually accepted agreement on some matter. It can be either competitive or cooperative.

Negotiation is the most important technique of MASs. Competitive negotiation is used in the event of independent goals among interacting agents, whereas cooperative negotiation is used when agents have a common goal or a single task to execute.

2.5 Development of Multi-Agent Systems

The concept of agents strongly resembles the objects from Object-Oriented Programming (OOP), since objects are defined as computational entities that encapsulate some state, are able to perform actions, or methods, in this state, and communicate by message passing [24]. Both agents and objects accept the principle of encapsulation and information hiding, and they communicate via message passing. On the other side, agents embody a stronger notion of autonomy than objects. Furthermore, the objects from OOP do not integrate the typical behavior of agents: reactiveness, proactiveness, and social ability. This led to the new programming paradigm: Agent-Oriented Programming (AOP).

The term Agent-Oriented Programming was coined by Shoham in 1989 [33] and it can be understood as a specialization or an extension of OOP. The new computational framework was proposed in [34]. The main differences between OOP and AOP are summarized in Table 2.1.

Table 2.1: Comparison of OOP and AOP [34].

Element	**OOP**	**AOP**
Basic unit	object	agent
Parameters defining state of basic unit	unconstrained	beliefs, commitments, capabilities, choices, ...
Process of computation	message passing and response methods	message passing and response methods
Types of message	unconstrained	inform, request, offer, offer, promise, decline, ...
Constraints on methods	none	honesty, consistency, ...

AOP models an application as a collection of agents that are characterized by autonomy, proactiveness and the ability to communicate. The architectural model of an agent-oriented application is intrinsically peer-to-peer, as any agent is able to initiate communication with any other agent or can be subject to an incoming communication at any time [27].

The most widespread agent-oriented middleware in use today is JADE, Java Agent DEvelopment framework. JADE, which is thoroughly described in [27], facilitates the development of complete agent-based applications. JADE is written completely in Java and is distributed as open source. With respect to interoperability, JADE is fully compliant with FIPA standards which define the reference model of an agent platform and a set of services that should be provided [35].

There are many agent platforms available these days. An up-to-date comparative review of the most promising existing agent platforms was performed by Kravari and Bassiliades [36] or more recently by Pal et al. [37]. For the purpose of this work, the multi-method simulation platform Anylogic [38] was selected. Chapter 3 deals with this platform in more detail.

2.6 Multiagent-Systems for Highway Management

Nowadays, MASs are used in a wide variety of applications including highway management. In 1997, Burmeister, Haddadi and Matylis [39] gave an overview of a diverse range of the potential applications of MASs in traffic and transportation. Among other applications, the control of traffic flow on highways is mentioned. In 2010, Chen and Cheng [40] provided a survey of real applications for agent technology in traffic and transportation or their proposals. Agent-based traffic control and management system architecture, and platforms developed by various research groups are presented in this work. One section is focused on agent-based systems for roadway transportation. A more recent review done by Přibyl, Koukol and Kuklová [41] deals with the computational intelligence in highway management. One part is dedicated to the usage of distributed intelligence for highway management. The authors concluded that most reviewed studies were solved only on the architectural level and the transfer of proposed solutions into practice is, currently, the most challenging issue. Nevertheless, the newest trends in this domain tend toward the implementation of active traffic management through autonomous and/or cooperative vehicles [42, 43].

As mentioned in Chapter 1, the MAS approach represents a novel research trend in highway management in the CR. Based on the research review [41], several advantageous reasons for introducing MASs can be listed. MAS has a modular architecture concept which corresponds to the framework of highway management introduced in the CR, particularly within the INEP project (see Figure 1.1). Moreover, it allows to introduce various highway management strategies gradually and it can thus be extended in the future. MAS is decentralized, which means that it is able to face the single point failure problem associated with centralized systems. The nature of highway management is distributed, as the gantries together with detectors and VMSs are distributed, along highways which are hundreds of kilometers long. In addition, a highway system consists of heterogeneous entities. The state of a highway is described by traffic flow detectors, meteo-stations, floating cars, etc. At the same time, drivers are influenced by means of informational messages, warning signs (work zone or traffic jam ahead), or restrictive signs (speed limits). Such information may be either displayed by VMSs or sent via RDS-TMC or transferred to the drivers by other information channels.

Considering a highway as an environment, it can be classified as an inaccessible, non-deterministic, non-episodic, dynamic, and continuous environment [24]. It is inaccessible, since the agents in highway management will obtain only that information which is measured by detectors and, in some cases, information from human operators. Such information can never be complete and fully accurate. As the highway represents the real physical world, the same agent's actions will not have a single guaranteed effect which, moreover, has to be considered with respect to future tasks. Thus, the highway will be treated as a non-deterministic and non-episodic environment. It is also dynamic, since agents have a certain impact on drivers' behavior and, thus, an impact on the overall highway traffic flow. Finally, the nature of a highway is conclusively continuous and the data from the highway detectors are gathered continuously.

A general description of an agent in highway management can be provided in the background of the general block diagram from Figure 2.1. The environment corresponds to a highway section and its associated traffic flow. The state s_t corresponds to the state of a given highway section in time t. The state s_t includes both traffic flow parameters (average speed, intensity, etc.) and the real state of the highway section (weather conditions, restrictions on the highway, etc.). The sensors either correspond to the real detectors (inductive loops, meteo-stations, cameras, etc.) or are represented by floating cars or VMSs, which provide data about their current state. The agent's actions a_t, which are intended to influence the environment, are represented by messages on VMSs (speed limit, warning signs, re-routing instructions, etc.).

2.7 Proposed Architecture

The proposed architecture of the agent-based highway management in the CR is shown in Figure 2.2. This proposal arises from a comprehensive study of the architectures designed abroad, which is documented in [41].

According to Hernández et al. [44], traffic management architectures are composed of three levels (from bottom to top):

- Sensor & Actuator level;
- Data level; and
- Knowledge level.

Inductive loops, cameras, meteo-stations, floating cars, VMSs, RDS-TMC, and other measuring and actuating devices belong to the sensor/actuator level in which signal processing is performed. Data captured at the first layer are processed at the data layer. The knowledge layer then provides the control actions and sends them to the actuator level. In this proposal, all three layers are solved by agents. While the data capturing and processing is performed by simple reactive agents, the control actions represent the result of intelligent agents' negotiation. Thus, the knowledge is embedded into intelligent agents. The knowledge level is supplemented by the operators at the traffic control center. Their main task is to supervise the automated control and, if necessary, they can also intervene in the automated control via corresponding agents.

As shown in Figure 2.2, the control strategies are solved at the horizontal level, while the scenario measures are solved at the vertical level [44, 45]. According to the number of gantries, the highway is divided into sections. Each highway section has an associated set of agents. The agents at the sensor layer capture traffic data and send them to the agents at the data layer. At the same time, these data are monitored by the operators

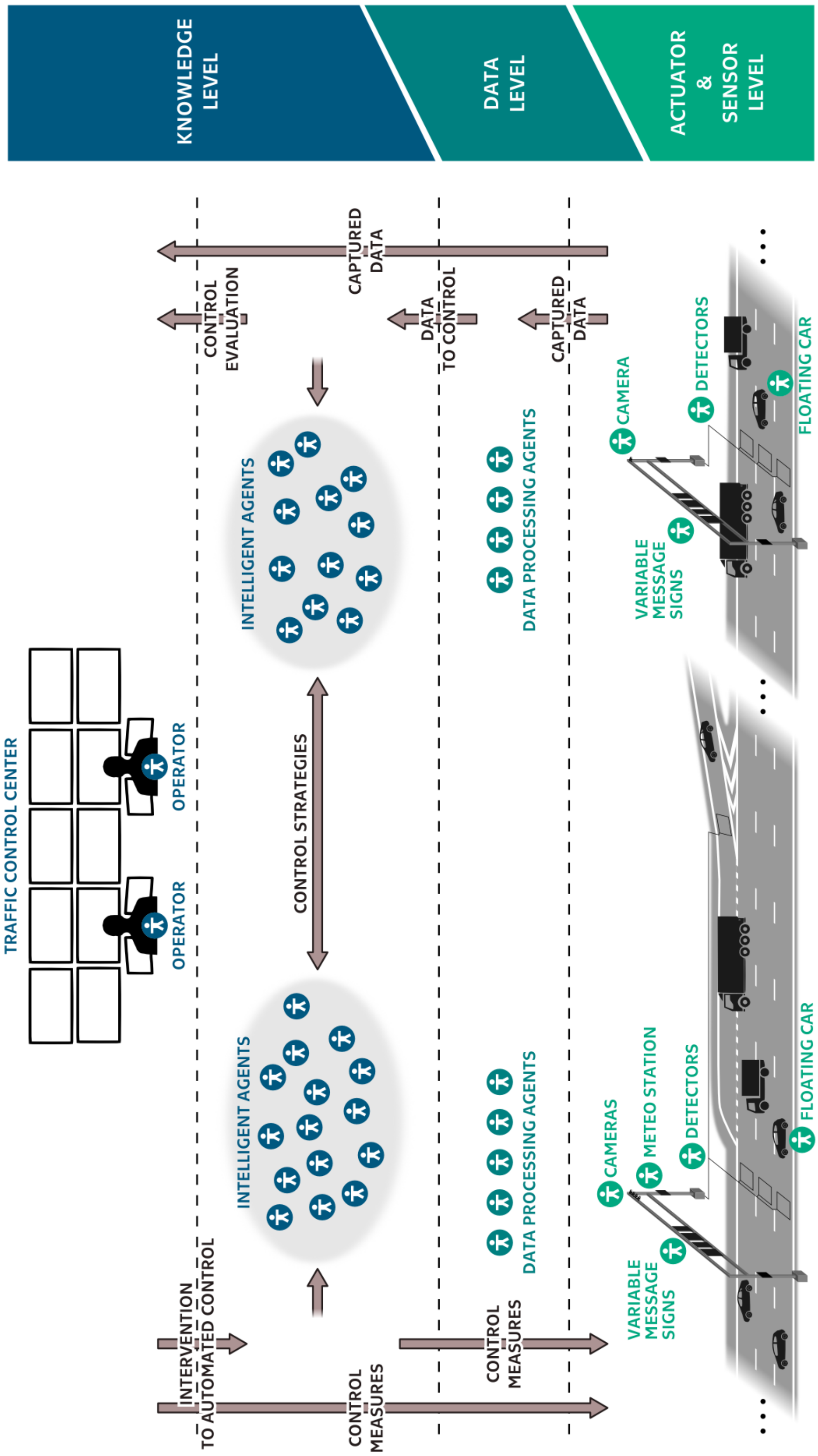

Figure 2.2: Proposed architecture of agent-based highway management.

at the traffic control center. Each sensor provides a different type of data, thus, the data processing is divided into several agents. The processed traffic flow data are subsequently provided to the intelligent agents. Each agent selects only those data which are required for its autonomous tasks. The intelligent agents associated with one gantry communicate together in order to find a control measure for a given section. At the same time, they have to communicate with the agents from downstream and upstream sections in order to comply with control strategies. Once the agents find the most appropriate control measure, they send it to the actuator level and it is then transmitted to the drivers by means of VMSs. At any time, operators can intervene in such automated control. They can temporarily switch off the automated control and directly send the control actions to the actuator level.

CHAPTER 3

Highway Management in Anylogic

Anylogic [38] is a multi-method simulation tool with an intuitive graphical user interface. It is completely written in Java and enables to combine process modeling, system dynamics, and agent based modeling in one model. Moreover, it has a built-in Road Traffic Library. On the one hand, Anylogic seems to be a perfect tool for this research since a microsimulation model of traffic flow can be created directly in Anylogic. On the other hand, several difficulties had to be overcome when modeling a highway. As an alternative, traditional microsimulation instruments, such as Aimsun or Vissim, could be used. However, the usage of these tools would require to develop an additional interface for testing different approaches to the control. Hence, Anylogic was eventually selected.

Anylogic developers provide some tutorials and example models which are available in Anylogic Cloud [46]. There is one section focused especially on traffic models. Although the Anylogic web-page declares that the software is suitable for the modeling of highway traffic, there are not many highway examples. The most inspiring and useful model created by Anylogic developers is Highway Junction [47]. Besides Anylogic developers, Benčat and Janota [48] provide their experience with Road Traffic Library in Anylogic. However, they focus on the modeling of traffic flow in urban areas.

For the purpose of this work, a model of Prague Ring Road was required, more precisely, a model of one-direction segment from Ruzyně (km 22.5) to Lochkov tunnel (km 13.5). Prague Ring Road is a highway equipped with detectors and VMSs which always form detector-gantry pairs. The selected segment consists of five entrance ramps, four exit ramps, and five gantries equipped by cameras, inductive loops and VMSs (see Figure 3.1).

Figure 3.1: Modeled highway section with the location of gantries. Map source: Mapy.cz

The development of the highway model in Anylogic, including all mentioned elements, is described in Appendix A. The first part deals with the physical and logical levels of highway infrastructure. The second part is dedicated to the traffic flow settings, its behavior and its interaction with the infrastructure. The mentioned appendix can be considered an Anylogic tutorial for the development of highway models intended for highway management, since the description is general enough [49].

This chapter is focused on the highway management itself. Figure 3.2 shows the entire schema of the Anylogic model which is proposed as a framework model for the testing of different control approaches to highway management [50]. It was designed in accordance with the architecture in Figure 2.2 and with the formulation of a multi-agent framework presented by Monteil et al. [42]. It should be understood that the schema primarily shows the processes at one gantry-detector pair. All the displayed agents represent one instance of an population living in the model. Figure 3.2 then shows the custom types of those populations.

The agent-based microsimulation model of traffic flow on a highway represents the lowest level of the model. Inductive loops in the path, modeled as pairs of stop lines, are considered agents at the sensor level. Such agents have assigned their data processing agents of the type *Detector* and continuously send them raw data. *Detector* works at the data level and performs data processing. It prepares both required input for control algorithms and data for further evaluation of traffic flow.

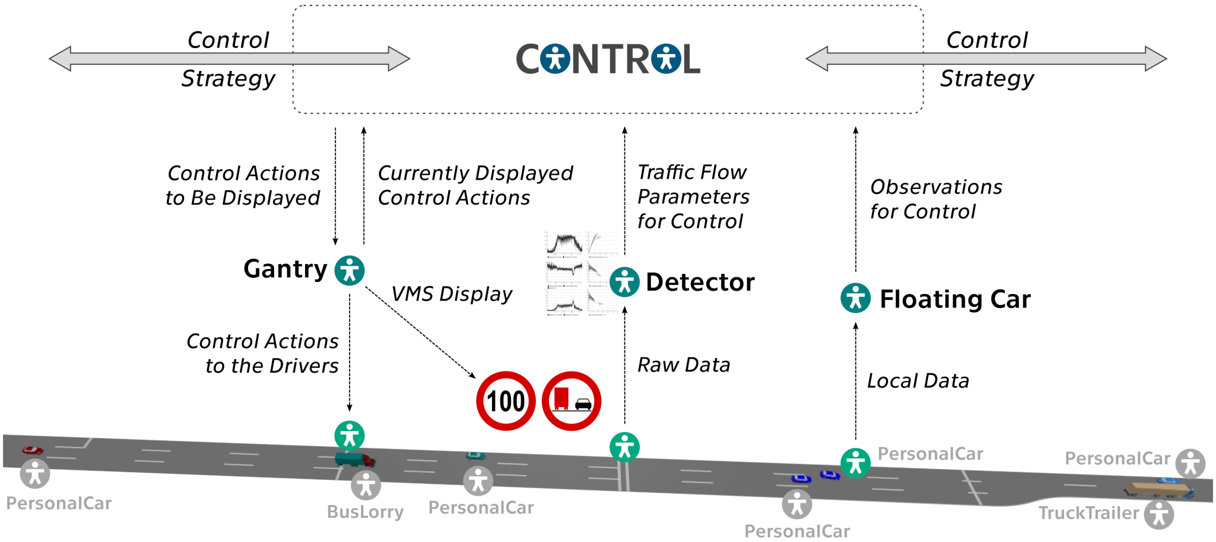

Figure 3.2: Schema of Anylogic model.

The traffic flow mapping by inductive loops can be reinforced by floating cars. In such a case, the vehicles themselves are agents at the sensor level. Each vehicle representing a floating car has assigned its agent of the type *FloatingCar* and continuously sends it local data. *FloatingCar* works at the data level and transforms the received data to required input for control algorithms.

Gantry is the last agent type working at the data level. It receives control measures from control algorithms and transforms them into an understandable format for vehicles. The control measures are then transmitted to all vehicles passing the corresponding stop line, which can be considered an agent at the actuator level. The other role of *Gantry* is to display the appropriate VMSs at a specified time and to provide control algorithms with the current control measures. This feedback is introduced to the model since some control approaches can be solved by several agents for which the final control measure is out of their scope.

The block CONTROL in Figure 3.2 corresponds to the knowledge level in Figure 2.2. It can be solved by any number of agents depending on the selected control approach. Since Anylogic is a multi-method simulation tool, the block CONTROL with horizontal control strategies can represent both a centralized and a decentralized control system. Moreover, it can be solved either by traditional tools, such as decision trees, or by negotiating among intelligent agents. The next section deals with the control approaches in more detail. It is assumed that the data from detectors, floating cars and gantries are available as required by CONTROL and, at the same time, CONTROL returns the output corresponding to control actions.

It should be emphasized that the control actions have to follow certain rules. In the case of Prague Ring Road, such rules are required by the Road and Motorway Directorate of the CR and are summarized in [51, 52]. Czech highways have a speed limit of 130 km/h. The speed limit can be modified by VSLs to 60, 80, 100, or 120 km/h. In order to maintain the safety of traffic flow the VSLs have to be synchronized in time and space. In a simplified form, any driver should not experience the decrease of speed limit higher than 30 km/h between two adjacent gantries.

The following sections of this chapter deal with the different control approaches which were implemented into the Anylogic model. In other words, it is focused on the solution of the block CONTROL and control strategies in Figure 3.2. Three general approaches can be delimited within this work:

- Manual control,
- Control by decision trees,
- Agent-based control.

3.1 Manual Control by Operator

The manual control represents a centralized control approach. It simulates an intervention from the Traffic Control Center (TCC) where operators can observe traffic flow through cameras placed at gantries. In the real world, it can be used in case of a sudden traffic excess which is noticed by an operator before it is detected by an automated algorithm. In this work, this control approach was implemented in the model in order to perform the model validation. The response of traffic flow on particular control measures was tested.

Since the control is fully centralized, TCC is simulated at the main agent of the model. A user can play the role of an operator: to observe the highway traffic flow at gantries and to intervene by setting the pre-defined VMSs as desired. Figure 3.3 shows a simple interface for the manual control. Each column represents one gantry. A set of buttons enable users to manually set five speed limits from the set $\{130, 120, 100, 80, 60\}$ km/h, considering that the speed limit of 130 km/h induces the speed limit sign to switch off.

The other two buttons are intended to switch on/off the sign end of all limits. At the bottom of Figure 3.3, the corresponding camera views with currently displayed VMSs are shown. They are supplemented by the instantaneous speed of the last vehicle which passed the gantry.

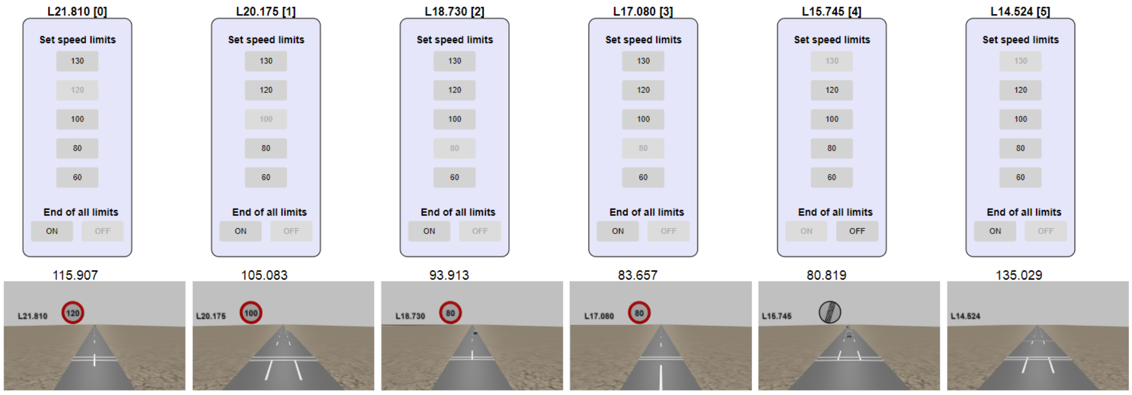

Figure 3.3: Control interface for operators.

Table 3.1 provides an example of possible settings of VMSs in the modeled highway section. It corresponds to the settings displayed in Figure 3.3. It should be emphasized that there is no additional cross-check and the control measures rely only on the operator. Thus, there is a danger of pointless settings between each pair of following gantries.

Table 3.1: Example of VMS setting at gantries.

Gantry label	**Speed limit**
L21.810	120
L20.175	100
L18.730	80
L17.080	80
L15.745	130
L14.524	130

The interface was further extended with analogical buttons like those seen in Figure 3.3. However, they are intended for all gantries at once. They thus enable an operator to set the same VMSs at all gantries.

3.2 Control by Decision Trees

Decision trees were used in speed harmonization algorithms within the project INEP [20]. This control approach has been implemented at Czech highways. Thus, the speed harmonization algorithm was also implemented in the Anylogic model and it is further used as a reference approach to the newly developed speed harmonization algorithms.

INEP solves the speed harmonization in three levels which are thoroughly described in [51, 52]. The first level evaluates the traffic flow at each gantry and proposes limit signs for any given gantry. The remaining two levels respectively solve the synchronization of gantries and the so-called dynamical sequence. This approach can be considered as partially distributed. Whereas the first level is essentially distributed, the second and the third levels are fully centralized.

At the first level, there is one decision tree upon each detector-gantry pair. It determines a new speed limit using unit intensity, mean speed, local density, and the currently displayed speed limit as input. The algorithm arises from a set of decision tables published in [51]. The decision tables provide thresholds of unit intensity Q_u, mean speed $\overline{V}$, and local density LD for both the increase and the decrease of the speed limit. Table 3.2 provides a survey of thresholds and conditions for the speed limit decrease, considering that the lower the speed limit to be displayed, the higher the priority of the corresponding condition.

Table 3.2: Decision table for speed limit decrease [20].

General condition $Q_u \vee (\overline{V} \wedge LD)$			**Speed limit to be displayed**	**Preference order**
$\geq Q_u^{\text{ON 120}}$	-	-	120	4
$\geq Q_u^{\text{ON 100}}$	-	-	100	3
$\geq Q_u^{\text{ON 80}}$	$\leq \overline{V}^{\text{ ON 80}}$	$\geq LD^{\text{ ON 80}}$	80	2
-	$\leq \overline{V}^{\text{ ON 60}}$	$\geq LD^{\text{ ON 60}}$	60	1

The decision table for the speed limit increase is very similar to Table 3.2. In accordance with Papageorgiou, Kosmatopoulos and Papamichail [5], the decision tables meet the conditions to create the so-called hysteresis for the switching process. Each activation threshold is accompanied by a different deactivation threshold which should avoid switching oscillations caused by the nature of traffic flow.

The primary logic of the decision tables arises from the fundamental diagrams. In general, it can be observed that unit intensity controls the stable traffic flow, whereas the combination of mean speed and local density detects and controls the congested traffic flow. The values of all thresholds have to be further calibrated for a given highway and drivers' behavior. The mentioned decision tables result in a fairly complex decision tree (see Figure 3.4 for an illustration).

Since the first level is logically distributed, it is solved by a population of agents of custom type *DetermineLimitsDT*. The number of agents within this population corresponds to the number of detector-gantry pairs. Each agent is then associated with two respective agents of type *Detector* and *Gantry*. Every 5 minutes, *Detector* perform data processing,

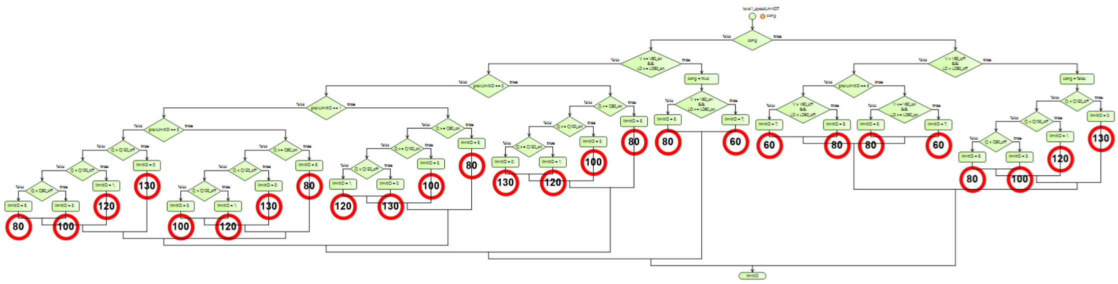

Figure 3.4: Decision tree for speed limit determination.

it sends data to *DetermineLimitsDT*, load the currently displayed sign from *Gantry*, and trigger the decision tree embedded within *DetermineLimitsDT*. This process is identical for the corresponding agents from all respective populations.

The second level performs the synchronization of speed limit signs. There is a set of rules defining which sequence of limits is allowed. Reference [52] describes the algorithm to achieve the synchronization according to the mentioned rules. It goes step by step through all gantries and performs synchronization with up to four upstream gantries. In general, it results in one decision tree with a for loop where the number of iterations corresponds to the number of gantries on a controlled highway decreased by one. The output is then a set of limits proposed for all gantries along the controlled highway section.

Since this level is fully centralized, it is solved by a single agent of custom type *SynchronizationDT* with one decision tree embedded within it. This is run once it receives the proposed limits from all the agents from the first-level population of the type *DetermineLimitsDT*. The chart in Figure 3.5 schematically displays the decision tree with the for loop.

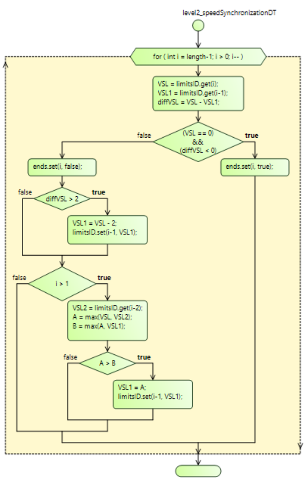

Figure 3.5: Decision tree for the synchronization of speed limits.

The third level performs the dynamical sequence which controls the speed limit sequence in time and space. It is thoroughly described in [52]. Unlike the second level, this algorithm considers the limit change dynamics. It does not allow any driver complying with speed limits to experience a speed limit decrease of more than 30 km/h between two following gantries. This algorithm can result in the delay of a speed limit display which occurs only if a sudden and significant decrease of speed limit is required. The algorithm goes step by step through all gantries and always sets the dynamical synchronization of up to two upstream gantries.

The algorithm generally results in a decision tree with one for loop in which the number of iterations corresponds to the number of gantries on a controlled highway decreased by one. It uses currently displayed speed limits and newly required speed limits as input. Then, it returns a set of delays in seconds for all required speed limits, considering that zero-delay means that the required speed limit can be displayed immediately. Similarly to the second level, it is fully centralized, thus, it is solved by a single agent of custom type *SequenceDT* with one decision tree embedded within it (see Figure 3.6). This decision tree is triggered by the synchronization agent *SynchronizationDT* once the speed synchronization is performed. The agent *SequenceDT* eventually sends the final speed limits to all agents of type *Gantry* with the respective delay.

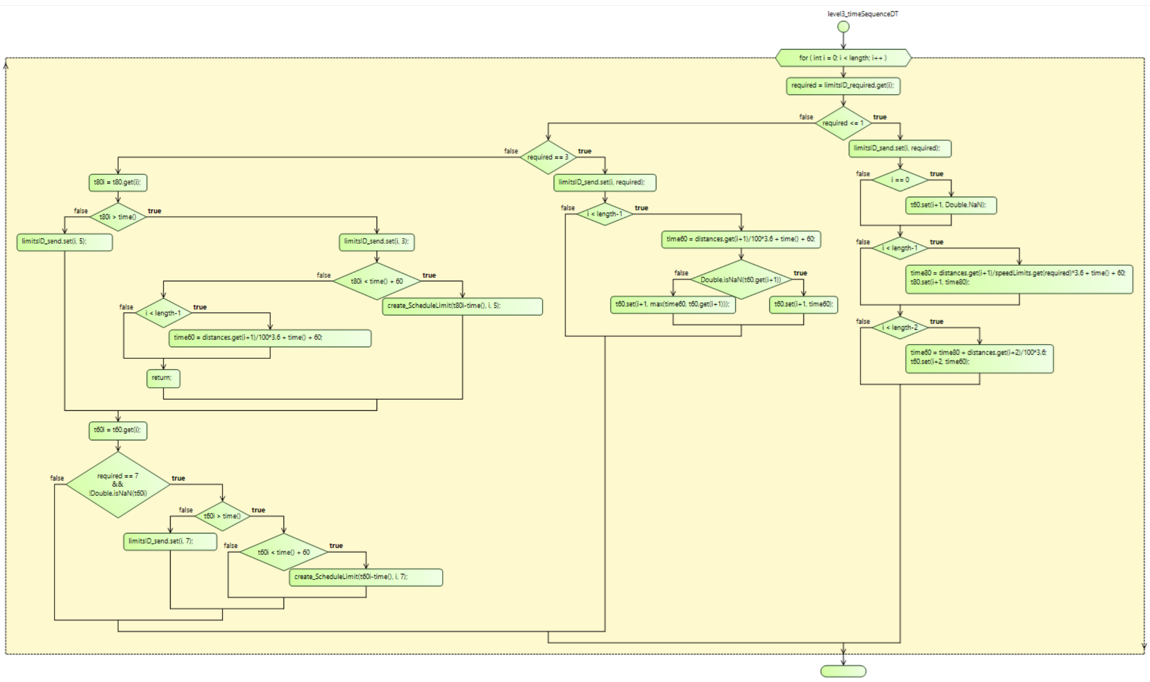

Figure 3.6: Decision tree for the dynamical sequence.

Figures 3.4-3.6 provide an illustration of the complexity of decision trees used in INEP. As mentioned above, they were implemented in the Anylogic model using one population of agents *DetermineLimitsDT* and two single agents *SynchronizationDT* and *SequenceDT*. The communication represented by data transmission is shown in Figure 3.7. Embedding this control solution to the framework model in Figure 3.2, the first-level population receives traffic flow data from the population of agents *Detector*, the third-level agent of the type *SequenceDT* provides limits in time to the population of agents of the type *Gantry*.

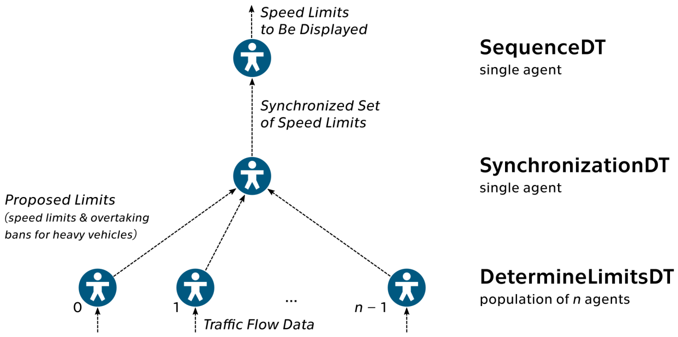

Figure 3.7: Agent communication schema for the INEP control.

3.3 Agent-Based Control

This section describes the original agent-based control developed within this dissertation. The logic of the traffic control arises from the decision tree approach used within INEP. The main goal of the new approach was to fully decentralize the control and thus facilitate a control extension and the implementation of new control rules.

For the agent-based control, five different agent custom types were developed. Their main role in the control system can be directly related to the decision trees from INEP as shown in Table 3.3. However, the communication among them is considerably different.

Table 3.3: Agent-based control related to the decision trees from INEP.

INEP	**Agent-based control**
1st level	Stability Agent StableTrafficAgent CongestionAgent
2nd level	SynchronizationAgent
3rd level	DisplayAgent

A population of agents was created for each custom type in Table 3.3. All these populations have as many instances as the number of detector-gantry pairs in the model. Thus, each gantry is always associated with one agent from each population. The indices of the instances within each population increase according to the traffic flow downstream, analogically to gantries and detectors.

In this section, the general functions of all agent types and communication among them are described. Unless otherwise stated, the communication among different agent types corresponds to the communication among the instances associated with one gantry-detector pair.

The overview of control agents for speed limit determination is shown in Figure 3.8. It represents a new control solution which can be embedded into the framework model in Figure 3.2. It should be emphasized that there is a certain difference in communication representation between Figures 3.7 and 3.8. Whereas the arrows in Figure 3.7 indicate data transmission which is always performed, the arrows in Figure 3.8 indicate only possible communication channels which do not have to be used. Moreover, traffic flow data from the population of agents *Detector* are not automatically sent to any agent as in the decision-tree approach. Here, traffic flow data can be actively loaded by any agent if required. Similarly, the currently displayed speed limits can be loaded from agents *Gantry*.

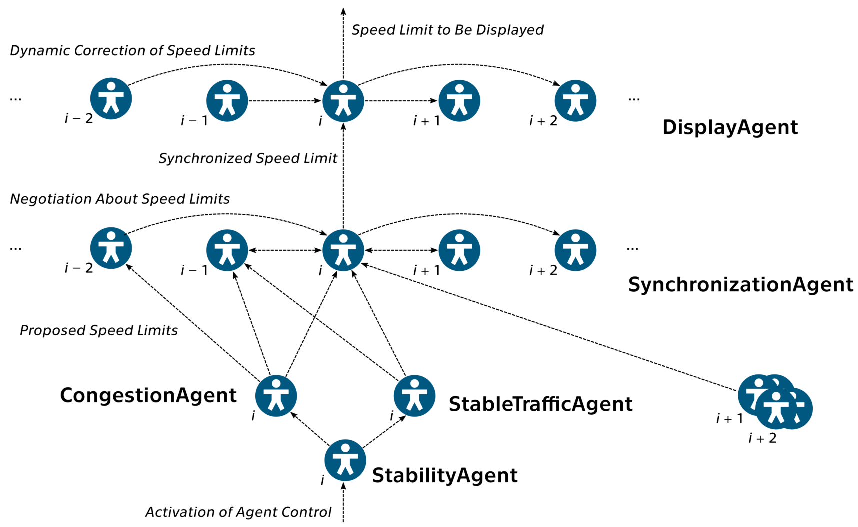

Figure 3.8: Agent communication schema for the agent-based control.

A trinity of agent types was designed to solve the first-level determination of speed limits. As mentioned above, it was observed in INEP that unit intensity controls the stable traffic flow, whereas the combination of mean speed and local density detect and control the congested traffic flow. Moreover, the conditions for lower speed limits, and thus for congested traffic flow, generally have higher priority. All these aspects are covered by *StabilityAgent*, *StableTrafficAgent*, and *CongestionAgent*.

First, *StabilityAgent* is activated by a message from *Detector* and it tests whether traffic flow is stable or not. Figure 3.9 shows the state diagram of this agent type. The message received from *Detector* induces the decision whether it remains in the same state or not. When an agent enters to a stable or congested state, it respectively activates *StableTrafficAgent* or *CongestionAgent* by sending a message. Then, one of these two agents proposes a speed limit for its corresponding gantry using the decision tables from INEP.

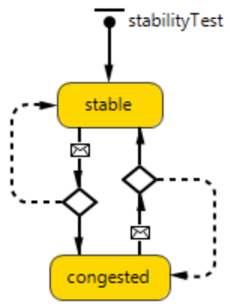

Figure 3.9: State diagram of *StabilityAgent*.

Table 3.4 provides an overview of the first-level agents. Inputs are the input data which are needed for their decision making. Thus, they actively load such data from *Detector* and *Gantry* once they are activated. Outputs in the case of *StabilityAgent* are the possible agents' states. Otherwise, the outputs are the sets of possible speed limit proposals.

Table 3.4: Basic characteristics of the first-level agent types.

Agent type	**Inputs**	**Outputs**
StabilityAgent	mean speed local density	"stable", "congested"
StableTrafficAgent	current speed limit unit intensity	130, 120, 100, 80
CongestionAgent	current speed limit mean speed local density	80, 60

The described trinity of agents operate upon each detector-gantry pair independently as the first-level decision tree from INEP. Both approaches provide an identical result, which is a set of speed limits proposed for each gantry. The only difference between them is that the agent solution distributes the conditions from one decision tree among three agents. It is believed that the agent-based solution is a more transparent and intuitive solution for users.

Whereas the first-level determination of speed limits is very similar in both approaches, the remaining levels differ considerably since they are completely centralized in INEP. The synchronization of speed limits and the dynamical sequence are respectively performed by the populations of agent types *SynchronizationAgent* and *DisplayAgent*.

In general, three main phases were identified in the synchronization process. First, the speed limits of 100, 80 and 60 km/h at one gantry impose maximum allowed speed limits at up to two upstream gantries in order to achieve a gradual decrease of speed limits downstream. Second, it is observed whether the speed limit at one gantry is higher than both speed limits at neighboring gantries. In such a case the speed limit of the middle gantry is decreased to the higher neighboring speed limit. Third, the speed limit of 130 km/h is decreased to 120 km/h in the event that less than three consecutive gantries propose this speed limit.

The mentioned rules were implemented by *SynchronizationAgent* (see Figure 3.10). At its first state *synchronizationProcess*, the agent is awaiting the speed limit proposals and maximum allowed speed limits from the first-level agents. Each instance of *StableTrafficAgent* and *CongestionAgent*, which is entrusted with the determination of the speed limit, sends its resulting proposal to its corresponding respectively to the first downstream instance of *SynchronizationAgent*. Moreover, it imposes speed maximums to the upstream synchronization agent(s) according to its own speed limit proposal. In general, the first state *synchronizationProcess* is designed to collect the outputs from the first-level agents.

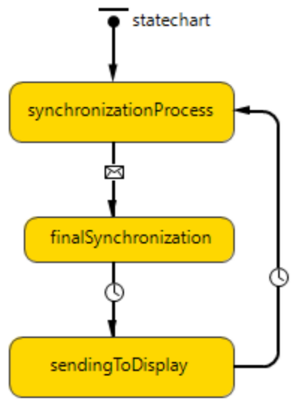

Figure 3.10: State diagram of *SynchronizationAgent*.

Once all the proposals and maximum allowed speed limits are sent, all the synchronization agents receive a message which induces their transition to the second state *finalSynchronization*. In other words, the message represents an impulse to start to negotiate with neighboring agents using the information obtained in the previous state. In some special cases, two downstream agents are involved to the negotiation. The negotiation is performed by a sequence of simple rules which leads to the same synchronization as in the centralized decision-tree solution. Then, each instance of *SynchronizationAgent* suggests one speed limit as a result of the negotiation. After a short time interval (less than 0.1 s), which ensures that the negotiation among all instances is finished, the agent is automatically switched to the third state *sendingToDisplay*. At the entrance to this state, the synchronization process is finalized by sending an activation message with the suggested speed limit to *DisplayAgent*. After that, it immediately switches to the first state and waits for the following proposals.

The state diagram of *DisplayAgent* is shown in Figure 3.11. The states of this agent reflect all possibilities in the speed limit display. In the event that any change in the state diagram of *DisplayAgent* occurs, it automatically sends the corresponding speed limit to *Gantry* to display it.

The first composed state corresponds to the situation when either no limit or the sign end of all limits is displayed. All inner states lead to the identical control action which is the speed limit of 130 km/h. The next two simple states are active in the cases that the speed limits of 120 or 100 km/h are displayed. The last two composed states correspond to the situation when the speed limits of 80 or 60 km/h are required. However, their display can be delayed by the dynamical sequence. The inner states represent the allowed speed limits. For instance, the state *display100to80* means that the speed limit of 80 km/h is required. Nevertheless, the speed limit of 100 km/h is displayed since it is currently the maximum allowed speed limit.

The dynamical sequence itself is solved within the population of agents *DisplayAgent*. Each agent is in the state which corresponds to the currently displayed speed limit. In the case of the states *noControl*, *display120*, and *display100*, it reacts to the activation message from *SynchronizationAgent*. It contacts up to two downstream agents as shown in Figure 3.8. It communicates to them the earliest moments of time when they are allowed to display the speed limits of 80 and 60 km/h. This time is calculated from the distance between the respective gantries and the currently displayed speed limit.

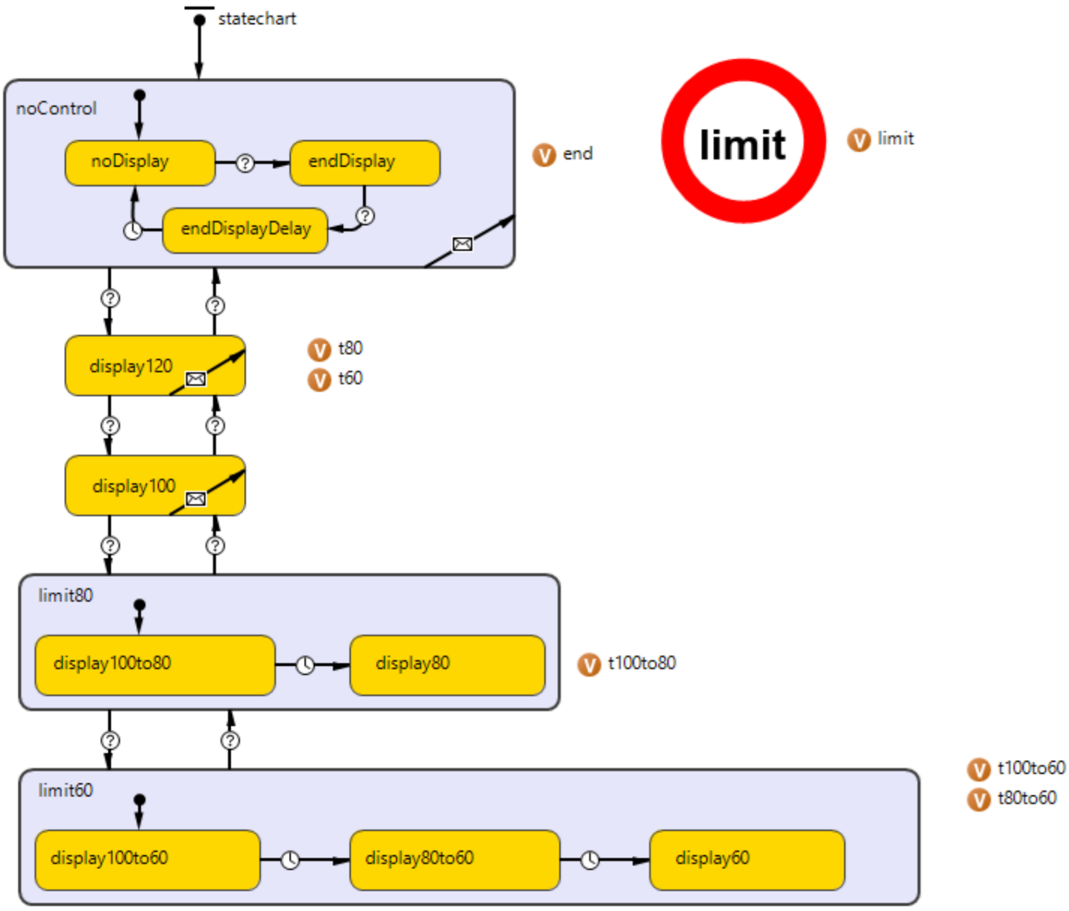

Figure 3.11: State chart of *DisplayAgent*.

Independently on the received activation message, each agent changes its state according to the speed limit received from *SynchronizationAgent*. In the case of the last two composed states, the inner transitions are controlled by the time allowances from the upstream agents.

The transitions within the first composed state for no control are given by conditions which arise from an additional communication with the nearest upstream agent. The state *endDisplay* corresponds to the display of the sign end of all limits. The transition to this state is given by the condition that no control measure is required at this gantry and any speed limit is displayed at the nearest upstream gantry. Then, in the event that none of the communicating agents display any speed limit, the condition for the sign's switch-off is fulfilled. If an agent is in the state *endDisplay* and the condition for the switch-off is met, the agent is first switched to the state *endDisplayDelay* where a time delay is applied. The sign end of all limits is still displayed until the potentially slowest vehicles arrive from the upstream gantry.

3.4 Modified Agent-Based Control

The logic of the agent-based control presented in the previous section is identical with the logic used in INEP. It means that both algorithms result in the same effects on the traffic flow behavior. In this section, an extension of the model was proposed. It should demonstrate that in comparison with the decision tree approach, the agent approach enables us to extend the algorithm without major interventions to the original solution.

3.4.1 Reduction of the Input Data Aggregation Interval

The present decision tree algorithm (INEP) uses three main tools to face the oscillation of speed limits: five-minute input data aggregations, exponential smoothing of input data, and the hysteresis for activation and deactivation thresholds. All three tools were described in the literature, e.g. [53, 54, 5], as well as in the description of the algorithm [51, 52].

The five-minute input data aggregations were identified as the greatest weakness of the INEP algorithm. Figure 3.12 provides a time plot displaying mean speed in afternoon peak hours. The exponentially smoothed five-minute data aggregations (used in INEP) are compared to the one-minute data without smoothing. The regular fluctuations in data are reduced as expected [54]. However, the data disturbances observed between 1:00 PM and 2:00 PM are almost entirely neglected and a considerable delay is visible on the rapid data changes.

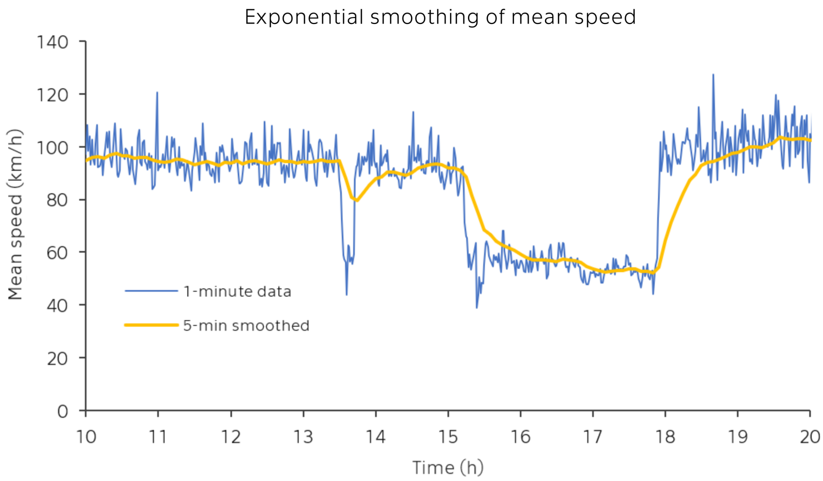

Figure 3.12: Comparison of smoothed five-minute data aggregations to minute data.

According to Guo, Smith and Williams [53], different smoothed data aggregations were displayed in one graph. Figure 3.13 compares smoothed one-, three-, and five-minute data aggregations with minute data without smoothing. The delay in the reaction on rapid data changes decreases proportionally to the aggregated time intervals. However, the regular data fluctuations in data are not reduced sufficiently for the one-minute aggregations.

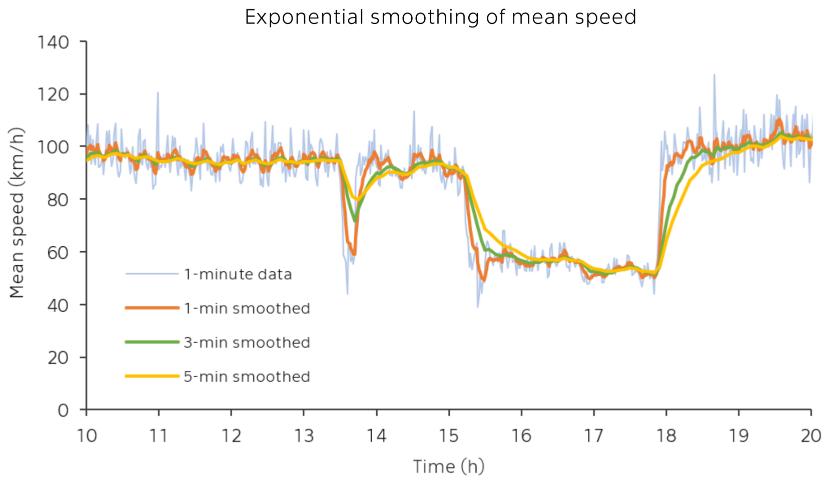

Figure 3.13: Exponential smoothing applied to different data aggregations.

Based on the previous analysis, it was decided to use three- instead of five-minute aggregations. On one hand, it resulted in the most suitable reaction of the algorithm on the traffic flow, especially at the transitions between stable and unstable traffic flow. On the other hand, a considerably higher oscillation of speed limits was observed as expected. For this reason, a new agent type, *PenalizationAgent*, was developed. As the name suggests, a penalization algorithm was implemented into the agent-based control through a population of this agent type. A similar algorithm was previously proposed for fuzzy-logic algorithms where the hysteresis for activation and deactivation thresholds could not be applied [22].

As shown in Figure 3.14, *PenalizationAgent* interacts with the agents at all three levels, considering that all of them are associated with the same gantry-detector pair. *DisplayAgent* informs *PenalizationAgent* about all changes sent to *Gantry*. By comparing this information with the limit requests from the first-level agents, *PenalizationAgent* can detect a possible oscillation in speed limits. It then intervenes in the negotiation at the level of synchronization and the oscillation is thus prevented. For this algorithm, the time interval after which a backward change in speed limits is not considered an oscillation has to be determined. In the case of the tested algorithm the interval of 6 minutes was selected.

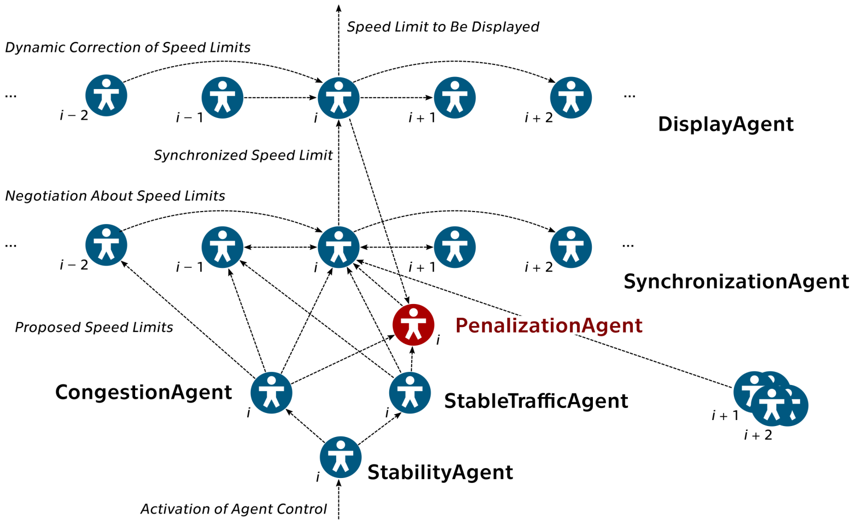

Figure 3.14: Implementation of *PenalizationAgent* into the agent-based control.

3.4.2 Implementation of Floating Cars

The speed harmonization algorithms (like INEP) are generally designed to face regular congestion which is usually developed upstream from common highway bottlenecks such as on- and off-ramps [55]. The effectiveness of these algorithms is then dependent on the distribution of detectors along the highway [56]. It can be anticipated that the effectiveness of such algorithms decreases in the case of irregular congestion which is mostly caused by an accident at a less expected location. This can be faced by an extension of the speed harmonization algorithm.

A wide number of the algorithms for Automated Incident Detection (AID) using detector data were developed [57, 58]. Nevertheless, the usage of floating car data has expanded recently due to technological progress [59]. For this reason, the implementation of floating cars into the agent-based control was proposed. The main objective of the implementation was to demonstrate that the present agent-based control using input data from detectors can be extended by another algorithm using floating car data as input.

It was supposed that a total highway standstill is hard to detect by inductive loops, since permanently covered (or uncovered) loops do not provide any meaningful data. Thus, in the case of a heavy accident, the effect of an additional algorithm detecting a standing queue would be considerable.

It is assumed that floating cars provide their GPS position and instantaneous speed [59]. For the purpose of the algorithm, the highway was divided into sections logically delimited by the detectors. The GPS data are then used to identify the section where the given floating car is located. This process is simplified in the model since GPS data are not available. The method for location determination, as well as the principles of modeling floating cars themselves are described in Appendix A. This section only deals with the algorithm for the detection of a standing queue.

According to Figure 3.2, *FloatingCar* works at the data level where data processing is performed. In other words, it should convert data obtained from a real floating car to an input for control agents. Nevertheless, the proposed *FloatingCar* also acts as an intelligent agent. It has proven advantageous that the agent can directly intervene in the control. Thus, it works at the data level, as well as at the knowledge level.

It should be emphasized that a population of agents of the type *FloatingCar* lives in the model, considering that each agent of this type is associated with one physical floating car. Figure 3.15 shows the state chart of *FloatingCar*. The first state *notOnHighway* is not interesting from the control point of view. Once the floating car enters the highway, it switches to the state *driving* within the composed state *onHighway_*. The transition to the state *trafficExcess* is induced by a detected instantaneous speed of 0 km/h. It is formally performed by sending a message within the model. The reversal transition is activated when the instantaneous speed exceeds 30 km/h. Until this occurs, *FloatingCar* observes its highway section every 20 seconds and tests whether an intervention in the control should be performed.

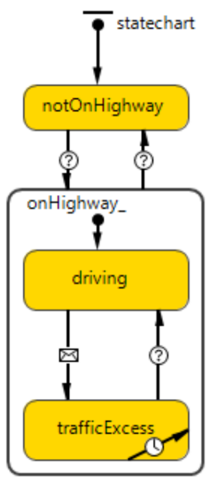

Figure 3.15: State chart of *FloatingCar*.

In the event that after the interval of 20 seconds its instantaneous speed is higher than 0 km/h, but lower than 30 km/h, an alert with a timestamp is sent to all agents of the same type in the same highway section. On the other side, if the instantaneous speed is still 0 km/h and another agent from the same section has sent an alert within the last 20 seconds, an intervention to the control is performed. Moreover, a new alert with a new timestamp is created and sent to all agents in the same highway section.

The intervention in the control consists of the activation of synchronization agents and the suppression of dynamical sequence. *FloatingCar*, according to the detected position, sends the request for a speed limit of 60 km/h, 80 km/h, and 100 km/h respectively to the first, the second, and the third nearest upstream agent of the type *SynchronizationAgent*. The limits are directly sent to be displayed without any further negotiation. Moreover, *FloatingCar* resets the delays for eventual dynamical sequence at the first two nearest upstream agents. The interaction of *FloatingCar* with other control agents is displayed in Figure 3.16.

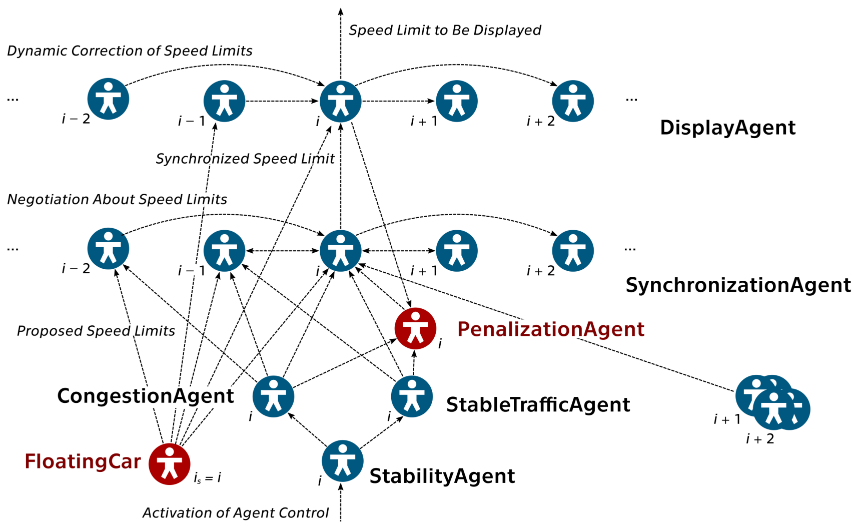

Figure 3.16: Implementation of *FloatingCar* into the agent-based control.

CHAPTER 4

Simulation Results

The main aim of the simulations was to demonstrate that the agent-based control has a potential to substitute and even improve the traditional centralized approach. It was performed by means of available simulation outputs which can be divided into four groups.

First, non-smoothed one-minute data of unit volume, average speed, speed variance within a one-minute interval, and local density from all six detectors were obtained for each simulation. In general, such data can be used to draw time plots and time-space plots for each traffic flow parameter, fundamental diagrams, and other dependency diagrams between mentioned characteristics.

Second, smoothed aggregated data of the same traffic flow parameters as for the one-minute data were also available. However, they were not used for analyses of simulation results. Since the control algorithms use such data as input, they were used for the calibration of automated control which, is out of the scope of this work.

Third, the logs of speed limits from all six gantries are recorded, considering that a speed limit of 130 km/h corresponds to an uncontrolled state. The logs were also used for the calibration and further validation of the automated control. Moreover, they were utilized for an analysis of traffic flow safety at the automated control. This analysis is presented further in this chapter.

Fourth, the total number of stops and lane changes of all vehicles were obtained for each simulation. Moreover, the position and time were also logged at each stop. While the number of lane changes was used as an indicator of traffic flow harmonization, the logs of stops were used to determine the length of a standing queue.

Fifth, travel times of personal and heavy vehicles were measured for each highway segment separately. Table 4.1 provides a survey of all ramp-exit pairs which represent the segments in the present model. Since the distance of each segment is known, travel speeds of personal and heavy vehicles could also be calculated. Basic statistics as the mean, median, minimum, and maximum of travel times and travel speeds for each vehicle class and highway segment were calculated and used for the further evaluation of traffic flow.

Table 4.1: Survey of highway segments.

Segment	Ramp-exit pairs	Distance (km)
1	K Barrandovu – Pod Lochkovem	0.82
2	Poncarova – Ořech	1.07
3	Pod Lochkovem – Tunnel Lochkov	1.11
4	Třebonická – Poncarova	1.50
5	Ruzyně – Poncarova	1.86
6	K Barrandovu – Tunnel Lochkov	2.19
7	Třebonická – Ořech	2.87
8	Ořech – K Barrandovu	2.93
9	Ruzyně – Ořech	3.23
10	Poncarova – K Barrandovu	4.13
11	Ořech – Pod Lochkovem	4.27
12	Poncarova – Pod Lochkovem	5.46
13	Ořech – Tunnel Lochkov	5.64
14	Třebonická – K Barrandovu	5.93
15	Ruzyně – K Barrandovu	6.29
16	Poncarova – Tunnel Lochkov	6.84
17	Třebonická – Pod Lochkovem	7.26
18	Ruzyně – Pod Lochkovem	7.63
19	Třebonická – Tunnel Lochkov	8.64
20	Ruzyně – Tunnel Lochkov	9.00

4.1 Overview of Scenarios

In total, sixteen scenarios were designed in order to gradually perform the validation of control actions, compare the effects of different approaches to automated control, and test the implementation of floating cars to the agent-based automated control. The scenarios can be generally divided into two groups: all-day scenarios and morning scenarios. These two types of scenarios were respectively designed to study regular and irregular congestion. Each scenario is then characterized by a control strategy.

4.1.1 All-Day Scenarios

The main focus of all-day scenarios is put on the morning and afternoon peak hours upstream from Exit 16 – K Barrandovu. This highway exit is critical since heavy vehicles are obliged to continue on the highway while most personal vehicles exit (see Figure 3.1). This off-ramp represents a bottleneck which can result in traffic breakdowns [55]. The upstream highway section is then prone to regular congestion. This is usually detected at the first upstream detector L17.080. In some cases, it can also be detected at the next upstream detectors, i.e. L18.730 and L20.175. Since regular congestion represents the main point of interest in all-day scenarios, the data obtained at detector L17.080 were selected for the analyses intended for individual detectors.

The all-day traffic flow was first subjected to one-limit control. Gradually, the speed limits of 130, 120, 100, 80, and 60 km/h were applied throughout the whole day. The scenario with the speed limit of 130 km/h is actually the uncontrolled scenario, since the speed limit is given by law. It was further used as a reference scenario. The other one-limit scenarios were performed in order to validate the speed limits. In other words, the effects of speed limits on traffic flow were studied. The results from one-limit scenarios were also used for the calibration of the thresholds at the automated control and for the analysis of traffic flow safety.

Consequently, three all-day scenarios with automated control were performed. First, the control by decision trees adopted from INEP (see section 3.2) was applied and the results were used as a reference for the agent-based control. Then, the traffic flow was subjected to the original agent-control and the modified agent-control without floating cars. They are respectively described in section 3.3 and subsection 3.4.1. Since the aim of the first agent-based control was to achieve identical control actions as with the decision tree approach, it was only used to verify it. The original agent-based control was then excluded from the presented results.

The last all-day scenario was performed in order to test the proposed algorithm executed by floating cars. The algorithm was designed to detect irregular congestion caused by a highway standstill. Thus, it should ignore regular congestion and the floating cars should not intervene in the automated control. Thus, it was supposed that even if all vehicles behaved as floating cars, a very low or no number of interventions would be observed.

4.1.2 Morning Scenarios

The implementation of floating cars was further studied in the morning scenarios. They are comprised of an irregular congestion caused by a modeled accident. The accident is scheduled at 8:00 AM in front of the entrance to the tunnel Lochkov and causes a highway standstill. It lasts 10 minutes and the simulation is finished at 9:00 AM when the traffic flow is recovered. The selection of this time and location assumes morning peak hours and a section which is not used to being directly affected by a regular congestion. It was previously observed that the elevated traffic volumes in this highway section did not induce speed limits lower than 120 km/h at the automated control. The accident is thus less expected by drivers.

This scenario with the accident was gradually subjected to a one-limit control with the speed limit of 130 km/h, the automated control solved by decision trees, and the modified agent control with different proportions of floating cars in traffic flow, i.e. 0 %, 10 %, 25 %, 50 %, and 100 %. It was supposed that the detection of the standstill would be hard without floating cars since the creation of the queue is abrupt in such a case. The detectors suddenly become either permanently covered or do not detect any passing vehicle. Neither of these two situations induce any control measure by the speed harmonization algorithms, which is convenient for the analysis of the implementation of floating cars.

4.1.3 Summary of Scenarios

Table 4.2 provides a summary of all sixteen scenarios. Each row corresponds to one control strategy applied at one or both types of scenarios. The numbers in the second and the third column only express the order in which the scenarios were performed and described in this section. These numbers are never used as a reference in this text. On the other hand, labels were assigned to each control strategy for the purpose of referencing them in further analyses (see the first column).

Table 4.2: Overview of scenarios.

Label of control	**All-day scenarios**	**Morning scenarios**	**A brief description of the control strategy**
Lim130	1	10	No control
Lim120	2	–	Permanent limit: 120 km/h
Lim100	3	–	Permanent limit: 100 km/h
Lim80	4	–	Permanent limit: 80 km/h
Lim60	5	–	Permanent limit: 60 km/h
INEP	6	11	Adopted decision tree control
AGENTS	7	–	Original agent control excluded from results
AG-F0	8	12	Agent control with no floating car
AG-F10	–	13	Agent control with 10 % of floating cars
AG-F25	–	14	Agent control with 25 % of floating cars
AG-F50	–	15	Agent control with 50 % of floating cars
AG-F100	9	16	Agent control with 100 % of floating cars

4.2 Number of replications

At the first stage, an appropriate number of replications had to be found. For this, an algorithm presented by Burghout [60] was applied. The algorithm represents an iterative approach, which finds the minimal replication number. It is described in Appendix B in detail.

In brief, it requires firstly performing a certain number (higher than one) of replications for all proposed scenarios. In other words, all the scenarios are run with different random seeds. Then, percentage error is calculated for each observed value at each scenario. If all calculated errors are lower than a given threshold, the number of performed replications is enough. Otherwise, another replication with a new random seed has to be run for all scenarios and the relative error is calculated again.

The determination of the number of replications was performed individually for the all-day scenarios and the morning scenarios, since the results from these two groups of scenarios are totally independent and are not compared in any analysis. Nevertheless, the same characteristics were tested in both cases, namely the mean and median of travel times for personal and heavy vehicles, and for different segments separately. As shown in Table 4.1, there were identified a total of 20 segments in the model. However, some of them are very infrequently used and the results for such segments can be highly affected by the randomness. For this reason, it was decided to select for the analysis only the most used segments which encompass the respective studied occurrences, i.e. regular and irregular congestion in respective all-day and morning scenarios.

Table 4.3 provides a summary of those segments which meet the aforementioned conditions. In general, drivers of both types of vehicles most frequently use the segments 18 and 20. Moreover, these segments encompass both the area upstream from the critical Exit 16 and the area upstream from the scheduled accident. Then, both types of scenarios have been assigned other segment(s) frequently used by personal vehicles.

Table 4.3: Segments selected for the determination of the number of replications.

Scenario type	**Personal**	**Heavy**
All-day scenarios	15, 16, 18, 20	18, 20
Morning scenarios	6, 18, 20	18, 20

Reference labels for each segment and vehicle type in Table 4.3 were introduced, since the segments were not used only for the determination of the number of replications, but also at any further analysis of travel times. The labels consist of letter P or H and the segment's number, considering that the letter stands for personal or heavy vehicles respectively, e.g. segment 18 used by personal vehicles is further referenced as P18. It should be emphasized that the behavior of each vehicle type in traffic flow is considerably different. Thus one segment used by both personal and heavy vehicles should be treated as two different segments.

It was decided to start with five replications for both scenario types. This means that five different random seeds were gradually used for each of 8 all-day scenarios and 7 morning scenarios. Then, both the mean and median of travel times for all selected segments from all respective scenarios were subjected to the test on relative error from the algorithm. The analysis for all-day scenarios concerns a total of 96 tested samples at each iteration which is given by 2 observed values for 6 segments from 8 scenarios. Similarly, the analysis for morning scenarios concerns 70 tested samples given by 2 observed values for 5 segments from 7 scenarios.

Figure 4.1 graphically displays the results for all-day scenarios on the left side and the morning scenarios on the right side. The red horizontal line represents the threshold which corresponds to the allowable percentage error of 5 %. Each point represents one tested sample for a given number of replications at horizontal axes. Then, the lowest number of replications where all samples are below the blue threshold line is the resulting minimum number of replications.

Figure 4.1 shows that the minimum number of replications for all-day scenarios is 12 and for morning scenarios is 9. It can be observed that the percentage error generally decreases with the increasing number of replications as expected [61]. Finally, it was decided to perform 15 replications for all-day scenarios and 11 replications for morning scenarios, where the maximum adjusted percentage error is lower than 4 %.

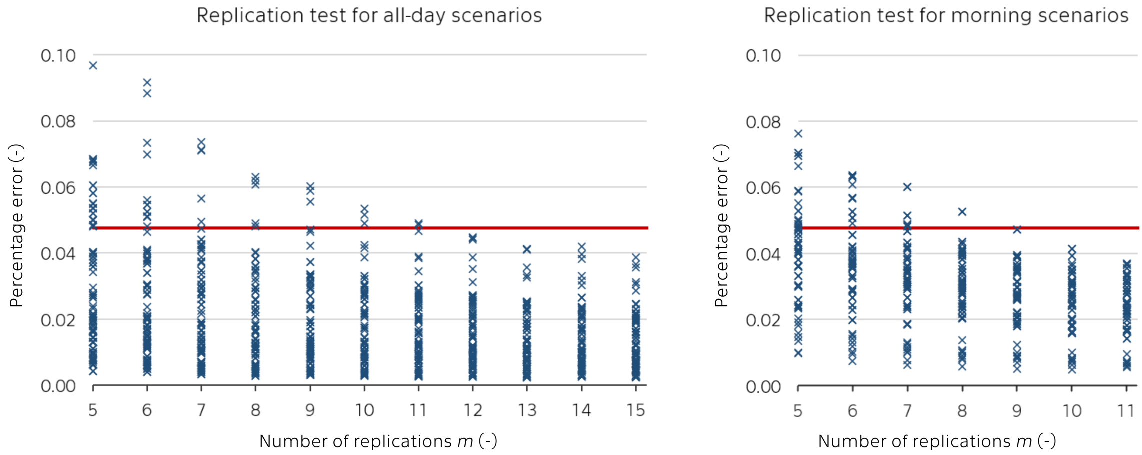

Figure 4.1: Results of the algorithm determining the number of replications.

4.3 Validation of the Model

The validation of the model was performed on two levels. First, traffic flow itself was validated by comparing data obtained from modelled detectors with the real data from Prague Ring Road. Second, the effects of speed limits on the behavior of traffic flow were analyzed by methods proved by other authors. As mentioned above, only all-day scenarios were used in this section.

4.3.1 General Traffic Flow Behavior

The validation of traffic flow itself is beyond the scope of this dissertation. Nevertheless, a general description of modelled traffic flow behavior which corresponds to the real situation is provided. Figure 4.2 shows time-space diagrams of local density, mean speed, and speed variance for one replication of the simulation without any control (Lim130). The diagrams were created based on minute data at five detectors. The first detector L21.810 was excluded since the data observed at this detector are affected by an initial setting of traffic flow in the model. The detectors L20.175 and L14.524 correspond to the borders of the diagrams and the other detectors L18.730, L17.080, and L15.745 are displayed by black lines. The direction of traffic flow is from top to bottom. The values of each traffic flow parameter are expressed by colors considering that the values between detectors are interpolated.

Several phenomena can be observed in the presented time-space diagrams. The commonly known features of free flow are visible at night hours. Traffic flow density is low, mean speed corresponds to the vehicles' preferred speed, and the values of speed variance are very variable [55]. Moreover, white patterns appear in the diagrams of mean speed and the speed variance. This means that no vehicle was detected within the given one-minute intervals.

Two traffic breakdowns and subsequent traffic congestion can also be distinguished in the diagrams. According to [55], a breakdown is characterized by an abrupt decrease in mean speed to a considerably lower speed in congested traffic. It is furthermore accompanied by increased density and increased speed variance. The last parameter indicates stop-and-go traffic [6] which has a negative impact on safety [15]. The mentioned congested traffic flow is detected at the L17.080 which is located upstream from the critical Exit 16 – K Barrandovu which confirms the aforementioned presumption.

The effects of speed limits were validated with the results from one-limit scenarios, i.e. scenarios with the speed limits of 120, 100, 80 and 60 km/h. The first insight is provided by time-space diagrams for the speed limit of 80 km/h. Figure 4.3 shows the measured data from one replication of the same randomness as the replication presented in Figure 4.2. Comparing the respective time-space diagrams, it can be observed that the speed limit results in increased densities, decreased mean speeds and considerably reduced speed variances. Moreover, the speed limit also provokes an earlier breakdown in this randomness.

4.3.2 Analysis of Travel Times

It was expected that travel travel times would increase with decreasing speed limits limits [5] and Figure 4.4 confirms this expectation. It should be understood that travel times are not normally distributed due to congestion. For this reason, the medians of travel times were selected as the main statistics to compare different scenarios. The line charts in Figure 4.4 show the average values of medians from all replications. All one-limit scenarios are included and data for the most used segments (see section 4.2) are displayed.

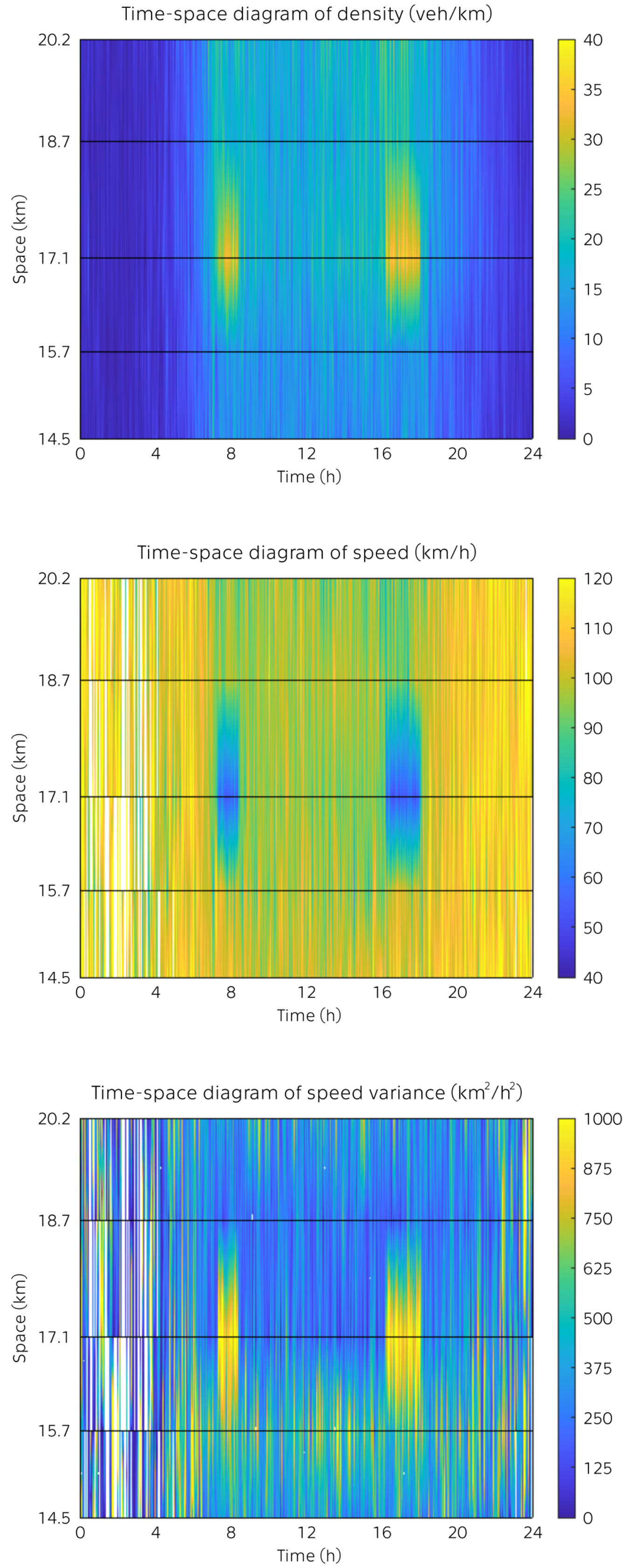

Figure 4.2: Time-space diagrams for all-day scenario Lim130, random seed 5.

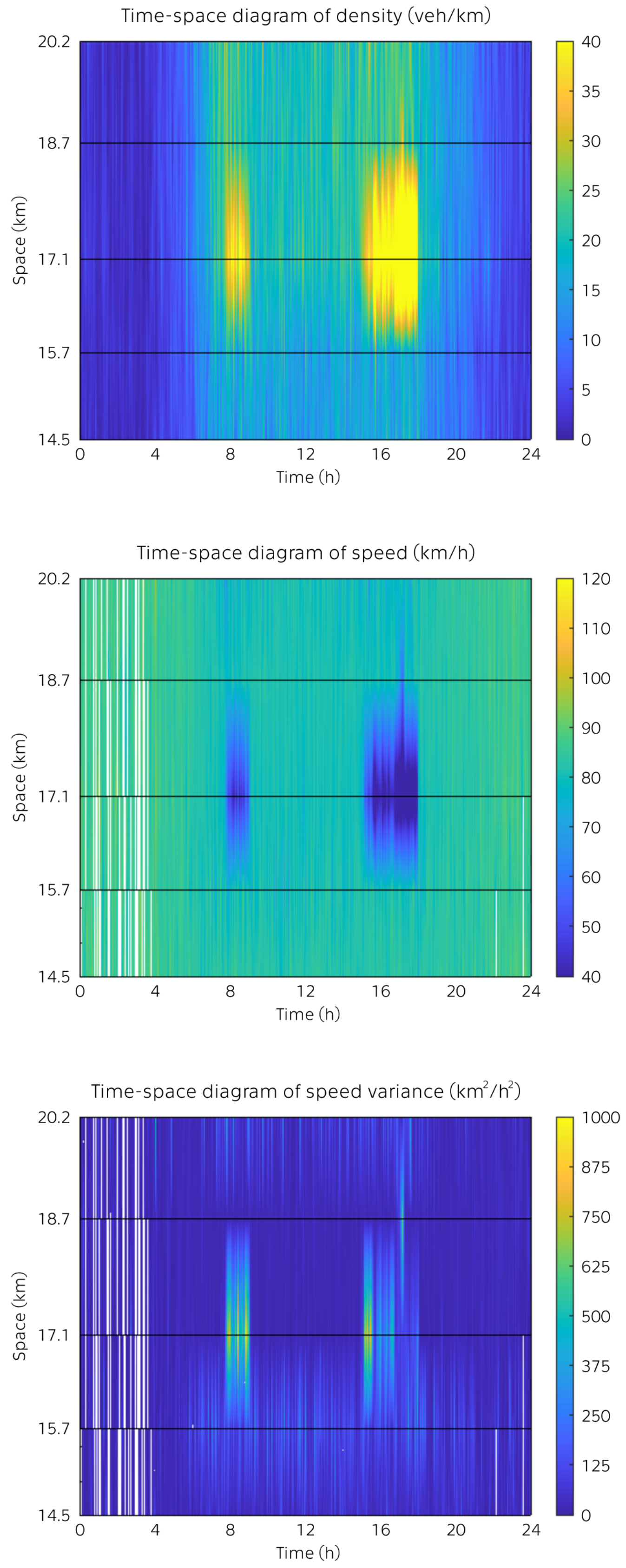

Figure 4.3: Time-space diagrams for all-day scenario Lim80, random seed 5.

While the left diagram provides the results in absolute values, the results in the right diagram are related to the values obtained for uncontrolled traffic flow.

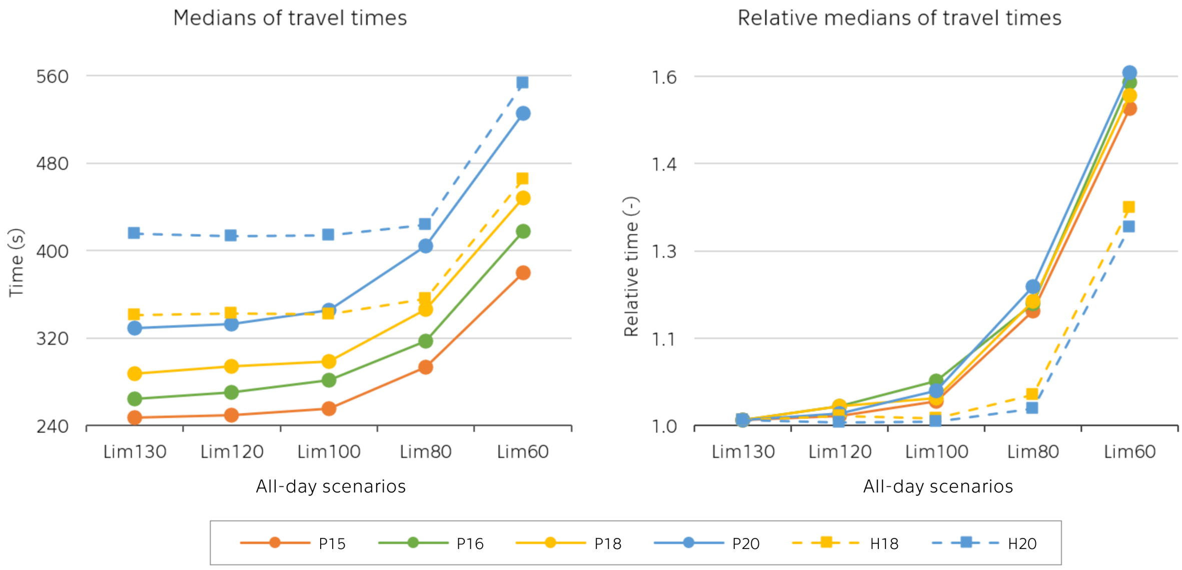

Figure 4.4: Medians of travel times for all one-limit scenarios.

A different trend among scenarios can be observed for heavy and personal vehicles. The speed limits of 120 and 100 km/h imply a slight increase in the travel times of personal vehicles whereas the travel times of heavy vehicles remain almost unchanged. It is caused by the fact that the preferred speed of heavy vehicles is in the range of 80 to 100 km/h independently of the highest speed limits. Then, the speed limit of 80 km/h implies a slight increase in the travel times of heavy vehicles. Furthermore, the absolute values of travel times for this scenario indicate that the personal and heavy vehicles became synchronized. The same phenomenon can be observed at the speed limit of 60 km/h. Nevertheless, this synchronization was achieved at the expense of disproportionately high travel times. It indicates that the permanent display of such a low speed limit on highways is inappropriate. It confirms the proposal from Gu et al. [7] that the low speed limits should be activated during congestion or when a traffic breakdown is expected.

4.3.3 Fundamental Diagram Analysis

The analysis arises from a quantitative model for the VSL-induced fundamental diagram change proposed by Cremer [62]. In brief, the model describes the theoretical curve in a flow-occupancy diagram as a function of the ratio of speed limit and the speed of free flow. According to this model, the highway capacity should be reached at higher occupancy and even increased for lower speed limits [5].

The analysis itself is inspired by the methodology implemented by Beneš and Přibyl [63]. They performed quadratic polynomial fitting on real data in a flow-occupancy diagram. It resulted in a set of parabolas at which the theoretical capacities (maximums) were observed. The usage of this type of fitting can be justified by the first theoretical traffic model proposed by Grienshiled [64]. That is because this generally known model uses parabola for the representation of traffic flow in flow-density diagrams.

In this work, the data obtained from one-limit scenarios with the speed limits of 130 and 80 km/h were used as input for the analysis. In other words, the traffic flow subjected to a speed limit of 80 km/h (Lim80) was compared to uncontrolled traffic flow (Lim130). Figure 4.5 displays the minute data obtained at the most observed detector L17.080. The graph includes data samples from both one-limit scenarios of the same randomness and the respective parabolas. Figure 4.6 then displays the averaged curves from all replications. As mentioned above, the maximums of parabolas represent the theoretical highway capacity. It can be observed that the capacity is shifted to a higher density and is increased in the case of a speed limit of 80 km/h. This corresponds to the results obtained by Cremer's model.

4.3.4 Safety Analysis

Lane-changing frequency and the speed variation in traffic flow are the parameters associated with traffic safety [14, 16, 17, 65]. It was tested whether the speed limits affect the mentioned parameters. In general, lower speed limits should decrease the number of lane changing maneuvers as well as speed variation [16, 17].

Table 4.4 provides the total number of all changes obtained as the average of all replications. The last column expresses the numbers of lane changes related to the uncontrolled scenario. The values demonstrate that all speed limits result in notably reduced lane changing maneuvers which is usually associated with higher densities [17]. However, a speed limit of 60 km/h does not bring any further reduction in comparison with a speed limit of 80 km/h. This coincides with an analysis performed by Soriguera et al. [16], where the speed limits of 80, 60, and 40 km/h were tested on a freeway. They proved that the effects of low speed limits are considerably different in the free flow and in the congested traffic flow. While decreased speed limits resulted in a reduced probability of lane changes in congestion, a sharp increase in the probability was observed in stable traffic flow.

The effect of speed limits on speed variance can be demonstrated by a diagram showing the relationship between the minute data of mean speed and speed variance. Gu et al. [7]

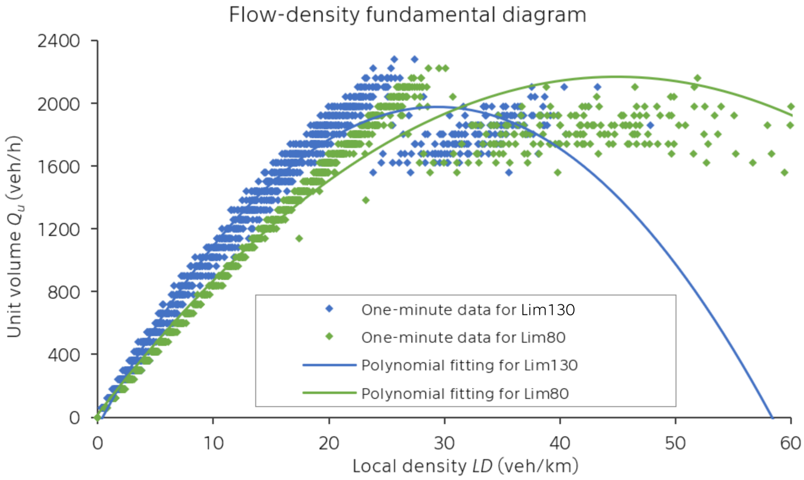

Figure 4.5: Polynomial fitting performed on minute data at L17.080; random seed 0.

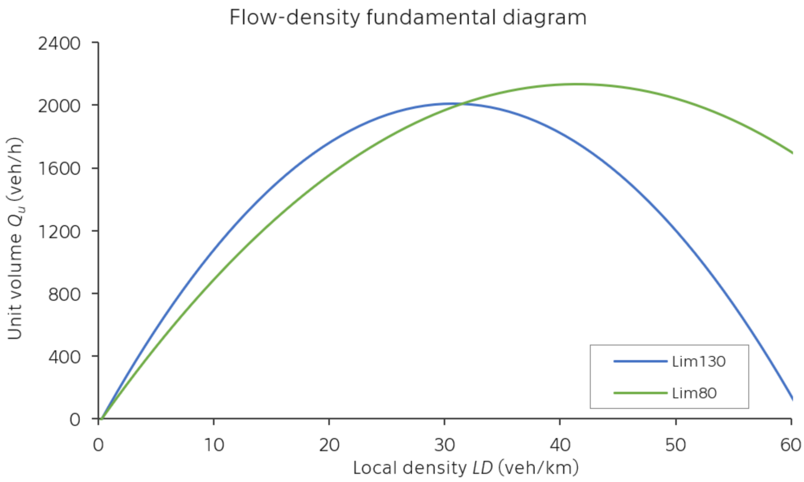

Figure 4.6: Analysis of theoretical highway capacity in fundamental diagram at L17.080.

Table 4.4: Average numbers of total lane changes for one-limit scenarios.

Scenario label	**Total number** (-)	**Relative number** (-)
Lim130	53 488	1.000
Lim120	50 912	0.952
Lim100	38 802	0.725
Lim80	31 917	0.597
Lim60	32 698	0.611

proposed to use such a diagram for the analysis of traffic flow and for the calibration of the thresholds at an automated control[1].

Figure 4.7 displays the mentioned relationship for uncontrolled traffic flow (Lim130). Two main clusters can be observed in this scatter plot. While the right cluster represents stable traffic flow, the left cluster represents congested traffic flow. The separated data samples on the right side correspond to a free flow and, finally, the data samples between the clusters are associated with the transition between a stable and unstable traffic flow. It is visible that the breakdown is accompanied by an increasing speed variance which can result in risky situations in traffic flow.

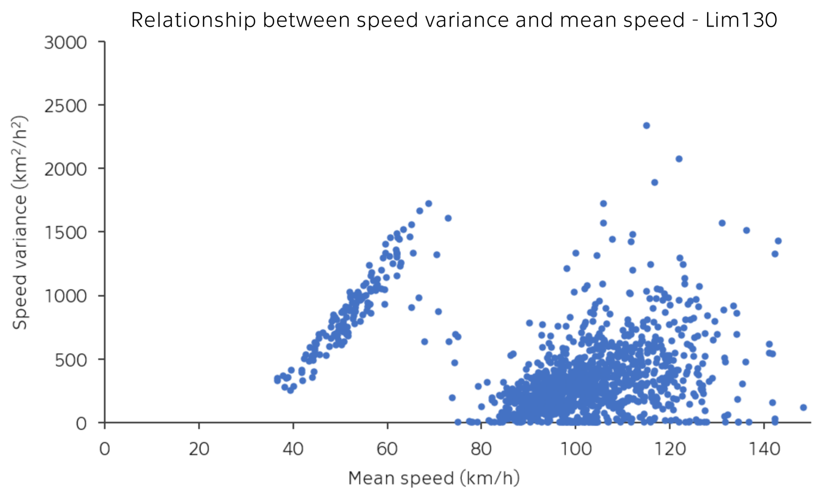

Figure 4.7: Speed-speed variance relationship for Lim130 at L17.080; random seed 13.

[1]The diagrams presented in this work display speed variance instead of speed standard deviation used in the diagrams proposed by Gu et al. [7]

The data from all one-limit scenarios of the same randomness were put together in Figure 4.8. It is apparent that speed limits of 120 and 100 km/h result in similar risky situations since the transition occurs accompanied by almost identical speed variance. On the other hand, the maximum variance at the transition is respectively lower and considerably lower for the speed limits of 80 and 60 km/h. In addition to Figure 4.8, the maximum variances at the transition are provided by Table 4.5. The average values from all replications are stated in the second column, whereas the last column expresses values related to an uncontrolled scenario.

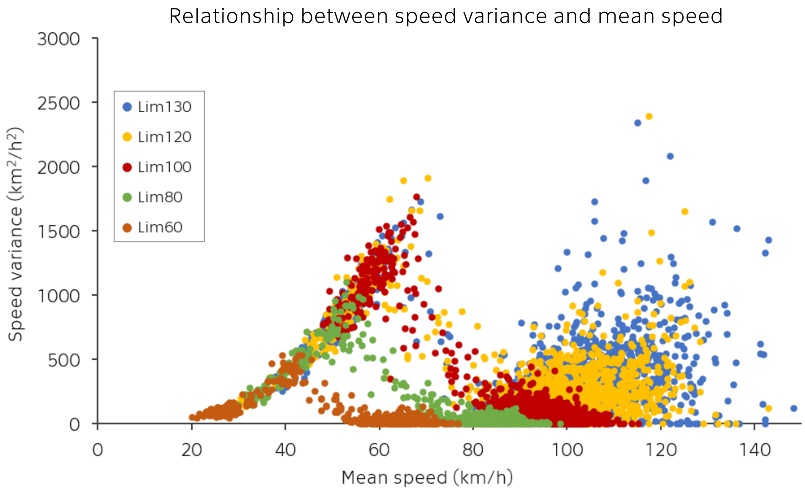

Figure 4.8: Speed variance analysis for one-limit controls at L17.080; random seed 13.

Table 4.5: Average maximum variances at the transition.

Scenario label	**Maximum variance** (km^2/h^2)	**Relative variance** (-)
Lim130	17 035	1.000
Lim120	17 875	1.049
Lim100	15 854	0.931
Lim80	10 337	0.607
Lim60	5 616	0.330

Moreover, observing the data for stable traffic flows in Figure 4.8, it can be stated that the speed dispersion decreases with lower speed limits. This statement is in accordance with the study performed by Gao et al. [14]

4.4 Automated Control

This section is dedicated to the evaluation of two automated control approaches described in the previous chapter. In general, the effects of VSLs controlled by decision trees (INEP), or by agents without floating cars (AG-F0), are compared from different perspectives. The scenario with uncontrolled traffic flow (Lim130) is always used as a reference for both control approaches. It should be recorded that only the modified agent model is considered in the evaluation since the original version provided identical results as with INEP.

4.4.1 Analysis of Travel Times

Similarly to the one-limit scenarios, a general increase in travel times was expected [5]. First, the medians of travel times were compared. The left chart of Figure 4.9 shows a slight increase, i.e. approximately of 15 s, for personal cars in the case of both controlled scenarios. At the same time, the change for heavy vehicles is not very noticeable among all scenarios. The differences are then more visible in the right chart displaying relative values. The differences between INEP and AG-F0 are always lower than 1 %. Nevertheless, in comparison with INEP, a certain increase for personal cars is observed at AG-F0. According to Papageorgiou, Kosmatopoulos and Papamichail [5], a higher increase in travel times usually indicates an earlier reaction by VSL to the approaching critical breakdown.

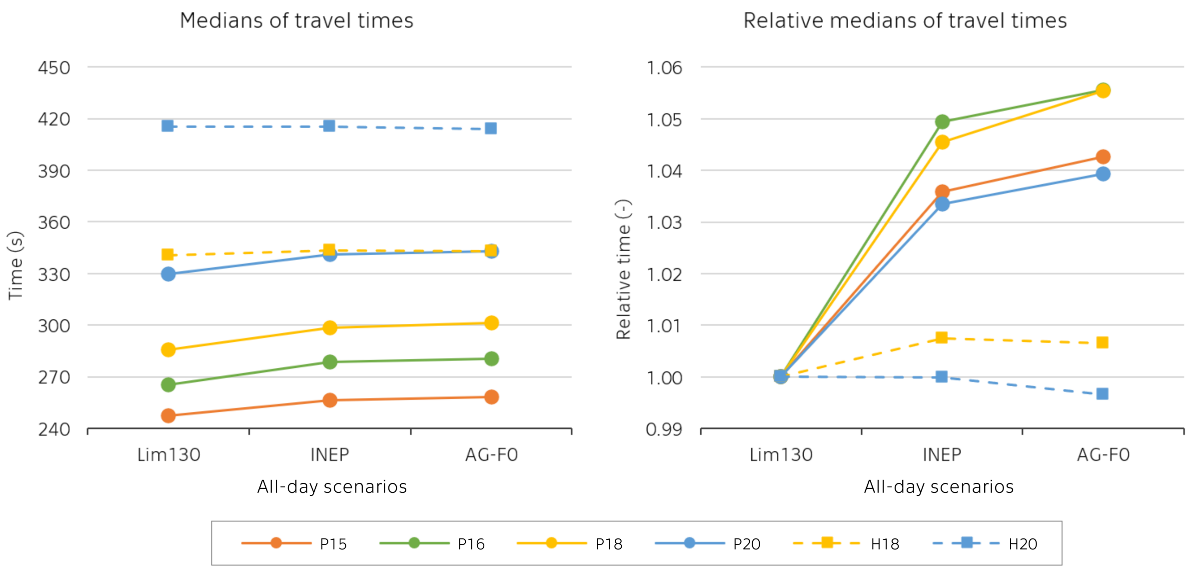

Figure 4.9: Medians of travel times – comparison of controlled scenarios to Lim130.

The analysis of travel times was extended by the observation of the difference between the respective minimums and maximums of travel times. A lower difference should then indicate a less negative impact of the congestion. Since the lowest travel times are generally associated with the free flow which is present in all scenarios, no remarkable changes in the minimums were observed. On the other hand, the highest travel times are recorded by the vehicles going through congestion. Thus, the maximums indicate, in a way, the heaviness of the congestion.

Figure 4.10 shows very similar results for all studied segments. The maximums are reduced by both control approaches. Whereas a decrease of 5-7 % is observed for INEP, a decrease of 15-19 % is achieved by AG-F0. This result indicates a better harmonization of traffic flow in congestion when the agent-based approach is used.

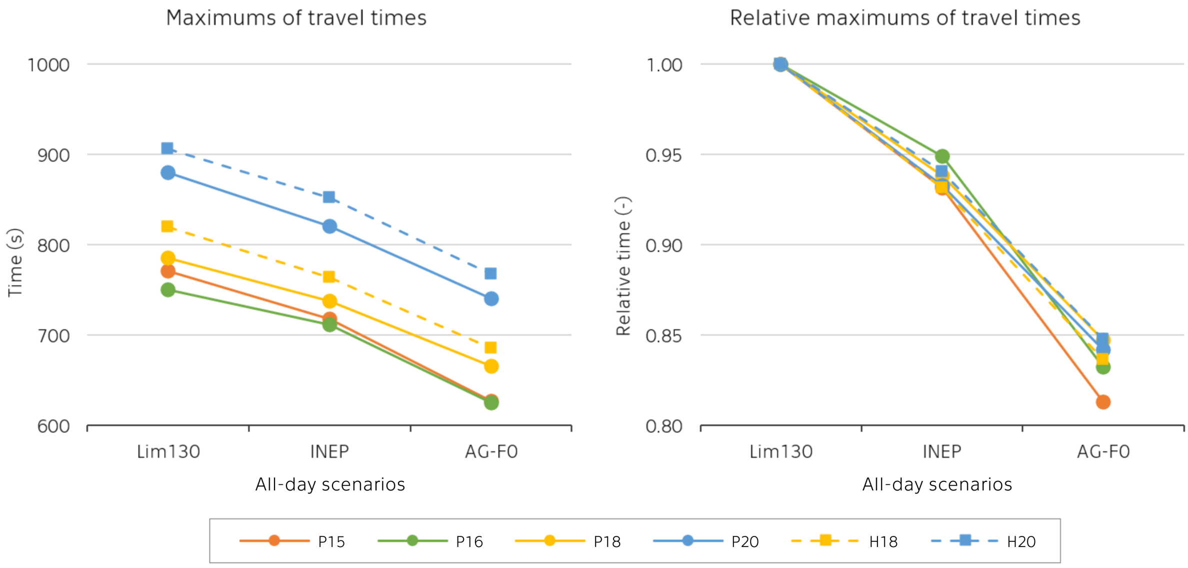

Figure 4.10: Maximums of travel times – comparison of controlled scenarios to Lim130.

4.4.2 Safety Analysis

The analysis of speed variance at the transition presented in the previous section was applied to the results from the controlled scenarios. Generally, the algorithms should detect the approaching breakdown and the speed limits of 80 or 60 km/h should already be displayed at the moment the breakdown occurs. Unfortunately, the data aggregating and smoothing (see Figure 3.12) can result in a delayed response of the algorithms. In such a case, the transition from stable traffic flow to congestion can occur under higher speed limits. This implies higher variances in speed and a riskier breakdown.

Scatter plots displaying the relationship between mean speed and speed variance at the detector L17.080 (as in Figure 4.7) were used to demonstrate the mentioned phenomenon. Figures 4.7, 4.11, and 4.12 display data from respective scenarios (Lim130, INEP, and AG-F0) of the same randomness. The colors represent the speed limits which were displayed when the corresponding data were logged.

Figure 4.11 demonstrates one case of INEP in which a speed limit of 100 km/h was displayed even if a breakdown occurred. In contrast to that, the whole congestion in the case of AG-F0 (see Figure 4.12) is under the speed limits of 80 or 60 km/h.

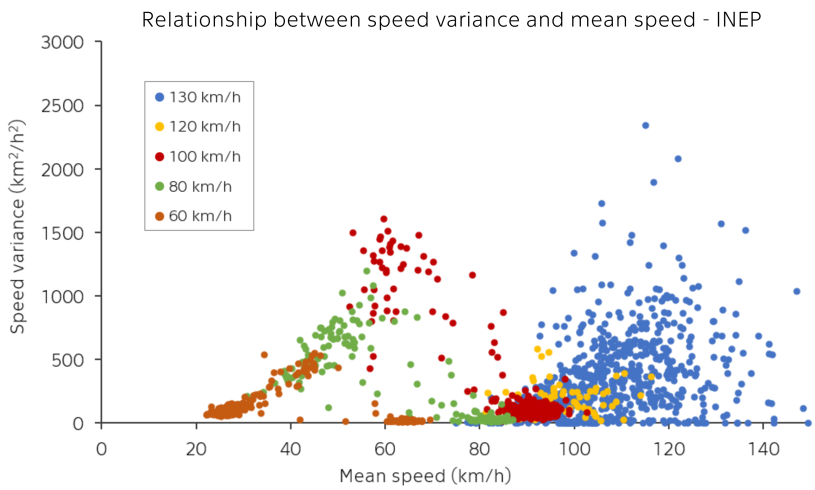

Figure 4.11: Speed variance analysis for INEP at L17.080; random seed 13.

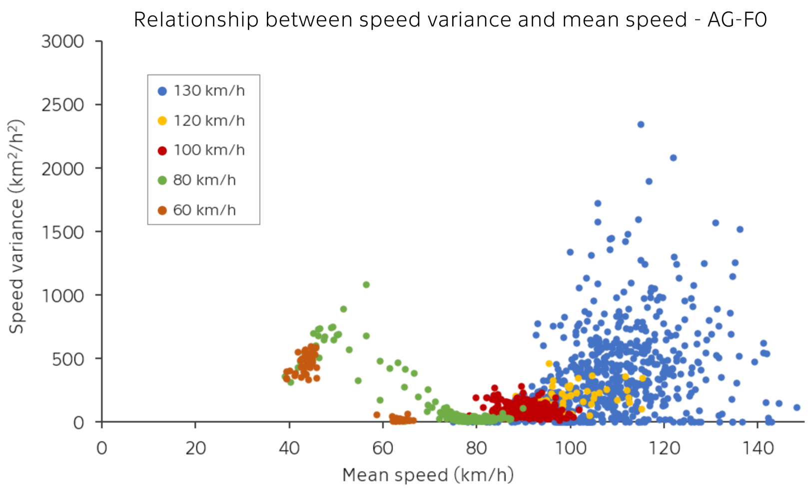

Figure 4.12: Speed variance analysis for AG-F0 at L17.080; random seed 13.

In order to compare the results from all replications, the maximum values of speed variance at the transition were retrieved from the data. The values from all replications were then displayed by means of box-plots in Figure 4.13.

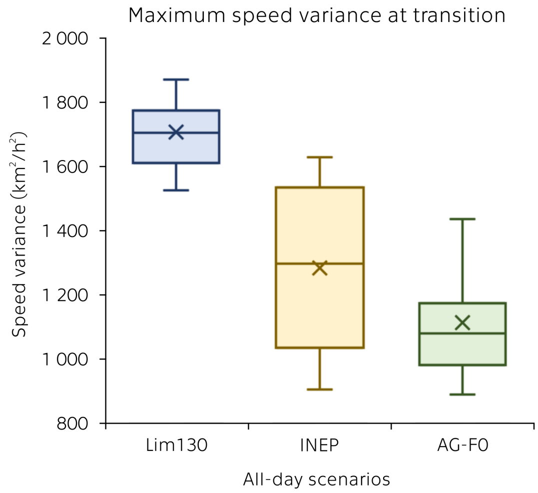

Figure 4.13: Maximum speed variances – comparison of controlled scenarios to Lim130.

It can be concluded that both algorithms improved the safety at the transition since the maximum speed variance in congested traffic flow was decreased. The higher dispersion in the results of INEP can be explained by the low sensitivity of the algorithm to the rapid data changes. INEP uses as input smoothed five-minute data aggregations. Thus, it is able to react appropriately to some gradual transitions to the congestion. On the other hand, data disturbances are usually neglected which causes a delayed reaction of the algorithm. This statement can be justified by the results associated with AG-F0, since the main change in the modified algorithm consisted in the decrease of the aggregation time from 5 to 3 minutes.

4.5 Implementation of Floating Cars

In addition to the change of the aggregation time, the implementation of floating cars to the agent control was proposed. The floating cars were designed to detect a highway standstill caused by an unexpected event. Thus, the agents should not intervene in the control in the case of regular congestion. This was tested in an all-day scenario controlled by agents where all vehicles are considered floating cars (AG-F100).

In total, 5 interventions were detected within all 11 replications, i.e. on average 0.5 per replication. Moreover, all interventions were logged in a congestion which had been already detected and controlled by the speed limit of 60 km/h. It means that the interventions did not result in any additional control actions. This confirms that the floating cars do not affect the automated control of regular congestion.

Hereinafter, this section deals only with the morning scenarios where an accident was scheduled. It was run for uncontrolled traffic flow (Lim130), for decision tree control (INEP) and for agent control with different proportions of floating cars in traffic flow (AG-FX where X stands for the ratio in percentages). The tested proportions were 0 %, 10 %, 25 %, 50 %, and 100 %.

As expected, it was observed that neither INEP nor AG-F0 reacted to the irregular congestion. Nevertheless, the length of queue created after the accident was analyzed for all scenarios. It was measured from the location of the accident to the last cars which had to stop in the queue. The box-plots in Figure 4.14 shows that the length of the developed queue is variable and random. The modeled accident, which caused a 10-minute closure of the highway, resulted in a queue whose length varied between 1000 m and 1800 m. This parameter is thus not affected by any control approach.

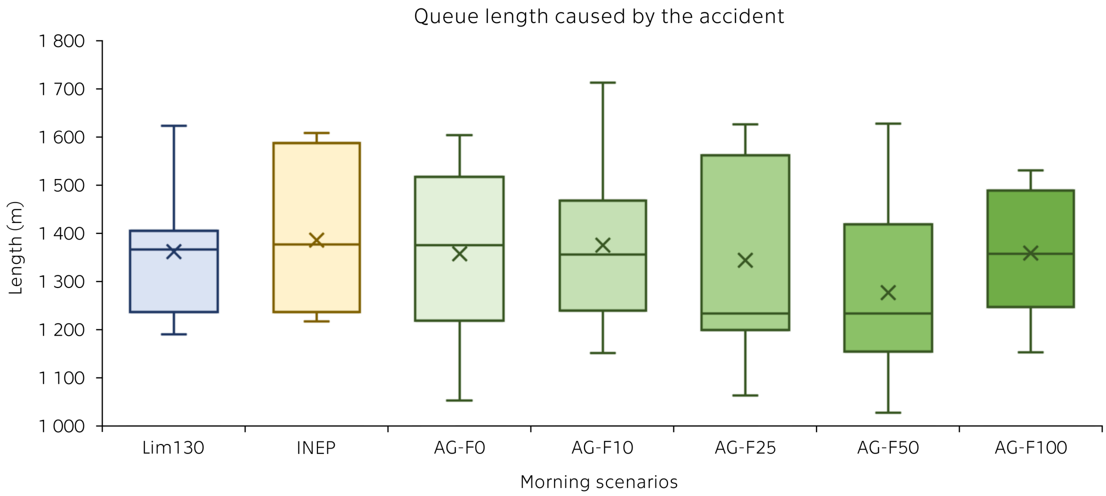

Figure 4.14: Length of queue caused by the accident.

As described in the previous chapter, once floating cars detect irregular congestion, they are induced to immediately display a speed limit of 60 km/h at the first upstream gantry. The time which lasts from the accident to the first display of the speed limit can be considered as the response time of the given algorithm. The response times for different proportions of floating cars are displayed through the box-plots in Figure 4.15. It demonstrates that the response time and its dispersion decreases with the increased proportion of floating cars. The dispersion in response time is given by the randomness of the location of floating cars. In the case of a 10% representation of floating cars, the average response time is approximately 5 minutes and the difference in response time reaches more than 7 minutes. On the other hand, the 100% representation of floating cars response on average in 1 minute and the dispersion is less than 2 minutes.

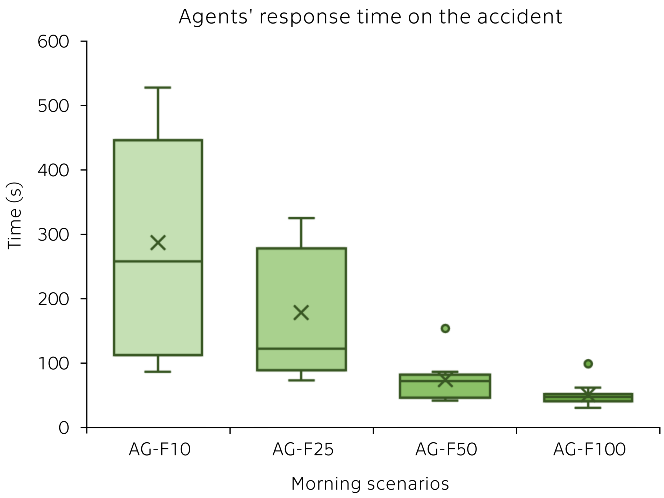

Figure 4.15: Agents' response time on the scheduled accident.

The possibility to provide information about an irregular congestion to other approaching drivers is considered to be the main contribution of the proposed floating cars. In this model, the drivers approaching the standing queue can be informed only via VMSs. Ideally, all drivers would see displayed the speed limit of 60 km/h displayed before arriving at the standing queue. This cannot ever be fully achieved, though the number of drivers without any information can be considerably decreased using floating cars. The effects of different representations of floating cars are provided by means of probabilities in Table 4.6.

Each row corresponds to one speed limit which can be displayed at gantries. The first row put together the speed limit 120 km/h with 130 km/h (no limit displayed) since the speed limit of 120 km/h is usually used for stable traffic flows in the time of a scheduled accident. It expresses the last displayed speed limit observed by a driver before arriving at the standing queue. The columns then correspond to different scenarios, i.e. different representations of floating cars in traffic flow. Each column represents one distribution of the drivers arriving to the standing queue over the last observed speed limits.

As mentioned above, AG-F0 did not react to the irregular congestion. Thus all drivers arrived at the standing queue at a speed limit 120 km/h or 130 km/h. In the case that 10 % of vehicles behaved as floating cars, only 24 % of all vehicles experienced such a situation. This proportion then decreases with the increasing representations of floating cars in traffic flow. The minimum is achieved by 100 % of floating cars, which is 4 % of all drivers. The last row in Table 4.6 corresponds to those drivers who have obtained

Table 4.6: Distributions of stops over speed limits.

Limit	**AG-F0**	**AG-F10**	**AG-F25**	**AG-F50**	**AG-F100**
≥120	1.00	0.24	0.16	0.07	0.04
100	0.00	0.06	0.01	0.00	0.00
80	0.00	0.02	0.08	0.05	0.03
60	0.00	0.68	0.75	0.88	0.93

the information about the unexpected standing queue before arriving there. It should be realized that VMS with a speed limit 60 km/h can be accompanied by another sign informing drivers about an unexpected queue. In the case that 10 % of vehicles behaved as floating cars, 68 % of all drivers had obtained this information. This proportion then decreases with the increasing representations of floating cars in traffic flow. The maximum is achieved by 100 % of floating cars, which is 93 % of all drivers.

The presented results demonstrate that floating cars were successfully implemented into the agent-based control. They further extended the agent-based control without any major change in the original algorithm. This would be hard to achieve with the original algorithm solved by decision trees.

CHAPTER 5

Discussion

The studied highway segment was selected, since the bottleneck at the off-ramp Exit 16 – K Barrandovu had been previously identified in real traffic flow on Prague Ring Road. As demonstrated in this work, speed harmonization is able to reduce the negative impact of developing congestion. However, this bottleneck still represents a challenge for other strategies of Active Traffic Management.

In addition to speed harmonization, INEP also includes the strategy for lane management known as the Heavy Goods Vehicle (HGV) overtaking ban [66, 67]. This strategy was also implemented into the model created in Anylogic (see Appendix A). It was verified that the display of corresponding VMS resulted in a considerably reduced number of heavy vehicles in the fast lane. However, automated control mostly failed.

The strategy generally results in decreased travel times for personal vehicles [66, 67] and this was not achieved by the algorithm proposed within INEP. There were not enough real data available to reveal the cause of the failure. For this reason, this strategy has been excluded from the simulations.

In my opinion, the strategy is not suitable for this type of bottleneck. In general, heavy vehicles are obliged to use the slow lane when higher volumes are detected. At the same time, a high proportion of personal vehicles exit the highway and they need to use the slow lane for merging. The slow lane thus becomes overloaded and the off-ramp represents an even greater bottleneck.

This reveals a weakness in the algorithms designed within INEP. It is due to the usage of the same control algorithm and the same thresholds at all gantries. According to Papageorgiou, Kosmatopoulos and Papamichail [5], the control thresholds should be set and calibrated for each gantry separately. Furthermore, in the case of such a critical bottleneck identified on a highway, it is believed that a particular control measure should be considered. In the world of MASs, it could be solved by a single agent. As proven in this work, such an agent could extend the model and intervene in another algorithm applied at all gantries. A similar intervention was actually tested in the case of floating cars.

In conclusion, it should also be understood that the modeled floating cars have the potential to behave like cooperative cars. Potentially, it offers a new direction in this research in accordance with recent trends [42, 43].

CHAPTER 6

Conclusions

This dissertation developed the idea proposed in 2014 [18, 19] that MAS could be introduced to highway management in the CR. First, the theory of MASs was studied and the application of MASs in highway management was reviewed. Then, the theoretical background gained in this research facilitated the proposal of an architecture for agent-based control for highway management.

On the basis of the proposed architecture, an agent-based simulation model was created in Anylogic. It was designed to test different control approaches in a single model. Therefore, a highway microsimulation model was also required, as well as the tools for highway management, i.e. detectors, VMSs. For the purpose of this research, the segment of Prague Ring Road from Ruzyně (km 22.5) to Lochkov tunnel (km 13.5) was modeled. Moreover, VMSs were prepared for the strategies of traffic flow harmonization, i.e. VSLs and the HGV overtaking ban.

In general, three different approaches were tested by a series of simulations in the same model: manual centralized control, automated centralized rule-based control, and automated decentralized agent-based control. Manual centralized control, modeling an operator in TCC, was implemented first. The results from the respective simulations were primarily used to validate and calibrate the traffic model.

Automated centralized rule-based control was taken from the INEP project. For the purpose of this work, a replication of this decision tree algorithm was created. It is a speed harmonization algorithm previously implemented on Czech highways. The results from simulations were used as a reference for newly developed control algorithms.

Automated decentralized agent-based control represented the main objective of this dissertation. The main goal of the first agent-based control was to decentralize the logic of the reference algorithm. Since the new algorithm provided identical results to the reference algorithm, two modifications for the control were proposed. The first modification consisted of the reduction of the input data aggregation interval and the implementation of a penalization algorithm at the same time. It was designed in order to achieve an earlier response for the control algorithm to rapid changes in input data. The second modification consisted of the implementation of floating cars. It was intended to show that the control algorithm can also be extended by another control strategy. In this case, an original algorithm for AID was successfully implemented.

Both newly developed algorithms demonstrated that an agent-based approach allows for the extension of the control algorithm without any major intervention to the original. Moreover, the results of the simulations showed certain improvements in comparison with the reference INEP algorithm, especially in terms of traffic safety.

Bibliography

[1] CARVELL JR., J. D.; BALKE, K.; ULLMAN, J.; FITZPATRIC, K.; NOWLIN, L.; BREHMER, Ch. *Freeway Management Handbook* [online]. 1997-08 [visited on 2021-02-26]. Report. Texas Transportation Institute. Available from: https://rosap.ntl.bts.gov/view/dot/3933/dot_3933_DS1.pdf.

[2] *Road Network Operations & Intelligent Transport Systems*. Highway Traffic Management [online]. PIARC (World Road Association), © 2021 [visited on 2021-02-26]. Available from: https://rno-its.piarc.org/en/network-control-traffic-management-traffic-control-measures/highway-traffic-management.

[3] *Traffic Management Services: Variable Speed Limits* [online]. 2015-12 [visited on 2021-02-26]. Deployment Guideline, TMS-DG02. European Comission (EasyWay, EIP, EIP+). Available from: https://portal.its-platform.eu/index.php?q=filedepot_download/1729/5382&_ga=2.7561842.1941821659.1619516863-38648972.1619516863.

[4] KHONDAKER, B.; KATTAN, L. Variable speed limit: an overview. *Transportation Letters*. 2015, **7**(5), 264–278. Available from DOI: 10.1179/1942787514Y.0000000053.

[5] PAPAGEORGIOU, M.; KOSMATOPOULOS, E.; PAPAMICHAIL, I. Effects of Variable Speed Limits on Motorway Traffic Flow. *Transportation Research Record*. 2008, **2047**(1), 37–48. Available from DOI: 10.3141/2047-05.

[6] WEIKL, S.; BOGENBERGER, K.; BERTINI, R. L. Traffic Management Effects of Variable Speed Limit System on a German Autobahn: Empirical Assessment Before and After System Implementation. *Transportation Research Record*. 2013, **2380**(1), 48–60. Available from DOI: 10.3141/2380-06.

[7] GU, S.; MA, J.; WANG, J.; SUI, X.; LIU, Y. Methodology for Variable Speed Limit Activation in Active Traffic Management. *Procedia - Social and Behavioral Sciences*. 2013, **96**, 2129–2137. ISSN 1877-0428. Available from DOI: 10.1016/j.sbspro.2013.08.240.

[8] KERNER, B. S. Three-phase traffic theory and highway capacity. *Physica A: Statistical Mechanics and its Applications*. 2004, **333**, 379–440. ISSN 0378-4371. Available from DOI: 10.1016/j.physa.2003.10.017.

[9] HABTEMICHAEL, F. G.; PICADO SANTOS, L. de. Safety and Operational Benefits of Variable Speed Limits under Different Traffic Conditions and Driver Compliance Levels. *Transportation Research Record*. 2013, **2386**(1), 7–15. Available from DOI: 10.3141/2386-02.

[10] ALLABY, P.; HELLINGA, B.; BULLOCK, M. Variable Speed Limits: Safety and Operational Impacts of a Candidate Control Strategy for Freeway Applications. *Intelligent Transportation Systems, IEEE Transactions on*. 2008, **8**, 671–680. Available from DOI: 10.1109/TITS.2007.908562.

[11] BEL, G.; ROSELL, J. Effects of the 80km/h and variable speed limits on air pollution in the metropolitan area of barcelona. *Transportation Research Part D: Transport and Environment*. 2013, **23**, 90–97. ISSN 1361-9209. Available from DOI: 10.1016/j.trd.2013.04.005.

[12] ACKAAH, W.; HUBER, G.; BOGENBERGER, K.; BERTINI, R. Assessing the Harmonization Potential of Variable Speed Limit Systems. *Transportation Research Record: Journal of the Transportation Research Board*. 2016, **2554**, 129–138. Available from DOI: 10.3141/2554-14.

[13] STRÖMGREN, P.; LIND, G. Harmonization with Variable Speed Limits on Motorways. *Transportation Research Procedia*. 2016, **15**, 664–675. ISSN 2352-1465. Available from DOI: 10.1016/j.trpro.2016.06.056. International Symposium on Enhancing Highway Performance (ISEHP), June 14-16, 2016, Berlin.

[14] GAO, Ch.; XU, J.; LI, Q.; YANG, J. The Effect of Posted Speed Limit on the Dispersion of Traffic Flow Speed. *Sustainability*. 2019, **11**(13). ISSN 2071-1050. Available from DOI: 10.3390/su11133594.

[15] LEE, Ch.; HELLINGA, B.; SACCOMANNO, F. Evaluation of variable speed limits to improve traffic safety. *Transportation Research Part C: Emerging Technologies*. 2006, **14**(3), 213–228. ISSN 0968-090X. Available from DOI: 10.1016/j.trc.2006.06.002.

[16] SORIGUERA, F.; MARTÍNEZ, I.; SALA, M.; MENÉNDEZ, M. Effects of low speed limits on freeway traffic flow. *Transportation Research Part C: Emerging Technologies*. 2017, **77**, 257–274. ISSN 0968-090X. Available from DOI: https://doi.org/10.1016/j.trc.2017.01.024.

[17] LIU, M.; SHI, J. Exploring the Impact of Differentiated Per-Lane Speed Limits on Traffic Safety of Freeways with Considering the Compliance Rate. *Journal of Advanced Transportation*. 2018, **2018**, 1–11. Available from DOI: 10.1155/2018/8368294.

[18] KUKLOVÁ, J.; PŘIBYL, O. Development of highway management in the Czech republic towards multi-agent approach. *World transport and technological machinery*. 2014, **4**(47), 109–115. ISSN 2073-7432.

[19] KUKLOVÁ, J.; PŘIBYL, O. Multi-Agent Systems for Highway Management. In: *Proceedings of the International Workshop Traffic Management and Road Safety Issues in European Cities*. CTU in Prague, 2014. ISBN 978-8-001-054857.

[20] *Projekt INEP* [online]. Eltodo, 2009 [visited on 2016-01-22]. Available from: http://inep.eltodo.cz/.

[21] *SIRID* [online]. Eltodo, 2012 [visited on 2016-01-22]. Available from: http://sirid.cz/.

[22] KUKLOVÁ, J.; PŘIBYL, O. Changeover from decision tree approach to fuzzy logic approach within highway management. *Neural Network World*. 2017, **27**(2), 181–196. ISSN 1210-0552. Available from DOI: 10.14311/NNW.2017.27.008.

[23] BOND, A.; GASSER, L. *A survey of distributed artificial intelligence* [online]. 1988 [visited on 2021-02-26]. Available from: http://www.exso.com/nsurvo.pdf.

[24] WEISS, G. *Multiagent systems*. Second edition. Cambridge, Massachusetts: The MIT Press, 2013. ISBN 978-0-262-01889 [xlviii, 867].

[25] SHOHAM, Y.; LEYTON-BROWN, K. *Multiagent systems: algorithmic, game-theoretic, and logical foundations*. Cambridge: Cambridge University Press, 2009. ISBN 978-0-521-89943-7 [xx, 483].

[26] WOOLDRIDGE, M.; JENNINGS, N. R. Intelligent agents: theory and practice. *The Knowledge Engineering Review*. 1995, **10**(2), 115–152. ISSN 0269-8889. Available from DOI: 10.1017/S0269888900008122.

[27] BELLIFEMINE, F. L.; CAIRE, G.; GREENWOOD, D. *Developing multi-agent systems with JADE*. Hoboken, NJ: John Wiley & Sons, Ltd., 2007. ISBN 9780470057476 [xiii, 286].

[28] VLASSIS, N. A Concise Introduction to Multiagent Systems and Distributed Artificial Intelligence. *Synthesis Lectures on AI and Machine Learning*. 2007, **1**(1), 1–71. ISSN 1939-4608. Available from DOI: 10.2200/S00091ED1V01Y200705AIM002.

[29] FININ, T.; FRITZSON, R.; MCKAY, D.; MCENTIRE, R. KQML as an agent communication language. In: *Proceedings of the third international conference on Information and knowledge management - CIKM '94* [online]. New York, USA: ACM Press, 1994, pp. 456–463 [visited on 2021-02-26]. ISBN 0897916743. Available from DOI: 10.1145/191246.191322.

[30] FININ, T.; LABROU, Y.; MAYFIELD, J. KQML as an Agent Communication Language. In: *Software agents*. Ed. by BRADSHAW, J. M. Cambridge, Mass.: MIT Press, 1997, pp. 291–316. ISBN 0262522349.

[31] *The Foundation for Intelligent Physical Agents* [online]. Foundation for Intelligent Physical Agents, © 2020 [visited on 2021-02-26]. Available from: http://www.fipa.org/.

[32] *FIPA Agent Communication Language Specifications* [online]. Foundation for Intelligent Physical Agents, © 2002 [visited on 2021-02-26]. Available from: http://www.fipa.org/repository/aclspecs.html.

[33] SHOHAM, Y. An Overview of Agent-Oriented Programming. In: *Software agents*. Ed. by BRADSHAW, J. M. Cambridge, Mass.: MIT Press, 1997, pp. 271–290. ISBN 0262522349.

[34] SHOHAM, Y. Agent-oriented programming. *Artificial Intelligence*. 1993, **60**(1), 51–92. ISSN 0004-3702. Available from DOI: 10.1016/0004-3702(93)90034-9.

[35] *Jade Site: Java Agent DEvelopment Framework* [online]. Telecom Italia, © 2016 [visited on 2021-02-26]. Available from: http://jade.tilab.com/.

[36] KRAVARI, K.; BASSILIADES, N. A Survey of Agent Platforms. *Journal of Artificial Societies and Social Simulation* [online]. 2015, **18**(1) [visited on 2016-02-17]. ISSN 1460-7425. Available from DOI: 10.18564/jasss.2661.

[37] PAL, C.-V.; LEON, Florin; PAPRZYCKI, Marcin; GANZHA, Maria. A Review of Platforms for the Development of Agent Systems. *arXiv preprint arXiv:2007.08961* [online]. 2020 [visited on 2021-02-26]. Available from: https://arxiv.org/ftp/arxiv/papers/2007/2007.08961.pdf.

[38] *Anylogic: Simulation Modeling Software Tools & Solutions for Business* [online]. Anylogic Company, © 2021 [visited on 2021-02-26]. Available from: https://www.anylogic.com/.

[39] BURMEISTER, B.; HADDADI, A.; MATYLIS, G. Application of multi-agent systems in traffic and transportation. *IEE Proceedings - Software*. 1997, **144**(1), 51–60. ISSN 1462-5970. Available from DOI: 10.1049/ip-sen:19971023.

[40] CHEN, B.; CHENG, H. H. A review of the Applications of Agent Technology in Traffic and Transportation Systems. *IEEE Transactions on ITS*. 2010, **11**(2), 485–497. ISSN 1524-9050. Available from DOI: 10.1109/TITS.2010.2048313.

[41] PŘIBYL, O.; KOUKOL, M.; KUKLOVÁ, J. Computational intelligence in highway management: a review. *PROMET – Traffic & Transportation*. 2015, **27**(5), 439–450. ISSN 0353-5320. Available from DOI: 10.7307/ptt.v27i5.1667.

[42] MONTEIL, J.; BILLOT, R.; SAU, J.; ARMETTA, F.; HASSAS, S.; EL FAOUZI, N.-E. Cooperative highway traffic: multi-agent modelling and robustness assessment to local perturbations. *Transportation Research Record: Journal of the Transportation Research Board*. 2013, **2391**(1), 1–10. Available from DOI: 10.3141/2391-01.

[43] HA, P.; CHEN, S.; DONG, J.; DU, R.; LI, Y.; LABI, S. *2021 Annual Meeting of the Transportation Research Board*. Leveraging the Capabilities of Connected and Autonomous Vehicles and Multi-Agent Reinforcement Learning to Mitigate Highway Bottleneck Congestion [online]. 2021 [visited on 2021-02-26]. Available from: https://arxiv.org/ftp/arxiv/papers/2010/2010.05436.pdf.

[44] HERNÁNDEZ, J. Z.; OSSOWSKI, S.; GARCÍA-SERRANO, A. Multiagent architectures for intelligent traffic management systems. *Transportation Research Part C: Emerging Technologies*. 2002, **10**(5-6), 473–506. ISSN 0968-090x. Available from DOI: 10.1016/S0968-090X(02)00032-3.

[45] VAN KATWIJK, R.; VAN KONINGSBRUGGEN, P. Coordination of traffic management instruments using agent technology. *Transportation Research Part C: Emerging Technologies*. 2002, **10**(5-6), 455–471. ISSN 0968-090x. Available from DOI: 10.1016/S0968-090X(02)00034-7.

[46] *AnyLogic Cloud: Online Simulation Tool* [online]. Anylogic Company, © 2021 [visited on 2021-02-26]. Available from: https://cloud.anylogic.com/.

[47] *AnyLogic Cloud: Online Simulation Tool*. Highway Junction – Simulation Models in Anylogic Cloud [online]. Anylogic Company, © 2021 [visited on 2021-02-26]. Available from: https://cloud.anylogic.com/model/a62f2580-9431-41cd-89cc-2abe3cc07b93.

[48] BENČAT, G.; JANOTA, A. Road traffic modelling based on the hybrid modelling tool AnyLogic. *Journal of Civil Engineering and Transport*. 2020, **2**(2), 73–89. ISSN 2658-1698. Available from DOI: 10.24136/tren.2020.006.

[49] KUKLOVÁ, J. Highway Modeling in Anylogic for Multi-Agent Approach to Smart City Management. In: *2021 Smart City Symposium Prague (SCSP)*. 2021, pp. 1–6. Available from DOI: 10.1109/SCSP52043.2021.9447402.

[50] KUKLOVÁ, J.; PŘIBYL, O. Framework Model in Anylogic for Smart City Ring Road Management. In: *2019 Smart City Symposium Prague (SCSP)*. IEEE, 2019. ISBN 978-1-7281-0498-0. Available from DOI: 10.1109/SCSP.2019.8805681.

[51] *SOKP, D1 Liniové řízení dopravy: Část B: Provozní řád*. 2010. Regulations. Eltodo.

[52] *SOKP, D1 Liniové řízení dopravy: Část A: Dopravní řád*. 2010. Regulations. Eltodo.

[53] GUO, J.; WILLIAMS, B.; SMITH, B. Data Collection Time Intervals for Stochastic Short-Term Traffic Flow Forecasting. *Transportation Research Record Journal of the Transportation Research Board*. 2008, **2024**, 18–26. Available from DOI: 10.3141/2024-03.

[54] NIST/SEMATECH e-Handbook of Statistical Methods. In: [online]. NIST, 2013, chap. 6.4.3 What is Exponential Smoothing? [Visited on 2021-02-26]. Available from DOI: 10.18434/M32189.

[55] KERNER, B. *Introduction to Modern Traffic Flow Theory and Control: The Long Road to Three-Phase Traffic Theory*. 2009. ISBN 9783642026041. Available from DOI: 10.1007/978-3-642-02605-8.

[56] NUGROHO, S. A.; VISHNOI, S. C.; TAHA, A. F.; CLAUDEL, Ch. G.; BANERJEE, T. Where Should Traffic Sensors Be Placed on Highways? *IEEE Transactions on Intelligent Transportation Systems*. 2021, 1–14. Available from DOI: 10.1109/TITS.2021.3119211.

[57] WILLSKY, A.; GERSHWIN, E. Chow anS.; GREENE, C.; HOUPT, P.; KURKJIAN, A. Dynamic model-based techniques for the detection of incidents on freeways. *IEEE Transactions on Automatic Control*. 1980, **25**(3), 347–360. Available from DOI: 10.1109/TAC.1980.1102392.

[58] CORBY, M. J.; SACCOMANNO, F. F. Analysis of Freeway Accident Detection. *Transportation Research Record*. 1997, **1603**(1), 80–89. Available from DOI: 10.3141/1603-11.

[59] HOUBRAKEN, M.; LOGGHE, S.; AUDENAERT, P.; COLLE, D.; PICKAVET, M. Examining the potential of Floating Car Data for Dynamic Traffic Management. *IET Intelligent Transport Systems*. 2018, **12**. Available from DOI: 10.1049/iet-its.2016.0230.

[60] BURGHOUT, W. *A note on the number of replication runs in stochastic traffic simulation models*. 2004-01. Technical Report CTR2004:01. Center for Traffic Research.

[61] CHIABAUT, N.; BUISSON, Ch. Replications in Stochastic Traffic Flow Models: Incremental Method to Determine Sufficient Number of Runs. In: APPERT-ROLLAND, C.; CHEVOIR, F.; GONDRET, P.; LASSARRE, S.; LEBACQUE, J.P.; SCHRECKENBERG, M. (eds.). *Traffic and Granular Flow '07* [online]. Berlin, Germany: Springer Berlin Heidelberg, 2009, pp. 35–44 [visited on 2022-03-05]. ISBN 978-3-540-77074-9. Available from DOI: 10.1007/978-3-540-77074-9_3.

[62] CREMER, M. *Der Verkehrsfluß auf Schnellstraßen* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 1979 [visited on 2021-02-26]. ISBN 978-3-642-95346-0. Available from DOI: 10.1007/978-3-642-95346-0.

[63] BENES, J.; PRIBYL, O. Effects of highway management on traffic flow characteristics. *Archives of Transport System Telematics*. 2014, **7**(2), 14–18.

[64] VAN WAGENINGEN-KESSELS, F.; VAN LINT, H.; VUIK, K.; HOOGENDOORN, S. Genealogy of traffic flow models. *EURO Journal on Transportation and Logistics*. 2015, **4**(4), 445–473. ISSN 2192-4376. Available from DOI: https://doi.org/10.1007/s13676-014-0045-5.

[65] TALEBPOUR, A.; MAHMASSANI, H. S.; HAMDAR, S. H. Speed harmonization: Evaluation of effectiveness under congested conditions. *Transportation Research Record*. 2013, **2391**(1), 69–79. Available from DOI: 10.3141/2391-07.

[66] BERENI, Matthieu. Heavy goods vehicle overtaking bans some information to assist decision-making. In: 2013, pp. 517–529. ISBN 9781848210585. Available from DOI: 10.1002/9781118557464.ch40.

[67] NAGY, E.; SÁNDOR, Z. Overtaking ban for heavy goods vehicle in Hungary on the national motorway network. *Pollack Periodica*. 2012, **7**, 83–95. Available from DOI: 10.1556/Pollack.7.2012.1.8.