MODELING, CONTROL, AND SIMULATION BASED TESTING FOR AUTOMATED ROAD TRAFFIC

Tamás Tettamanti, PhD

Faculty of Transportation Engineering and Vehicle Engineering
Budapest University of Technology and Economics

DSc dissertation

January 13, 2022

Acknowledgements

First of all, I would like to express my major gratitude for my family, Zita, Marcell, Benedek, and Levente for their constant love and support.

I am really thankful to the Department of Control for Transportation and Vehicle Systems which makes it possible to work simultaneously in higher education, academic research as well as industrial projects, providing me an exciting and challenging workplace.

Personally, I would like to thank the wonderful people I have worked together during the past 15 years, especially to Prof. István Varga, Prof. József Bokor, Prof. Péter Gáspár, Prof. Balázs Kulcsár, Dr. Balázs Varga, and Márton Tamás Horváth.

Contents

List of Figures

List of Tables

Nomenclature

ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance Systems
AI	Artificial Intelligence
ANN	Artificial Neural Network
AV	Autonomous Vehicle
BASt	Bundesanstalt für Straßenwesen
BP	Biproportional Procedure
CAV	Connected and Autonomous/Automated Vehicle
CACC	Cooperative Adaptive Cruise Control
CEN	Comité Européen de Normalisation
cKF-I	constrained Kalman Filter while matrix $W = I$
cKF-P	constrained Kalman Filter while matrix $W = P(k)$
DEAP	Distributed Evolutionary Algorithms in Python
DSRC	Dedicated Short Range Communications
ECU	Electronic Control Unit
EDF	Estimated Degrees of Freedom
ESP	Electronic Stability Program
FCD	Floating Car Data
FHWA	Federal Highway Administration
FMD	Floating Mobile Data
GA	Genetic Algorithm
GAM	Generalized Additive Model
GEH	GEH formula gets its name from Geoffrey E. Havers
GLM	Generalized Linear Model
GRE	Global Relative Error
GNSS	Global Navigation Satellite System
GUI	Graphical User Interface
HDV	Human Driven Vehicle
ICT	Information and Communication Technology
ISO	International Organization for Standardization
IT	Information Technology
ITS	Intelligent Transportation Systems
I2V	Infrastructure to Vehicle
LFR	Linear Fractional Transformation
LSTM	Long Short Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MPC	Model Predictive Control
MFD	Macroscopic Fundamental Diagram
NN	Neural Network
OD	Origin-Destination
NHTSA	National Highway Traffic Safety Administration

PCU	Passenger Car Unit
PN	Petri Net
RNN	Recurrent Neural Network
SAE	Society of Automotive Engineers
SPSA	Simultaneous Perturbation Stochastic Approximation
SUMO	Simulation of Urban Mobility
TLNN	Time Lagged Neural Network
TWPN	Time Weighted Passenger Number
VMS	Variable Message Sign
VOC	Volatile Organic Compounds
VSS	Variable Speed Signs
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
V2X	Vehicle to Everything
XML	eXtensible Markup Language

Chapter 1

Introduction

Autonomous or highly automated road vehicles are core elements of future's transportation and it is not just a vision anymore. In the starting phase of AV developments road tests have been focused only on the behavior of sole vehicles without exploiting the advanced ability of AVs such as cooperation with infrastructure or other vehicles. In our days, a plethora of new ideas are born day after day for automated vehicles. However, the impact of these new technologies must be carefully assessed preliminary together with extensive and thorough testing. This work is ensured by the evolution of traffic microsimulation and its use for CAV modelings [Raju and Farah 2021]. All these shall lead finally to a safe and reliable homologation process of CAVs/AVs in the future [Varga et al. 2020b]. Accordingly, the main aim of my research is to reveal open questions in this field especially from the perspective of transportation engineering as well as to find novel and practical ways for better exploitation of automated traffic towards a sustainable future traffic.

1.1 Concept and Levels of Automated Vehicles

Automated vehicles are vehicles that are capable of sensing their environment and navigating with less or without human input [Gehrig and Stein 1999]. The automated driving is grouped into six different levels by the international Society of Automotive Engineers (SAE) according to the amount of driver intervention and attentiveness required. It delivers a harmonized classification and taxonomy for automated driving systems, specifically SAE J3016 Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems [SAE International 2021].

Accordingly, levels of autonomy are shown by Fig. 1.1. The last two columns in Fig. 1.1 represent the cross-compliance of SAE levels compared to the levels of the German Federal Highway Research Institute (BASt: Bundesanstalt für Straßenwesen) as well as that of the National Highway Traffic Safety Administration (NHTSA) of the USA.

As an interpretation SAE phrased the following: *"These levels are descriptive rather than normative and technical rather than legal. They imply no particular order of market introduction. Elements indicate minimum rather than maximum system capabilities for each level. A particular vehicle may have multiple driving automation features such that it could operate at different levels depending upon the feature(s) that are engaged."* [SAE International 2021]

Basically, the defined levels indicate the balance of the dynamic driving tasks between human and machine from zero level (no automation) to fifth level (full automation). To achieve the full automation two evolution paths are possible: the concept of "something everywhere" or "everything somewhere" [Corporate Partnership Board 2015]. The first variation means that automated driving systems appear gradually in the traditional vehicles according to the levels in Fig. 1.1, i.e. the drivers give more and more driving tasks to the automated systems. The second

SAE level	Name and definition	Steering, acceleration, deceleration	Monitoring driving environment	Fallback performance of dynamic driving task	System capability (driving modes)	BASt level	NHTSA level
0	No Automation: the full-time performance by the human driver of all aspects of the dynamic driving tasks, even when enhanced by warning or intervention systems	Human driver	Human driver	Human driver	-	Driver only	0
1	Driver Assistance: the driving mode-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the human driver perform all remaining aspects of the dynamic driving task	Human driver and system	Human driver	Human driver	Some driving modes	Assisted	1
2	Partial Automation: the driving mode-specific execution by one or more driver assistance systems of both steering and acceleration/deceleration using information about the driving environment and with the expectation that the human driver perform all remaining aspects of the dynamic driving task	**Human driver and system**	Human driver	Human driver	Some driving modes	Partially automated	2
3	Conditional Automation: the driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task with the expectation that the human driver will respond appropriately to a request to intervene	System[1]	**System**	Human driver	Some driving modes	Highly automated	3
4	High Automation: the driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task, even if a human driver does not respond appropriately to a request to intervene	System	System	**System**	Some driving modes	Fully automated	3/4
5	Full Automation: the full-time performance by an automated driving system of all aspects of the dynamic driving task under all roadway and environmental conditions that can be managed by a human driver	System	System	System	**All driving modes**	-	

Figure 1.1. Levels of driving automation according to the SAE International

evolution concept assumes that the fully automated cars could be applied in driverless mode immediately together with traditional vehicles until a total market penetration is achieved.

1.2 The Expected Future of Road Transportation

It is expected that by 2030, 5-10% of road vehicles in European Union will be autonomous or at least allowed to be operated in autonomous mode on certain road sections [Alonso Raposo et al. 2021]. The break-in of fully automated vehicle technology is expected primarily in the areas of public transport, freight transport and taxi passenger transport, as these are transport modes whose cost-effective operation is a key issue: financing state / municipal public transport is a constant problem as it is typically unprofitable; on the other hand, transportation and taxi services are areas of the economic sector where cost-efficiency is also of crucial importance. The emergence of fully free-moving autonomous vehicles in cities is not yet expected by 2030 due to the complexity of traffic scenarios (e.g. pedestrians, intersections, traffic lights). At the same time, they will already be able to travel on fixed routes (e.g. in separated ”autonomous lane”) and with dedicated stops: as a public transport vehicle or even as a shuttle service for a vehicle-sharing service. As far as private passenger cars and lorries are concerned, their primary terrain by 2030 will be the dedicated lanes of the freeway network, as autopilot function is partly an available technology today.

The self-driving future is certainly on the verge of. At the same time, it is important to emphasize that in addition to automotive innovation, there are 3 other keys to intensive development: transport and telecommunications infrastructure, technical regulation, and catching up with the legal framework. Infrastructure development is typically a national competence and in practice one of the basic criteria for the wide spread of autonomous vehicles (e.g. appropriate road markings, roadside equipment, communication base stations). In terms of technical regulations, domestic standardization obviously shall follow the work of international (ISO) and European (CEN) standards organizations (requirements for automated vehicle functions and their methodology for attesting conformity are under constant development), so they need to localize the relevant standards as soon as possible. Another critical element in the proliferation of autonomous vehicles is the development and implementation of an appropriate legal background. The types of liability for road transport (civil, criminal and administrative) are completely new in the world of autonomous cars, and their interpretation will require the use of alternative approaches, all in line with international legal harmonization.

1.3 Development Trends of Road Vehicles and Transportation

The development of road vehicles has been accelerated in the last decades. One of the spectacular results of this process is the fast growing number of electronic control units (ECU) in vehicles. Nowadays, typical compact and mid-size cars contain over 100 ECUs. Beyond the basic functioning (e.g. engine control unit) these special elements also improve safety (e.g. Electronic Stability Program, ESP), assist the driver (e.g. Advanced Driver Assistance Systems, ADAS), as well as ameliorate passenger comfort. ECUs are also becoming increasingly intelligent devices, such that more and more functions are integrated inside them. Moreover, new communication systems are also appearing in modern vehicles, which are capable to make contact with other cars or infrastructure. This is called V2V (Vehicle to Vehicle) or V2I (Vehicle to Infrastructure) communication [Gáspár et al. 2014]. The development and standardization of V2V/V2I technologies are also ongoing processes of our days [Khan et al. 2022]. Beyond the progressive technical solutions specific to vehicles and transport, recently the data generated by travelers has also been shown up as a new key factor. Namely, more and more information arise which are mostly used separately or not utilized at all at the present time. For the future, huge opportunities open up concerning transport management by exploiting transport related big data. As an illustrative example, one can mention data fusion methods which enable more reliable traffic modeling and forecasting by applying different ”data crumbs” [Tettamanti et al. 2014b]. The ongoing transport research generally focuses on the implementation of intelligent transport systems (ITS). In the ITS concept intelligent infrastructures must be also emphasized which build up a complex traffic network together with the partly or fully autonomous vehicles. Another relevant research is the calculability of everyday life and therefore that of transport needs. Barabási [2010] investigated the predictability of future human mobility based on the observation of cellular phone locomotion among others [Tettamanti and Varga 2014]. These research directions contribute to a more extensive understanding and a better management of transportation processes. Definitely, the development of autonomous vehicles will strongly modify the transport needs and traffic behavior parameters which will finally trigger the emerging intelligent transport infrastructure.

1.4 The Holistic Approach: Automated Mobility Services

Automation in road public transportation has been facilitated by the emergence of infocommunication and vehicle technologies. The widely accepted definitions of automation level focus solely on the driving aspects of vehicles. However, automation covers even more fields: service planning and management, vehicle and traffic control, and passenger-handling functions [Csiszár et al. 2019; Földes et al. 2021].

A holistic approach is needed for automated mobility innovations in light of trends in automated vehicles. This is because current transportation systems are transforming into integrated mobility services based on advanced information management and automation. Novel assess-

	Level 1	Level 2	Level 3	Level 4
Driving control	By human driver; driving might be assisted indirectly (e.g., automatic warning)	Driving is assisted (e.g., braking in emergency); human driver has overall control	Majority of driving is automated (e.g., adaptive cruise control); human driver can intervene	Driving is full automated; human is only present in traffic control room as supervisor
Fleet control	Interaction of dispatcher and driver based on navigation system	Specific fleet management system using navigation and onboard unit; driver is instructed automatically (e.g., delay warning)	Automatic fleet management, i.e., cruise control considering requirements of timetable; driver only monitors vehicle	Dynamic fleet management is operated entirely autonomously; human is only present as supervisor
Traffic control (infrastructure)	Conventional infrastructure; digital maps with static data; mostly fixed-time signaling without public transport priority; driving is regulated by signals	Digital information is provided to vehicles (dynamic and static infrastructure information); adaptive signaling with public transport priority for dedicated intersections	Infrastructure is capable of perceiving and providing traffic situation directly; networkwide traffic-responsive system; driving is regulated by traffic signals or automatic cruise control	Real-time information on vehicle movements, infrastructure guides vehicles; human is only present in traffic control center as supervisor

Figure 1.2. Automation levels of vehicle and traffic control functions (source: Földes et al. [2021])

ment methods and automation levels for urban motorized mobility services make it possible to assess the operational functions, such as service planning and management, vehicle and traffic control, and passenger-handling functions, jointly in a novel, comprehensive way. Hence, mobility services can be analyzed and compared with possible areas of development and their automation potential. An example for the holistic approach for automated level definitions is shown by Fig. 1.2.

[Földes et al. 2021] introduced a method to be applied for the handling of complex automation levels, which gives a more comprehensive assessment of new transport technologies and mobility services. The method was demonstrated through specific, currently available mobility services and found that several functions (e.g. vehicle-passenger assignment, dynamic pricing, entitlement checking, handling of boarding process) have significant development potential. Consequently, human interactions in a future mobility service may be limited merely to supervisory roles. The more flexible a service is, the more functions need to be automated. Developments should be focused on whole function categories, rather than specific single functions. Accordingly, complex methods for system modeling and engineering are required now more than ever before.

Also as a part of the holistic approach of traffic engineering, one must consider the influencing as an effective tool to form future road transport, i.e. beside the direct control tools we can influence the traffic demand itself. Accordingly, Esztergár-Kiss et al. [2021] introduced a persuasive technique to promote sustainable daily traveling supported by an effective web application.

1.5 Overview of the Thesis

The structure overview of the thesis is summarized by Fig. 1.3.

In the introductory part (Chapter 1) background is discussed by enlightening the disruptive evolution of automated road traffic and the expectation of future traffic. Chapter 2 presents the contributions of *Thesis 1*. Advanced traffic estimation methods are introduced to cope with the problem of fusion of heterogeneous traffic data. Moving horizon estimation is proposed for roundabout traffic flow estimation. Uncertainty modeling is discussed in the context of urban road traffic. Artificial intelligence based spatial extension of traffic sensors is developed. Chapter 3 discusses the contributions of *Thesis 2*. Impacts of automated vehicles are investigated in relation with the conventional traffic flow modeling in urban and freeway traffic networks. Both the effect of penetration rate and the different levels of autonomy are studied. Chapter 4 introduces the contributions of *Thesis 3*. A dynamic road pricing based control scheme is proposed for real-time control in urban road traffic network. The solution of the nonlinear

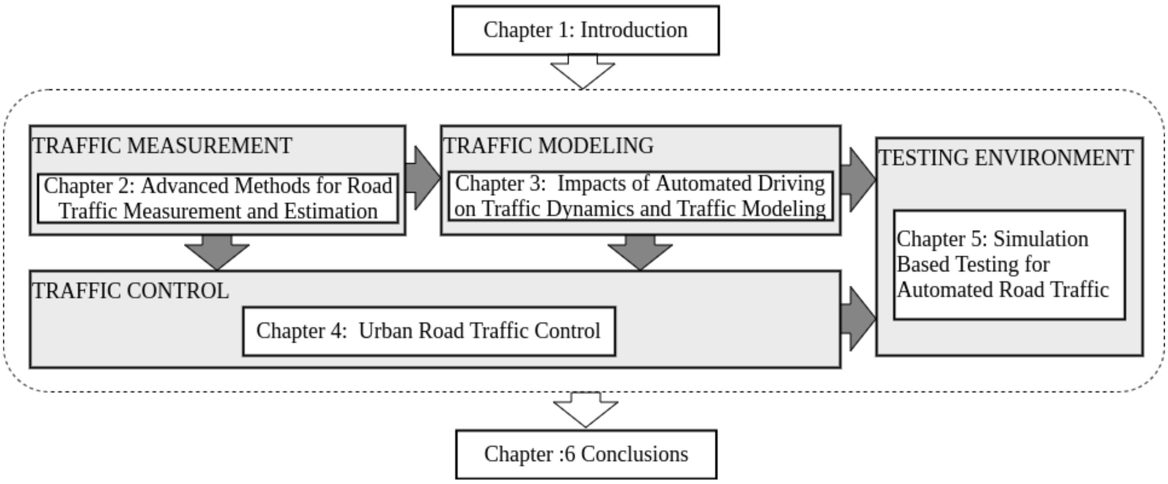


Figure 1.3. Structure overview of the thesis

predictive control intends to respond to the travelers' utility functions. A routing method for public buses is developed with dynamic paths between consecutive bus stops in order to meet the timetable based operation criteria. Distributed operation based traffic controller concept is created providing the possibility of operating wireless traffic signal controller. Chapter 5 presents the contributions of *Thesis 4*. A real-time calibration method is proposed for microscopic traffic simulation providing a practical framework for digital twin of real-world traffic dynamics. Co-simulation framework is also developed based on validated traffic simulators. Novel traffic control system is designed for test track especially focusing on a fully flexible control opportunities together with V2X/I2V technologies. Finally, Chapter 6 concludes the thesis results.

Chapter 2

Advanced Methods for Road Traffic Measurement and Estimation

The purpose of road traffic detection is twofold. On the one hand, automatically sensed traffic data serves for real-time traffic management [Papageorgiou 2004], e.g. actuated traffic light control, congestion monitoring, route guidance. On the other hand, road traffic infrastructure planning and development are also based on measured or manually counted traffic data.

In our days, a plethora of road traffic data are continuously collected producing historical and real-time traffic information as well. The available information, however, arrive from inhomogeneous sensor systems, possibly with incomplete or biased data. Accordingly, this section introduces data fusion and data filtering methodologies specialized for road traffic state measurement and estimation for the purpose of practical application. Another important way of utilization of traffic data is when it is applied for further modeling purposes (e.g. modeling traffic induced emission based on traffic data solely [Gressai et al. 2021; Kovács et al. 2021]) or for control goals [Tettamanti 2013].

2.1 Sensors for Road Traffic Data

Measurement systems applicable to road traffic data collection can be classified as:

1. traditional and
2. alternative sensor techniques.

Traditional techniques are sensors that have been developed to measure road traffic parameters. These devices are, e.g. loop-detector, magnetic detector, camera. Unlike the previous ones, alternative sensor technologies are not originally developed to measure road traffic parameters, even though they can provide these types of information, e.g. fleet management systems, cellular and GNSS data of mobile telephones, etc. The features of different measurement types are described in this section.

Traditional road traffic sensors are directly developed for accurate traffic data measurement. Basically, cross sectional measurements can be done on the traffic road network [Tettamanti et al. 2016b]. Therefore, time dependent parameters can be obtained. These are time occupancy, time headway, volume and time mean speed. By using a single cross-sectional detector, time mean speed can only be estimated (by calculating with a mean vehicle length). If more detectors are placed after each other within a short distance (usually on motorways), time mean speed can be measured, not just estimated, therefore the results are more precise. Even though video image processing systems are quite developed, these devices usually operate as virtual loop-detectors. The advantage of this method is that the number and size of placed virtual loop-detectors can be freely set. Therefore, some spatial parameters such as spatial occupancy and density can be calculated. In general, well-performing measurement systems can be built up by using traditional sensors (after adequate calibration and smoothing) for continuous measurement. The drawbacks are the high maintenance and installation costs that make it unrealistic to set up a system covering all important roads of a town. Another constraint is that measurements take place only on separated cross sections. In other words, traditional sensors are not capable of measurements that cover a large area, e.g. OD data or travel time data cannot be obtained by them.

Alternative sensor technology has become a real option for traffic parameter estimation due to IT developments of the last few years [Lana et al. 2018]. Nowadays, numerous historic or real-time databases exist that have relevant information for road traffic parameter estimation although they have not been directly generated for traffic estimation purposes. Even though these data are not available yet commonly for business or organizational reasons, they might be well applied for traffic control or service purposes. One of the most common data types is the GNSS data of vehicles that are part of a fleet management system. These information make real-time monitoring possible. During the operation of Floating Car Data (FCD) systems, an on-board unit provides data of the vehicle, e.g. actual speed, position or other information which are important basically for the operator, such as the fuel consumption. From these traffic data travel times, OD and route information can be estimated accurately. FCD are collected in numerous systems day-by-day either in private or public sector. Well-known examples for FCD data are Tom Tom Move or Waze application. An example for public transport FCD is the Hungarian FUTÁR project being developed for public transportation of Budapest. Another way of estimation is using Floating Mobile Data (FMD) that contains position data of traveling mobile phones. FMD can be divided into 2 classes: client-side and server-side information systems. Client-side FMD is collected from applications typically running on smartphones for which users get traffic information in exchange. The other way of gathering FMD information is using server-side technologies. The basis of this method is the observation of traveling mobile phones, but only at the operator-side. All of the cellular signals generated by users are automatically observed by the operator through base stations. Therefore mobile phones can be treated as detectors that do not require additional development of the infrastructure. Nevertheless, processing these rough data requires special algorithms [Tettamanti and Varga 2014]. The most important advantage of server-side FMD is the measurement capability on the whole network, e.g. estimation of OD matrices. Furthermore, the estimation of velocity is also possible only by using network mobile phone data (without GNSS data) [Nokia Solutions and Networks OY et al. 2014]. Bluetooth-based vehicle detection is another radiofrequency-based technology that is already applied in few cities [Qing 2011]. The system is able to estimate traffic parameters by following unique identifiers (MAC address) of wireless devices situated in vehicles.

The purpose of combined sensor technologies is based on the opportunity that fusion of different data types results in a better outcome than using only one type of sensor data. Integrated use of different measurement systems means data fusion. Mitchell [2007] defines data fusion as follows: "The theory, techniques and tools which are used for combining sensor data, or data derived from sensory data, into a common representational format. In performing sensor fusion our aim is to improve the quality of the information, so that it is, in some sense, better than would be possible if the data sources were used individually."

Data fusion can help all intelligent transportation subsystems to improve [El Faouzi et al. 2011]. The functions of these subsystems are usually measurement, estimation, forecasting, control and information collecting or providing [Qing 2011].

- Measurement systems: Data fusion techniques ensure standard platform for data having different semantics and syntax. The next step is the development of estimating and data completing algorithms that can also analyze those parts of the network from which in a current time step no sensor information have been collected.

- Information systems: Providing information for travelers and drivers is a basic criterion of transportation. Modern navigation systems are provided with traffic and road state information.
- Estimation and forecast: The knowledge of the present state of the network is crucial for traffic control. The result can be even better if the future state can also be forecast.
- Network traffic control: Besides providing information for travelers, the most important factor of traffic data fusion is providing data for traffic network control. In this case the state of traffic can be optimized based on traffic volume and OD data knowledge.

Certainly, these systems do not always operate separately. Moreover, they cannot often be separated, since the systems support or complete each other in many cases.

Many articles have been written recently that have introduced different data fusion techniques. The most common used data fusion technique among researchers is the Kalman Filter [Kalman 1960] with its variants (e.g. the extended Kalman Filter). The first time it was applied for road traffic was in 1972 by Szeto and Gazis [1972]. The basic theory of this research is that most analytic traffic models can be generated by state space representation. One of these solutions is described in Chu et al. [2005] claiming that even only measuring by more loop-detectors can meet the requirements of sensor fusion. All the same, some other researches use inhomogeneous data sources to improve estimation quality. In Herrera and Bayen [2007] the traditional cross sectional measurement is combined with moving sensors that actually means following GPS based trajectories of some specific vehicles. In this case from GPS data of moving vehicles, by using the conservation equation and the macroscopic model, density and speed of a specific location can be determined. Another way of traffic parameter estimation with low computing capacity is represented by interpolation techniques. These are similar to convolution image filtering techniques. A smoothing convolutional filter technique was introduced in Treiber and Helbing [2002] by using the interpolation technique in order to fuse stationary data (loop-detectors) and non-stationary (FCD) measurements. Previous techniques exploit the relationship between the measured data and the traffic model. During using data fusion techniques, by weighting the traffic model and the measured data, a balance can be set among them.

2.2 Switching Kalman Filter for Sensor Fusion

Different sensors produce inhomogeneous data during road traffic measurements, i.e. different reliability and diverse sampling frequencies. The Switching Kalman Filter can be applied to fuse these data; therefore, even the continuously changing number of sensors does not cause any difficulties, at the same time the benefits of Kalman Filter can be exploited.

2.2.1 Switching Kalman Filter for Travel Time Estimation

The Kalman Filter, introduced in Appendix A, can be directly applied for sensor fusion. The only condition is that correlation among different measurement noises is not permitted. If the results of sensor measurements are temporarily available, the state space representation of travel time estimation is:

$$x(k+1) = x(k) + v(k), \tag{2.1}$$

$$y(k) = Cx(k) + z(k), \tag{2.2}$$

where 2.1 describes the system dynamics based on the random walk model, that is $A = I$ and $B = 0$. The reason for it is the unknown changing behavior of travel time, therefore it is considered as a simple random process. Since in the example presented in this thesis travel time is estimated only for a single link, $x(k)$ and $v(k)$ are scalar variables. At the same time, $y(k)$ in the measurement equation describes a system of equations. The dimension of $y(k)$ equals the number of sensors. In the example, two different sensors, loop-detector and FCD are involved measuring travel time:

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} x(k) + \begin{bmatrix} z_1(k) \\ z_2(k) \end{bmatrix}. \tag{2.3}$$

The Kalman Filter can estimate state (i.e. travel time) $x(k)$ directly from measurements $y_1(k)$ and $y_2(k)$ that are automatically weighted according to their covariance. It is followed by their fusion that results in an integrated state estimation.

The Switching Kalman Filter (and also the basic Kalman Filter) can be applied if the system is observable, that is:

$$rank \begin{bmatrix} C^T_{\rho(k)} \\ C^T_{\rho(k)} A_{\rho(k)} \\ \vdots \\ C^T_{\rho(k)} A^{n-1}_{\rho(k)} \end{bmatrix} = \dim(x). \tag{2.4}$$

If no measurement data is available for a current time step, this requirement is not met. If the data of at least one sensor is at service, the requirement is met. Considering the frequency of sensor measurements and the number of sensors, only spatially fixed sensors are able to measure continuously by all means. As a consequence, at least one cross sectional loop-detector is required. On major roads of urban networks this might be usually available. Any other sensors (e.g. FCD, FMD, Bluetooth, etc.) are considered as potential data sources in the network, whereas their permanent operation is not a requirement. The main drawback of this approach is measuring travel times by cross sectional detectors, which obviously cannot be done directly. Therefore simulations have been done in a modeled environment in which it is possible to determine the mean travel time on a link within typical traffic volume conditions. The result can be either a function ($T = f(Q)$) or a simple determination of volume scales. In that case, as an example, a mean travel time can be determined for 0-200 PCU/h, another one for 201-400 PCU/h and so on. PCU means the number of vehicles expressed in Passenger Car Unit, i.e. the different types of road vehicles are expressed in the ratio of the private car [Lay 2009].

Travel times determined for links within different volume conditions during simulations can be applied as measurement results. For a volume data measured by a real detector, a simulation-based mean travel time can be assigned.

As a conclusion, it can be stated that loop-detector data are available in every measurement period, but they are not that reliable, because travel time is not directly measured. In contrast, FCD is much more reliable, but these data is not generated every time. This make the filter operate in the following way. If loop detector data is available, which is always true, the estimation has a higher uncertainty. If floating car data is also available, which is not always true, the estimation is much more accurate. Therefore, FCD are considered as a set of very reliable measurements, between which the state of traffic is estimated based on the loop-detector data with higher uncertainty.

The switching system applied in this thesis based on Eqs. 2.1 and 2.2 is as follows

$$\begin{aligned} x(k+1) &= x(k) + v(k), & (2.5) \\ y_{\rho(k)}(k) &= C_{\rho(k)} x(k) + z_{\rho(k)}(k), & (2.6) \\ \rho(k) \in S &= \{1, 2\}, & (2.7) \end{aligned}$$

where state variable $x(k)$ represents the estimated travel time of a specific link. Switching signal $\rho(k)$ has an effect only on the measurement equation. Set S contains four possible different measurement combinations using two different sensor types (loop-detector, FCD). These are shown in Table 2.1.

Table 2.1. Different measurement configurations

$\rho(k)$	$y_{\rho(k)}(k)$	$C_{\rho(k)}$	$z_{\rho(k)}(k)$
1	$\begin{bmatrix} T_1^{loop} \end{bmatrix}$	$\begin{bmatrix} 1 \end{bmatrix}$	$\begin{bmatrix} \sigma_1 \end{bmatrix}$
2	$\begin{bmatrix} T_1^{loop} \\ T_2^{FCD} \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix}$

Measurement vector $y_{\rho(k)}(k)$ contains the combination of travel times measured by sensors of different types. Note that the elements of $C_{\rho(k)}$ are always 1, according to the number of sensor measurements available, hence it satisfies Eq. 2.4. The Switching Kalman Filter switches according to the signal $\rho(k)$ that is a known value representing the set of sensor types that provided data in the measurement period. (See Table 2.1.)

Loop-detector measurements are guaranteed in every time step, even if there is no vehicle on the link. Therefore, the minimum size of matrices is 1×1. The size of matrices $y_{\rho(k)}(k)$, $C_{\rho(k)}$, and $v_{\rho(k)}(k)$ varies in connection with the number of sensor types providing information in a time step. The method can be extended to n sensor types. Table 2.2 shows how these measurements can be combined. The bottom line represents the case when all n sensor types provide information in a time step.

Table 2.2. Measurement configurations with n sensors

$\rho(k)$	$y_{\rho(k)}(k)$	$C_{\rho(k)}$	$z_{\rho(k)}(k)$
1	$\begin{bmatrix} T_1^{sensor_1} \end{bmatrix}$	$\begin{bmatrix} 1 \end{bmatrix}$	$\begin{bmatrix} \sigma_1 \end{bmatrix}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
q	$\begin{bmatrix} T_1^{sensor_1} \\ T_i^{sensor_i} \\ T_j^{sensor_j} \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$	$\begin{bmatrix} \sigma_1 \\ \sigma_i \\ \sigma_j \end{bmatrix}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
s	$\begin{bmatrix} T_1^{sensor_1} \\ T_2^{sensor_2} \\ \vdots \\ T_i^{sensor_i} \\ T_j^{sensor_j} \\ \vdots \\ T_n^{sensor_n} \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$	$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_i \\ \sigma_j \\ \vdots \\ \sigma_n \end{bmatrix}$

For a precise estimation measurement noise $v_{\rho(k)(k)}$ is assumed to be known. Theoretically, this can be determined based on the standard deviation of measurements. Travel times calculated by the cross sectional detector have the highest standard deviation. The noise of FCD/FMD measurements is much lower, which results in more precise estimations. As a consequence, measurement estimation is expected to be much reliable during the rush hours (when FCD/FMD are probably available) and less out of them. However, during off-peak periods travel time data can be calculated from static databases, since the effect of traffic is low. The described data fusion estimation method is shown by Fig. 2.1.

2.2.2 Simulation Example

The elaborated method was tested in a simulation environment in PTV VISSIM microscopic traffic simulator [Bede et al. 2020]. In order to be able to apply the Swithing Kalman Filter, the average travel time and its standard deviation should be known after each measurement period. The example is shown with loop-detector data and FCD, including the description of

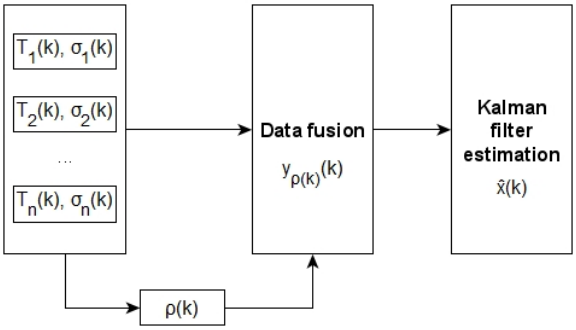


Figure 2.1. The sensor fusion method for the estimation of mean travel times on urban links

the data processing steps of the two sources. Simulation network was generated by modeling a specific test link on Villányi út and its network area around BAH-csomópont in the city of Budapest, Hungary. The test network was generated by taking into consideration the links that have influence on the traffic flow of the test link (Fig. 2.2). Signal programs were set according to morning weekdays plans of the real network, cycle time was 90 seconds. Loop-detectors in the simulation environment were also installed at the same location as it is in the real network. One detector was set at each lane of the test link. It is an important issue, as some traffic parameters vary in connection with the position on the link. Figure 2.3 shows an example for this.

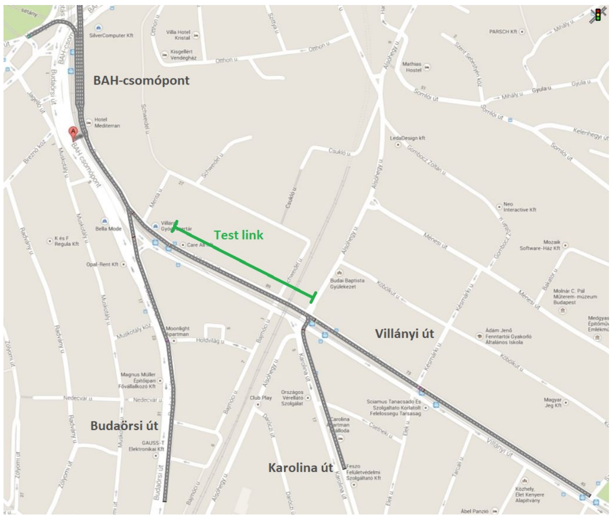


Figure 2.2. Test network in PTV VISSIM

Overall 3600 ninety-second-long simulation periods were evaluated on a filled network with different vehicle inputs ranging from almost empty links to gridlock situation. Traffic volume and time-occupancy, which is the proportion of time when any vehicle is over the detector, were measured by the loop-detector in PTV VISSIM in 90 seconds. Volume was calculated as the sum of the values of two detectors, whereas occupancy is the average of the two measurements in the cross section.

Travel time can be derived from loop-detector measurements by generating a look-up-table that contains volume and occupancy data in the cross section as input and an estimated average time plus its standard deviation as the output. These can be composed by running simulations or processing historical data.

In order to reach accurate estimation results, tuning of the Switching Kalman Filter is an important step. Parameters of Q and R of the filter in different operation modes can be

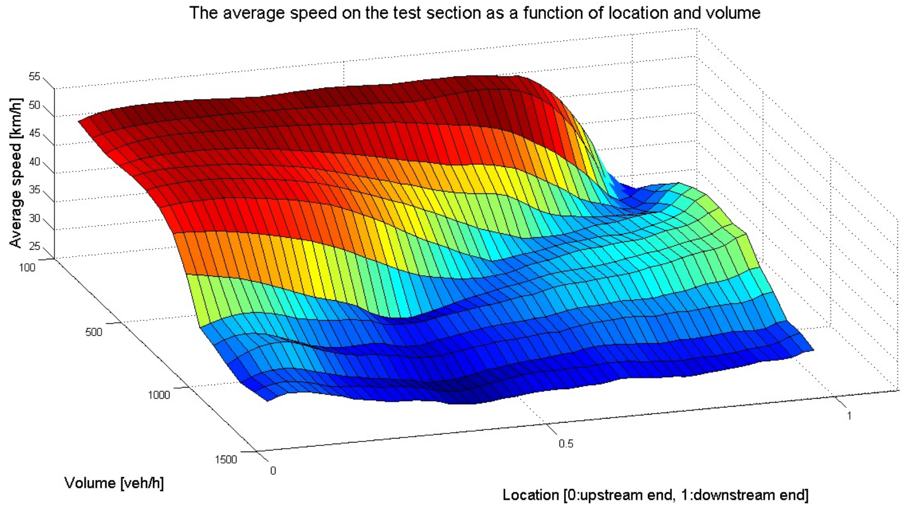


Figure 2.3. Average speed as the function of volume and location

adjusted according to Böker and Lunze [2002]. Q and R can be treated as tuning parameters using practical assumptions. Matrix R contains the standard deviations of measurement results in the two configurations as follows. If $\rho(k) = 1$, i.e. only loop-detector data is available at period k, then $R_1(k) = \left[\sigma^2_{loop}(k)\right]$. If $\rho(k) = 2$, i.e. both loop-detector data and FCD are available at the period, then $R_2(k) = \begin{bmatrix} \sigma^2_{loop}(k) & 0 \\ 0 & \sigma^2_{FCD}(k) \end{bmatrix}$. Matrix Q changes its form similarly and the its values are based on empirical tuning.

The Switching Kalman Filter was implemented in Matlab software. An operation example is depicted by Fig. 2.4. The uncertainty of loop-detector-based travel time estimation is much higher than FCD-based, which makes the filter primarily believe the FCD results. The effect of a floating car measurements last for 4-5 periods, after that the filter gradually returns to loop-detector data.

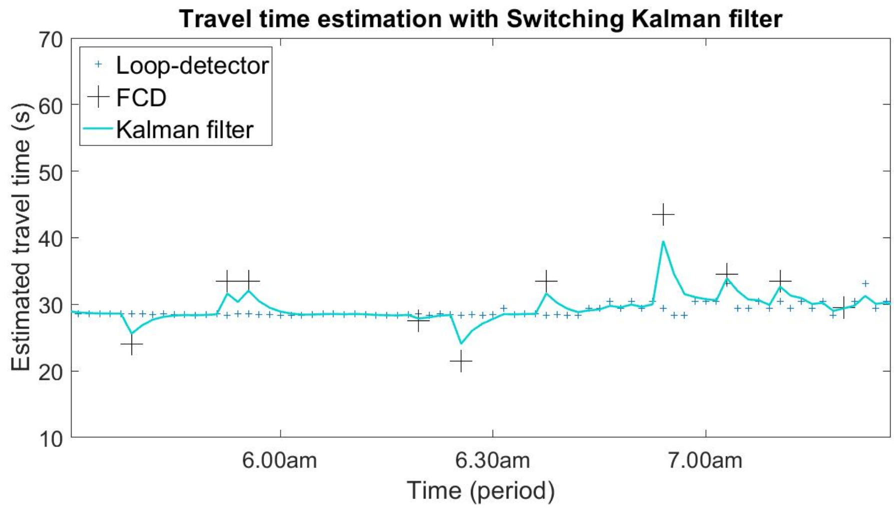


Figure 2.4. Operation example of the Switching Kalman Filter

2.3 Moving Horizon Estimation of Traffic Flows in Roundabouts with Missing Observation

A roundabout is a circular type of intersection in which vehicles are permitted to flow in one direction around a central island. Fig. 2.5 shows possible turning movements in a circular intersection for vehicles arriving at the entrance of Leg 1. V_{1j} denotes the traffic flow from entrance 1 to exit j, whereas $V_{1,in}$ and $V_{1,out}$ are the total traffic volumes entering and exiting at Leg 1. Using the notation in Fig. 2.5, turning rates can be expressed as follows (from entrance i to exit j):

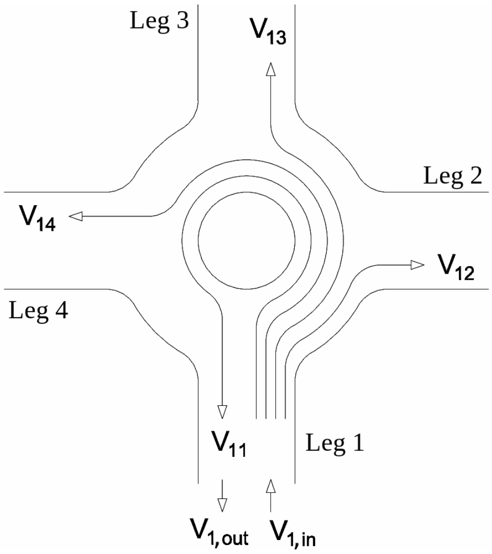


Figure 2.5. Turning movements in a four-legged roundabout

$$x_{ij} = \frac{V_{ij}}{\sum_{j=1}^{n_D} V_{ij}} = \frac{V_{ij}}{V_{i,in}}, \tag{2.8}$$

where n_D is the number of exits.

Observing turning rates in roundabouts is a real issue in traffic engineering practice due to the special geometry and the size of this type of junctions. Therefore, turning flows are generally counted by human resources, which is quite costly (typically, more than one person is needed to perceive all movements). Automated methods are also available for turning flow counts. For instance, it can be carried out by installing cameras on the spot and evaluating the footage subsequently by using artificial intelligence [Taylor et al. 2016]. However, placing cameras or shooting aerial videos with drones [Salvo et al. 2014] are also costly. Moreover, the legal background of drones is not yet clarified for this to be a practical alternative [Budinska 2019]. Another suggested method is to use vehicle trajectory data which has a drawback that beside input and output traffic the circular flows must be measured as well [Eisenman and List 2005].

Traffic volumes at cross-sections can be straightforwardly measured manually or with the help of a wide variety of traffic sensors, e.g. inductive loop detectors, cameras, ultrasonic detectors. Thus, for roundabout turning rate detection, a hybrid solution is suggested, i.e. using simple cross-sectional detection (manual or automatic) combined with advanced estimation procedures. Counting traffic on the legs of a roundabout and then adequately estimating turning rates based on the collected data has the potential to substitute laborious turning flow counts in a cost-effective way.

As a cost-effective solution to this problem, a hybrid solution is suggested, i.e. using cross-sectional detection combined with Moving Horizon Estimation (MHE, see Appendix C).

In case of estimating turning rates in roundabouts based on the traffic flow on the legs, the elements of the state vector are the turning rates [Tettamanti et al. 2019a]. The applied dynamical model is the random walk model, meaning that in each period the state variable takes a random step which is independent of the previous state value. In this approach, matrix A in state equation (A.1) is an identity matrix, and matrix B can be substituted with 0 as there is no control vector. Thus, the dynamical model is simplified to:

$$x(k+1) = x(k) + v(k). \tag{2.9}$$

The state vector to be estimated is as follows:

$$\hat{x}(k) = \begin{pmatrix} \hat{x}_{11} \\ \hat{x}_{12} \\ \vdots \\ \hat{x}_{n_O,n_D} \end{pmatrix}, \tag{2.10}$$

where $\hat{x}_{ij}$ denotes the estimated turning rate from entrance i ($i = 1,\ 2,\ ...,\ n_O$) to exit j ($j = 1,\ 2,\ ...,\ n_D$). $C(k)$ in measurement equation (A.2) contains the measured entering traffic flows (marked by q_m where m denotes the m^{th} leg of the roundabout).

$$C(k) = \begin{pmatrix} q_1(k) & & & q_2(k) & & & & q_m(k) & & \\ & \ddots & & & \ddots & & \cdots & & \ddots & \\ & & q_1(k) & & & q_2(k) & & & & q_m(k) \end{pmatrix}. \tag{2.11}$$

Thus, exiting traffic flows appear in vector $y(k)$ as measured parameters.

2.3.1 Evaluation and Benchmark of the MHE Method on Real-World Data

In order to validate and benchmark the estimation algorithms, real-world test fields have been investigated. By using drone technology, video recordings were made at two different roundabouts in Kecskemét, Hungary (Roundabout 1 at GNSS coordinates: 46.92971, 19.663997; Roundabout 2 at GNSS coordinates: 46.881503, 19.707799). Then, based on the drone footage real turning movement volumes were counted as ground truth.

In accordance with the drone's maximal flight time, 26-minute aerial video recordings were taken at the two four-legged intersections. The counts took place at different times of the day (morning and afternoon). The 26-minute counts are adequate to be divided into 1, 2, and 5 minute intervals (in the latter case, only 25 minutes are examined). The traffic count was therefore conducted for 1-minute interval, so that 2 and 5 minute intervals could be calculated afterward.

The tendency of error measures is similar in all cases, irrespective of the location or the time of the day. Therefore, for the sake of transparency and a more general result, error values are averaged over the different traffic counts. The average values form the basis for the comparison of different estimation procedures.

During the evaluation of estimation procedures the Mean Absolute Error (MAE) error metrics was applied:

$$MAE = \frac{\sum_{k=1}^{n} |\hat{x}_k - x_k|}{n}, \tag{2.12}$$

where n is the number of samples (intervals), $\hat{x}_k$ is the state estimation in interval k, and x_k is the actual state. In the case of turning rate the MAE is a unitless value between 0 and 1. Based on the average MAE values, it can be stated that the longer the interval, the more accurate the estimation. Additionally, the MAE values for all MHE horizon lengths are highlighted by Fig. 2.6. It is observable that higher horizon numbers result in less estimation performance which is caused by the simple system modeling (random walk model in Eq. (2.9)).

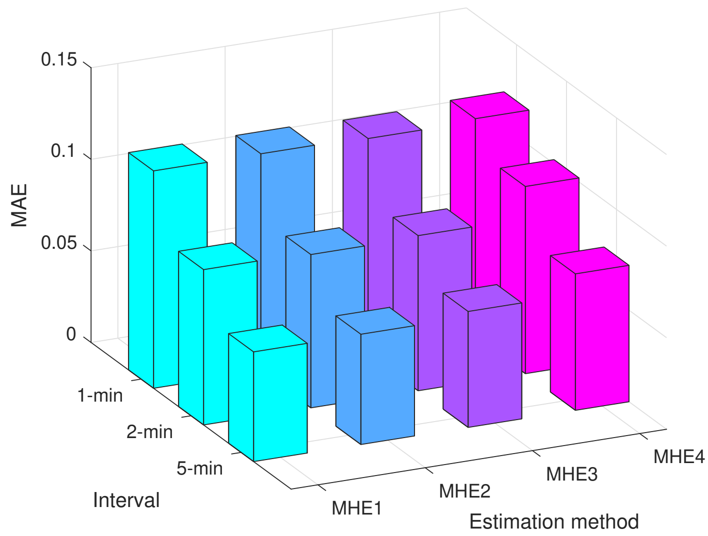


Figure 2.6. MAE values for all MHE horizon lengths and intervals

The 5-minute interval led to the smallest errors in the case of every examined method. A possible explanation for this is the following. If the sampling intervals are short, it is more frequent that a specific turning movement is not executed during that brief time period. This can result in sharp fluctuations in turning rates, which is harder to track for an estimator. This implies that 1 or 2 minutes sampling intervals are not suggested to be applied in this practical problem.

The order of estimation procedures with 5-minute intervals based on MAE values is the following: MHE1, MHE2, cKF-P (constrained Kalman Filter while $W = P(k)$, see Appendix B), MHE3, BP (Biproportional Procedure, see Appendix D), cKF-I (constrained Kalman Filter while $W = I$), KF (Appendix A), MHE4.

In the case of 5-minute intervals, the MHE and the Kalman Filter with constraints outperforms the BP procedure and the unconstrained Kalman Filter. The performance of the MHE decreases as the horizon length increases. It is also observable that the shorter estimation intervals of 1 or 2 minutes provide higher errors in every estimation procedure. This clearly means that on longer time intervals, the algorithms can result in smoother estimations.

2.3.2 Turning Flow Estimation in Case of Missing Detection

The temporary unobservability of certain traffic flows represents a likely situation in real practice due to the potential operational failure of traffic detectors. This is also called as intermittent sensor data problem. The proposed estimation procedure requires full observability of traffic flows in the roundabout intersection legs. Therefore, the required detection can be replaced by exploiting the previous step's estimation $\hat{x}(k-1)$ as a practical solution based on the definition of the measurement equation $y(k) = C(k)x(k)$ [Kalman 1960]. For example, in case of missing value of $y_1(k)$ the following substituting calculation is applied:

$$y_1(k) = \begin{pmatrix} q_1(k) & 0 & 0 & 0 & q_2(k) & 0 & 0 & 0 & q_3(k) & 0 & 0 & 0 & q_4(k) & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \hat{x}_{11}(k-1) \\ \hat{x}_{12}(k-1) \\ \vdots \\ \hat{x}_{44}(k-1) \end{pmatrix}. \tag{2.13}$$

Similarly, another example for missing value replacement on $y_3(k)$ is:

$$y_3(k) = \begin{pmatrix} 0 & 0 & q_1(k) & 0 & 0 & 0 & q_2(k) & 0 & 0 & 0 & q_3(k) & 0 & 0 & 0 & q_4(k) & 0 \end{pmatrix} \begin{pmatrix} \hat{x}_{11}(k-1) \\ \hat{x}_{12}(k-1) \\ \vdots \\ \hat{x}_{44}(k-1) \end{pmatrix}. \tag{2.14}$$

In order to investigate the effect of missing observation (Fig. 2.5) two special scenarios have been simulated assuming no direct traffic outflow detection at certain roundabout legs:

- Scenario 1: no detection of $y_1(k)$ at Leg 1;
- Scenario 2: no detection of $y_1(k)$ and $y_3(k)$ at Leg 1 and 3, respectively.

In both scenarios the MHE1 algorithm with 5 min sample time was applied considering the baseline when no missing observation was used. The missing detections were substituted according to Eq. (2.13) and (2.14). The results (Table 2.3) show that in case of only one missing observation point, the error metrics remained in an acceptable range.

Table 2.3. Performance in case of missing detection on intersection leg in case of MHE1 algorithm with 5 min sample time

Missing detection points	MAE	Change in MAE
0	0.0599	0
1 (Leg 1)	0.0678	0.0079
2 (Leg 1 & 3)	0.1255	0.0656

2.4 Uncertainty Modeling in Urban Road Traffic

Advanced dynamic control involves traffic models which contain the most important characteristics of the network, i.e. topology and further dynamic parameters. In general, these models then become bases for model-based control solutions. With an appropriately chosen and parameterized model, future traffic states can be predicted resulting in an optimal traffic control. However, the applied traffic model can be biased by non-measurable vehicle flows. In the sequel, uncertainty modeling is investigated specifically focusing on urban road traffic. Based on the proposed uncertainty approach, robust schemes can be applied for traffic estimation or control.

2.4.1 Uncertainty Modeling in General Urban Road Link

In a general urban road link (see Fig. 2.7) between two signalized intersections (M and N) different potential traffic streams can be observed.

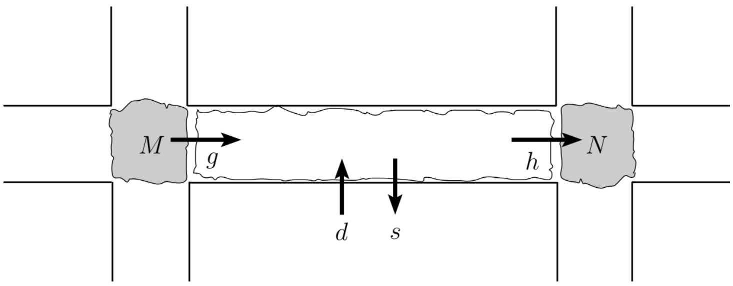


Figure 2.7. Traffic flows in a link

g and h represent entering and exiting vehicle flows. In an advanced traffic management system g and h are usually measured by detectors. Contrarily, entering and exiting flows d and s are not controllable and might not always be measured. d and s are able to induce uncertainty in the traffic modeling and consequently to corrupt the traffic control. In urban road traffic network uncertainties may appear for several reasons. Unexpected traffic fluctuations can be caused for instance by parking garages or on-street parking. Moreover, demand flow (entering at the boundary of the network) may increase the uncertainty if it is imprecisely determined, e.g. based on inappropriate historical data. These potentially ambiguous traffic flows are depicted by the circles in Fig. 2.8.

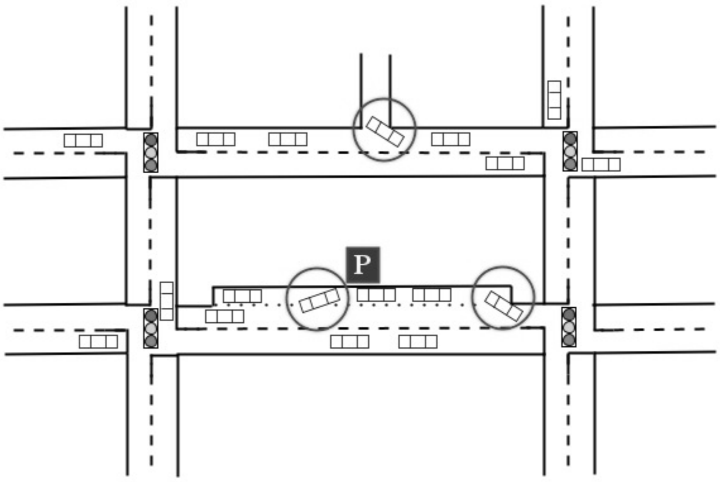


Figure 2.8. Potential uncertainties in urban road links

Disturbances d and s are usually assumed to be available as constant nominal values for traffic light control strategies, see Aboudolas et al. [2009, 2010]; Diakaki et al. [1999]. In these contributions s is expressed as the ratio of g and considered fixed and known. Similarly, input d is defined as a constant nominal demand. In de Oliveira and Camponogara [2010] exiting traffic s and demand d are lumped together creating a single (and known) disturbance term. In Lin et al. [2011], a mixed integer linear programing based MPC is proposed which is also based on nominal traffic network model with known traffic demand. Naturally, these approaches use practical assumptions as the overall and accurate measurement of all vehicle flows would lead to enormous cost increase.

If certain vehicle flows are not precisely available (e.g. by measurement), a robust approach can be applied for modeling and control purposes. Ambiguous traffic is proposed to be handled as time-varying but bounded model-mismatch based on the store-and-forward macroscopic modeling. The state equation for generalized urban link z (see Fig. 2.7) can be formulated as follows:

$$x_z(k+1) = x_z(k) + T\big(g_z(k) - h_z(k) + d_z(k) - s_z(k)\big), \tag{2.15}$$

where $x_z(k)$ is the number of vehicles expressed in PCU, $g_z(k)$ the inflow, $h_z(k)$ the outflow, $d_z(k)$ the demand traffic, and $s_z(k)$ the exit traffic (traffic flows in PCU/h) during $[kT, (k+1)T]$. Moreover, k denotes the discrete time step index and T is the sampling time. $s_z(k)$ can be defined as the ratio of $g_z(k)$:

$$s_z(k) = \kappa_z g_z(k), \tag{2.16}$$

where κ_z is the exit rate, considered fixed and known. Thus, Eq. (2.15) can be rewritten as:

$$x_z(k+1) = x_z(k) + T\big((1-\kappa_z)g_z(k) - h_z(k) + d_z(k)\big), \tag{2.17}$$

which represents a nominal traffic model. A direct way to consider traffic ambiguity in Eq. (2.17) is to use additional terms:

$$x_z(k+1) = x_z(k) + p_z^x(k) + T\big((1-\kappa_z)g_z(k) - h_z(k)\big) + Td_z(k) + p_z^d(k), \tag{2.18}$$

where $p_z^x(k)$ denotes the uncertainty component of state $x_z(k)$ and $p_z^d(k)$ that of traffic demand $d_z(k)$ (both expressed in PCU) with the following definitions:

$$p_z^x(k) \;=\; \delta_z^x(k)\gamma_z^x x_z(k),\;\; |\delta_z^x(k)| \le 1, \tag{2.19}$$
$$p_z^d(k) \;=\; \delta_z^d(k)\gamma_z^d T d_z(k),\;\; |\delta_z^d(k)| \le 1. \tag{2.20}$$

$\delta_z^x(k)$ and $\delta_z^d(k)$ express unknown, time-varying, and bounded uncertainties. γ_z^x and γ_z^d are the uncertainty weights, not necessary constant scaling parameters; $\gamma_z^x,\ \gamma_z^d \in [0,1]$. Hence, product $\gamma_z^x x_z(k)$ denotes the maximal potential uncertainty defined relative to the state variable $x_z(k)$. Similarly, $\gamma_z^d T d_z(k)$ is the a maximal bound for demand uncertainty. The uncertainty components are only known to be bounded. This means practically that uncertainty $p_z^x(k)$ may arbitrarily vary between $-\gamma_z^x x_z(k)$ and $\gamma_z^x x_z(k)$. Respectively, $p_z^d(k)$ changes between $-\gamma_z^d T d_z(k)$ and $\gamma_z^d T d_z(k)$. In virtue of the above definitions, Eq. (2.18) can be recast as follows:

$$x_z(k+1) = \big(1+\delta_z^x(k)\gamma_z^x\big)x_z(k) + T\big((1-\kappa_z)g_z(k) - h_z(k)\big) + \big(1+\delta_z^d(k)\gamma_z^d\big)Td_z(k). \tag{2.21}$$

Uncertain traffic model (2.21) captures traffic ambiguity caused by any off-nominal disturbances, defined in terms of scaled state or demand dependent uncertainty. In fact, both uncertainty components intend to express the uncertainty of state $x_z(k+1)$, i.e. the calculated link queue length in the following time step. The modeling concept (2.21) is chosen by reason of the different types of uncertainties. $p_z^x(k)$ can typically model traffic fluctuations of on-street parking or non-measured side streets in a state dependent way. Thus, the more intensive the traffic is, the larger the ambiguous vehicle mass becomes within the traffic network, and vice versa. This approach is especially reasonable for the links where nominal demand $d_z(k)$ is negligible (zero or not available). Contrarily, $p_z^d(k)$ intends to capture uncertainty around the nominal traffic demand specifically. $d_z(k)$ is often determined from prior measurement and thus involved in the traffic models. The application of historical data for $d_z(k)$, however, may cause modeling error. Hence, traffic flows of parking garages or entering demands (at the boundary of the network) having nominal $d_z(k)$ can be augmented by uncertainty in the model via $p_z^d(k)$. The applied structure of the ambiguity given by Eqs. (2.19-2.20) expresses a state (queue) and demand multiplicative uncertainty description, i.e. uncertainty varies relative to the nominal traffic states and traffic demands. Naturally, the determination of γ_z^x and γ_z^d is of capital importance in this approach. In fact, their value can be estimated precisely enough based on prior observations in a traffic network. Moreover, γ_z^x and γ_z^d can be defined as time-varying parameters.

2.4.2 Uncertainty Modeling Extended to Signalized Traffic Network

Equation (2.21) describes a single link dynamics. In order to obtain an overall network model with traffic lights, each of the link equations of the traffic network is required. Moreover, traffic lights must be incorporated into the model by expressing traffic flows g_z and h_z as functions of the green times.

A network can be represented by directed graphs formed of nodes and arcs. Nodes $j \in J$ denote the controlled intersections and arcs $z \in Z$ the links. Equation (2.21) can be embedded therefore into a network dynamics. The equation for vehicle conservation between signalized junctions M and N is given by:

$$
\begin{aligned}
x_z(k+1) = {} & x_z(k) + p_z^x(k) \\
& + T\left((1-\kappa_z)\sum_{w\in I_M}\alpha_{w,z}\frac{S_w\sum_{i\in v_w}u_{M,i}(k)}{C} - \frac{S_z\sum_{i\in v_z}u_{N,i}(k)}{C}\right) \\
& + Td_z(k) + p_z^d(k),
\end{aligned}
\tag{2.22}
$$

where C is the cycle time and $T = C$ is applied. I_M represents the set of incoming links of junction M. $\alpha_{w,z}$ denotes the turning rate of link w entering junction M towards link z. S_w and S_z are the saturation flows representing the capacity of the outflow of the link during its green time. The values of α and S are considered known and constant in the sequel. Nevertheless, they can be assumed to be time-variant and may be continuously measured or estimated, e.g. by online state estimation method [Kulcsár et al. 2004]. $\sum_{i\in v_w}u_{M,i}(k)$ and $\sum_{i\in v_z}u_{N,i}(k)$ represent the green times of intersections M and N, respectively. v_w and v_z denote the set of green stages for links w and z.

The derived model (2.22) must satisfy state and control input constraints. This practically means that a well designed controller is able to choose suitable green times to ensure all constraints. It is evident that the maximum number of vehicles in a link (x_z^{max}) is determined by the length of link z between two junctions. Thus, the states are subject to hard physical constraints:

$$0 \le x_z(k) \le x_z^{max}. \tag{2.23}$$

u_z is limited by the following constraints:

$$u_z^{min} \le u_z(k) \le u_z^{max}, \tag{2.24}$$

$$\sum_{i=1}^{O_j} u_{j,i}(k) \le T_j^{max}, \tag{2.25}$$

where O_j is the number of stages at junction j, and $T_j^{max} = T - L_j$ (L_j is the fixed lost time resulted from the geometry of junction j).

The conservation property is a crucial point of model (2.22). Originally, store-and-forward approach was interpreted with the assumption of saturated traffic condition[1] [Diakaki et al. 1999; Gazis and Potts 1963]. Therefore, the conservation equation may fail under oversaturated traffic condition. A potential solution was provided by Aboudolas et al. [2009] by proposing a state dependent control input in the control design and allowing $u_z^{min} = 0$. This can be used to help avoiding overspilling effect, i.e. in case of oversaturated traffic. Thus, the zero length minimal green time is adopted producing the following expression instead of (2.24):

$$0 \le u_z(k) \le u_z^{max}. \tag{2.26}$$

Hence, the conservation property of the model can be guaranteed in saturated and oversaturated cases through (2.23) and (2.26) indirectly.

The application of (2.22) to an urban traffic network yields a general (vectorized) state space form by:

$$
\begin{aligned}
x(k+1) &= Ax(k) + Bu(k) + Ed(k) + Gp(k), && (2.27)\\
p(k) &= \Delta(k)(D_x x(k) + D_d Ed(k)),\ \|\Delta(k)\|_2 \le 1. && (2.28)
\end{aligned}
$$

[1]In saturated traffic the green time intervals are fully utilized. This means that arriving vehicles are always present during the green time period.

Equations (2.27-2.28) describe a Linear Fractional Transformation (LFT) with norm-bounded uncertainty model frequently used in control theory, e.g. Boyd et al. [1994]. Each term of (2.27) has a real traffic meaning. The elements of the state vector $x(k) = [x_1\, x_2\, \ldots\, x_n]^T$ represent the number of vehicles (expressed in PCU) on each of the controlled links (by traffic lights). The values of $x(k)$ may not be directly measurable but estimated by appropriate methods [Vigos et al. 2008]. State matrix A is an identity matrix. Control input vector $u(k)$ contains the green times (in seconds) for all stages. The latters' numerical values are the results of the corresponding controller actions. The number of control inputs is equal to the number of the controlled links. Matrix B consists of three basic elements: turning rates, exit rates and saturation flows. Although B is a time-varying matrix in the reality, it may be considered as a constant one with fixed nominal rates. If the involved saturation flows or turning rates are estimated real-time, B is time-variant and can be recalculated for shorter time periods (e.g. in every 15 minutes) but not necessarily for each time step. Constant B matrix is used in the following parts. Disturbance vector $d(k)$ represents the non-controlled traffic demand which is a time-varying term. In practice, however, it is more reasonable to redefine $d(k)$ for shorter time periods based on continuous real-time measurements or historical data. Coefficient matrix E is a constant diagonal matrix with T in its diagonal. $G = [I|I]$ is a non-quadratic hyper-matrix containing two identity matrices and $p(k)$ is the uncertainty vector. D_x and D_d are the constant diagonal scaling hyper-matrices containing weights, such as:

$$D_x = \left[\begin{array}{ccc} \gamma_1^x & & \\ & \ddots & \\ & & \gamma_n^x \\ \hline & \mathbf{0} & \end{array}\right], \tag{2.29}$$

$$D_d = \left[\begin{array}{ccc} & \mathbf{0} & \\ \hline \gamma_1^d & & \\ & \ddots & \\ & & \gamma_n^d \end{array}\right], \tag{2.30}$$

where $\gamma_1^x, \ldots, \gamma_n^x$ and $\gamma_1^d, \ldots, \gamma_n^d$ are predefined design parameters introduced in Section 2.4.2 and n denotes the number of states. Hyper-matrix $\Delta(k)$ is organized as:

$$\Delta(k) = \begin{bmatrix} \Delta^x(k) & \\ & \Delta^d(k) \end{bmatrix} = \begin{bmatrix} \delta_1^x(k) & & & & & \\ & \ddots & & & & \\ & & \delta_n^x(k) & & & \\ & & & \delta_1^d(k) & & \\ & & & & \ddots & \\ & & & & & \delta_n^d(k) \end{bmatrix}. \tag{2.31}$$

By substituting (2.28) in (2.27), the traffic model can be recast as a dynamics with state and demand multiplicative uncertainty under the form:

$$x(k+1) = (A + G\Delta(k)D_x)x(k) + Bu(k) + (I + G\Delta(k)D_d)Ed(k), \;\; \|\Delta(k)\|_2 \leq 1. \tag{2.32}$$

The derived traffic model (2.32) is only valid with constraints. Constraints (2.23), (2.25), and (2.26) determine two sets of linear inequalities; one set for state and one for control input. These are denoted by $\mathbb{X}$ and $\mathbb{U}$, respectively. The dynamic equation described in (2.32) is restricted along its trajectory by the sets $x_z(k) \in \mathbb{X}$ and $u_z(k) \in \mathbb{U}$ for all k.

In the previous parts, the uncertainty was demonstrated as an extension of the well-known store-and-forward traffic model. Another example for traffic uncertainty is presented by Appendix E where a nonlinear traffic model is applied for uncertainty definition. The more, a robust (H∞ Filter) estimation is demonstrated as an application example on the uncertain traffic dynamics.

2.5 Spatial Extension of Sensors in Urban Traffic Network

Network-wide traffic monitoring has increased importance in the current and future panorama due to the verge of adoption of smart mobility technologies, i.e. monitoring all links in a network is a general desired goal. However, installation and maintenance of sensors across the whole network are not cost-effective. Therefore, traffic networks are frequently suffering from the lack of well-operating and reliable traffic detectors. Accordingly, the employment of neural networks based models is suggested to virtualize the measurements on road links without detectors. The proposed method applies the measurements of monitored links as input to the deep learning model in order to estimate virtual measurements on the unmonitored road links. Several neural network models differing in architecture (Artificial Neural Network, Time Lagged Neural Network and Long Short Term Memory Neural Network) have been implemented and their hyper-parameterization were optimized using Bayesian search. The methodology is proposed for spatially extended traffic data, and consequently the quality of the monitoring system through the use of artificial intelligence. That is, using the measurements of monitored links to infer the values of unmonitored ones.

Artificial Intelligence (AI) based methods have been applied in urban traffic for a long time. Among the AI methods, Neural Networks (NNs) stand out as the most promising approach, as depicted in Do et al. [2019], which comprehend for the majority of approaches towards traffic forecasting. Regarding traffic estimation, some authors divide it into two categories: i) Temporal estimation (long or short-term prediction, being the latter more relevant and advanced) and ii) Spatial estimation (extension of traffic data links to links) [De Luca and Gallo 2017]. Temporal extension is out of the scope from this work, therefore the literature review in this topic will be omitted. Thorough literature review about this topic can be found in De Luca and Gallo [2017]; Do et al. [2019]; Lana et al. [2018].

The use of AI to solve the spatial extension task was investigated only recently. The pioneer in doing this was Gallo et al. [2016], employing shallow Neural Network to perform the extension of traffic flow in a synthetic grid network. A hand-made feature construction, which links would be considered as monitored (input) and which would be considered unmonitored (output), was performed. Further work was depicted in Iannella et al. [2017], comparing several regression methods, such as Linear Regression, Kernel Regression, Support Vector Machine, Generalized Least Squares and NN, for the same grid network.

2.5.1 Deep Learning Models Applied for Road Link Selection

The research work was supported by traffic simulations carried out in SUMO traffic simulator [Krajzewicz et al. 2012]. A grid network with 80 road links was used for the simulation. The demand patterns which generated the flows in the network were randomized and varied during the simulation. Moreover, the perimeter edges in the corners of the network were considered as possible origin destination points. The choice of the traffic network and the origin-destination points was made to match the setup used by [Gallo et al. 2016], providing a baseline for comparison. To build up the training and testing database, 30 simulations were performed, each one lasting one day. The values were aggregated in 60 seconds interval.

Although, short sampling intervals typically produce noisy data [Houpt et al. 1978], monitoring systems are usually setup for aggregation intervals between 5 and 10 minutes [Park et al. 2009]. For that reason the loop detector data was aggregated in 10 minutes intervals and smoothed by a moving average filter for less erratic values [Coifman 1996].

The related works discussed previously were mostly based on artificial neural networks with one or two hidden layers. The input of these models consists of the immediate measurement of the links with sensors. The problem formulation can be expressed in a mathematical form:

$$V_o(t) = f(V_i(t)), \tag{2.33}$$

where t is the time step, V_o is a traffic variable vector for all unmonitored links and V_i is the traffic variable vector for every monitored links and $f(.)$ is the NN mapping the input to the output.

Once past measurements are not taken into account, temporal relation between the measurements is ignored. However, it is known that traffic variables are deeply related to past values. One of the simplest NN architectures dedicated to time series tasks is the Time Lagged Neural Network (TLNN) [Werbos 1990]. In this architecture, the input construction process takes into consideration not only the instant values but an arbitrary number of past values. In this way, the formulation of the task can be expressed as follows:

$$V_o(t) = f(V_i(t), V_i(t-1), ..., V_i(t-D)), \tag{2.34}$$

where D is the maximum number of past values presented in the input of the model.

Even though this approach can yield good results, a time dependence between the time steps is not learned explicitly. To this end, Recurrent Neural Networks (RNNs) are commonly used in time depending problems [Robinson and Fallside 1987]. In this type of network the output of each layer is fed back as part of the input in the next time step. However, the classic RNN training process suffers from exploding or vanishing gradient problem [Hochreiter and Schmidhuber 1997]. Long Short Term Memory (LSTM) Neural Networks is a variety of RNN that can overcome this limitations by using gates (Input, Output and Forget).

Neural Network based methods performance can be very sensible to hyperparameter setting (e.g. number of neurons, dropout rate, regularization rate). This parameterization can be performed manually based on empiric knowledge allied with trial and error fine-tuning, even though it is a common and valid practice it does not assure an optimal solution.

Bayesian search presents a viable solution in fine-tuning deep learning models, requiring acceptable computing power and great optimization capabilities [Snoek et al. 2012]. In general lines, the performance of the model is assumed to be a Gaussian process, expressed by the surrogate function $g(.)$ dependent of the hyperparameters θ. The optimization process is defined by

$$\theta^* = arg \max_{\theta \in \Theta} g(\theta), \tag{2.35}$$

where Θ corresponds to the domain of the parameters. The guesses of θ can be made in a more informed manner, choosing the best performing point in the surrogate function $g(.)$, evaluating in the model and updating $g(.)$ iteratively until the maximum iteration or other stop criteria is met.

In the research, a Bayesian search was applied to find the optimal parameters in a given architecture with a set of number of layers. Changing the number of layers introduces new parameters to the optimization process, such as new number of neurons, activation type and so on. In this way, the choice of numbers of hidden layers was made exhaustively, varying from 1 to 3 hidden layers. The time window for TLNN and LSTM was also exhaustively searched in the range of 2 to 20 time lags. Moreover, three activation functions were considered (Sigmoid,

Table 2.4. Parameter optimization according to the architecture: "✓" stands for Bayesian optimized, "-" stands for exhaustively optimized and × for not applied

	ANN	TLNN	LSTM
Number of neurons	✓	✓	✓
Number of layers	-	-	-
Time window	×	-	-
Activation type	✓	✓	×
Regularization type	✓	✓	×
Regularization rate	✓	✓	×
Dropout rate	✓	✓	✓

Hyperbolic tangent (Tanh) and Rectifier linear unit (Relu) and two regularizations (L1 and L2). Table 2.4 shows the parameters optimized according to the architecture.

For the spatial extension of sensors, the process of road link selection is a crucial step. The grid network focus of this work presents 80 links. Even in this small network, the number of possible combinations exceeds 1.2×10^{24} if ratio of monitored and unmonitored links is not set, and if a fixed ratio the possibilities can surpass 10^{23}. In a real traffic network sensors are opportunely located to provide maximum information about the network. Here, a different approach is used. The road link selection was made in a constructive manner, starting from a very small number of monitored links (the eight corner links). The unmonitored link that presented worse performance in this configuration was considered to be monitored in the next iteration. The process continued until stop criteria was met. Initial tests show results favorable to the LSTM network, so the selection of links was only based on it. The Bayesian optimization showed similar results between iterations, for that reason it was only performed when a stagnation or loss in performance was noticed. The complete training process is provided in the pseudo-code, see Algorithm 1. After establishing the input and output links, TLNN and ANN parameter were optimized and trained in concern this input/output set.

```
BestEvaluation ← -∞ ;
LastEvaluation ← 0 ;
InLinks, OutLinks ← InitialIO() ;
Initialize θ ;
for lag ∈ LAGS do
    for layer ∈ LAYERS do
        while R² < 0.8 do
            x, y ← updateData(lag, InLinks, OutLinks);
            Evaluation, WorstLink ← Train(θ, x, y);
            if LastEvaluation ≥ Evaluation then
                θ ← Bayesian Optimization ;
            end
            InLinks ← InLinks + WorstLink ;
            OutLinks ← OutLinks - WorstLink ;
            if Evaluation > BestEvaluation then
                BestEvaluation ← Evaluation ;
            end
        end
    end
end
Result: BestEvaluation, InLinks, OutLinks
```

Algorithm 1: Training process

2.5.2 Simulation Results

In this section, the results achieved by the proposed approach are presented. All the results presented below are regarded to the testing set, which corresponds to 20% of the whole data set. The traffic variable chosen as target of the prediction was average traffic speed of the given road

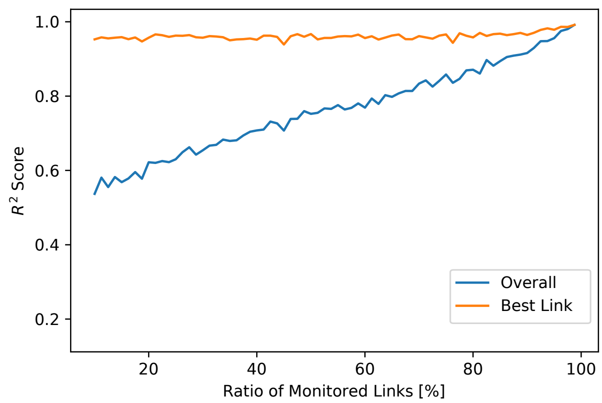


Figure 2.9. Evolution of performance according to the ratio of monitored links (LSTM Neural Network)

link (i.e. space mean speed). The speed values were normalized in a standard score manner, setting the values to present zero mean and the standard deviation equal to one before the training process.

The results are presented in terms of the coefficient of determination ($R^2 = 1 - \sum_i \frac{(y_i - v_i)^2}{(y_i - \bar{y})^2}$). For each road link R^2 can be defined where v is the predicted speed in a specific link, i is the index of the value in the test dataset, y and $\bar{y}$ are the observed data in the specific link and the average value respectively. The overall R^2 score can be calculated as the average of all links R^2.

Naturally, as well as in traditional monitoring systems, there is a positive correlation between the number of monitored points and the monitoring quality. Firstly, a greater number of monitored links provide more information about the network and therefore enabling better estimations. Secondly, a smaller number of unmonitored links simplifies the task in hand. Fig. 2.9 shows the evolution of the overall performance and the performance of the best estimated links with respect to the ratio of monitoring links for the LSTM model.

To establish a comparison between approaches, the same input/output configuration must be considered. It was selected following the procedure as explained previously, until the overall performance reach results above 0.8. This outcome was achieved with 63.75% of monitored links in the network. Once the quality of the fitting varies greatly from link to link, a box-plot is appropriated because it allows a visualization of the result's dispersion. Fig. 2.10 shows the results for LSTM, TLNN, ANN and the best ANN parameterization from Gallo et al. [2016] referred by the author's name as *Gallo*.

It can be observed that approaches that take into consideration past values (LSTM and TLNN) outperforms approaches which consider only instant values. The LSTM model presented itself superior among all others models, achieving not only better estimations in all links in network, but also more concise results with less dispersion. The NN configuration of Gallo et al. [2016] showed poor results for the task in hand. Although, the ANN method and Gallo's approach are similar, the ANN outperformed Gallo's parameterization because of the Bayesian optimization. Table 2.5 shows the Bayesian optimization parameter results and the numerical results generated by this configuration in terms R^2 score where "Best" means the best performing road link, "Worst" is the link with the worst estimation result and "Overall" denotes the average performance on all links. For the LSTM model, the optimization process rejected the use of dropout between layers, which corroborates the results found in Cheng et al. [2017], disfavoring the use of per element dropout in LSTM networks. Both LSTM and TLNN agreed on the time window size, showing 8 time steps as optimal window size for the estimation. Although all the

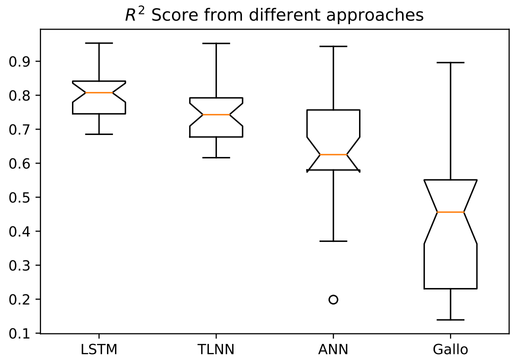


Figure 2.10. Comparison between approaches

	LSTM			TLNN			ANN			*Gallo*		
Number of neurons	[127, 47]			[59, 59, 47]			[25,11]			[10]		
Number of hidden layers	2			3			2			1		
Time Window	8			8			1			1		
Activation type	Relu			Sigmoid			Tanh			Tanh		
Regularization type	-			l2			l1			-		
Regularization rate	-			[0.0, 0.05, 0.0]			[0.0, 0.0]			-		
Dropout rate	[0.0, 0.0]			[0.0, 0.218, 0.0]			[0.0, 0.0]			-		
	Worst	Overall	Best	Worst	Overall	Best	Worst	Overall	Best	Worst	Overall	Best
Result of R^2	0.685	0.805	0.953	0.616	0.745	0.952	0.198	0.633	0.943	0.139	0.419	0.896

Table 2.5. Parameterization and results between approaches

NN models could effectually achieve acceptable results on the best-performing link, only LSTM could maintain acceptable average results. The results for the worst link in the LSTM model were also superior compared to the results of the other approaches.

2.6 Contributions

As new scientific contributions advanced methods have been elaborated to enhance the estimation of road traffic parameters in case of noisy or intermittent data. The more the notion of uncertainty has also been formalized for urban traffic modeling.

Thesis 1

Measurement of road traffic data is possible via several technologies with different reliability. The measured data might be biased, sparse, or incomplete due to sensor failure. To overcome these problems, I have developed estimation and modeling techniques applicable in practice. In this work, I have justified the applicability of the Switching Kalman Filter based data fusion for traffic parameter estimation. I have formalized the uncertainty caused by measurement bias or incompleteness in urban road traffic modeling, so that it can be directly used for traffic estimation and control. Finally, as a model-free approach the applicability of artificial intelligence has been proved for the problem of incomplete traffic measurement.

Thesis 1.1

I demonstrated that the problem of inhomogeneous data of road traffic measurements (i.e. diverse reliability and sampling frequencies of different sensors) can be handled by the Switching Kalman Filter technique. The method allows to efficiently fuse different sensor data; therefore, even the continuously changing number of sensors can be managed. The switching system applied with the random walk model approach is as follows:

$$\begin{aligned} x(k+1) &= x(k) + v(k), && (2.36)\\ y_{\rho(k)}(k) &= C_{\rho(k)}x(k) + z_{\rho(k)}(k), && (2.37)\\ \rho(k) \in S &= \{1, 2, \dots\}, && (2.38) \end{aligned}$$

where $x(k)$ is the traffic parameter to be estimated, $y_{\rho(k)}(k)$ denotes the measurements, $v(k)$ and $z_{\rho(k)}(k)$ are state and measurement noise terms with zero mean Gaussian distribution, $\rho(k)$ is the switching signal having effect only on the measurement equation. Set S contains possible different measurement combinations according to the different sensor types.

Thesis 1.2

I demonstrated the applicability of Moving Horizon Estimation for roundabout traffic flow estimation together with the special case of missing measurement data. The temporary unobservability of certain traffic flows represents a likely situation in real practice due to the potential operational failure of traffic detectors. As a practical solution, in the proposed estimation procedure the required detection can be replaced by exploiting the previous step's estimation $\hat{x}(k-1)$ based on the definition of the measurement equation:

$$y(k) = C(k)x(k). \quad (2.39)$$

Thesis 1.3

Advanced dynamic control or estimation involves traffic models which contain the most important characteristics of the network, i.e. topology and other dynamic parameters (e.g. traffic light program). With an appropriately chosen and parameterized model, future traffic states can be predicted. However, the applied traffic model can be biased by noisy or incomplete sensor data. Uncertainty modeling is introduced specifically for urban road traffic, i.e. the traffic model is formalized as a dynamics with state and demand multiplicative uncertainty under the form:

$$x(k+1) = \big(A + G\Delta(k)D_x\big)x(k) + Bu(k) + \big(I + G\Delta(k)D_d\big)Ed(k), \|\Delta(k)\|_2 \leq 1, \tag{2.40}$$

where $\|\Delta(k)\|_2 \leq 1$ expresses the bounded uncertainties on traffic state $x(k)$ and non-controlled traffic demand $d(k)$; $u(k)$ is the control input; D_x, D_d and G are weighing matrices; A, B, and E denote system matrices of the state equation. Based on the proposed uncertainty approach, robust schemes can be applied for traffic estimation or traffic control.

Thesis 1.4

The applicability of artificial intelligence for incomplete traffic measurement has been demonstrated using simulation. The main achievement of the research is the method of spatial extension of sensors in urban road traffic monitoring by proposing a road selection algorithm to find the proper set of monitored links regarding to the performance (R^2) of the estimation. The employment of Long Short Term Memory Neural Network to perform spatial extension of traffic sensor points is a novelty in the field of road traffic prediction and yields better results than the other approaches proposed in the literature.

Related journal and conference papers

- T. Tettamanti, T. Bécsi, and I. Varga. A közúti forgalom becslésére felhasználható mérési adatok és együttes alkalmazhatóságuk. *Közlekedéstud. Szemle*, LXIV(3):29–42, 2014a;
- T. Tettamanti, M.T. Horváth, and I. Varga. Road traffic measurement and related data fusion methodology for traffic estimation. *Transport and Telecommunication*, 15(4):269–279, 2014b. doi: 10.2478/ttj-2014-0023 **(Q3)**;
- T. Tettamanti. Innovatív közlekedési kutatás - korszerű mérési és irányítási módszerek városi közúti közlekedési hálózatban. *Közlekedéstud. Szemle*, LXIV(1):41–48, 2014;
- A. Csikós, Zs. J. Viharos, K.B. Kis, T. Tettamanti, and I. Varga. Traffic speed prediction method for urban networks - an ANN approach. In *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pages 102–108, June 2015b. doi: 10.1109/MTITS.2015.7223243;
- T. Tettamanti, M.T. Horváth, and I. Varga. Közúti eljutási idő becslésének lehetősége adatfúziós technikával városi úthálózaton. *Közlekedéstud. Szemle*, LXVI(3):46–56, 2016a;
- T. Tettamanti, M.T. Horváth, and I. Varga. Nonlinear traffic modeling for urban road network and related robust state estimation. In *Proceedings of the 9th European Nonlinear Dynamics Conference*, page ID 247, 2017a. ISBN 978-963-12-9168-1;
- T. Tettamanti, A. Csikós, K.B. Kis, Zs.J. Viharos, and I. Varga. Pattern recognition based speed forecasting methodology for urban traffic network. *Transport*, 33(4):959–970, 2018a. doi: 10.3846/16484142.2017.1352027 **(Q2)**;
- A.C. Piazzi and T. Tettamanti. Deep learning approach for spatial extension of traffic sensor points in urban road network. In *2019 IEEE 13th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pages 81–86, 2019b. doi: 10.1109/SACI46893.2019.9111522;

- A.C. Piazzi and T. Tettamanti. LSTM Approach for spatial extension of traffic sensor points in urban road network. In *hEART 2019 - 8th Symposium of the European Association for Research in Transportation*. European Association for Research in Transportation, 2019a. Paper 81;
- R.P. Tóth, M. Szalai, and T. Tettamanti. A HU-GO elektronikus útdíjrendszerből származó adatok forgalombecslési és forgalomirányítási célú felhasználási lehetőségei. *Közlekedéstudományi Szemle*, LXX(6):208–214, 2020. doi: 10.24228/KTSZ.2020.6.1;
- M.T. Horváth and T. Tettamanti. Real-time queue length estimation applying shockwave theory at urban signalized intersections. *Periodica Polytechnica Civil Engineering*, 65(4): 1153–1161, 2021a. doi: 10.3311/PPci.17022 **(Q3)**;
- A. Kovács, Á. Leelőssy, T. Tettamanti, D. Esztergár-Kiss, R. Mészáros, and I. Lagzi. Coupling traffic originated urban air pollution estimation with an atmospheric chemistry model. *Urban Climate*, 37:100868, 2021. ISSN 2212-0955. doi: 10.1016/j.uclim.2021.100868 **(D1)**;
- M. Gressai. and T. Tettamanti. Turning rate estimation in roundabouts: Analysis and validation of different estimation methods. In *Proceedings of the 7th International Conference on Vehicle Technology and Intelligent Transport Systems - VEHITS*, pages 65–71. INSTICC, SciTePress, 2021. ISBN 978-989-758-513-5. doi: 10.5220/0010405700650071;
- M.T. Horváth and T. Tettamanti. Robust vehicle count estimation on urban signalized links. *Measurement*, 181:109581, 2021b. ISSN 0263-2241. doi: 10.1016/j.measurement.2021.109581 **(Q1)**;
- T. Tettamanti. Advanced methods for turning rate estimation in roundabouts. *Measurement*, 181:109676, 2021. ISSN 0263-2241. doi: 10.1016/j.measurement.2021.109676;
- B. Varga and T. Tettamanti. Városi járműforgalom térbeli becslése kernel módszerek segítségével. *Közlekedéstud. Szemle*, 5(LXXI):37–43, 2021. doi: 10.24228/KTSZ.2021.5.2;
- M. Gressai, B. Varga, T. Tettamanti, and I. Varga. Investigating the impacts of urban speed limit reduction through microscopic traffic simulation. *Communications in Transportation Research*, 1:100018, 2021. ISSN 2772-4247. doi: 10.1016/j.commtr.2021.100018.

Related books

- T. Tettamanti, I. Varga, and A. Csikós. *Közúti mérések, Eszközök és módszerek a közúti járműforgalom megfigyelésére.* Typotex Kiadó, 2016b. ISBN 978-963-279-916-2;
- T. Tettamanti, T. Luspay, and I. Varga. *Road Traffic Modeling and Simulation.* Akadémiai Kiadó, 2019a.

Related patent

- Nokia Solutions and Networks OY, H. Demeter, N. Vékony, T. Tettamanti, I. Varga, and Á. Ludvig. Determining travel information (method and system for real-time travel time calculation in road traffic network using radio signaling data), 2014. Invention Publication No.: WO 2014/023339 A1.

Chapter 3

Impacts of Automated Driving on Traffic Dynamics and Traffic Modeling

As a revolutionary technology, Automated Vehicles (AVs) might have great potential to reduce traffic collisions, increase transportation system performance, and improve environmental sustainability. In order to introduce future policy measures and control for traffic in a conscious way, it is significant to know how the introduction of AVs in everyday traffic will influence the road capacity compared to our current knowledge on traffic dynamics. Therefore, the effects that automated vehicles bring to the macroscopic fundamental diagram (MFD) have been investigated through microscopic traffic simulation. This is a key issue as the MFD is a basic model to describe road capacity in practical traffic engineering, i.e. used in strategic traffic planning or even in real-time traffic control. Additionally, the perspective of fuel economy and emission reduction have also been investigated in relation with the AV penetration.

3.1 Impacts on Urban Macroscopic Fundamental Diagram

Traffic congestion is now a part of our daily life with examples aplenty. Autonomous vehicles will change our conventional transportation frameworks. It can possibly fundamentally change the driver interactions and give huge chances to boost traffic capacity, efficiency, stability, and safety of existing mobility systems. Accordingly, my research investigates the impacts of AVs, with a special spotlight on the efficiency of utilizing the existing infrastructure.

Based on the literature review, the advent of AVs to the road network can improve capacity by i) keeping traffic flow parameters stable, and ii) taking into account more tightly spaced vehicles. Using simulations, various studies investigated the shifts of driving behavior in AVs. For example, the reaction time, acceleration, deceleration, platoon size, and their impacts on road capacity were studied. If AVs were operated at shorter headway, then maximum throughput could generally increase. If vehicles were operated at lower speeds so as to make the vehicle flow more stable, this could cut down the capacity of a bottleneck. At the same time, the delay and travel time would increase [Jerath and Brennan 2012; Kesting et al. 2008; Talebpour and Mahmassani 2016; Talebpour et al. 2017; Van Arem et al. 2006]. Van Arem et al. [2006] investigated the impacts of vehicle platoon size on the flow stability and capacity on a freeway with a lane drop by using a microscopic traffic simulation. Most of these research revealed that an intermediate (or even lower) AV penetration rate could contribute to a considerable capacity upgrading [Jerath and Brennan 2012]. With the help of simulation software, Talebpour et al. [2017] found the throughput was enhanced considerably when the AV penetration rate exceeded 30% as they studied the influences of assigning a lane of a four-lane freeway to AVs on the traffic flow dynamics and trip time reliability. Van Arem et al. [2006] found that significant impacts were noticed only when the AV penetration exceeded 40%. Likewise, Jones and Philips [2013] found that the positive impact of Cooperative Adaptive Cruise Control (CACC) vehicle on the traffic flow stability and throughput was realized if the CACC vehicle penetration surpassed 40%. In contrast, Van Arem et al. [2006] spotted that the capacity increment was negligible unless the AV penetration rate was above 50% in simulation. Shladover [2012] obtained the consistent outcome by utilizing field experiments. While, Tientrakool et al. [2011] found that capacity improved slightly until the CACC penetration rate exceeded 85%. 100% AV penetration scenario contributes a more noteworthy positive impact on the road capacity. For instance, Friedrich [2016] found that a capacity increase of 40% could be achieved with purely autonomous vehicles in city traffic, while capacity could be improved on highway sections by about 80%. Olia et al. [2018] found that the purely cooperative vehicle scenario can raise the highway capacity by 300% compared to the traditional vehicles' scenario.

In most of the articles mentioned above, AVs were considered and these works focused on freeway traffic. Solely Tientrakool et al. [2011] investigated the AVs impacts on urban road capacity, but his work concentrated only on the impacts at a single intersection and not on a whole urban traffic network. In the next sections the effects of AVs without connected technology are focused by defining the vehicles with different parameters compared to the conventional cars in order to assess the possible impacts on the urban traffic network capacity.

3.1.1 Macroscopic Modeling of Urban Traffic Flow

In this research, the impacts of AV penetration on the MFD have been investigated. The work was carried out with SUMO simulations [Lopez et al. 2018] by applying different percentage of both conventional cars and full automation AVs. The simulations were operated in a real-world traffic network and a virtual grid road network considering different penetration rates. The simulator virtually measured the traffic volume on each link and the whole network's throughput. Data was obtained through a macroscopic link-level measurement that called "edgeData" measurement in SUMO. The outcomes were calculated to understand the development of different situations and to make known how the traffic network capacity changes along with different penetration levels of self-driving cars.

The macroscopic fundamental diagram of traffic flow defines the relationship among the traffic flow $Q(vehicles/h)$, the vehicle concentration ρ $(vehicles/km)$ and the space mean speed V (km/h) [Williams et al. 1987]. The MFD is based on the fundamental equation:

$$Q(\rho) = \rho \cdot V(\rho)\,. \tag{3.1}$$

As described in the previous section, the fundamental diagram can be applied in both network-level and link-level. The network-level MFD models the throughput of the traffic network per hour:

$$Q_N(\rho_a)\,, \tag{3.2}$$

where Q_N is the number of vehicles that pass through the network. ρ_a is the average density of the network, and it simply equals to the known total number of vehicles in the network divided by the sum of all link lengths of the road network, i.e.

$$\rho_a = \frac{\sum_{i=1}^{n} \rho_i l_i}{\sum_{i=1}^{n} l_i}\,, \tag{3.3}$$

where l_i is the length of link i, n is the number of links [Tettamanti et al. 2015a]. The second approach interprets the MFD of one single road link of the network, i.e.

$$Q_i(\rho_i) = \rho_i \cdot V_i(\rho_i)\,, \tag{3.4}$$

where Q_i is the flow, ρ_i means the density, $V_i(\rho_i)$ defines the mean velocity, and Q_i is the flow on link i.

SUMO traffic simulator was applied for MFD estimation. All trip histories of the vehicles going through the traffic network were collected to utilize these observations in the MFD parameter estimation later. The edge-based measurement of SUMO gave link-level concentration, stop time, overlap travel time, sample second and average speed.

To draw the MFD, concentration ρ, network average velocity V and flow Q are needed. The following formula were used for the calculation of these values.

$$\rho = \frac{\sum_1^n N_{vi}}{\sum_1^n l_i}\,(veh/km), \tag{3.5}$$

$$V = \frac{\sum_1^n V_i N_{vi}}{\sum_1^n N_{vi}}\,(km/h), \tag{3.6}$$

$$Q = \sum_1^n 3.6 \cdot V_i \rho_i\,(veh/h), \tag{3.7}$$

where n is the number of the links in the traffic network, V_i and N_{vi} are the average speed and vehicle numbers on the i^{th} link during the measurement interval T.

3.1.2 Simulation Based Methodology

Detailed simulation studies were carried out with SUMO to analyze the effect of AVs on the urban road system capacity. To have a better observation, the simulations were carried out in two road networks, namely a virtual grid traffic network and a real-world urban road system. In order to determine the impacts of variations in market penetration of conventional and autonomous vehicles, the simulation scenarios were characterized to represent various combinations of both types of vehicles. The penetration of autonomous vehicles varied from 0% to 100% by stepping with 20%. The results were handled with the Generalized Additive Model (GAM) to find the relationship between average speed V and vehicle density ρ. Then, with the help of MFD theory, the flow-density relationship with respect to AV penetration rate could be identified.

For the study a 8×8 grid network was created having 60 nodes and 36 intersections in it. The lengths between adjacent nodes were 300 m. The network links were bidirectional roads with single lanes. As a traffic-responsive control, the SUMO built-in tool, named ”time gap based” traffic signal method, was applied. To realize an optimized change of traffic light phases dynamically, the controller switches to the next phase when it detects an adequate time opening between successive vehicles. In the simulations all vehicles get automatically routed at insertion. Routing was created by using trip file of SUMO to form traveling demand, i.e. origin and destination links. In this situation, vehicles choose the fastest paths according to the current traffic conditions of the network when they enter. Therefore, the distribution of vehicles is more flexible and homogeneous compared to the demand modeling with fixed routes. The origin and destination links were located at the perimeter of the network. The demand increased gradually till a traffic congestion formed in the network then decreased to zero smoothly. The simulation scenarios were generated to obtain significant traffic jams but gridlock was avoided.

As shown in Figure 3.1, the real-world study area was located in the 11^{th} district of Budapest. The network contained five arterial roads: Bartók Béla út, Karinthy Frigyes út, Irinyi József út, Műegyetem rkp., Budafoki út. Simulations were run in the whole road network shown by Figure 3.1, but the results were evaluated concerning 30 selected road links only (depicted by blue color) forming an intrinsically homogeneous sub-network for investigation. The speed limits for all roads were set to 50 km/h. Link length average was 0.116 km. In these simulations, traditional fixed traffic light signal method was applied as it works in reality.

The vehicle flows came from the arterial roads then ran through the investigated area. According to the real-world traffic data, heavy traffic demands were applied to the arterial roads, while small vehicle flows to other side streets. Moreover, for the purpose of MFD estimation, simulations were carried out with varying traffic loads, including the situations from the free flow to the rush hours flow. At the beginning of the simulation, the traffic demand was low. Then, a series of light additional flows were introduced into the network continuously in order to arrive at a congested situation. Finally, demand started decreasing. The amount of vehicles is defined by the departure time in the trip file of SUMO.

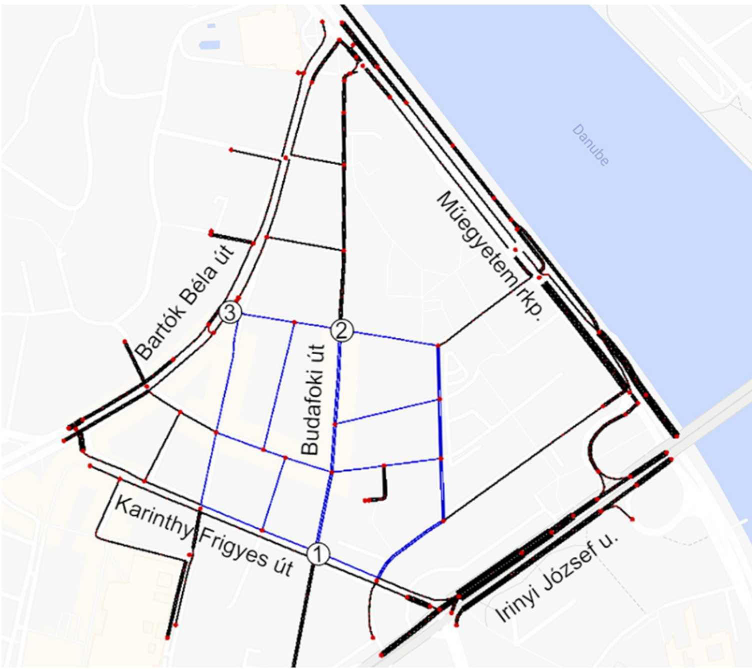


Figure 3.1. Real-world traffic network (GPS coordinates: 47.47733, 19.05358)

The vehicle modeling method is the same with the work of Lu and Tettamanti [2018]. Default SUMO parameters have been modified in order to model a plausible future for AVs. The default car-following model (Krauss Model) was applied. The parameter selection was related to longitudinal movement, acceleration, deceleration and gap acceptance. These behaviors were formalized as parameters in the car-following model of SUMO. The implemented model followed the idea that let vehicles drive as fast as possible while maintaining perfect safety (always being able to avoid a collision if the leader starts braking within leader and follower maximum acceleration bounds). The following list shows the editable parameters of the Krauss car-following model [Krauß 1998]:

- Mingap: the offset to the leading vehicle when standing in a jam (in m).
- Accel: the acceleration ability of vehicles of this type (in m/s^2).
- Decel: the deceleration ability of vehicles of this type (in m/s^2).
- Emergency Decel: the maximum deceleration ability of vehicles of this type in case of emergency (in m/s^2).
- Sigma: the driver imperfection (between 0 and 1).
- Tau: the driver's desired (minimum) time headway (reaction time) (in s).

For no and full automation vehicles, the deceleration and the emergency deceleration remained the same, considering the safety. The emergency deceleration was set to 8 m/s^2. This value was based on the study of Kudarauskas [2007]. While, the mingap, acceleration and time headway were taken from Atkins Ltd. [2016]. The parameters were tabulated to Table 3.1.

Table 3.1. Parameters of the driver model used in SUMO simulations

	Mingap (m)	Accel (m/s^2)	Decel (m/s^2)	Emergency decel (m/s^2)	Sigma	Time headway (s)
No automation	2.0	2.6	4.5	8	0.5	1.0
Full automation	0.5	3.8	4.5	8	0	0.6

The output of the simulation experiment introduced earlier in this section is a series of observations at various levels of AV penetration rates and network-level traffic conditions. The ultimate goal is to estimate the flow-density relationship of the MFD and observe the way how the share of AVs affects its shape, with a focus of any deviations in the critical density and the corresponding flow level.

To achieve this goal, firstly the speed-density function was estimated using the simulation outcomes. This segment of the MFD is closer to a linear relationship, and therefore can be estimated more reliably than the inverse U-shaped flow-density function. A semiparametric approach was applied to model the speed-density relationship. This allows to relax the assumption that the speed-density function is perfectly linear.

Average speed is modeled as

$$V(r, \rho) = \alpha + s(\rho) + \beta \cdot r + \gamma \cdot r \cdot \rho \, , \tag{3.8}$$

where α can be considered as average speed under free-flow conditions, i.e. the intercept of the speed-density function, and $s(\rho)$ is a spline estimated together with the prespecified parameters of the model: α, β, and γ. The ratio of AVs among all vehicles (r) enters the regression equation twice. First, it affects the free-flow speed if the estimate of β will become statistically significantly different from zero. Second, it affects the slope of speed-density function through the interaction term between r and ρ, providing that the coefficient of this interaction (γ) will differ from zero after model fitting. That is, γ tells whether the presence of AVs induces changes in the way in which increasing traffic density affects average speed. If this parameter is positive and statistically significant, then the negative impact of traffic density on speed will be somewhat weaker with the introduction of AVs. This semiparametric representation can be powerful in prediction because assumptions are not needed about the baseline functional form of the speed-density curve *a priori*.

The estimation based on semiparametric regression was applied via GAM (Generalized Additive Model) implemented in R software. GAM is basically a generalized linear model in which the predictor depends linearly on covariates that enter the regression with predefined functional form, and unknown smooth functions of other predictor variable(s); in this study a smooth function of traffic density.

For a given level of AV penetration rate, informative flow-density curve can be recovered with a simple algebraic transformation of the estimated speed-density function. By substituting V in Eq. (3.1) into Eq. (3.8), one gets the corresponding flow-density relationship $Q(\rho; r)$.

In practice, one generates the full sequence of potential density values from zero to the highest density observed in the data, and compute the corresponding speed levels using the estimated model. Then, the average speed values are transformed into flow, based on the algebraic relationship detailed above. This way one can reproduce the $Q(\rho; r)$ segment of the MFD for any given AV ratio $r \in (0, 1)$. Identifying the critical density and the corresponding traffic flow level (i.e. the maximum throughput of the network) is a straightforward numerical task, so this approach enables us to express the key parameters of the simulated MFDs as function of the penetration rate of AVs.

3.1.3 GAM Regression for Urban MFD

With the help of GAM regression model, the simulation data were processed. The results of the grid network and the real-world network are displayed separately in this section. In order to verify the rationality of the GAM, the statistic coefficients were firstly inspected. After the validation of the regression model, the maximum capacity and the corresponding density of different AV penetration are also investigated.

Table 3.2 shows the summary of speed–AV ratio–density relationship estimation of grid network simulation results of all scenarios. R^2 indicates the goodness of fitting, which was 0.983. So the proposed model is validated for the speed–AV ratio–density relationship estimation. The *t-value* and *p-value* reflect the significance levels of the independent variables. The *t-value* is a measure of how many standard deviations the estimate coefficient is far away from zero. It is expected to be far away from 0 because this would indicate the null hypothesis is rejected. That is to say, a relationship between velocity and the independent variables mentioned above exists. In this analysis, the *t-values* are relatively far away from 0 and are large relative to the standard error, which could indicate a relationship exists. The *p-value* is the probability when the null hypothesis is true. A small *p-value* for the intercept and independent variables indicates that the null hypothesis is rejected. This allows us to conclude that there is a relationship between capacity and the investigated variables. Actually, the *p-value* can be calculated from *t-value*. These two values give equivalent conclusions that all investigated terms have a relationship with the response average speed V because the *p-values* of all terms are much less than 0.001.

Table 3.2. GAM regression of the grid network simulation

coefficient	estimate	standard error	*t-value*	*p-value*	R^2
α	16.237	0.05988	271.129	$< 2E-16$	0.983
β	1.298	0.2073	6.264	5.56E-10	
γ	0.024	0.00723	3.372	0.000776	
$s(density)$				$< 2E-16$	

A spline is a piece-wise function defined by polynomials. The spline regression is shown in Fig. 3.2. The Estimated Degrees of Freedom (EDF) of the smooth term s(density) is 6.603. EDF is the estimated degrees of freedom of the spline. It can be understood as how much a given variable is smoothed. Higher EDF value suggests higher complexity of the spline. The dashed lines in the figure are the confidence lines which indicate two standard error bounds. The rug on the horizontal axis is used to visualize the distribution of the data. Fig. 3.3 shows the speed - density relationship of the data from grid network simulation of all scenarios. These data were obtained in six scenarios with a full range of AV ratios (0%, 20%, 40%, 60%, 80% and 100%). The AV ratio was also considered to be a variable in the speed-density relationship estimation. Two lines in Fig. 3.3 represent the regressions of two different AV penetration. The black line is the speed-density relationship regression of 0% AV scenario. The blue line is the fitted curve of 100% AV penetration.

After obtaining the speed–AV ratio–density, the flow–AV ratio–density is easily to get by substituting the speed V in Eq. (3.1) with the fitted function $V(r, \rho)$. The fitted flow–AV ratio–density function is illustrated in Fig. 3.4. The black line is the fitted curve of 0% AV scenario. The blue line is the regression of purely driverless car scenario data. Based on the fitted flow–AV ratio–density relationship, the maximum flows and the corresponding densities of different AV

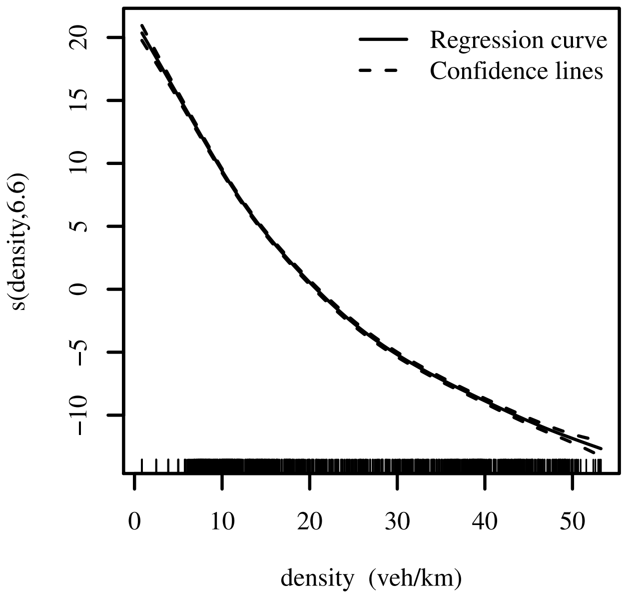


Figure 3.2. Grid network: spline regression where the EDF value is 6.6 in s(density, EDF)

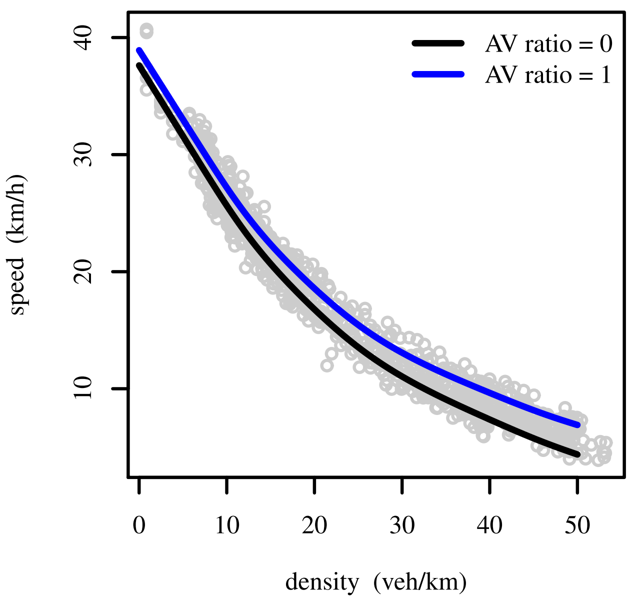


Figure 3.3. Grid network data: speed-density relationship

penetration scenarios are marked as black dots, as shown in Fig. 3.4. They are connected to show the change of the capacity and the critical densities with different AV ratio. From low to high, the ratios of the self-driving cars corresponding to these points are 0%, 10%, 20%,30%,40%, 50%, 60%, 70%, 80%, 90%, and 100% respectively. The flow–AV ratio–relationship and density–AV ratio relationship were extracted and plotted in Fig. 3.5 and Fig. 3.6. The relative changes of the scenarios comparison with the zero AV penetration case are also provided by Table 3.3. It is observable that the maximum flow is augmenting along with the increase of AV penetration in an almost linear way. But the final gain of 100% AV penetration is only 16.01%, which is less than the theoretically calculated result (40%) by Friedrich [2016] formerly. This can be explained by the fact that the calculated result by Friedrich [2016] merely took one intersection into consideration which eliminated the accumulation of vehicles on the adjacent roads. The maximum flow has an approximately linear relationship with AV penetration. The value of the critical density raises slowly in the beginning. Then the increase accelerates after 40% AV penetration.

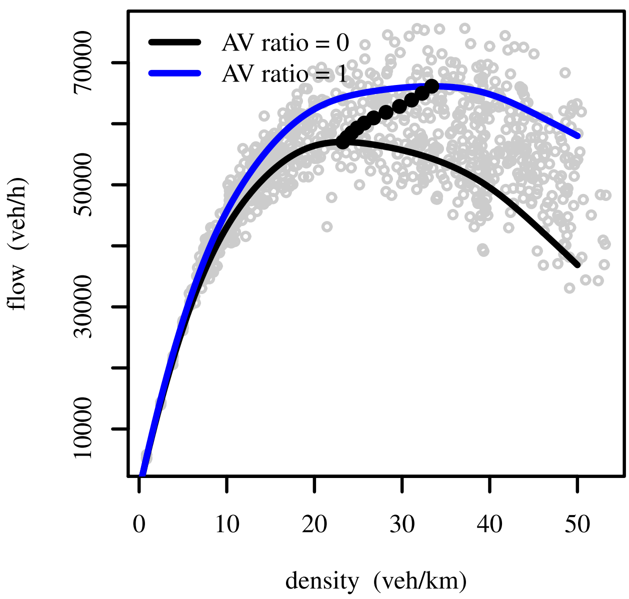


Figure 3.4. Grid network data: estimated flow-density relationship

Table 3.3. Grid network data: maximum flow, critical density, and their change

AV penetration rate	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Maximum traffic flow (veh/h)	57016	57753	58511	59293	60106	60958	61865	62839	63881	64983	66142
Relative change compared to the zero penetration case	-	1.29	2.62	3.99	5.42	6.91	8.50	10.21	12.04	13.97	16.01
Critical traffic density (veh/km)	23.24	23.7	24.24	24.87	25.68	26.75	28.16	29.7	31.08	32.28	33.38
Relative change compared to the zero penetration case	-	1.98	4.30	7.01	10.50	15.10	21.17	27.80	33.73	38.90	43.63

For the real-world network, the calculation and estimation processes were identical with the methods used for the grid road network (presented in previous section). Table 3.4 shows the speed–AV ratio–density relationship estimation of the real-world network simulation results in all scenarios. The high value (0.789) of R^2 means a high quality of the fitting. In terms of significance test, *t-values* and *p-values* demonstrate all the terms having relationship with the

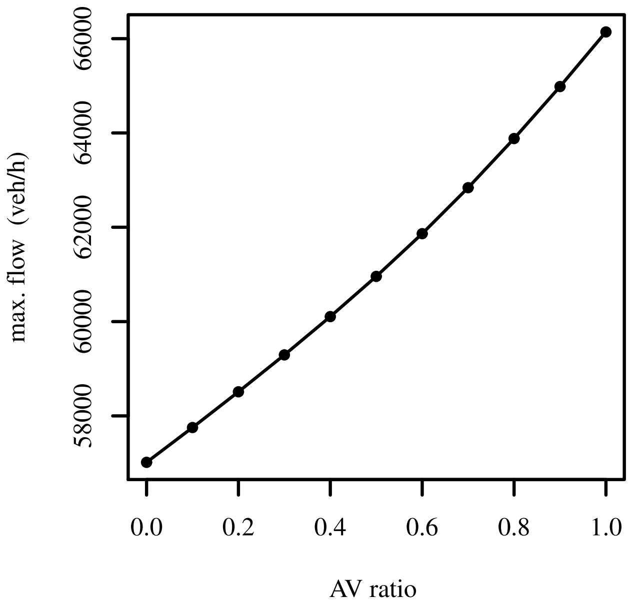


Figure 3.5. Grid network data: change of the maximum traffic flow according to the AV penetration ratio

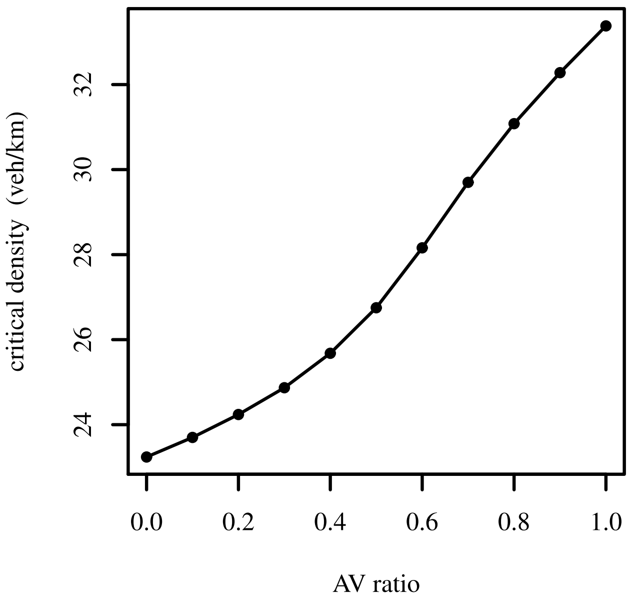


Figure 3.6. Grid network data: change of the critical traffic density (the density at the maximum traffic flow) according to the AV penetration ratio

response variable (average speed). The *t-values* are far away from zero and the *p-values* are less than 0.001. This means that the model is also validated for the real-world road network.

Table 3.4. GAM regression of the real-world network simulation

coefficient	estimate	standard error	*t-value*	*p-value*	R^2
α	21.815	0.1784	122.284	$< 2E-16$	0.789
β	1.724	0.51106	3.374	7.57E-04	
γ	0.07	0.02066	3.399	0.000693	
$s(density)$				$< 2E-16$	

The EDF values of the smooth term $s(density)$ was 3.853. The spline regression is illustrated in Fig. 3.7. The confidence lines are depicted as dashed lines in the figure, indicating two standard error bounds. The visualization of the data distribution is illustrated as a rug on the horizontal axis. The estimated speed-density relationship can be found in Fig. 3.8. The grey circles are the simulation data of real-world network in all scenarios. The curves are monotonically decreasing with a concave shape rather than a linear one. The black curve is the regression of all conventional vehicle scenarios. The blue line is the fitted speed-density relationship of the purely driverless vehicle scenario.

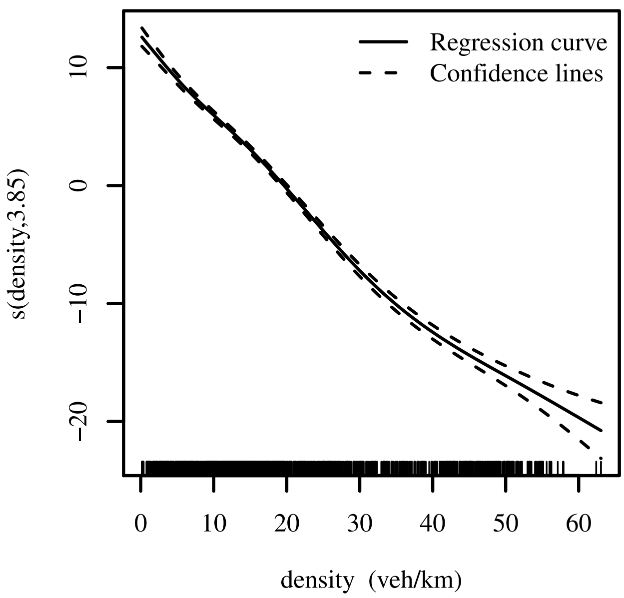


Figure 3.7. Real-world network: spline regression where the EDF value is 3.85 in s(density, EDF)

Fig. 3.9 shows the fitted flow–AV ratio–density relationship. The gray circles are the simulation data of real-world network in all scenarios. The black curve is fitted with 0% AV ratio scenario data. The blue line shows the fitted flow–density relationship of purely AV penetration scenario. In the 100% AV penetration scenario, a plateau occurs when the flow is around the maximum value. This shows a beneficial property of having AVs in a network. In a range, the density of vehicles can increase without congestion. The maximum traffic flow and the corresponding densities of different AV penetration scenarios are marked as black dots in Fig. 3.9. These points represent the capacity and critical densities of a range of self-driving cars penetrations which are 0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, and 100% respectively

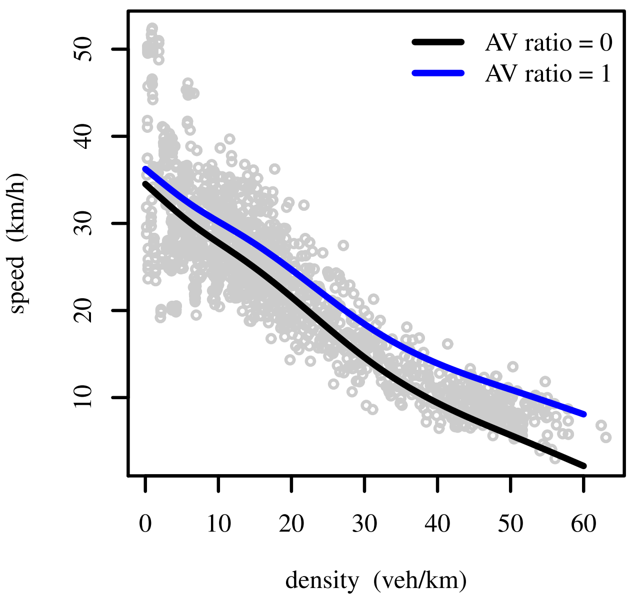


Figure 3.8. Real-world network data: speed–density relationship

from the bottom to the roof. A line through these dots shows the change of the capacity and the critical densities of different AV penetration scenarios in the real-world network simulation. Both the maximum flow and the corresponding critical density have an increasing trend with higher AV penetration. From Fig. 3.10, one can see that the maximum flow and AV ratio also have a quasi-linear relationship in the real-world network simulations. The maximum flow boosts with the raise of AV penetration. Fig. 3.11 shows the traffic concentration changes with the different AVs market penetration scenarios. The critical density is increasing with a higher driverless car penetration. Table 3.5 shows the maximum flow, the corresponding critical traffic concentration, and their relative comparison with the purely traditional regular vehicle scenario of different AV penetration scenarios. It shows that a capacity increment value of 23.81% is achieved with purely autonomous traffic, which is a bit higher than that of the grid network simulations. This may stem from the fact that the real-world network is smaller. The smaller the network is, the less the accumulation of vehicles is present. This fact may also lead to higher critical concentration of same AV penetration than that of the grid network. It also reflects that the increase of the critical traffic density is moderate at the beginning. After 50% of AV ratio, however, it becomes steep. The benefits of driverless car are more and more obvious with an increasing penetration. In all, the critical traffic concentration has a 48.45% raise with 100% AV penetration.

Table 3.5. Real-world network data: maximum flow, critical density, and their change

AV penetration rate	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Maximum flow (veh/h)	13497	13765	14040	14324	14616	14919	15234	15563	15910	16283	16710
Relative change compared to the zero penetration case (%)	-	1.99	4.02	6.13	8.29	10.54	12.87	15.31	17.88	20.64	23.81
Critical density (veh/km)	25.22	25.67	26.15	26.68	27.25	27.9	28.65	29.57	30.75	32.54	37.44
Relative change compared to the zero penetration case (%)	-	1.78	3.69	5.79	8.05	10.63	13.60	17.25	21.93	29.02	48.45

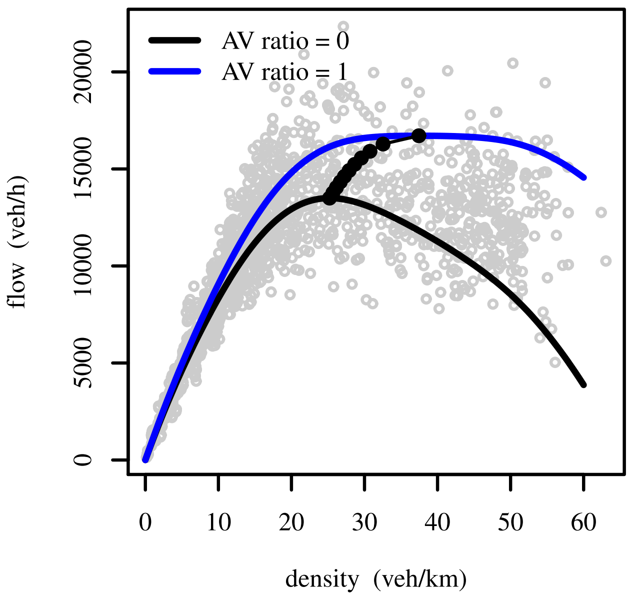


Figure 3.9. Real-world network data: estimated flow–density relationship

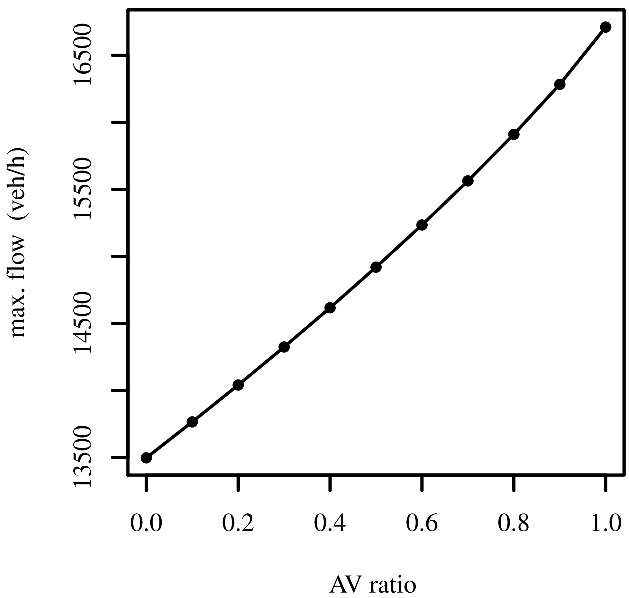


Figure 3.10. Real-world network data: change of the maximum traffic flow according to the AV penetration ratio

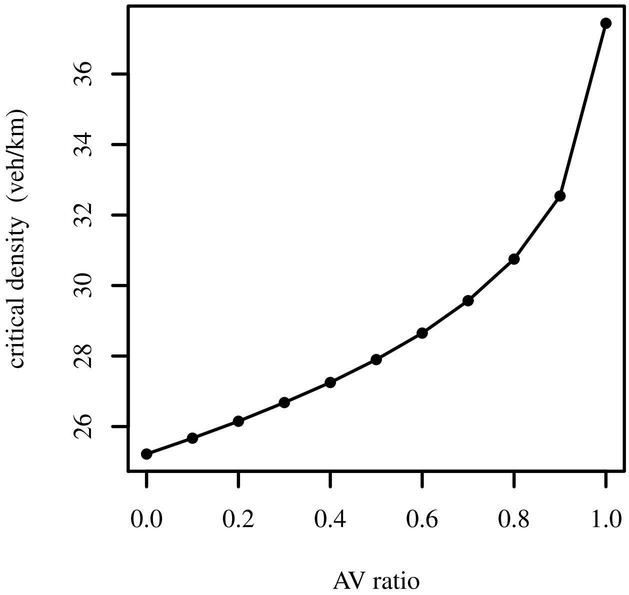


Figure 3.11. Real-world network data: change of the critical traffic density according to the AV penetration ratio

Table 3.6. The AV penetration scenarios used in this simulation

Scenario nr.	Scenarios	The ratio of traditional cars	AV penetration composition				
			Level 1	Level 2	Level 3	Level 4	Level 5
1	Base	100%	0%	0%	0%	0%	0%
2	25% penetration	75%	15%	5%	5%	0%	0%
3	50% penetration	50%	25%	10%	10%	5%	0%
4	75% penetration	25%	25%	20%	15%	10%	5%
5	100% penetration	0%	15%	20%	20%	25%	20%
6	Upper bound	0%	0%	0%	0%	0%	100%

3.2 Impacts of Different Levels of Autonomy of AVs on Urban Traffic

In the previous section, only the AV penetration rate's impact was investigated for the urban MFD modeling without the consideration of the different levels of autonomy. In the sequel, the impacts of the levels of autonomy was analyzed together with the penetration change. Here another approach, the cubic polynomial curve fitting, was used to model the relationship between density and traffic flow:

$$Q(\rho) = a \cdot \rho^3 + b \cdot \rho^2 + c \cdot \rho, \tag{3.9}$$

where $a,\ b,\ c$ are polynomial coefficients.

In this study the same grid network was applied for the simulation as discussed in Section 3.1.2 with the same traditional "time gap based" traffic signal control (a built-in tool of SUMO). At low market penetration, the technical capability is limited (for example, to driver assistance which means low autonomous driving level). As market penetration increases, consumer confidence also augments, and better use of connected and automated technology prevails. Measurements in one link and in the whole network were realized. The modeled scenarios are summarized in Table 3.6.

Table 3.7. Parameters of the driver model for different levels of autonomy

Capacity level	Mingap (m)	Accel (m/s^2)	Decal (m/s^2)	Emergency Decel (m/s^2)	σ (driver imperfection)	τ (s)
Level 0	2.0	2.6	4.5	8	0.5	1
Level 1	2.0	3.05	4.5	8	0.4	0.95
Level 2	1.5	3.5	4.5	8	0.3	0.9
Level 3	1.25	3.6	4.5	8	0.2	0.8
Level 4	0.75	3.7	4.5	8	0	0.7
Level 5	0.5	3.8	4.5	8	0	0.6

For level 0, the default values were taken for all parameters. But the emergency deceleration was set to 8 m/s^2. This value is based on the study of Kudarauskas [2007]. For other autonomous driving levels, the deceleration and the emergency deceleration remained the same, considering the safety. For level 2 and level 5, the mingap, acceleration, time headways were taken from Atkins Ltd. [2016]. For level 1, the values of these items were set as the average value of level 0 and level 2. For level 3 and level 4, the values of these items were changed linearly between level 2 and level 5. The driver imperfection for level 5 and level 4 was set to 0 because these levels do not need human driver's intervention. It was assumed to be 0.4, 0.3, and 0.2 for level 1, level 2, and level 3, respectively. The parameters for all levels are tabulated in Table 3.7.

The main simulation results are provided by Figs. 3.12 and 3.13. From the results for the

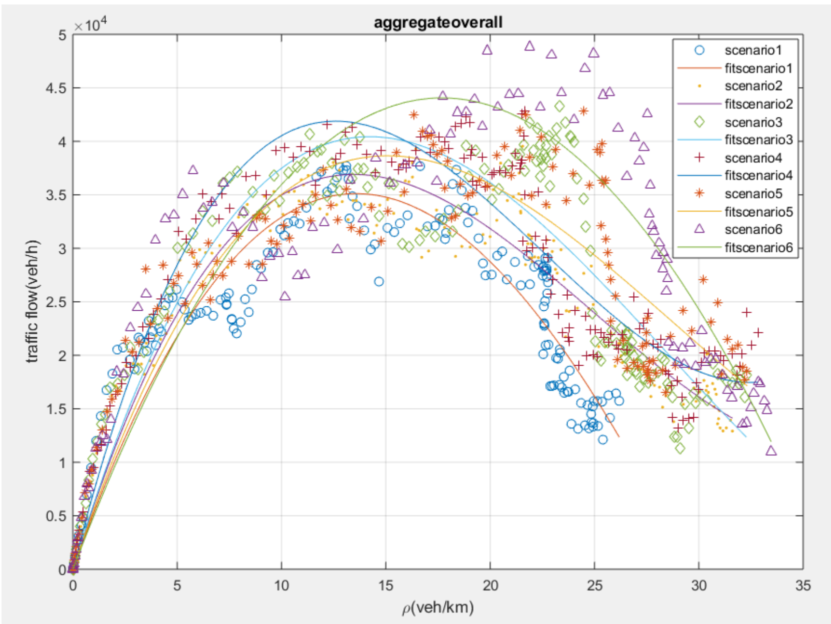


Figure 3.12. Simulation results for the whole network

whole network, one can see that from scenario 1 to scenario 6 the capacity of the whole network and the critical density vary. Scenario 6 has the largest critical density straightforwardly. The same tendency can be found in the whole network for critical density and capacity. They go up in the beginning, then decrease, roar up at the end.

From the results for one single link, one can see that the capacities for scenarios 1, 2, 3, and 4 are similar and relatively smaller, and the capacities for scenarios 5 and 6 are bigger and have an increasing trend. The same change can be found in the critical densities.

In all, the results justified regularity in the change of the urban MFD (network and link level as well) along with the autonomous technology evolution. The results are also important

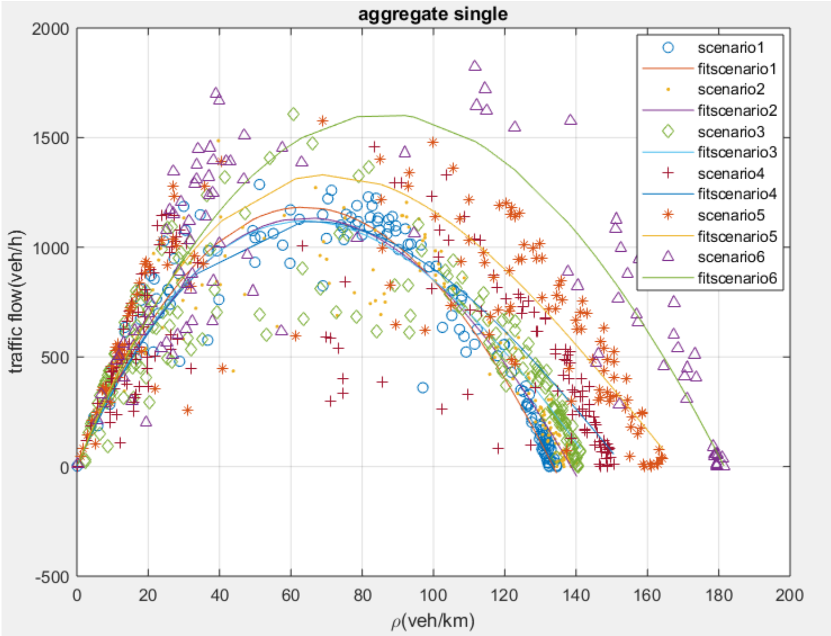


Figure 3.13. Simulation results for a single link

from the point of view of practical traffic engineering as the fundamental diagram is a common modeling approach when planning or analyzing a road network.

3.3 Impacts on Freeway Macroscopic Fundamental Diagram

Cooperative Adaptive Cruise Control (CACC) and Adaptive Cruise Control (ACC) are essential technologies towards fully automated driving. They can be used to simulate the cruising behavior of AVs to study how AVs can beneficially impact transportation systems [Makridis et al. 2018]. At present, there is no consensus on the right value for the headway of self-driving cars. AV developers confronted a dilemma about the proper setting of safe time headway based on the driver's response times [Goodrich and Boer 2003]. Abdulsattar et al. [2020] simulated the AVs with a time headway of 0.5 seconds. While Berrazouane et al. [2019] used a large time headway (even large than the conventional vehicles) to model an AV behavior demonstrating that the preferred driving behavior of AVs is still uncertain. Therefore, it is important to do a sensitivity analysis on time headway.

The speed limit of freeways varies from country to country. The speed limit has strong impacts on the road network performance, especially when the times of AVs arrives. Ye and Yamamoto [2018] concluded that placing a higher speed limit for AVs on the dedicated lane can improve the performance of AV specific lane. With the improvement of the safety performance of self-driving cars, it might become a trend to increase the freeway speed limit. Therefore, studying the effect of different speed limits on AV performance is important. Various researchers examined the impacts of microscopic driving behavior in AVs with the help of simulation. In general, a shorter headway could boost maximum throughput. Contrarily, an increasing time headway would lead to a decreasing road capacity. Lower speed vehicles could contribute to the stability of vehicle flows. However, as a consequence, they would cut down the bottleneck's capacity. The more, the delay and travel time would increase at the same time [Talebpour et al. 2017; Van Arem et al. 2006]. Some of these studies revealed that AV penetration rate in the intermediate (or even lower) range could result in a significant capacity upgrade. Talebpour et al. [2017] found that the throughput and trip time stability were improved considerably when AV penetration rate surpassed 30% assuming dedicated AV lane on a four-lane freeway.

However, road capability increased marginally until the CACC ratio surpassed 85%, according to Tientrakool et al. [2011]. Olia et al. [2018] found that the situation with the 100% penetration of cooperative vehicles could boost the highway capacity by 300%.

Autonomous vehicles have two opposing effects on the environment. On the one hand, good driving technology would reduce fuel consumption and exhaust emissions. On the other hand, increased transportation demand would increase pollution in transportation [Brown et al. 2014]. The primary aim of AV technology is to improve traffic safety and make available better mobility services. Though, AV would predictably and significantly transform the transportation sector environmental profile as well. Taiebat et al. [2018] classified the impacts of AVs on the environment into four levels, which are vehicle, transportation system, urban system, and society levels. As for the vehicle level, operation, electrification, design, and platooning could affect the vehicles' consumption and emission. This research focused on the impacts of the vehicle behavior. Greater driving efficiency might be obtained with AVs through an assortment of mechanisms, like, counting optimum driving cycle, dynamic system optimum routing, less idling, reducing cold starts, and speed harmonization.

Fewer idling or less cold start could help decrease fuel consumption and reduce emissions. Cold starts are significant causation of some air pollutants from the transportation area, for instance, NOx, CO, and Volatile Organic Compounds (VOCs) [Barth et al. 2014]. A few properties of AVs might lead to more energy waste. Radars, sensors, information exchanges, and high-speed network connections need more auxiliary energy from cars, which exposes as extra noteworthy power draw and subsequently more fuel consumption. Power used in sensing, connectivity units, and computing accessories could considerably change the AVs energy efficiency. Moreover, advanced protection in AVs might bring about higher freeway speeds. Because aerodynamic drag forces build up quadratically with speed, therefore, higher speeds bring about higher energy waste above a certain threshold. For example, a velocity increasing from 70 to 80 miles per hour was announced to raise average power usage by 13.9% per mile [Thomas et al. 2013]. Even though, it is convincing that improved security in AVs can empower relaxation of speed limits for freeways. In most countries vehicles are currently restricted to below optimal speeds, which lead to some fuel conservation. This aspect gains less consideration in the literature.

3.3.1 Macroscopic Modeling of Freeway Traffic Flow

The fundamental diagram is a well-acknowledged theory to characterize freeway traffic dynamics in a macroscopic way. In this approach averaged traffic flow variables are considered on a given road stretch, i.e. traffic density ρ (vehicles/km), space mean speed $V(\rho)$ (km/h) depending on ρ, as well as traffic flow $Q(V(\rho))$ (vehicles/h). The fundamental diagram of traffic flow is selected to show the impacts on the freeway capability. The theory of fundamental diagram includes three important relationships among the macroscopic variables, which are flow-density, speed-density, and speed-flow. These three diagrams are related via the following equation adopted from fluid dynamics:

$$Q(V(\rho)) = V(\rho). \tag{3.10}$$

A wide range of diverse forms of fundamental diagrams are available in the literature [Carey and Bowers 2012]. One practical form of the fundamental diagrams (which well reflects the real-world observations on freeways) is proposed by Newell [1993], i.e. the triangular flow-density relationship:

$$Q = \min \{v \cdot \rho;\ w \cdot (\rho_j - \rho)\},\ for\ 0 \leq \rho_j, \tag{3.11}$$

where v is free-flow speed, ρ is the vehicle density, w means the backward wave speed, as well as ρ_j stands for the jam density. The freeway link is assumed to follow the kinematic wave theory with all lanes having the same free-flow speed v and the same backward wave speed w, see Fig. 3.14. Q_m is the maximum flow rate, and the corresponding density ρ_c is referred to as the critical density. From the other parameters, $\rho_j = Q_m(1/v + 1/w)$ or $\rho_j = 1000/(l_v + minGap)$, the jam density can be estimated [Gayah et al. 2014]. Here, l_v is the length of vehicle, and $minGap$ is the minimum distance between two consecutive vehicles. Within the triangular flow-density relationship, the proportion of jam density ρ_j to critical density ρ_c is $1 + v/w$:

$$\rho_j/\rho_c = 1 + v/w. \tag{3.12}$$

The maximum flow Q_m is derived as:

$$Q_m = \frac{\rho_j w v}{w + v}. \tag{3.13}$$

In this study, the simulated traffic flow is in line with the kinematic wave theory and the triangular fundamental diagram is applied for practical reasons [Cassidy et al. 2011].

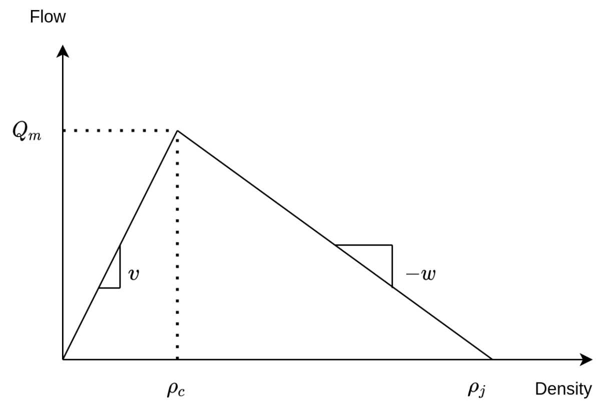


Figure 3.14. Triangular macroscopic fundamental diagram

3.3.2 Assessing AV Impacts on Freeway MFD

For realistic simulation based analysis SUMO simulator was applied. As shown in Fig. 3.15, a real-world network was selected for simulation based analysis. The test network is a 10 km section of the European designation E60 (Hungarian highway M1) near Herceghalom city. The base data for geometry, average traffic flow, as well other road features were gathered directly from the open database of Hungarian Public Roads.

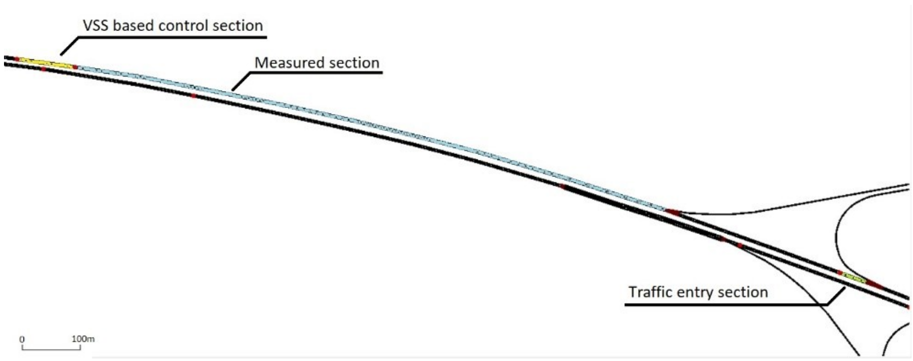


Figure 3.15. The network used in the simulations

In this research, a sensitivity analysis was carried out investigating three parameters: reaction time, AV penetration, and the speed limit to find their impacts on the MFD.

The realization of AV was accomplished through the parameterization of the vehicle behavioral models which are the Krauss car-following model and the LC2013 lane-changing models. The execution of different type of cars into SUMO simulation was realized by different vehicle types definition (declaring sets of cars with the same parameters and performance in transportation system).

In this work, 2 vehicle types were applied, which are Human Driven Vehicles (HDV) and AV. The parameters are listed in Table 3.8.

Table 3.8. The constant attributes of the different vehicle types

Vehicle Type	**Acceration** (m/s^2)	**Deceleration** (m/s^2)	**speed limit (km/h)**	σ	τ**(s)**	**speed factor**
HDV	2.6	4.5	80 - 200 with an increment of 20km/h	0.5	2	normc(1,0.1,0.2,2)
AV	2.6	4.5	80 - 200 with an increment of 20km/h	0	0.1-1.9 with a 0.2s increment	normc(1,0.1,0.2,2)

The acceleration and deceleration are considered as constant for HDV and AV. Speed limit means the speed limit of freeway for both HDV and AV. It varies from 80 km/h to 200 km/h with an increment of 20 km/h. Theoretical high speed limits are investigated. As the technologies are developing, a higher speed limit may be possible in the future. The σ in SUMO is the driver imperfection for which the value 0 denotes perfect driving. The AV is assumed have the perfect driving behaviors. The τ is the minimum desired time headway of drivers. The time headway for HDVs is a constant that equals to 2 seconds. The time headway for AVs vary from 0.1 s to 1.9 s with a 0.2 s increment. The speed factor is used to model the desired driving speed variation. It represents the driver's attitude towards the road's speed limit, by introducing a randomly generated multiplier. The speed factor in the study is a normal distribution which is given as "normc(mean, deviation, minimum, maximum)" form.

A full MFD should be built at each observed speed limit to realize the intentions stated at the start of this part. Relevant data were collected from the measured section in Fig. 3.15. The collected data are mainly the volume of vehicles on the measured segment as well as their average velocity. The traffic flow increased up continuously till the maximum road network capacity. Then an outside factor was applied to destabilize the flow. After multiple tests, the Variable Speed Signs (VSS) based traffic flow interference was chosen as it was the most reliable way to simulate full MFDs. The length of the measured edge is 1033.12 m, and it has two lanes, as shown in Fig. 3.15. Data is being collected every ten simulated seconds and was valid for the given time step (0.1 seconds). The nominal traffic flow was set based on the Hungarian Public Road's real-world measurement.

In order to obtain a full MFD, the simulation began with a low flow rate of traffic. The cars were introduced with Poisson distribution to guarantee randomization of the simulations. After planned periods, the vehicle flow increased up until reached the network's peak capacity. And then, the determined flow control program, VSS, was triggered to generate growing amounts of barrier on the road network, until the simulation reached a near grid-lock state. At this time, the interference terminated, and the flow started to release. When the jammed traffic flow disappeared, this simulation completed. Subsequently, a new simulation began with higher AV penetration. Fig. 3.16 shows the automatized simulation process of the work. The simulation was operated under three levels of nested loops. The variables of three-level loops from inside to outside are reaction time (τ), AV Penetration ($Pene$), and Speed Limit (SL). The reaction time of AV starts at 1.9 s and goes down to 0.1 s with a 0.2 s step length. The middle loop changes AV penetration rates starting from 0% to 100% with a 10% increment. The outer loop

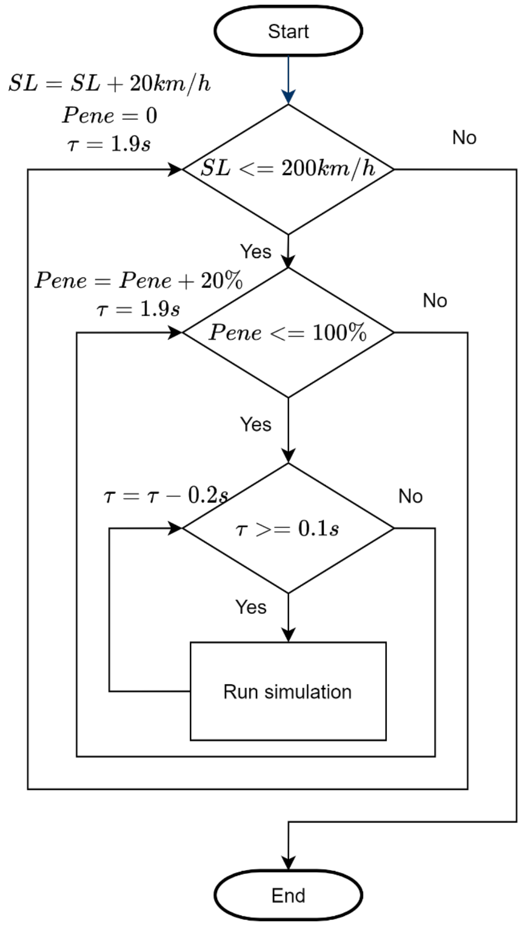


Figure 3.16. The automatized simulation process

defines the speed limit of the simulation, starting from 80 km/h with an increment of 20 km/h to 200 km/h. The simulation started with the scenario where the speed limit equals 80 km/h, AV penetration is 0%, and reaction time equals 1.9 s. After different reaction time scenarios simulated, the simulation goes on with the next predefined AV penetration and reset the reaction time to be 1.9 s. After running a full penetration range, the simulation goes on with a higher speed limit. The AV penetration is reset to 0; reaction time to 1.9 s. This procedure repeats until the outer loop finishes, which means all scenarios are simulated.

3.3.3 Simulation Results on AV Penetration Change on Freeway

To assess the impacts of AV penetration, reaction time variation, and speed limits, different simulation scenarios were carried out via the sensitivity analysis on reaction time, AV penetration, and the speed limits.

Fig. 3.17 shows the fundamental diagram of one of the simulations when the AV penetration is 40%, AV reaction time is 1.1 s, and the speed limit is 80 km/h. They show the fundamental diagram's typical shape, shared between all other simulation scenarios. From the flow-density relationship, one can see that the simulations have a triangular fundamental diagram. Therefore, a triangular fundamental diagram model was chosen to fit the collected data. The capacity of an edge is defined by the maximum vehicle number that passes the network per hour. It can be defined as the product of the traffic density and the space mean speed [Friedrich 2016].

Fig. 3.18 depicts the flow-density relationship of the simulations when the speed limit is 80 km/h and AV penetration is 60%. The time headway of AV varies from 1.9 s to 0.1 s. It is obvious that the maximum flow is increasing with the decrease of time headway. As the time

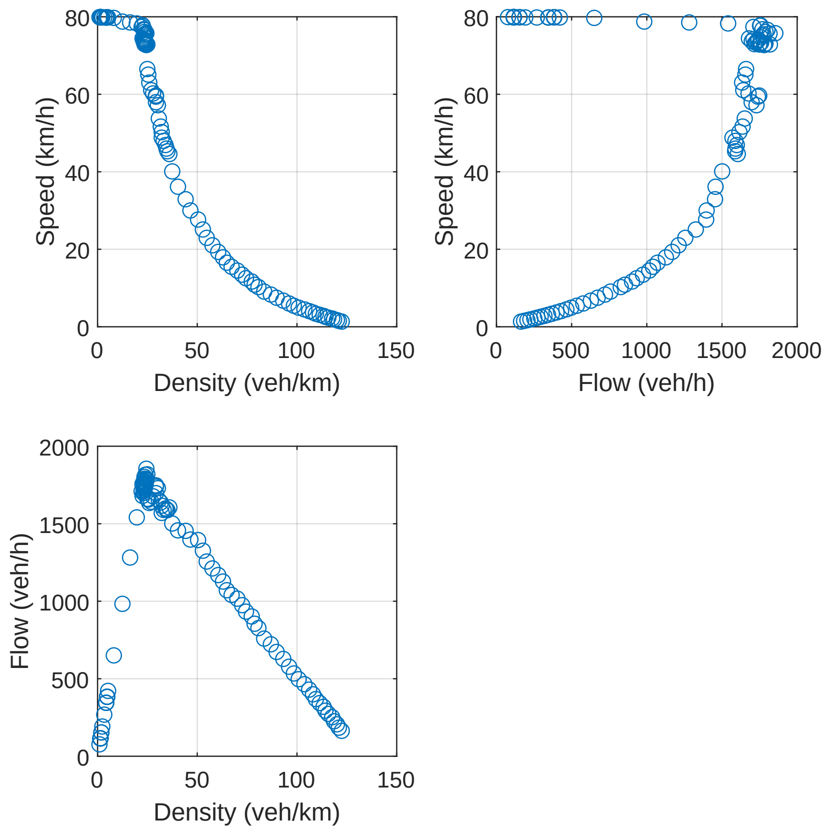


Figure 3.17. Fundamental diagram of a simulation (penetration = 40%, τ = 1.1 s, speed limit = 80 km/h)

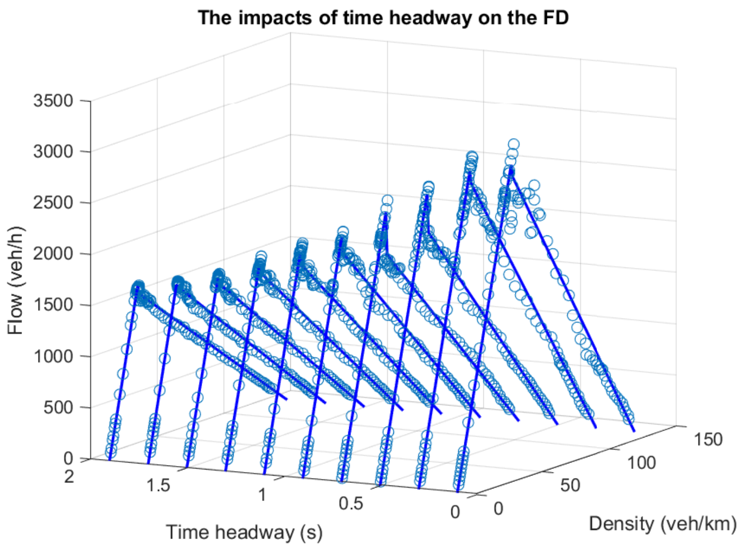


Figure 3.18. Flow-density relationship changes with reaction time

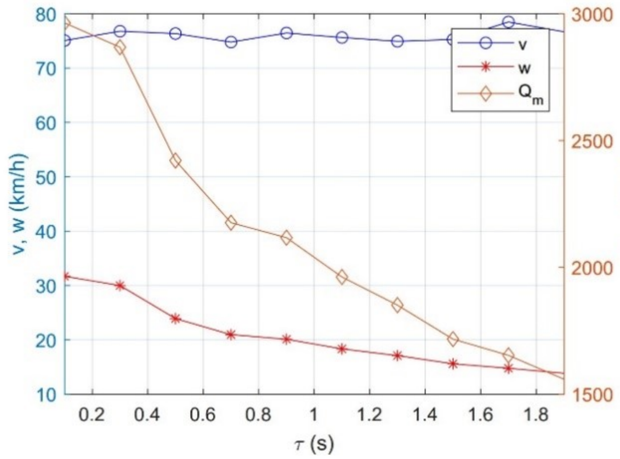


Figure 3.19. The triangular fundamental diagram parameters change with the time headway

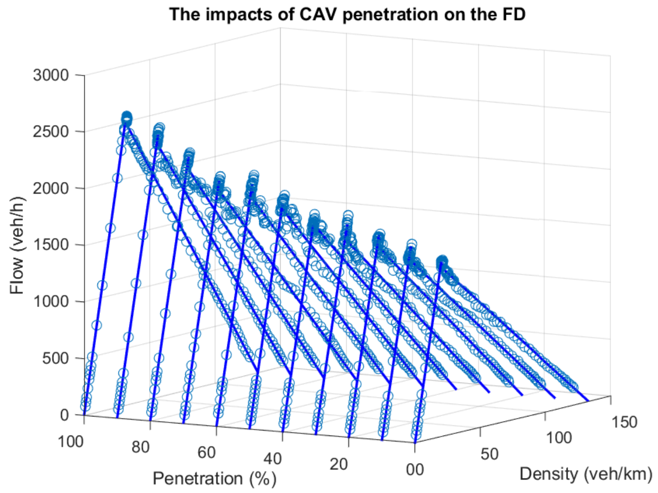


Figure 3.20. Flow-Density relationship changes with AV penetration

headway decreases, the points at the maximum flow rate become more and more scattered. On the one hand, it increases the capacity of the road, on the other hand, it reduces the stability of the traffic flow. Fig. 3.19 reveals the values of the fitted fundamental diagram. The free flow speed v is relatively stable. The wave speed w increases substantially with a shorter time headway. The maximum flow volume increases when the time headway becomes shorter.

Fig. 3.20 shows the impacts of AV penetration rate on the flow-density relationship when the speed limit is 80 km/h and reaction time is 1.1 seconds. Obviously, with the gradual increase in the AV penetration, the maximum flow is also rapidly augmenting. However, when the penetration rate is less than 40%, the advantage is not so apparent. This observation is in line with the results of Van Arem et al. [2006] and Jones and Philips [2013]. The highest traffic flow of pure AV scenario is almost twice that of pure traditional vehicle scenario. Fig. 3.21 depicts the changes of fitted triangular fundamental diagram parameters. The free flow speed v almost keeps the same. On the contrary, the wave speed w rises sharply. The maximum flow Q_m also increases drastically when there are more and more AVs on the road. Fig. 3.22 reveals the variation of the observed maximum traffic flow with AV penetration and time headway. From the impact of traffic flow, the shorter the time headway and the higher the market penetration is, the better the road performance becomes. Table 3.9 shows the observed maximum flow volume changes with different AV penetration and time headway. The simulations with 0% of

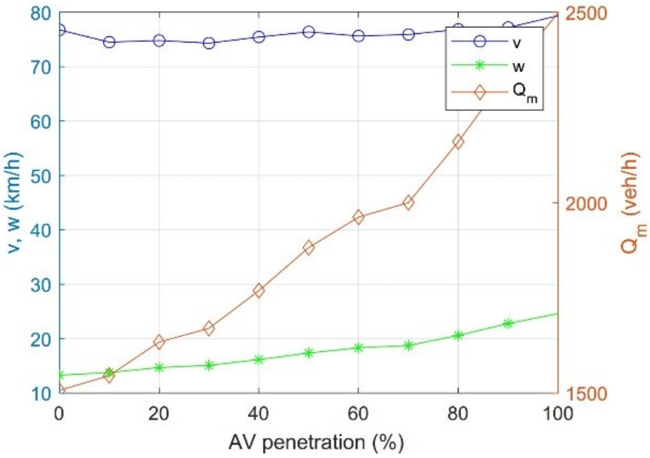


Figure 3.21. The triangular fundamental diagram parameters change with the AV penetration

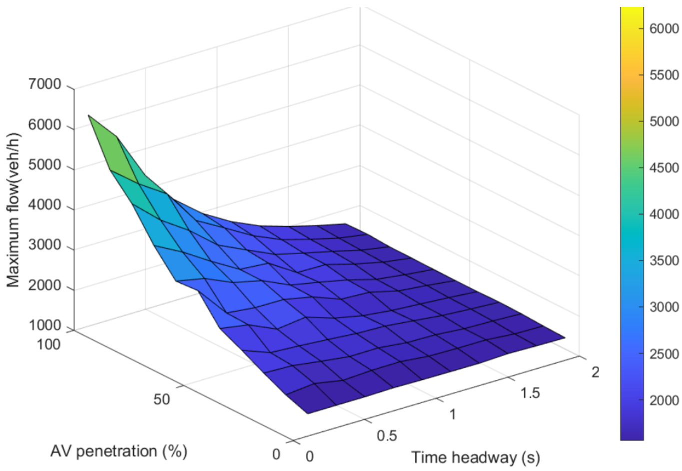


Figure 3.22. Maximum flow changes

Table 3.9. Maximum flow [veh/h] changes along with the AV penetration and time headway

Penetration	**1.9 s**	**1.7 s**	**1.5 s**	**1.3 s**	**1.1 s**	**0.9 s**	**0.7 s**	**0.5 s**	**0.3 s**	**0.1 s**
0%	1559	1570	1587	1558	1562	1567	1580	1566	1566	1565
10%	1580	1593	1656	1657	1672	1746	1740	1702	1833	1763
20%	1609	1640	1659	1718	1733	1814	1809	1902	2005	2029
30%	1622	1663	1713	1793	1871	1896	1957	2011	2053	2323
40%	1610	1672	1724	1773	1855	2004	2241	2153	2470	2589
50%	1611	1700	1774	1954	1991	2258	2406	2413	2492	3259
60%	1615	1693	1789	1984	2138	2279	2600	2762	3075	3218
70%	1623	1731	1894	2092	2103	2549	2606	3016	3458	3831
80%	1627	1754	1935	2102	2282	2604	2936	3510	3978	4596
90%	1663	1780	1962	2163	2471	2877	3273	4144	4606	5160
100%	1650	1817	2008	2231	2548	2960	3519	4331	5506	6257

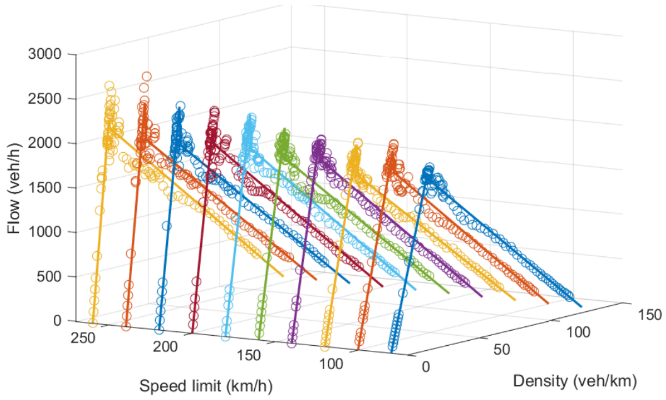


Figure 3.23. The impacts of speed limit on fundamental diagram

AV penetration rate are treated as the baseline in this research. As the AV penetration rises, the maximum flow volume grows as well. The shorter the time headway is, the higher the growth is. The speed limit has a direct impact on the free flow speed. With higher speed limit, free flow speed v is expected to increase. As an example, consider Fig. 3.23 and Table 3.9 displaying the simulation results with 1.1 s time headway and 50% AV penetration. In Fig. 3.23, the scatter points represent the collected data from the simulations. The solid lines are the fitted triangular fundamental diagrams. Fig. 3.23 reveals the free flow speed and road capacity increase when the speed limit gets higher. When the speed limit is higher than 160 km/h, the measured points around the critical density are more scattered. Therefore, in order to maintain the stability of the flow around the critical density, the speed limit should not exceed 160 km/h.

As shown in Fig. 3.24, the free flow speed v increases with higher speed limit. However, the value of w, the backward wave speed, remains almost the same while the speed limit varies. From Eq. (3.12) it is easy to derive that the critical density ρ_c moves to the left side slightly when the speed limit grows. This is consistent with the result shown in Fig. 3.23. The maximum traffic flow has a monotone growing trend, but the maximum difference is only 284 veh/h.

3.3.4 Impacts of AVs on Fuel Consumption and Emissions

In the simulations, it was assumed that the travel demand remains the same when AV penetration increases. The aggregate fuel consumption and CO_2 emission during the whole simulation run were investigated. The emission model used in the simulation was HBEFA3. For simplicity, the emission class for all vehicles was gasoline driven passenger car of European emission stan-

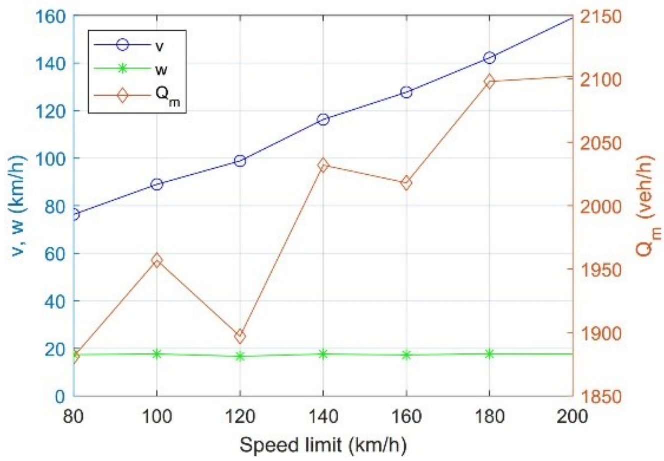


Figure 3.24. The triangular fundamental diagram parameters change with the speed limit

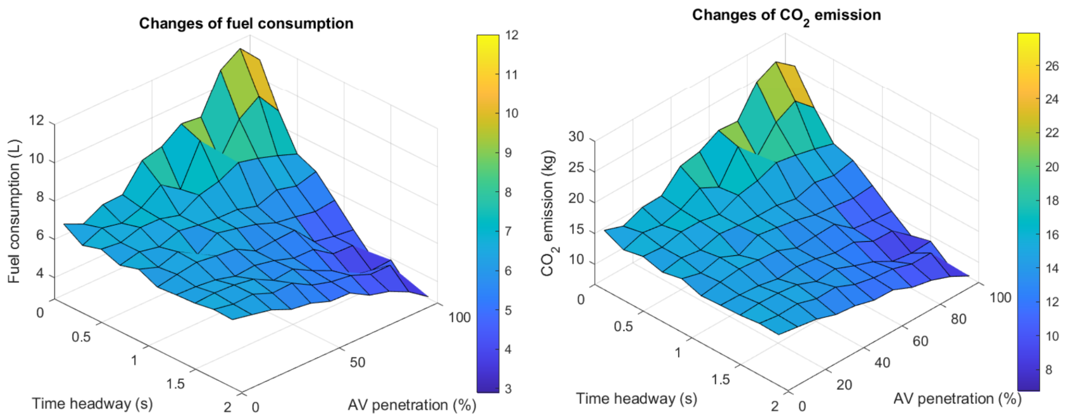


Figure 3.25. Fuel consumption changes and CO_2 emission changes

dards 5. Left of Fig. 3.25 indicates the aggregate fuel consumption over the entire simulation when the speed limit is 80 km/h. When the AV time headway is 1.1 seconds, the consumption does not change much, no matter what the AV penetration is. When the time headway less than 0.5 seconds, fuel consumption increases sharply with the rise in AV penetration. The all HDV scenarios serve as the baseline. One can observe that when time headway is longer than 0.5 seconds the aggregate fuel consumption drops with the increase of AV penetration. When the reaction time becomes shorter than 0.5 seconds, there will be more frequent braking and starting and more changes in velocity, especially in traffic jams. As mentioned in the introduction, breaks, starts, and velocity variations may contribute to increasing fuel consumption. From the perspective of reducing fuel consumption, 0.5 seconds is the lower recommended limit of the reaction time setting for AVs.

The results of CO_2 emission share a similar shape with the fuel consumption. Right of Fig. 3.25 indicates the aggregate CO_2 emission during the whole simulation. When the AV response time is under 1.1 seconds, the emission does not change much no matter what the AV penetration is. However, when the time headway is less than 0.5 seconds, carbon dioxide emissions increase sharply with the increase of AV market share. The all HDV scenarios serve as the baseline. One can see that when time headway is longer than 0.5 seconds the aggregate CO_2 emission is dropping with the increase of AV penetration. From the perspective of reducing carbon dioxide emissions, 0.5 seconds is the lower limit of the reaction time setting for AVs when speed limit is 80 km/h.

3.4 Contributions

The thorough analysis of the AV impacts on traffic dynamics resulted in new scientific contributions in terms of urban and freeway macroscopic fundamental diagram fitting and related scientific experiences.

Thesis 2

The impacts of Autonomous Vehicles (AVs) on road traffic have been investigated using microscopic traffic simulation technique via the macroscopic fundamental theory. I have proposed a methodology to determine the urban and freeway macroscopic fundamental diagrams (MFD) in case of the presence of AVs in traffic flow. Additionally, I have found relationship between the AV penetration and the MFD theory in both urban and freeway contexts.

Thesis 2.1

I demonstrated that AVs have significant potential to improve traffic capacity, efficiency, stability, and safety of existing urban mobility systems. A thorough sensitivity analysis has been carried out to find the impacts with different AV penetration rate applying grid shape (common in the USA) and irregular (typical European) urban traffic networks. A new approach for macroscopic average speed has been proposed as follows

$$V(r,\rho) = \alpha + s(\rho) + \beta \cdot r + \gamma \cdot r \cdot \rho, \tag{3.14}$$

where $s(\rho)$ is a spline (depending on density ρ) estimated together with the model parameters (α, β and γ). The ratio of AVs among all vehicles (r) enters the regression equation twice. First, it affects the free-flow speed if the estimate of β becomes statistically significantly different from zero. Second, it affects the slope of speed-density function through the interaction term between r and ρ. This semiparametric representation is powerful in prediction of macroscopic average speed considering the ratio of AVs. To fit the speed - AV ratio - density relationship Generalized Additive Model (GAM) based regression method was applied.

Thesis 2.2

R^2 and *p-values* in the estimation of Eq. (3.14) showed that the GAM regression is a reliable method to determine the speed - AV ratio - density relationship. This also implies the estimation of urban MFD as well via the well-known fundamental equation of $Q(\rho) = \rho \cdot V(\rho)$ adopted from fluid dynamics. The results of the urban MFD fittings showed that the capacity is increasing quasi-linearly with higher AV penetration for both grid and real-world networks. This improvement is due to shorter headway and less reaction time of autonomous vehicles. When self-driving cars start dominating the roads, a plateau occurs around the maximum flow. This phenomenon means that no specific maximum is obtained, i.e. maximum throughput is available on a longer area and not at a dedicated point as (critical density of conventional MFDs).

Thesis 2.3

I investigated the impacts of AVs on freeway traffic via the triangular flow-density relationship, i.e. the traffic flow is

$$Q = \min\left\{v \cdot \rho;\ w \cdot (\rho_j - \rho)\right\},\ for\ 0 \leq \rho \leq \rho_j, \tag{3.15}$$

where v is the free-flow speed, ρ is the vehicle density, w means the backward wave speed, as well as ρ_j stands for the jam density. I demonstrated that the parameters of the freeway MFD vary in correlation with the change of time headway, AV penetration and speed limit. I also found that reducing time headway in a certain range may result in lower fuel consumption and emission. At the same time, when the time headway is lower than this range, the energy consumption and emission increase sharply. In conclusion, from the perspective of fuel economy and emission reduction, a recommended optimal time headway must be found.

Related journal and conference papers

- T. Tettamanti and I. Varga. A jövő intelligens járművei és az infokommunikáció hatása. *Híradástechnika*, LXXI:59–63, 2016. „Smart City a célkeresztben" különszám;
- T. Tettamanti, I. Varga, and Zs. Szalay. Impacts of autonomous cars from a traffic engineering perspective. *Periodica Polytechnica ser. Transp. Eng.*, 44(4):244–250, 2016c. doi: 10.3311/PPtr.9464 **(Q2)**;
- Q. Lu and T. Tettamanti. Impacts of autonomous vehicles on the urban fundamental diagram. In *5th International Conference on Road and Rail Infrastructure, CETRA 2018,*, pages 1265–1271, 17-19. May 2018;
- M.T. Horváth, T. Tettamanti, and I. Varga. Az autonóm járműforgalom modellezhetősége mikroszkopikus forgalomszimulációs szoftverben. *Közlekedéstud. Szemle*, LXVIII(2):34–44, 2018. doi: 10.24228/KTS.2018.2.3;
- T. Tettamanti and I. Varga. Az autonóm járművek forgalmi hatásai: a jármű- és forgalomirányítás kihívásai. *Közlekedéstud. Szemle*, LXIX(1):35–41, 2019. doi: 10.24228/KTSZ.2019.1.4;
- B. Maximcsuk, Q. Lu, and T. Tettamanti. Determining maximum achievable flows of autonomous vehicles based on macroscopic fundamental diagrams. *Perner's Contacts, Special Issue: "36th International Colloquium on Advanced Manufacturing and Repair Technologies in Vehicle Industry"*, pages 192–199, 2019;
- Q. Lu, T. Tettamanti, D. Hörcher, and I. Varga. The impact of autonomous vehicles on urban traffic network capacity: an experimental analysis by microscopic traffic simulation. *Transportation Letters*, 12(8):540–549, 2020. doi: 10.1080/19427867.2019.1662561 **(Q2)**;
- H. Lengyel, T. Tettamanti, and Zs. Szalay. Conflicts of automated driving with conventional traffic infrastructure. *IEEE Access*, 8:163280–163297, 2020 **(Q1)**;
- Q. Lu and T. Tettamanti. Impacts of connected and automated vehicles on freeway with increased speed limit. *International Journal of Simulation Modelling (IJSIMM)*, 20(3), 2021b. doi: 10.2507/IJSIMM20-3-556 **(Q1)**.

Chapter 4

Urban Road Traffic Control

Automated vehicles have the advantage of interactive communication possibilities towards other vehicles or the infrastructure. Autonomous road vehicles with this option are also called Connected and Autonomous/Automated Vehicles (CAVs). In this part novel control methods are proposed for both private and public road vehicles. Additionally, the novel concept of distributed road traffic signal controller is introduced.

4.1 Dynamic Pricing Based Urban Traffic Control

The routing problem of road traffic, when only a few alternatives exist between an origin (O) and a destination (D) point (Fig. (4.2), is a common road traffic issue. Consider for example the accessibility of a city center from the end of a given freeway or the typical suburban traffic flow which intends to reach a specific destination zone. Another frequent case to describe this problem is the routing between two traffic nodes, e.g. the proper route choice between an important freeway node and a bridge. This setup is indeed very common in Budapest (Hungary) where Variable Message Signs (VMS) in several points of the network suggest to drivers the shortest path in an indirect way by displaying travel times of important destinations.

In our days, the high degree of ICT (Information and Communication Technology) device utilization or ITS (Intelligent Transport System) tools help travelers to make their route choice in order to avoid congestion, thus minimize travel time. These tools are, on the one hand, web based applications such as Google Maps providing color coded speed categories on links. On the other hand, VMS may display travel time information by a traffic management center. Although drivers continuously check real-time traffic information via smartphone applications or they are directly informed by VMS, the traffic topology introduced above induces the problem of uneven utilization of the possible routes, i.e. bad route choice. This phenomenon is mainly due to the inertia characteristics of the traffic mass, i.e. when a given route is more preferred and so increasing traffic demand enters, then there is a notable time delay until the steady state will be reached. Simply, the increased traffic volume at the beginning part of the specific route will not cause immediate travel time increase on the whole route. While the travel time estimation is typically provided as an online service intending to reflect the actual state of the transportation system, the utilization of this information cannot be real-time as the traffic state can change after picking a given route. Furthermore, ICT tools or VMS displays do not give short term traffic forecast yet and generally do not deal with the delay in the provided information. In practice the chosen path might become congested in contrast with the previous information due to the information given to a large number of travelers who respond similarly and therefore change the traffic state can lead to oscillation on short run. Therefore, traffic flows on the possible routes (Fig. 4.5) start an oscillatory behavior in case of heavy (peak hour) traffic demand when the traffic network is saturated or close to the saturated state. Adequate road toll system might be a possible solution to achieve an optimal route choice in such situation in system level on short run. At the same time, the adequate determination of the applied tolls is not a straightforward problem. Fixed tolls result in a rigid system which might be only effective in case of a more or less constant traffic demand. In case of variable traffic (for example in rush hours) dynamic toll system is needed. In general, user optimum has higher total cost than system optimum, sometimes much higher [LeBlanc and Abdulaal 1984]. Previous studies have shown that system optimum [Long et al. 2018; Zhao and Leclercq 2018] is more difficult to achieve than user optimum [Ma and Gao 2016]. In case of system optimum, the total utilities are maximized, meanwhile in user optimum the individual user utilities are maximized. In case of system optimum, some individual utilities will decrease for the common good [Du et al. 2015]. Road pricing has been recognized as an efficient approach to traffic demand management and control which can be introduced incrementally, according to need, with the feasible aim of reducing traffic congestion and keep it on an acceptable level [Metz 2015]. Study of Zhu et al. [2017] proposed to use road pricing as a tool to capture the value of travel time savings in order to induce better road usage patterns. By using road pricing as a tool to spread out the peak demand, it could improve the utility of travelers. Although a time-varying toll could achieve full efficiency, its implementation is a big challenge. In reality, almost all practices follow either a uniform toll, or a step-toll (peak vs. non-peak, or with a smaller time step). Based on the classical consumer behavior theory the users would maximize their utility meanwhile the traffic control could influence this utility due to pricing. Dynamic tolls are shown to be meaningful from both economic and behavioral viewpoints, and therefore proposed to be a good alternative toward system optimum [Yang 1999].

4.1.1 Control Design Based on the Concept of Traveler's Utility Function

Utility represents satisfaction experienced by the consumer of a good or service. Utility maximization is the main goal of rational choice theory in economics. It is very hard to directly measure satisfaction or happiness from consuming a good or service, therefore economists introduced utility functions [Houthakker 1950]. Mathematically let us consider the simple linear utility function (proposed by Koppelman [1981]) for traveling:

$$\mu = -\alpha T - \beta C \tag{4.1}$$

where T is the travel time [min], C is the the toll price concerning the specific route choice (e.g. in EUR), α and β are appropriate weighting parameters. Considering the utility function of the i^{th} route Eq. (4.1) can be generalized as follows:

$$\mu_i = -\alpha T_i - \beta C_i L_i \tag{4.2}$$

where i refers to the i^{th} route alternative and c_i is the unit toll price [EUR/km].

Note that in Eq. (4.2) toll cost is linked only to distance traveled but not to travel time, i.e. distance-based pricing is applied. The main reason for this is that a tolling system with time-based prices can yield several problems. On the one hand, parking time has to be differentiated from travel time, because the pricing schemes of the two activities differ. On the other hand, it can be problematic from traffic safety aspect, as time-based pricing forces road users to hurry on user optimum.

The road traffic dynamics of urban or metropolitan area can be described by the urban MFD. An example for the MFD is depicted by Fig. 4.1. where x counts the number of vehicles within the traffic network, Q^{out} denotes the output traffic flow [vehicle/hour].

In the vicinity of the critical number of vehicles (x_{crit}) the network is operated at maximum capacity. Moreover, x_{crit} splits the fundamental diagram to a stable (left side) and an unstable part (right side). Apparently, the unstable part of the MFD represents traffic congestion as

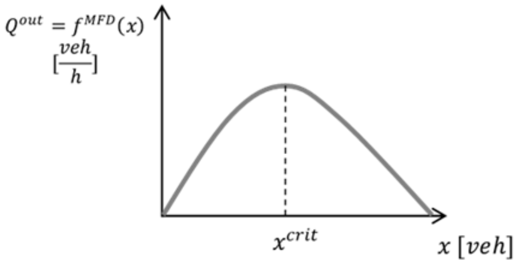


Figure 4.1. Urban macroscopic fundamental diagram

network throughput Q_{out} tends to decrease. By using the MFD theory together with the basic rule of the conservation law, the following nonlinear and discrete-time traffic model can be deduced:

$$x(k+1) = g(x(k), u(k)) = x(k) + T_s Q^{in}(k) - T_s Q^{out}(k) \tag{4.3}$$

where g is the function of state variable $x(k)$ and control input $u(k)$ (defined below). k denotes the discrete time step [sec] and T_s is the discrete sample time (for simplicity T_s is equal to the control measure period) [sec]. Practically, Eq. (4.3) depicts the state variation over the time period $[kT_s, (k+1)T_s]$. State variable x can also be represented as passenger car unit [PCU] (the different types of road vehicles can be expressed in the ratio of private car). Inflow traffic $Q^{in}(k)$ compasses the sum of entering vehicles. Total outflow traffic is defined by using the function of the MFD based on Keyvan-Ekbatani et al. [2014]:

$$Q^{out}(k) = f_{\mathrm{MFD}}(x(k)). \tag{4.4}$$

The analogy of MFD based traffic model can also be applied for a given route considered as traffic network. Practically, in this approach a given MFD is equivalent to a given route option between an origin and a destination point, e.g. two different MFDs (f_1^{MFD} and f_2^{2MFD}) can be defined for Route 1 and 2 as illustrated by Fig. 4.2. In this case, Eq. (4.3) can be generalized for the i^{th} route:

$$x_i(k+1) = g_i(x_i(k), u_i(k)) = x_i(k) + T_s q_i^{in}(k) - T_s q_i^{out}(k). \tag{4.5}$$

Similarly to Eq. (4.4) the outflow can be defined as

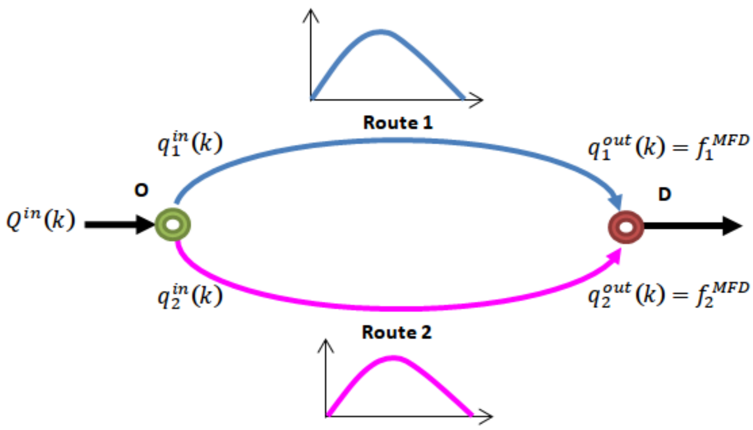


Figure 4.2. MFDs for Route 1 and 2 between origin and destination pair

$$q_i^{out}(k) = f_i^{\mathrm{MFD}}(x_i(k)). \tag{4.6}$$

The controlled input traffic flow of the i^{th} route is given as:

$$q_i^{in}(k) = Q^{in}(k)u_i(k) \tag{4.7}$$

where $u_i(k)$ is the control input which attributes weighting to the total entering flow (Q^{in}). Note that in this system formulation the input traffic flow is the control, while the output traffic flow is determined by the characteristics of the system, namely its MFD.

As equivalent utility function is assumed for all drivers, $u_i(k)$ is defined as a binary control input variable for the i^{th} route:

$$u_i(k) \in \{0, 1\}. \tag{4.8}$$

It is assumed that within time interval $[kT_s, (k+1)T_s]$ vehicles decide identically concerning the route choice, i.e. in a typical peak hour time workplace/school travel is dominated. As $u_i(k)$ depends on the utility function (defined by Eq. (4.2)), control input $u_i(\mu i(k))$ is calculated based on the maximal achievable utility function:

$$u_i(\mu i(k)) = \begin{cases} 1 & \text{if } \ \mu_i(k) = \max\{\mu_1(k), \mu_2(k), \dots, \mu_n(k)\} \\ 0 & \text{otherwise.} \end{cases} \tag{4.9}$$

Practically, index i is sought where $\mu_i(k)$ obtains the maximal value for $i = 1, 2, \dots, n$. The above formula needs the assumption that human drivers have relevant traffic information through ITS/ICT tools for all route options $i = 1, 2, \dots, n$ before they make their route choice decision. By considering the example shown by Fig. 4.2, Eq. (4.9) means practically that $u_1(k) = 1$ and $u_2(k) = 0$ if $\mu_1(k) > \mu_2(k)$, or $u_1(k) = 0$ and $u_2(k) = 1$ if $\mu_1(k) < \mu_2(k)$.

Utility function of the i^{th} route is given by Eq. (4.2). However, it is modified with a time delay (h) [min] in the first term:

$$\mu_i(k) = -\alpha T_i(k-h) - \beta c_i(k)L_i. \tag{4.10}$$

The replacement of $T_i(k)$ by $T_i(k-h)$ reflects the driver's slightly outdated perception, i.e. when traveler makes a route decision, he/she considers the currently available travel time information of the route options. However, the average travel time of the i^{th} route always reflects the state shaped by an antecedent traffic demand entered the network by time interval h before. It practically means that travelers decide based on delayed travel time information. Therefore, it is a reasonable operation to consider this delay in the utility function as well as in the urban traffic dynamics.

If there is no road toll or it is uniformly applied as a fixed (i.e. time-invariant) toll for the different routes, the traffic dynamics can be described as self-controlled system based on the model Equations (4.6)-(4.10), i.e. in this case drivers make their choice to the best of their abilities (knowledge on the traffic system) with cost parameter $c_i(k) = constant$. Therefore, it is practically a user self-controlled system which will be analyzed later by simulation.

To obtain better efficiency, the dynamic control system can be applied to represent a time-variant toll function $\varphi(T_i)$ as depicted in Fig. 4.3. T_i^{free} is an important point of $\varphi(T_i)$, i.e. the average travel time at free-flow traffic concerning the i^{th} route:

$$T_i^{free} = \frac{L_i}{v_i^{free}} \tag{4.11}$$

where v_i^{free} is the generalized average free-flow speed (km/h), i.e. an idealized case without any congestion on the network. It is reasonable to apply a minimum and a maximum toll charge

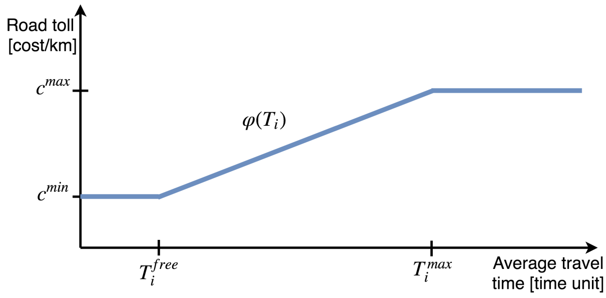


Figure 4.3. Toll function

(c^{min}, c^{max}) for the system. Of course, T_i^{max} is a practical value to pair c^{max} and maximum travel time can be infinite theoretically. According to Calfee and Winston [1998], transport policy is resistant to use congestion tolls to minimize the social costs from urban congestion. Therefore, a linear charging is considered model to find the short-term equilibrium. With the increase of average travel time, the road tolls are increasing linearly. The determination of the toll can be summarized mathematically as follows:

$$c_i(k) = \begin{cases} c_i^{min} & \text{if } T_i < T_i^{free}, \\ \varphi(T_i) & \text{if } T_i^{free} \le T_i \le T_i^{max}, \\ c_i^{max} & \text{if } T_i > T_i^{max}. \end{cases} \tag{4.12}$$

The previously introduced fixed and dynamic control schemes have a common drawback: both are affected by the time delay of the available travel time information. Thus, a different approach is needed which is capable to take into consideration the time-delayed system model. For this purpose, MPC (Model Predictive Control), also called rolling-horizon control [Maciejowski 2002] can be applied as one of the most powerful adaptations for online optimization with built-in predicitive modeling. At each time step, this receding horizon technique:

1. predicts the traffic conditions along a future time window based on the system model (even with a time delay);
2. minimizes an *a priori* defined network-related performance index by finding the appropriate control input; and
3. applies the first sequence of optimal decision variables.

Accordingly, the optimal control may take shape as a non-linear MPC (NMPC), where the control inputs are computed by minimizing an objective function $J(k)$ over the prediction horizon K. This problem is formulated as a non-linear optimization task.

$$\begin{aligned} \min_{\mathbf{u}(k+l-1)} \quad & J(k), \\ \text{subject to} \quad & \mathbf{u}(k+l-1) \in \mathbb{U}, \\ & \mathbf{x}(k+l) \in \mathbb{X}, \quad l = 1,\ 2,\ \ldots,\ K. \end{aligned} \tag{4.13}$$

where $\mathbb{U}$ and $\mathbb{X}$ denote the constraint sets for control input vector $\mathbf{u}$ and state vector $\mathbf{x}$, respectively. $J(k)$ is defined as a quadratic objective function

$$J(k) = \sum_{l=1}^{K} \left\| \mathbf{x}(k+l) - \mathbf{x}^{crit} \right\|_2^2 \tag{4.14}$$

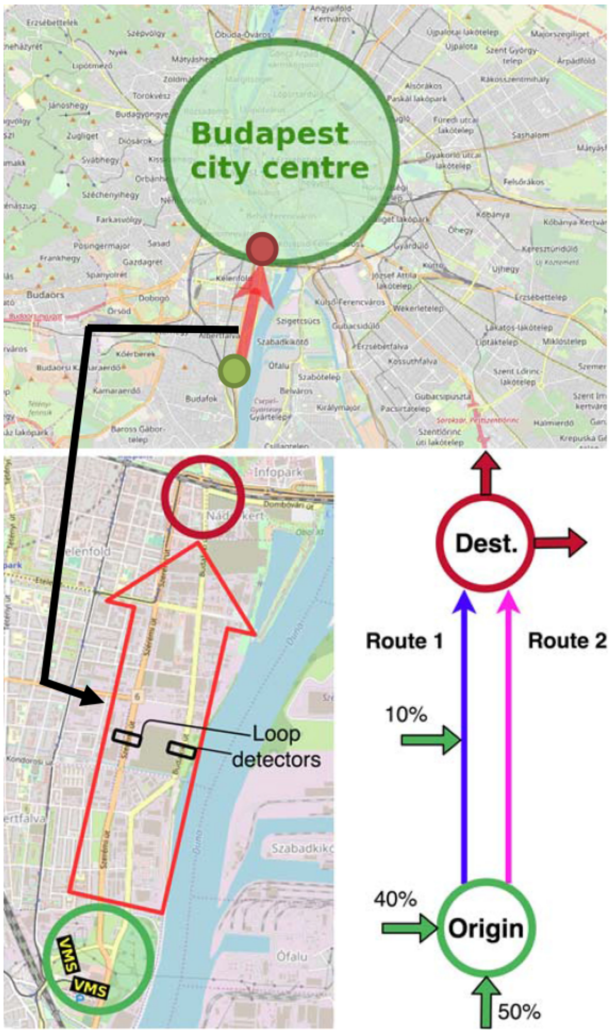


Figure 4.4. Test network (area covered by the red arrow) in Budapest (Hungary), global positioning system: 47.438778, 19.043547

where $\mathbf{x}^{crit}$ represents the vector of objective values concerning the different MFDs ($i = 1,\ 2,\ \dots$) (Fig. 4.2), i.e. the objective function intends to minimize state deviation from the optimal value ($\mathbf{x}^{crit}$).

$$x_i^{crit} = \arg\max_x f_i^{\mathrm{MFD}}(x). \tag{4.15}$$

The control input vector is composed of the binary control variable $u_i(k)$ (as defined by Eqs. (4.8)-(4.9) which is dependent on the utility function $\mu_i(k)$ and $\mu_i(k)$ influenced by road toll $c_i(k)$

$$\mathbf{u}(k) = \begin{bmatrix} u_1(\mu_1(c_1)) \\ u_2(\mu_2(c_2)) \\ \vdots \\ u_n(\mu_n(c_n)) \end{bmatrix},\quad i = 1,\ 2,\ \dots,\ n. \tag{4.16}$$

$J(k)$ is minimized subject to the constraints listed below:

- The non-linear difference equation of the state dynamics (originally given by Eq. (4.5)) concerning the i^{th} route must be held during the NMPC optimization over the prediction horizon K :

$$x_i(k+1) = x_i(k) - Tf_i^{\mathrm{MFD}}(x_i) + TQ^{in}(k)u_i(\mu_i(k)). \tag{4.17}$$

- To ensure binary control input $u_i(k) \in \{0, 1\}$, one should apply Eq. (4.9). However, it is not a continuous formula; thus, cannot be directly applied for the NMPC solver. Hence, an exponential formula is applied as non-linear equality constraint for $u_i(k)$:

$$u_i(\mu_i(k)) = \frac{e^{\kappa\mu_i(k)}}{\sum_{i=1}^{n} e^{\kappa\mu_i(k)}} \tag{4.18}$$

  where κ is a weighting parameter. Via appropriately chosen κ , Eq. (4.18) only results in values 0 or 1.

- Utility function is also given as equality constraint:

$$\mu_i(k) = -\alpha T_i(k-h) - \beta c_i(k) L_i. \tag{4.19}$$

- $c_i(k)$ is also restricted as

$$c^{min} \leq c_i(k) \leq c^{max}. \tag{4.20}$$

- Optionally, it can be ensured that the optimal tolls do not differ too much from the previously calculated tolls

$$|c_i(k) - c_i(k-1)| \leq \delta \tag{4.21}$$

  where δ is a design parameter. Simply, Eq. (4.21) helps to avoid toll oscillation. It is emphasized that the system model is applied to the consideration of the time-delay effect. The control term in Eq. (4.17) is depending on the travel time estimates with delay h, i.e. $u_i(T_i(k-h), c_i(k))$. Therefore, the NMPC solver must be configured such that it works with delayed travel time $T_i(k-h)$.

4.1.2 Application Example Using Real-World Traffic Network Data

For simulation purposes, a real-world network within Budapest was chosen (Fig. 4.4) as a realistic study area. Traffic data of the test area was provided by the Centre for Budapest Transport and Budapest Public Road Plc. This area perfectly reflects the problem illustrated by Fig. 4.2, i.e. the corridor traffic between two specific city zones. Namely, the daily morning traffic from the South to North (from the suburban directions toward the city center) flows via two route options: route 1 or 2. The choice of this zone is also justified by the fact that VMS displays are applied in the vicinity of the South traffic input point. Now, these VMSs provide the current travel times of the two route options, i.e. the problem of this traffic zone is an existing issue. For realistic simulation setup, the integrated multimodal transport model of Budapest was applied. The distribution of input traffic demand is shown in Fig. 4.4. Also, for the determination of the MFDs, real-world data were used obtained from loop detector measurements on the two alternative routes ((Fig. 4.4). The MFDs were finally defined as fourth-order polynomial functions with appropriate parameters A, B, C, D as follows:

$$f^{MFD}(x) = Ax^4 + Bx^3 + Cx^2 + Dx. \tag{4.22}$$

In the literature typically second, third or even higher-order approximations are used for MFD fitting, e.g. Keyvan-Ekbatani et al. [2014] or Csikós et al. [2014]. In the simulation study, fourth-order polynomial was applied in order to better catch the dynamics of the traffic system. The MFDs are shown in Fig. 4.5.

To run thorough simulation tests, a MATLAB/Simulink test environment has been realized with 3 options for road pricing based control:

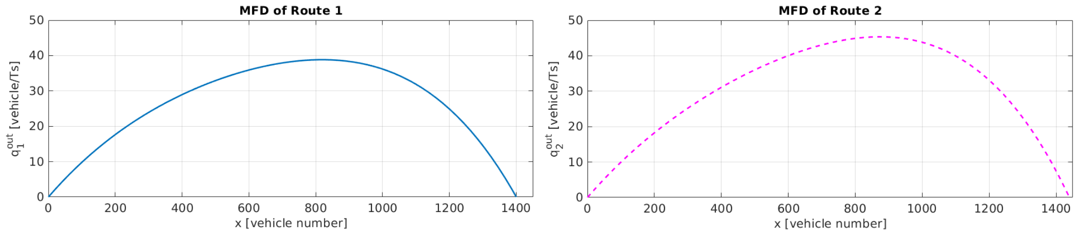


Figure 4.5. MFDs of Route 1 and 2

1. Fixed (time invariant) control;
2. Dynamic control;
3. NMPC.

To simulate travel time $T_i(k-h)$ for Eq. (4.19), a well-known exponential formula for the macroscopic average speed was applied, proposed in the literature [Papageorgiou et al. 1990] with model parameter a

$$v_i(x_i) = v_i^{free} e^{-\frac{1}{a}\left(\frac{x_i}{x_i^{crit}}\right)^a}. \tag{4.23}$$

The average travel time on the i^{th} route was calculated in all simulation scenarios simply as follows:

$$T_i(k) = \frac{L_i}{v_i(x_i(k))}. \tag{4.24}$$

The simulation scenario with fixed control was basically created to mimic a human driver's behavior when on both route options fixed tolls are applied. In this scenario, it is assumed that drivers simply choose the fastest route if the road toll is static. In the simulations, the normalized road price was used ($c_1 = c_2 = 0.5km^{-1}$) in order to avoid modeling real currency. The utility was, therefore, calculated based on Eq. (4.10) with the fixed costs for route $i = 1,\ 2$ as

$$\mu_i(k) = -\alpha T_i(k-h) - \beta 0.5 L_i. \tag{4.25}$$

The logic of the fixed control used in the simulation is as follows:

$$u_1(k) = \begin{cases} 1 & \text{if } \ \mu_1(k) > \mu_2(k) \\ 0 & \text{else} \end{cases} \quad u_2(k) = \begin{cases} 1 & \text{if } \ \mu_1(k) \le \mu_2(k) \\ 0 & \text{else.} \end{cases} \tag{4.26}$$

Then, the controlled input traffic flow of the i^{th} route can be calculated based on Eq. (4.7).

The applied simulation scenario for dynamic control was modeled similarly to the fixed control case with the only difference that dynamic prices were applied. By using the toll function, defined by Fig. 4, the price of 1 km road usage was calculated as follows:

$$c_i(k) = \begin{cases} c^{min} = 0.1 & \text{if } \ T_i < T_i^{free} \\ \varphi(T_i) = mT_i & \text{if } \ T_i^{free} \le T_i \le T_i^{max} \\ c^{max} = 1 & \text{if } \ T_i > T_i^{max} \end{cases} \tag{4.27}$$

where m is an appropriately chosen parameter, i.e. the slope of the linear function $\varphi(T_i)$, and T_i^{free} is the free-flow travel time defined previously by Eq. (4.11). Thus, the utility can be determined as given by Eq. (4.10) with the costs calculated in Eq. (4.27). The controlled input traffic flow of the i^{th} route can be finally given by Eq. (4.7).

The simulation scenario with NMPC was applied according to the Eqs. (4.13)-(4.21). The applied solver for the NMPC optimization was borrowed from Grüne and Pannek [2011]. The weighting parameter κ was 10^2. Both prediction and control horizon lengths were set to 6 in the NMPC optimization.

To better understand the effect of time delay h on the network dynamics, a sensitivity analysis was carried out. Simulations were run with three different time-delay values ($h = 0,\ 2,\ 5,\ min$) on the three simulation scenarios defined previously. The results are compared through the average speeds (see Table 4.1).

Control method	$h = 0$ min	$h = 2$ min	$h = 5$ min
Fixed	30.4	29.3	26.5
Dynamic	30.7 (+1%)	29.6 (+1%)	28.3 (+3%)
NMPC	30.9 (+2%)	30.6 (+4%)	30.6 (+16%)

Table 4.1. Average traveling speeds (km/h) with different time delays (in parentheses the relative changes are given compared with the fixed control case)

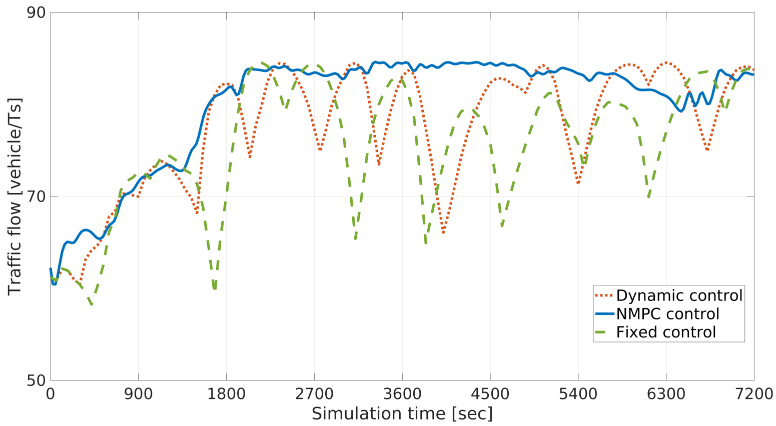


Figure 4.6. Traffic flows achieved with different control methods

The analysis clearly reflects the efficiency of the non-linear predictive control as NMPC incorporates the future modeling of states when calculating the actual control input. The evolution of the aggregated traffic flows in the network (i.e. flows on route 1 and 2 are summed) for the three different control cases are shown in Fig. 4.6. It is clearly observable that the maximal throughput of the network can be guaranteed the best by the NMPC method.

In this study the effect of utility-based traffic control was demonstrated on a closed-loop test system, which can provide an effective usage of road infrastructure without congestion. Moreover, the time-delay effect in response to vehicle concentration has been investigated. The simulation results showed that time delay causes oscillation on the route alternatives in case of fixed and dynamic control. However, the NMPC method described is insensitive to delay effect, and thus can be used as an efficient regulator type for utility-based traffic management. Concerning the practical implications of the results, one can emphasize that the technology is already given to apply dynamic road tolling.

4.2 Dynamic Routing for Automated Public Transport Buses

Control for public transport buses is an important issue in order to improve service quality [Polgár et al. 2013; Polgár et al. 2011a,b]. Control methods to avoid bus bunching are deeply investigated in the literature [Varga et al. 2017, 2018b, 2020a]. However, the published methods are based on fixed routes of buses. As a new approach, dynamic routing methodology is presented for autonomous or highly automated buses that have to reach certain stops at given times. The route choice process is based on minimizing a generalized cost function and also taking into consideration the expected departure times from stops.

4.2.1 Routing Methods for Public Buses with Timetable-Based Operation

Central management of public buses is a general technology of our days: it is expected that public transport vehicles are continuously monitored, and the drivers can be instructed real-time depending on traffic conditions. Even though it would be possible, public transport vehicles do not leave their predefined route. Therefore, the edge-cost dynamic routing method, introduced in the dissertation, can also be applied to traditional vehicles as an on-board advisory system followed by the driver before it is used in AVs. In these cases, it may be sensible to suggest the "simplest path" (with as few turns as possible) instead of the shortest path so that the routing information can be easily followed by the driver [Duckham and Kulik 2003].

In urban networks heuristic shortest path algorithms can be efficiently applied for routing. As Fu et al. [2006] conclude, these algorithms can be generally classified into four strategies: (i) limit the area searched, (ii) decompose the search problem, (iii) limit the links searched, and (iv) combination of the previous methods.

When taking more objectives into consideration, Pareto optimization can be used to find relatively fast the set of routes that are obviously better than the other options, since this requires no objective aggregation which decreases the complexity of such algorithms. The longer the computational time is, the more solutions can be found that represent all choices [Diakonikolas and Yannakakis 2009]. But as Disser et al. [2007] state, even if the Pareto optima are found, the search algorithms need more time to find a certificate that no further solutions exist.

4.2.2 Modeling Framework

This section introduces a routing approach that allows timetable-based automated vehicles to travel on different paths between given points, minimizing the generalized cost of the route. Between public transport stops, possible routes are modeled as a continuously updated weighted directed graph. The weights represent relevant parameters of links, collected from surrounding sensors and monitoring systems of the network. Route optimization is done by Yen's algorithm [Yen 1970].

The model uses a special edge-cost algorithm [Storandt 2012], generally it proposes the route with the least generalized cost, but if the vehicle is late, it chooses the fastest way. The proposed route planning methodology consists four main stages:

1. General preparation is to calculate offline as many steps as possible in advance to spare computing capacity and data traffic. Between consecutive stops a subnetwork is determined, which is monitored if the vehicle has to find an alternative route. The subnetworks are determined by the physical attributes of the vehicle and the streets, especially width (narrow turns) and traffic rules (turn prohibitions and one-way streets). Expected departure times from stops are also collected and offline traffic models are prepared (see the details below).

2. The sequence of stops is determined in advance, each station has to be reached at a specific time. As the vehicle moves, the state of the subnetwork between the previous and

the following stop is queried periodically and after each query the k-shortest paths are calculated using Yen's algorithm [Yen 1970] taking into consideration a generalized cost that can be determined by the user depending on the relevant aspects of route choice. If there are only fewer than k paths, then the calculation continues with this set of paths.

3. Travel times on these paths are calculated and paths exceeding the travel time limit (i.e. to reach the next stop on time) are excluded. The path with the least generalized cost is chosen from the remainder. If no paths among these k-shortest paths can guarantee the arrival on time, another shortest path search is done based only on travel time, in order to choose the fastest way to minimize delay.

4. Steps (2) and (3) are repeated until the next stop is reached. The whole process ends when the last stop is reached.

The flowchart of the process is shown in Fig. 4.7.

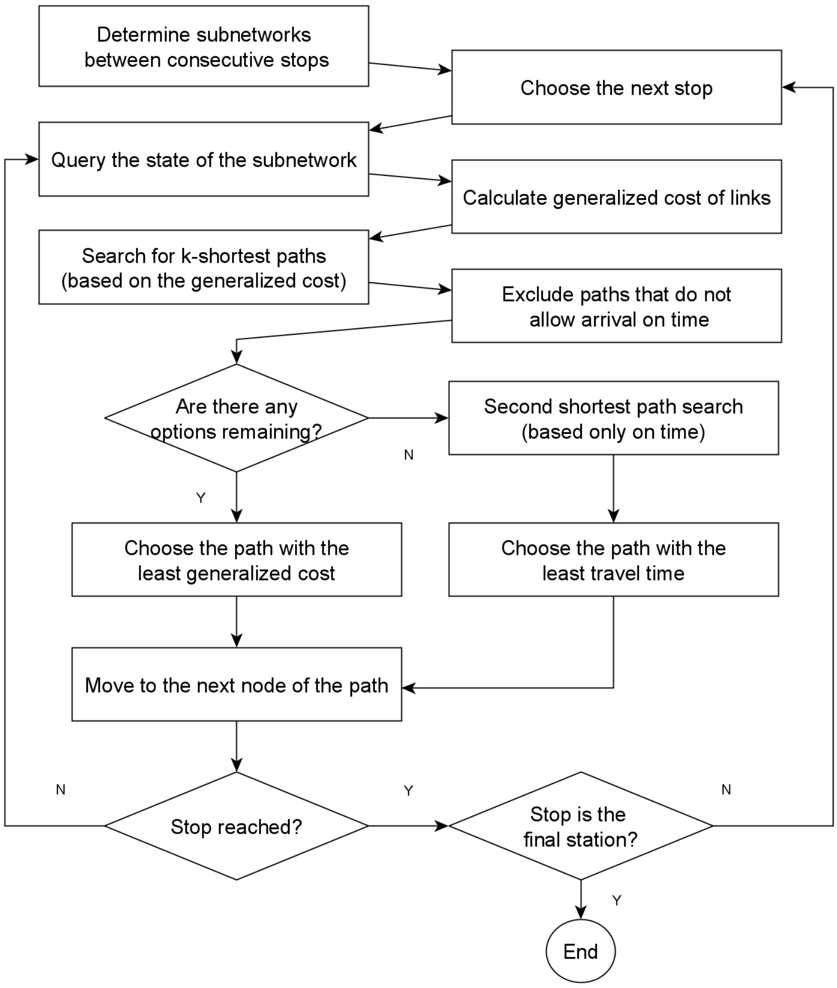


Figure 4.7. The routing process

The route-search algorithm can be written as follows. Subnetwork G between stops s and $s+1$ is considered as a weighted directed graph

$$G = (N, L, w) \tag{4.28}$$

with N nodes and L links and a cost function $w : L \to \mathbb{R}^+$, which gives each link $(i, j) \in L$ a positive cost (i.e. weight): $w(i, j) \in \mathbb{R}^+$. Formally, the cost function is extended as

$$w : N \times N \to \mathbb{R}^+ \cup 0 \cup \infty \tag{4.29}$$

with $w(i, i) = 0$ for all $i \in N$ and $w(i, j) = \infty$ for all $(i, j) \notin L$. The chosen route is selected by comparing paths $x_0 l_1^p,\ x_1^p l_2^p, \dots,\ x_n^p l_n^p$ for $i = x_0 \in N$ to $j = x_n \in N$ in subnetwork (graph) G from $p = 1$ to P, where P is the number of possible paths between i and j in subnetwork G. The cost of path p is calculated as

$$C^p = w(x_0, x_1^p) + \sum_{m=2}^{n} w(x_{m-1}^p, x_m^p). \tag{4.30}$$

There are two types of costs calculated in the application example: a time cost (C_t^p) that is actually the time needed to get to the next station and a generalized cost (C_g^p), which can be determined by the user by combining relevant factors. In the example showed later in this thesis, it is the combination of trip duration and trip length, but it can include numerous different factors as well, e.g. monetary cost or the user satisfaction [Apáthy 2017], or even external costs, such as consequences of air pollution on health [Jadaan et al. 2018]. As the first step of the route-search algorithm the k-shortest paths ($p = 1,\ 2,\ \dots\ \ k \in K$) are chosen based on the generalized cost ($C_g^1,\ C_g^2, \dots,\ C_g^k$). Then trip durations of these paths are calculated and those that exceed the limit (t_{limit}) to reach the next stop on time are excluded. The alternative with the least generalized cost is chosen from these remaining paths: $min(C_g^p),\ p \in (P \cap K);\ C_t^p \le t_{limit}$. If there is no path among the k-shortest paths that make it possible to reach the next stop on time, another route search is done based only on time cost. In this case the shortest path of this search is chosen: $min(C_t^p),\ p \in P$.

4.2.3 Application Example

The introduced routing methodology can be operated between stops of which the arrival and departure times are previously determined. Therefore, the whole journey of the vehicle can be divided into equivalent sub-processes between consecutive stops. In this example the route planning methodology is shown between such two stops. The key point is that the vehicle cannot leave any stop earlier as it is in the timetable.

A subnetwork is defined between each stop pair, in the example from stop 1 (S1) to stop 2 (S2). The vehicle departs from the starting stop and queries the state of the traffic on the links of the subnetwork (see the black links in Fig. 4.8).

The k (in this case 3) shortest paths with the least generalized cost, A (dash-dotted line), B (dashed line) and C (solid line) are queried and also the predicted arrival times are calculated (see Fig. 4.9). The vehicle chooses the alternative with the least generalized cost if the arrival is predicted on time. In this case, it is route A.

As the vehicle approaches the next node, it calculates again the 3 shortest paths in the predefined subnetwork. In the example an incident occurred on the previously decided route (marked with an 'x' mark in Fig. 4.9), therefore, route A is no longer among the best 3 choices. The new set of choices is B (dashed line), C (solid line) and D (dash-dotted line) of which the only one predicting arrival on time is B, so the vehicle chooses this option, even if the generalized cost of this route is not the lowest one. Note that as the vehicle approaches S2, the generalized cost decreases in normal cases, since the values of distance and travel time left decrease (see Fig. 4.9).

Then, the vehicle starts to move on route B and as it reaches the next node, it calculates again the 3 shortest paths based on the generalized cost, checks predicted arrival times, excludes alternatives with late predicted arrival and chooses the route from the remainder with the least generalized cost. If there is no such route, a second shortest path search is done based only on

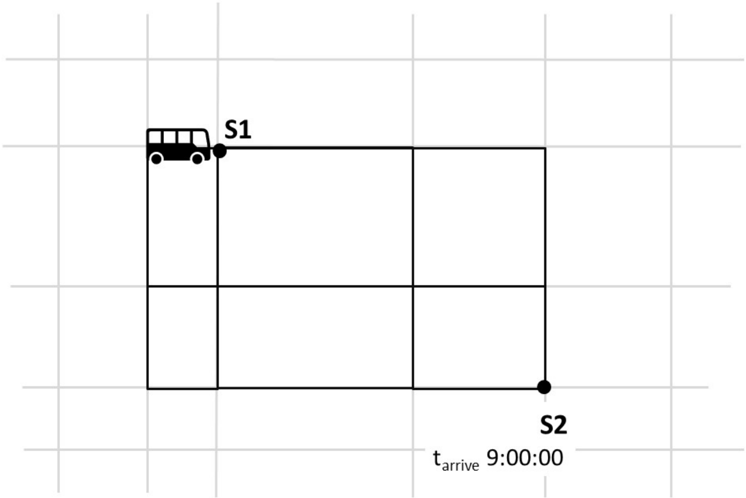


Figure 4.8. The monitored subnetwork

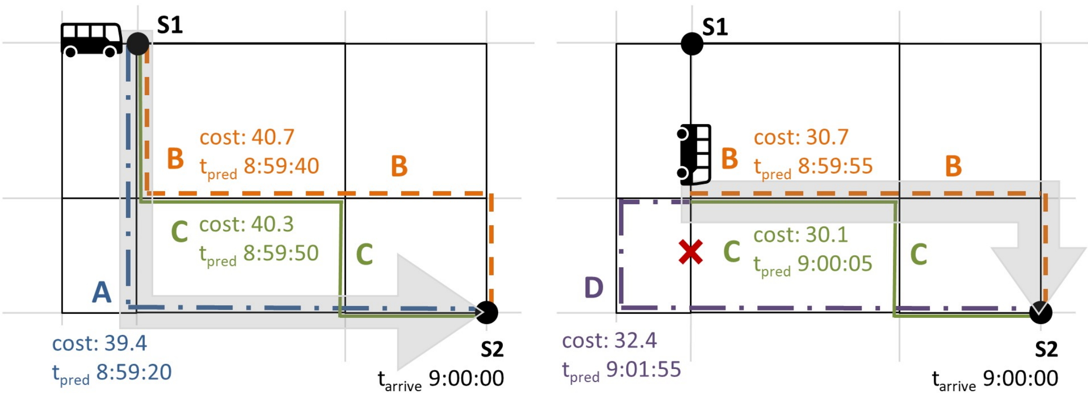


Figure 4.9. First route choice (left) and second route choice (right)

travel time. The sub-process ends when the vehicle reaches S2. The whole process ends when the vehicle reaches the final stop.

The introduced example has been extended and coded in Matlab software environment to investigate the performance of the algorithm. The route search was done between S1 and S2 on the same subnetwork as shown in Fig. 4.8. There was a reference route (route A in Fig. 4.9) representing a traditional transport route when the path between stops is fixed and no deviation is possible. The reference route has the lowest generalized cost and fastest travel time on an empty network. Simulation runs started from S1 and lasted until S2 was reached. At the end of each simulation run the generalized cost and the duration of the actually completed path were compared to the measures of the original reference route. Overall 1100 scenarios were generated, 990 with regular traffic and 110 with an incident on the link marked with the 'x' mark (4.9). The scenarios were generated by assigning random numbers of velocity to links between a lower bound of 3 m/s and an upper bound of 4 m/s, taking into consideration the average speed of public transport buses in heavy traffic conditions in off-street areas [Oskarbski et al. 2015]. When an incident occurs the travel time on the marked link will be five times longer than as usual. Velocity values were always re-generated after a vehicle reached the next node, therefore, the state of the network changed dynamically.

4.2.4 Simulation Results

Three aspects of the algorithm were investigated: (1) what is the effect of the number of shortest paths searched on the simulation time and how does the (2) generalized cost and (3) travel time of routes change compared to the predefined route with different weighting values α and β (see the details later).

Having run the simulations for the generated scenarios with each k-shortest path values from $k = 1$ to $k = 20$ the average run times are taken into consideration. Certainly, run times depend on the complexity of the network. Therefore, a relative value of run times is generated by dividing the average run time of case $k(\overline{T_k})$ by the average run time of case $k = 1(\overline{T_1})$. The average relative run times $(\overline{T_k^r})$ searching the k-shortest paths are generated as follows:

$$\overline{T_k^r} = \frac{\overline{T_k}}{\overline{T_1}} \tag{4.31}$$

where $\overline{T_k}$ is the average run time of simulations when k-shortest paths are searched and $\overline{T_1}$ is the average run time of simulations when only the absolute shortest path is searched. The average relative run times are shown in Fig. 4.10.

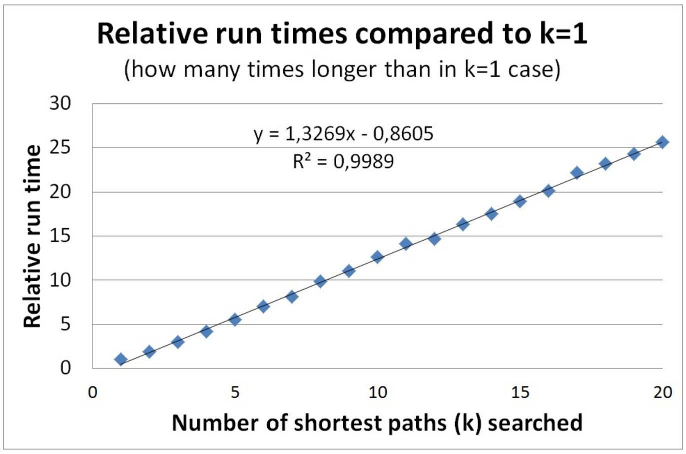


Figure 4.10. Average relative run times with different k values

The average relative run time is almost a linear function of the number of shortest paths searched, therefore, the number of shortest paths searched can be decided arbitrarily based on the performance of the computer when using the algorithm. In the example case, $k = 3$ is chosen, because the size of the subnetwork is relatively small and computing with more alternatives does not lead to better results, but in bigger networks it is worth to increase this number. The effect of weighting factors α and β is investigated using $k = 3$ configuration.

The generalized cost function can be constructed by the user, depending on their preferences, what factors they consider important. In the example case, it is constructed as follows:

$$C_i = \alpha l_i + \beta t_i \tag{4.32}$$

where C_i is the generalized cost of link i, l_i, and t_i are the length and the current travel time on link i, $\alpha[1/m]$ and $\beta[1/s]$ are weighting factors of the lengths and travel times of links. In the example the connection between the weighting factors is the following:

$$\alpha + \beta = 1. \tag{4.33}$$

Eq. (4.33) is not necessary, α and β can be completely independent of each other, but it is useful to have some constraints, because for example increasing both values simultaneously at the same time will not have any effect on the route choice. In the application example presented, the lengths of links are shorter than 100 meters, whereas travel times are mostly between 25 and 33 s during incident-free conditions. So if the values of α and β are equal, then $\alpha l_i > \beta t_i$, i.e. the length of the link will have a stronger influence on the generalized cost. As α decreases and β increases, the travel times of links will be more dominant. Certainly, travel time is also in connection with the length of the link, but it is true for many factors that can be taken into consideration, for example emission, tolls etc.

It should be noted that the overall computational time depends only marginally on the complexity of the cost function. Analyzing the run time of the sub-processes results in that calculation of the cost function takes only approximately 5% of the total run time. This does not increase significantly even if more complicated objective functions are deployed. The most time-consuming process is running Yen's algorithm which takes 81.5% on average. The internal processes of the software take the remaining 13.5% of the total simulation time. The effect of weights on generalized cost and travel time were evaluated by comparing the cost of the route the vehicle completed to the cost of the originally defined route by calculating with current travel times in each period and the route lengths, which are permanent. By comparing the predefined and the realized routes the costs can decrease, increase or stagnate depending on whether the realized route has a lower, the same or a higher generalized cost than the predefined one.

Overall 1100 simulations were run, 100 with each 11 combinations of α and β for $\alpha = 1$ and $\beta = 0$ to $\alpha = 0$ and $\beta = 1$ (see Figures 6 and 7). Both the generalized cost and the travel time decreased in most cases with the combination of $\alpha = 0.1$ and $\beta = 0.9$, a bit more than 50% of simulations resulted in increased performance, however, in this case almost 20% of simulations provided weaker results than without using the algorithm (see Figures 6 and 7).

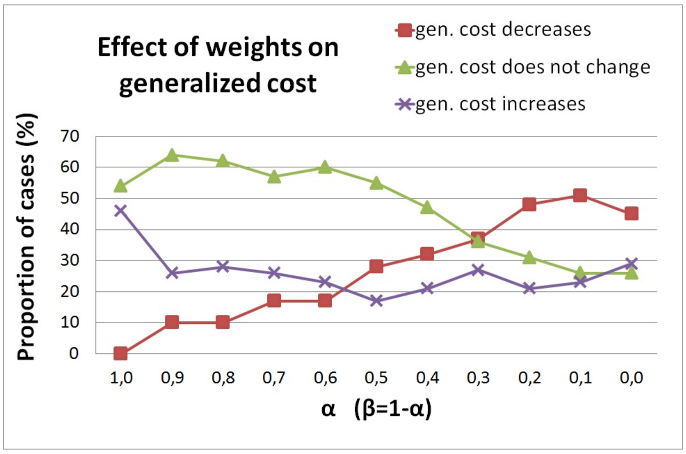


Figure 4.11. Effect of weighting parameters on generalized cost (chosen route vs predefined reference route)

As a general consequence using a practical engineering approach $\alpha = 0.5$ and $\beta = 0.5$ values seem to be a sensible compromise, since both the generalized cost and travel times improve in about 30% of the simulations, whereas there are only 10% when performance decreases (see Figs. 4.11 and 4.12). Nevertheless, the proper choice of α and β values is always in connection with the characteristics of the actual network, e.g. topology.

The overall performance of the algorithm was compared to Dijkstra's algorithm [Dijkstra 1959] which gives the basis of the method. The reference case was Dijkstra's algorithm optimized for travel time. The difference in travel times and covered distances of the proposed algorithm and Dijkstra's algorithm were analyzed during the 1100 simulation runs, the results are shown in Fig. 4.13.

As the value of α increases (until $\alpha = 0.6$), the proportion of simulation runs where the

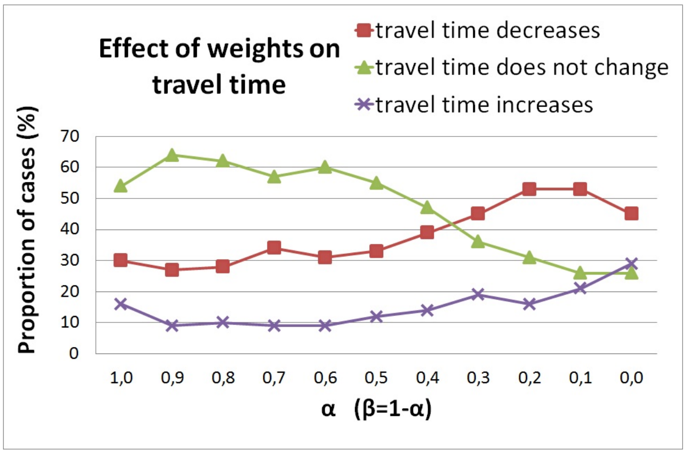


Figure 4.12. Effect of weighting parameters on travel time (chosen route vs predefined reference route)

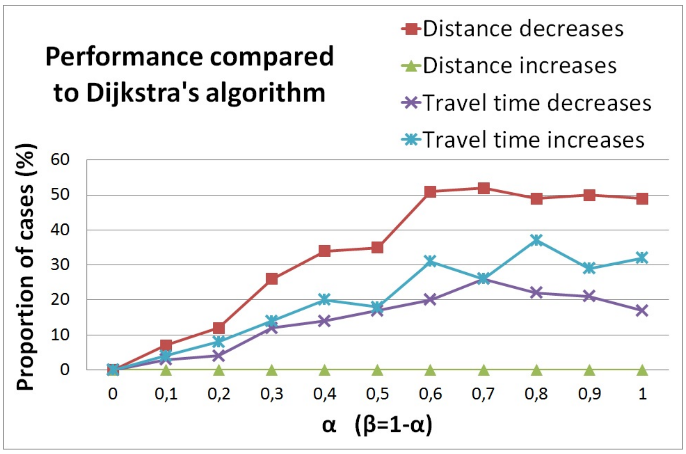


Figure 4.13. Performance compared to Dijkstra's algorithm

vehicle ran on a shorter route increases compared to the reference Dijkstra case. The reference Dijkstra algorithm is based only on travel time, but the proposed algorithm also includes distance during calculations, therefore, travel distance never increases. Note that the $\alpha = 0$ and $\beta = 1$ case equals to the reference Dijkstra case, therefore, the two methods result in the same route.

There is only difference in travel times when the two algorithms suggest different routes, i.e. the proposed algorithm suggests a shorter route. Travel time decreases in a bit less than half of these cases, in the other part it increases. When a routing decision is made, the future cannot be foreseen exactly, therefore, sometimes the proposed methodology, which is not clearly optimized for travel time, results in a faster route than the Dijkstra case, which focuses only on travel time. Certainly, when the vehicle is predicted to arrive late at the next scheduled stop the proposed algorithm is also based only on travel time, which equals to the reference Dijkstra case.

4.3 Wireless Traffic Signal Controller with Distributed Control System Architecture

Traffic signals (also called as traffic lights) are control devices at road intersections or pedestrian crossings to ensure safe and efficient traffic flows. The world's first traffic light (with gas-lit signals) was designed by J.P. Knight, a railway engineer, and launched in London in 1868. Then, the first electric traffic lights, similar to today's traffic signal heads, started operating at the beginning of the 20th century in the USA. Since then, this technology has spread everywhere especially due to the persistent expansion of road traffic. A signalized road junction basically consists of a central controller unit, traffic signal heads, as well as electric power cables, realizing a fully centralized system (Fig. 4.14). In this concept all signal commands are sent to the light sources directly from the central controller unit by switching the corresponding relays. Hence, the traffic lights are electrically energized according to the central controller's command which is a one bit information practically, i.e. current does or does not flow to the light sources. This traditional concept has been in use from the beginning of the traffic light's history, for more than 100 years (the conventional architecture is depicted by Fig. 4.14. The technology of our days, however, enables completely different system architecture for signalization in which the signal heads can be controlled not only by simple electric power but via digital messages (which is more than a single bit of information obviously).

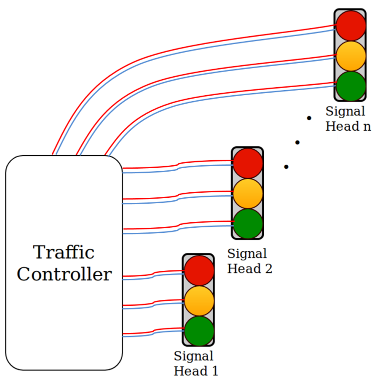


Figure 4.14. The architecture of the traditional centralized traffic signal controller with pair power cables to each light source

An innovative realization in this field is the CAN-communication (Controlled Area Network) based traffic signalization which is just prior to practical introduction provided by the Swarco Group [Swarco Group 2018]. Also, the Siemens company uses CAN interfaces for external communication for the control of motorway traffic management systems [Siemens AG 2019]. As CAN protocol has mainly been used in the automotive industry, it is a safe and reliable technology for controlling signal heads as well. In the practical applications of [Siemens AG 2019], a central architecture is used where a main controller unit is in charge of control and power switching respectively via CAN. In this setup the electric power for the light sources can be supplied from the central controller unit (as conventionally used) or locally from the traffic signal pole (the latter case is illustrated by Fig. 4.15.

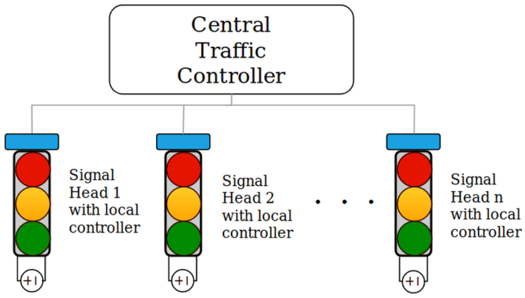


Figure 4.15. The architecture of the CAN bus communication based (centralized) traffic signal controller with a single CAN cable to each signal head (power supply is provided in the traffic signal poles)

This technology naturally indicates that some logic is necessary at the signal head directly. The digital information must be processed between the central unit and the signal head unit (local controller). Thus, the light sources are controlled according to the relay switching of the local controllers. Another unconventional approach is straightforwardly resulted from the emergence of wireless communication technology, i.e. signal heads can be controlled without direct physical connection to the central controller unit. Moreover, the advent of Autonomous Vehicles (AV) make new technologies possible to be applied in traffic control. AVs need to visually sense the signal heads to gather information of signal heads, or it can be provided via wireless communication.

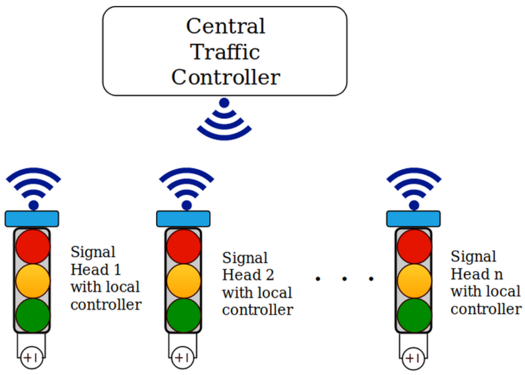


Figure 4.16. The architecture of the centralized wireless traffic signal controller (power supply is provided in the traffic signal poles)

The basic concept of the wireless traffic light with central control system architecture (see Fig. 4.16) has already been introduced by [Bo and Fusheng 2013; Thatsanavipas et al. 2011]. All these works presented a master-slave operation where a central controller (as a master unit) controls all signal heads (slaves) through wireless data transmission (the electric power is provided locally at each signal head). In this setup, again some logic is necessary at the signal head level identically to the CAN bus based concept (presented previously by Fig. 4.15).

Although these papers have shown the basic idea of wireless traffic lights, the presented concepts rest on central control architecture and are limited in terms of reliable and safe engineering design according to the standards of road traffic signal control.

4.3.1 Distributed Control Architecture for Traffic Light Controller

In practice, the traditional traffic light controllers work on the basis of a central control unit. Similarly, the papers investigating the wireless concept [Bo and Fusheng 2013; Thatsanavipas et al. 2011] introduced centralized control system (see Fig. 4.16). Although in their approach some local processor is also applied at the signal heads, the control logic is operated in a strict centralized way.

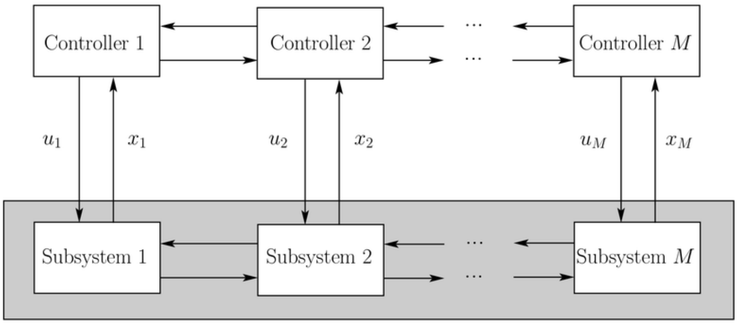


Figure 4.17. Distributed control architecture

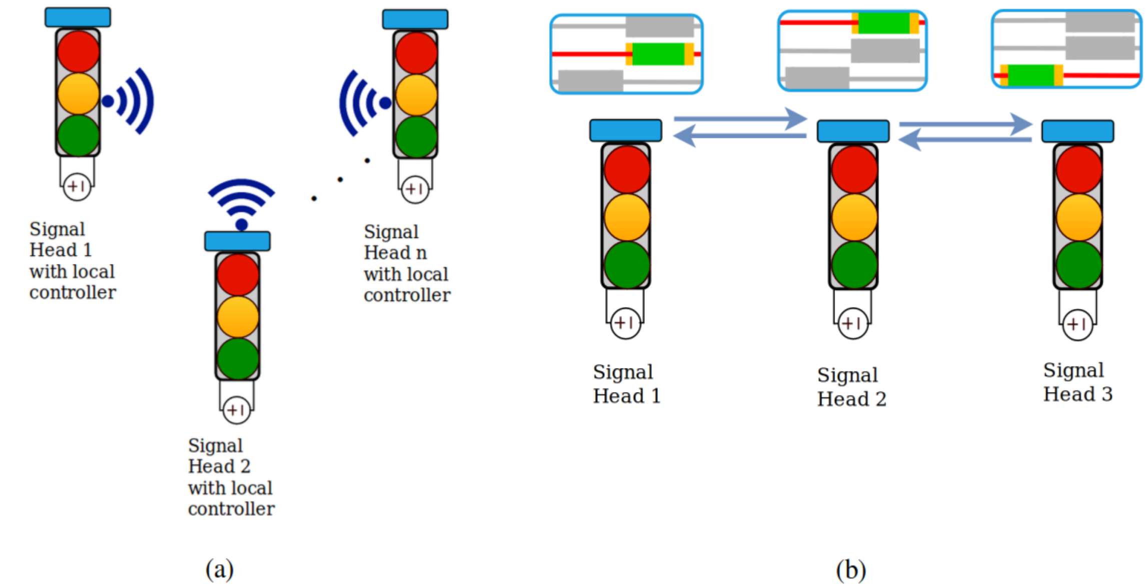


Figure 4.18. The architecture (a) of the wireless traffic signal control with distributed control system (power supply is provided in the traffic signal poles) and an example for the distributed functioning (b) representing the common knowledge of the actual signal program

Identically to the notions used in control engineering, one can distinguish central and distributed system architectures for road traffic controllers as well. Centrality means that all information about the system is gathered at a single point, where all the calculations are executed based on this information. Contrarily, in a distributed realization the computational tasks are divided among the local units [Tettamanti and Varga 2010]. Distributed control scheme is depicted by Fig. 4.17 where u and x mean control and state signals, respectively for $i = 1, 2, \ldots M$ subsystems. Moreover, communication among the controllers and among the subsystems are also applied. Accordingly, a novel concept (the concept is under national trade-mark protection: [Hujber et al. 2019]) is introduced for wireless traffic signalization with a clear distributed control system architecture where the central controller unit is eliminated (see Fig. 4.18 (a)), i.e. the local controllers of the signal heads shown in Fig. 4.18 correspond to the subsystems given in Fig. 4.17. As an illustrative example, Fig. 4.18 (b) represents a case of a simple T-junction containing three signal heads with three corresponding signal phases. The figure presents the basic functioning of the distributed concept. Each signal head control unit knows the whole traffic signal program and only uses its own phase. Furthermore, every unit is able to check the proper functioning of the others (explained in detail in the sequel). The architecture of the distributed traffic light is already presented in Fig. 4.18 with power supply provided in the traffic signal pole. Traditionally, the electric power is typically supplied by the public electricity network for traffic lights. However, as a new energy efficient approach solar power system can also be used. An innovative concept of intelligent signal heads with wireless distributed traffic control has been introduced first by Tamaskovics et al. [2016], where the energy consumption was served with solar cells for each signal head. The intelligence means that the signal head is not only used to show the specific signal but it also has an own logic that serves control and communication tasks. The solution of Tamaskovics et al. [2016] is further developed in the sequel by directly providing the main safety algorithms to ensure safe functioning even with wireless technology. The concept is shown by Fig. 4.19.

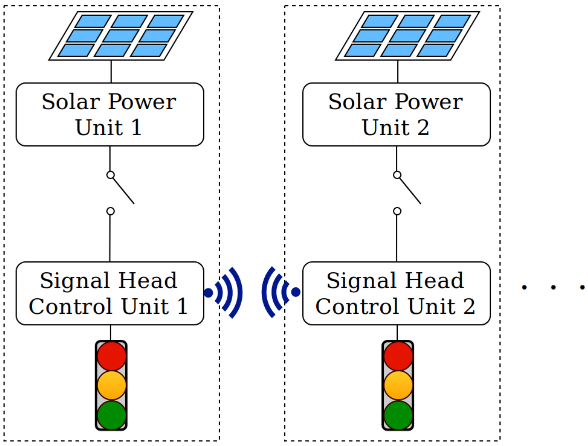


Figure 4.19. Flowchart of the redundant functioning for distributed traffic signal control (the picture considers only two signal head units for simplicity)

In order to make a consistent design of the distributed wireless traffic light and be technically correct, all safety critical aspects must be properly addressed with respect to the technical standards, i.e. the system must be able to ensure all safety functions given in Appendix G. As the communication is wireless, the main guarantee for the fulfillment of all requirements is the safe and reliable communication among the signal head units (see Figs. 4.18-4.19) and the fail-safe event handling in case of communication loss. Therefore, the proposed control concept is capable of handling the most critical situation, i.e. when any of the signal head units crashes. A spectacular example for such failure is the situation when the signal head unit “freezes” due software error, and thus the signal head cannot produce any light (or even the signal freezes as well). In this case all units of the system must switch off immediately, including the “frozen” unit. As a safe solution for this requirement a redundant control unit is applied in the system, i.e. the solar power unit is not only responsible for power control but also constantly checks the error-free functioning of the signal head unit. If a critical failure occurs, this redundant unit can interrupt the power supply between the battery and the signal head via an emergency relay. At the same time, the remaining signal head units switch off automatically due to the lack of communication from the crashed unit. The algorithm of this redundant safety process is summarized by Fig. 4.20 Note that the figure only represents the case of two parallel signal heads for simplicity. In the case of more units, every unit is involved in the communication respectively. The communication depicted in Fig. 4.20 is quasi continuous, i.e. the exchange of messages among the signal head units as well as that of between the solar power unit and its related signal head unit must be repeated with high frequency. Of course, the proposed system can be built without solar power system as well, e.g. by connecting to the public electricity network. In this case, the safety function of the solar power unit can be substituted with a similar control unit used for the cabled power supply.

4.3.2 The Base of Safe Distributed Operation

According to the standard EN 12675:2018 (Traffic signal controllers. Functional safety requirements. [CEN 2018a]) any signal state endangering the road traffic must be prevented during operation, i.e. a safeguarding facility shall lead to a safe state of operation as defined in [CEN 2018a]. Beyond the typical hardware/software errors in the traffic light system, the distributed architecture might also effect additional hazard for safe operation. Therefore, this necessitates a different safety concept compared to that of the traditional road traffic controller.

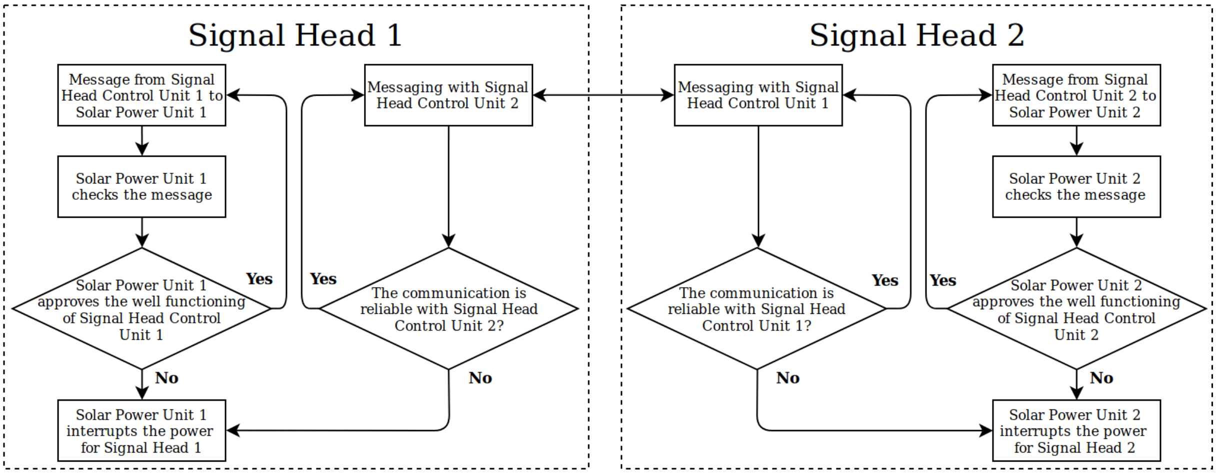


Figure 4.20. Flowchart of the redundant functioning for distributed traffic signal control (the picture considers only two signal head units for simplicity)

The distributed logic is ensured in a way that each signal head unit is identical considering the hardware as well as the software. It also means that the units know the whole signal program and this makes it possible that no central unit is needed to dictate the next signal states. The sole master function dedicated to one of the units is the check and control of time synchronization together with program change. If any of the units is delayed or is in hurry compared to that master time, the master shall ensure synchronization. The program changes (e.g. when dynamic signal programs are used) are also controlled by that master unit, i.e. after the confirmation of all other units for program change (and the time of change), the master starts the program change process. The master unit has no any other master function, i.e. every signal head control unit has the same privilege. This also means that any signal head control unit can start a fail-safe process in the case of failure. In order to make the distributed signal operation safe as possible proper checks have to be performed prior to the physical signal visualization on the light sources. Accordingly, every second (as the smallest discrete time interval in traffic signalization is typically 1 second) just before the light switching each signal head control unit verifies and confirms if the following points (according to the requirements listed in the Appendix G) are valid regarding the previous signal states:

1. no conflicting green signals;
2. no failure in safety timings (intergreen time or minimum green time);
3. no display of unintended signal and no failure in correct signal timing;
4. no failure in displaying of correct signal sequence.

On the one hand, the signal head units must go through the above checklist concerning itself (for which no communication is needed with the other units). On the other hand, every unit has to check the error-free functioning of all other signal heads in parallel based on the wireless communication. Obviously, the frequency of the communication is critical in this checking process. Beside the technical capability of the radio unit, one has to consider the standard EN 50556:2018 (Road traffic signal systems [CEN 2018b]) which defines 7 different classes for handling dangerous failures: from 100 ms up to 850 ms intervals. The time intervals defined by the standard mean the maximum times from the dangerous signal is present until the state has been removed. Accordingly, the safeguarding operation shall become active within 850 ms at most. This value has to be prudently considered when setting the frequency of the communication for the distributed traffic signal control system.

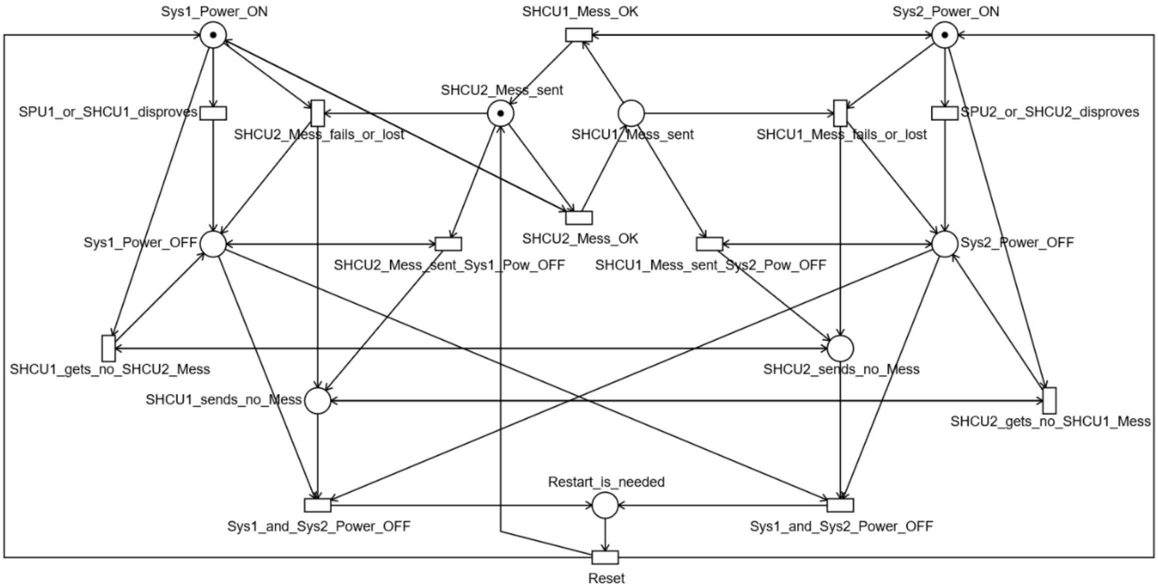
4.3.3 Safety Analysis of the Fail-Safe Distributed Traffic Controller Using Petri Net

Petri Net (PN) modeling is a powerful mathematical technique for the description of discrete event dynamic systems [Peterson 1981]. Moreover, PN can be used for the analysis of safety-critical systems. As a justification for practical applicability of the proposed distributed traffic control system, the redundant operation (Fig. 4.20) was modeled by Petri Net. For this reason PetriDotNet, a PN editor and analysis tool was used [Peterson 1981]. Fig. 4.21 shows the Petri Net model of the redundant functioning for distributed traffic signal control.

The model only considers two signal head units for the sake of clarity. Nevertheless, the same operation can be extended for further signal head units due to the identical safety protocol of the units, i.e. any of the units can lead the whole system to a fail-safe state. The operation modeled by Petri Net in Fig. 4.21 assumes periodic processes inside, i.e.

- SPU1 / SPU2 periodically checks the error-free functioning of SHCU1 / SHCU2,
- SHCU1 / SHCU2 periodically sends messages towards SHCU2 / SHCU1,
- SHCU1 / SHCU2 periodically receives and checks the messages from SHCU2 / SHCU1.

Based on the Transition-Invariants (T-Invariants) of the modeled Petri Net, one can justify that the whole system goes to fail-safe state whenever a critical problem occurs in any of the subsystems, i.e. any of the signal head control units fails or communication is lost among the units of the system. The calculated T-Invariants simply showed that all firing series of the Petri Net induced by any error result in the "Power OFF" state of both subsystem, i.e. SPU1, SPU2, SHCU1, and SHCU2 are switched off. On the other hand, as the modeled Petri Net is deadlock-free, it is confirmed that the proposed system might work infinitely if error-free operation is guaranteed. Finally, the Petri Net based analysis also showed that the system is bounded (1-bounded safe net) which means that the number of tokens is limited in the state space. Thus, the state space is bounded as well.



Nomenclature for the figure:
- SPU1 / SPU2: Solar Power Unit 1 / 2
- SHCU1 / SHCU2: Signal Head Control Unit 1 / 2
- Sys1_Power_ON / Sys2_Power_ON: the electric power of system 1 (SHCU1 and SPU1) / system 2 (SHCU2 and SPU2) is on
- Sys1_Power_OFF / Sys2_Power_OFF: the electric power of system 1 (SHCU1 and SPU1) / system 2 (SHCU2 and SPU2) is off
- SHCU1_Mess_sent / SHCU2_Mess_sent: the message of SHCU1 / SHCU2 is sent
- SHCU1_Mess_OK / SHCU2_Mess_OK: the content of the message from SHCU1 / SHCU2 is confirmed by SHCU2 / SHCU1
- SHCU1_Mess_fails_or_lost / SHCU2_Mess_fails_or_lost: the content of the message from SHCU1 / SHCU2 fails or the message is not arrived
- SPU1_or_SHCU1_disproves / SPU2_or_SHCU2_disproves: SPU1 or SHCU1 / SPU2 or SHCU2 disproves error-free functioning of SHCU1 or SHCU2
- SHCU1_Mess_sent_but_Sys2_Pow_OFF: the message of SHCU1 is sent, but the electric power of system 1 (SHCU1 and SPU1) is off
- SHCU2_Mess_sent_but_Sys1_Pow_OFF: the message of SHCU2 is sent, but the electric power of system 2 (SHCU2 and SPU2) is off
- SHCU1_sends_no_Mess / SHCU2_sends_no_Mess: SHCU1 / SHCU2 is not able to send message to SHCU2 /SHCU1
- SHCU1_gets_no_SHCU2_Mess / SHCU2_gets_no_SHCU1_Mess: SHCU2 / SHCU1 gets no message from SHCU1 / SHCU2
- Sys1_and_Sys2_Power_OFF: the electric power of both systems (SHCU1 and SPU1 as well as SHCU2 and SPU2) / is off
- Reset: reset operation is necessary to restart both systems (SHCU1 and SPU1 as well as SHCU2 and SPU2)

Figure 4.21. The Petri Net of the redundant functioning for distributed traffic signal control (the model considers the operation of two signal head units)

4.4 Contributions

Scientific contributions have been achieved in the field of traffic control. Traffic-responsive control algorithms have been designed for private cars and public buses. Additionally, as a practical result in this field, a novel concept for road traffic controller was developed realizing a distributed control concept using wireless communication.

Thesis 3

I have developed a dynamic road toll based control technique for the traffic management problem of alternative routes using a nonlinear model predictive control via the utility function of individual travelers. A methodology for dynamic, traffic-responsive route planning of automated public buses has been elaborated to improve the reliability of scheduling. I have developed an operational algorithm for wireless and distributed road traffic controller capable of coordinated control of intelligent units at individual signal heads without a central control unit. The practical, fail-safe operation has been demonstrated by formal method.

Thesis 3.1

Dynamic pricing based control scheme has been elaborated for the problem of alternative routes. The effect of utility-based traffic control was demonstrated on a closed-loop test system showing its effectivness on congestion reduction. Moreover, the time-delay effect in response to vehicle concentration has been investigated. The simulation results showed that time delay causes oscillation on the route alternatives in case of fixed and dynamic control. However, the applied nonlinear model predictive control method is insensitive to delay effect, and thus can be used as an efficient regulator type for utility-based traffic management applying the following utility function:

$$\mu_i(k) = -\alpha T_i(k-h) - \beta c_i(k) L_i, \tag{4.34}$$

where L_i is the length of the i^{th} route $T_i(k-h)$ is the average travel time of route i (where h reflects the driver's slightly outdated perception), and c_i is the unit toll price.

Thesis 3.2

A dynamic, traffic-responsive routing methodology has been elaborated for automated public transport buses allowing them to travel on not predefined paths between stops, minimizing the generalized cost of the route. The routes between these fixed points have been modeled as a continuously updated weighted directed graph. The weights represent relevant parameters (e.g. length, fee) of links, collected from surrounding sensors and monitoring systems of the network. Route optimization has been done by a k-shortest path search algorithm depending on the timetable: if the vehicle will reach the next stop on time, the alternative with the lowest generalized cost is chosen; else the fastest route is followed.

Thesis 3.3

Beside the development of control algorithms, I also worked out the background for a practical application of control. A novel concept for wireless and distributed traffic control system has been introduced, which enables that siganl heads of intersections are controlled without a central control unit. The system realizes a distributed control method based on the signal head control units. The solution can reduce installation and maintenance costs, especially if electric power supply is ensured by solar panels. As a justification for practical applicability and corresponding to the relevant standards, the fail-safe and redundant operation of the proposed distributed traffic control system has been modeled by a formal method (Petri Net).

Related journal and conference papers

- I. Varga, B. Kulcsár, T. Luspay, and T. Tettamanti. Korszerű szabályozások a közúti forgalomirányításban. *A Jövő Járműve*, 1-2:34–36, 2008
- T. Tettamanti and I. Varga. Traffic control designing using model predictive control in a high congestion traffic area. *Periodica Polytechnica ser. Transp. Eng.*, 37(1-2):3–8, 2009c. doi: 10.3311/pp.tr.2009-1-2.01 **(Q2)**;
- T. Tettamanti and I. Varga. Elosztott közúti forgalomirányító rendszer. *Városi Közlekedés*, XLIX(6):338–341, 2009a;
- T. Tettamanti and I. Varga. Városi forgalomirányító rendszer prediktív szabályozással. *Városi Közlekedés*, XLIX(3):131–135, 2009b;
- T. Tettamanti and I. Varga. Distributed traffic control system based on model predictive control. *Periodica Polytechnica ser. Civil Eng.*, 54(1):3–9, 2010. doi: 10.3311/pp.ci.2010-1.01 **(Q3)**;
- T. Tettamanti and I. Varga. Robusztus városi forgalomirányítás. *Városi Közlekedés*, LI (1-2):80–84, 2011;
- A. Csikós, T. Tettamanti, and I. Varga. Nonlinear gating control for urban road traffic network using the network fundamental diagram. *Journal of Advanced Transportation*, 49 (5):597–615, 2015a. ISSN 2042-3195. doi: 10.1002/atr.1291 **(Q1)**;
- T. Tettamanti, A. Csikós, and I. Varga. Macroscopic modeling and control of emission in urban road traffic networks. *Transport*, 30(2):152, 161 2015a. doi: 10.3846/16484142.2015.1046137 **(Q2)**;
- G. Tamaskovics, T. Tettamanti, and I. Varga. Az intelligens jelzőfej koncepciója: vezeték nélküli, elosztott rendszerű jelzőlámpás forgalomirányítás. *Közlekedéstud. Szemle*, 6(LXVI): 45–54, 2016;
- T. Tettamanti, A. Mohammadi, H. Asadi, and I. Varga. A two-level urban traffic control for autonomous vehicles to improve network-wide performance. *Transportation Research Procedia*, 27:913 – 920, 2017b. doi: 10.1016/j.trpro.2017.12.160. 20th EURO Working Group on Transportation Meeting, EWGT 2017, 4-6 September 2017, Budapest, Hungary;
- B. Varga, T. Tettamanti, and B. Kulcsár. Optimally combined headway and timetable reliable public transport system. *Transportation Research Part C: Emerging Technologies*, 92:1 – 26, 2018a. doi: 10.1016/j.trc.2018.04.016 **(D1)**;
- T. Tettamanti. Wireless traffic signal controller with distributed control system architecture. *Periodica Polytechnica ser. Civil Engineering*, 63(3):918–925, 2019. doi: 10.3311/PPci.13974 **(Q3)**;
- M.T. Horváth, T. Tettamanti, and I. Varga. (Smart CPS) Multiobjective dynamic routing with predefined stops for automated vehicles. *International Journal of Computer Integrated Manufacturing*, 32(4-5):396–405, 2019c. doi: 10.1080/0951192X.2018.1535197 **(Q1)**;

- T. Tettamanti, Á. Török, and I. Varga. Dynamic road pricing for optimal traffic flow management by using non-linear model predictive control. *IET Intelligent Transport Systems*, 13:1139–1147(8), 2019b. doi: 10.1049/iet-its.2018.5362 **(Q2)**;
- B. Varga, T. Tettamanti, and B. Kulcsár. Energy-aware predictive control for electrified bus networks. *Applied Energy*, 252:113477, 2019. doi: 10.1016/j.apenergy.2019.113477 **(D1)**;
- Q. Lu, T. Tettamanti, and D. Hörcher. Implications of user and system optimum based traffic control considering autonomous fleets. In *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, pages 1–5, 2019. doi: 10.1109/ICCVE45908.2019.8965210;
- T. Tettamanti. Vezeték nélküli, napelemes jelzőlámpa. *Élet és Tudomány*, LXXV(39), 2020;
- B. Varga, T. Tettamanti, B. Kulcsár, and X. Qu. Public transport trajectory planning with probabilistic guarantees. *Transportation Research Part B: Methodological*, 139:81 – 101, 2020c. doi: 10.1016/j.trb.2020.06.005 **(D1)**
- B. Kővári, T. Tettamanti, and T. Bécsi. Deep reinforcement learning based approach for traffic signal control. *Transportation Research Procedia*, 2021. 24th EURO Working Group on Transportation Meeting, EWGT 2021, 8-10 September 2021, Aveiro, Portugal;
- Q. Lu and T. Tettamanti. Traffic control scheme for social optimum traffic assignment with dynamic route pricing for automated vehicles. *Periodica Polytechnica Transportation Engineering*, 49(3):301–307, 2021a. doi: 10.3311/PPtr.18608 **(Q3)**.

Related book chapter and book

- T. Tettamanti, I. Varga, and T. Péni. MPC in urban traffic management. In *Model predictive control*, pages 251–268. Open Access Books, IntechOpen, 2010. doi: 10.5772/9922;
- T. Luspay, T. Tettamanti, and I. Varga. *Forgalomirányítás, Közúti járműforgalom modellezése és irányítása.* Typotex Kiadó, 2011. ISBN 978-963-279-665-9.

Related patent

- R. Hujber, T. Tettamanti, and I. Varga. Intelligent road traffic light system with distributed control, 2019. Registration number: 5034, NSZO: G08G 1/095 , Case Number: U1800160/10

Chapter 5

Simulation Based Testing for Automated Road Traffic

The continuous development of road traffic control systems is indispensable due to the increasing demands. The control measures can be implemented in three different areas: on freeways, in urban areas or in integrated networks (when the control of freeway and urban roads are managed together). Traffic control strategies may consist of providing basic information (e.g. congestion warning) or use complex logical algorithms realizing adaptive operations. The actuators are also very heterogeneous, e.g. variable message sign, ramp metering, traffic light, intelligent on-board system, route guidance, etc. In simple static systems (e.g. fixed time ramp metering), the control strategy is realized based on historical measurement data. At the same time, in case of intelligent traffic control the advanced use of informatics is evident. As consequence, when traffic engineers design intelligent traffic control the help of specific computer tools are indispensable. The use of traffic simulators and mathematical optimization software is more and more expected during the development process and the validation phase as well.

5.1 Microscopic Traffic Simulation Practice Considering Automated Vehicles

Traffic simulation is the mathematical modeling of traffic dynamics through the application of computer software to support planning, operation and development of transportation systems. Simulation models can be classified into macroscopic, mesoscopic, and microscopic models according to the level of detail [de Dios Ortúzar and Willumsen 2001]. Macroscopic models have applications when detailed information about a single vehicle's behavior is not required. It only provides a general evaluation of traffic flows in a network. These models are often used for regional transportation planning. Microscopic models describe each vehicle's behavior and interactions in the traffic system, making more detailed modeling for each movement of the vehicle. For this reason, microscopic models can be applied with a much higher level of detail. The microscopic model has the following advantages: by tracking a single vehicle on the road, it can not only reflect the interaction between vehicles but also predict traffic performance indicators such as vehicle travel time, delay and emission while avoiding the impact on actual road traffic; through the microscopic simulation model, the impact of a specific parameter on traffic can be reflected; through the animation interface of the simulator, one can intuitively visualize the changes in road traffic, and provide a good platform for understanding the traffic operation status under different traffic demands. It has superiority that traditional mathematical models cannot match in describing and evaluating the traffic flow of the road network. Microscopic models are becoming an increasingly important and popular tool in the transportation field. It has been used for a wide range of applications in network design, analysis of transportation problems, the evaluation of Intelligent Transportation System (ITS), and traffic management strategies formulation. Even though there is a large number of microscopic models, unfortunately, none of them can be considered as an ideal or, at least, a universal one. It is mainly because every model has different parameters to describe a different traffic situation and vehicle behavior. The early research focused on maintaining the existing distance with the vehicle in front [Ni et al. 2004]. Car-following models [Brackstone and McDonald 1999] are the most popular approach to model the interaction between vehicles. Car-following theories examine the longitudinal movement of each vehicle and are extended by lane-changing maneuver models. With the development of computer science, the extension extends to the use of cell automation and multi-agent systems. Continuing these efforts, the expansion was being conducted to get a more realistic behavior model by adding a stochastic method for making decisions based on a given environment of the road. Furthermore, the most adopted methodology is to apply the Monte Carlo procedure to generate random values to show the driving behavior in traffic conditions. The basic steps involved in the development are the same irrespective of the type of model described above [Brackstone and McDonald 1999].

5.1.1 Microscopic Traffic Simulation Considering Autonomous Driving

Before the autonomous vehicles are officially launched into the market, it must be fully tested in the different traffic environments, thoroughly verify the autonomous driving function, and achieve collaboration with roads, traffic facilities, and other transportation participants. Validation is a necessary step in the development and application of autonomous vehicles. The research and development of autonomous driving systems have been developing rapidly, but the industry and the governments have not yet reached a clear consensus on how to conduct safety testing and reliable proving in the real-world. Because dangerous traffic scenes are difficult to exhaust, there are technical bottlenecks in scene-based actual vehicle testing methods. According to statistics from the Federal Highway Administration (FHWA), a driver needs to travel 850,000 kilometers on average to experience a police report accident and close to 150 million kilometers to experience a fatal accident. The industry generally believes that each autonomous driving system requires 16 billion kilometers of driving data to optimize the system. It would take about 50 years for a fleet of 1,000 autonomous driving test vehicles to complete a sufficient mileage test. Therefore, the general consensus in the industry is that virtual testing and evaluation of autonomous driving systems based on simulation technology is required. Microscopic traffic simulation is widely used both by the traffic engineering industry and the academia research community. In the traffic engineering industry, a microscopic simulation is a powerful tool in transport development studies, feasibility studies as well as for concrete development/construction of infrastructures. In research and development, it is used to study traffic management and traffic estimation methodologies. Furthermore, in our days traffic simulation is also applied in the development of autonomous vehicle systems beside vehicle dynamics simulators.

Traffic simulation is a mature field; several microscopic road traffic simulators are available. Each simulator has its own advantages and aims mimic realistic traffic based on car-following models. Typical microscopic traffic simulators applied both in academic and industrial fields are for instance Paramics, CORSIM, PTV VISSIM, or SUMO. It is significant to specify the microscopic modeling issues of autonomous vehicles because automated functions truly affect simulation results. The microscopic simulation software development is inevitable. The traffic impacts of autonomous vehicles should be examined before its implementation. The safety, mobility, and environmental sustainability of the AVs shall be checked. With the emergence of AVs, new vehicle models are needed to simulate them, which means practically new vehicle classes on the simulation software level. Besides, the connected autonomous vehicle (CAV) has a promising prospect. To simulate CAV, Vehicle to Everything (V2X) communications technology also shall be considered. Traffic control features are also expanding. Specific autonomous driver models of different manufacturers should also be implemented in software. There are currently two ways to develop a microscopic traffic simulation model in software. The first way is that the software developer tries to refine the model and features as much as possible. Another is that the user of the software "developes". For example, they can apply their own vehicle tracking model in traffic simulation applications (e.g. VISSIM API, SUMO TRACI interfaces), or they fine-tune the default driver model to automation properly. The classical process of traffic control development concludes data collecting, model development, and finally model calibration. First, traffic engineers make manual traffic counts or get automatically measured traffic data (i.e. detector data). Then, the microscopic simulation model can be created. The last step is to calibrate the model with the data from the reference cross-sections of the test field. When the reference traffic volumes are fixed, one can tune the model by modifying the simulation parameters, such as the car-following model, turning rates or dynamic traffic inflow. Validation ensures that the software represents reality at a satisfying confidence level by comparing simulation results with real-life observation. Based on reference cross-sections or full network parameters, the GEH-index based validation is applicable. The GEH index is commonly used in traffic engineering, traffic forecasting, and traffic modeling to compare two sets of traffic volumes. The formula used to calculate GEH-index is:

$$\mathrm{GEH} = \sqrt{\frac{2(M-C)^2}{M+C}}, \tag{5.1}$$

where M is the simulated data, and C is the real-world collected data. For traffic modeling, a GEH of less than 5.0 is considered a good match between the modeled and observed hourly volumes. 85% of the volumes in a traffic model should have a GEH less than 5.0. And the GEH value generally should be smaller than 10.0. If the GEH is over 10.0, it is likely that the traffic demand model or the measured data are biased. In practice, a lower GEH index is applicable for smaller traffic volumes. Validation also varies from autonomous to highly automated vehicle simulation. The GEH index is classically cross-referenced based on traffic volume. With the appearance of AVs, the validation methodology should be extended. Appropriate statistical data is required for different automated vehicles. The future outlook for the use of microscopic traffic simulation is also required.

There are many speculations about the impact of autonomous vehicles on the transportation system. Some researchers pointed out that AVs would reduce road congestion, greenhouse gas emissions, economic loss and revolutionize the transportation system. Based on the above background, to have a deeper understanding of the impact of the emerging autonomous driving technology on the microscopic traffic simulation models, the traffic performance in the micro-simulation technology was studied via the gap acceptance model parameter. With the help of PTV VISSIM (as a high-fidelity microscopic simulation software) a detailed sensitivity analysis was carried out to quantify the traffic performance under different model parameter settings. The goal of the conducted simulations is to show that the common practice of traffic simulation requires a thorough revision and modification when it is applied with the presence of autonomous vehicles.

5.1.2 Simulation-Based Sensitivity Analysis of Model Parameters

The traffic flow model in PTV VISSIM is based on the work of Wiedemann, including a psycho-physiological car-following model [Higgs et al. 2011]. To simplify the experimental model the following assumptions were made.

- The traffic flow distribution of the road network remains unchanged. Although AVs can obtain real-time vehicle status information on the road network and optimize vehicle flow distribution on the road network, with the continuous development of intelligent transportation systems, travel information services tend to be intelligent and dynamic. Travelers can also obtain road network outbound information and optimize travel routes based on a series of devices such as onboard networking equipment, smart navigators, and smartphones. Therefore, the impact of AVs is ignored in optimizing travel routes. That is, the distribution of traffic flow in the simulated test network is unchanged.

- To simplify the simulation model, the impact of other vehicle types is ignored in the road network since this study focuses on the urban network, and normally mostly passenger cars exist on it.
- Since the current autonomous technology is still developing, and the relevant supporting data is insufficient, the related parameter values in this research were set based on the current theory of autonomous driving technology.

Traffic flow is defined as the interactions between travelers (passenger cars, pedestrians, heavy-duty vehicles, etc.) and road infrastructure systems (traffic lights, traffic signs, etc.). AVs have the following obvious advantages: smaller gap acceptance, shorter headway, no reaction time in front of the signal system, maintenance of a constant desired speed, and stable acceleration and deceleration. The main difference between the AVs and CAVs in the simulation study was the selected parameters of the car-following model. The gap acceptance was chosen as a main parameter to be changed due to the emerging autonomous driving technology. "Standstill Distance" represents the base value for the average desired distance between two stationary cars in Wiedemann 74 model in PTV VISSIM. The default value is 2.0 meters for passenger vehicles. In the simulations the following "Standstill Distance" values have been investigated: 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0.

5.1.3 Case Study

To analyze the changes in traffic performance, a typical signalized intersection in the city of Hefei (China) was modeled as a simulation scenario based on an open database (OpenITS). This network contains two arterial roads, Huangshan road and Kexue avenue. The test network model was created in VISSIM. The detailed layout of the intersection is shown in Fig. 5.1.

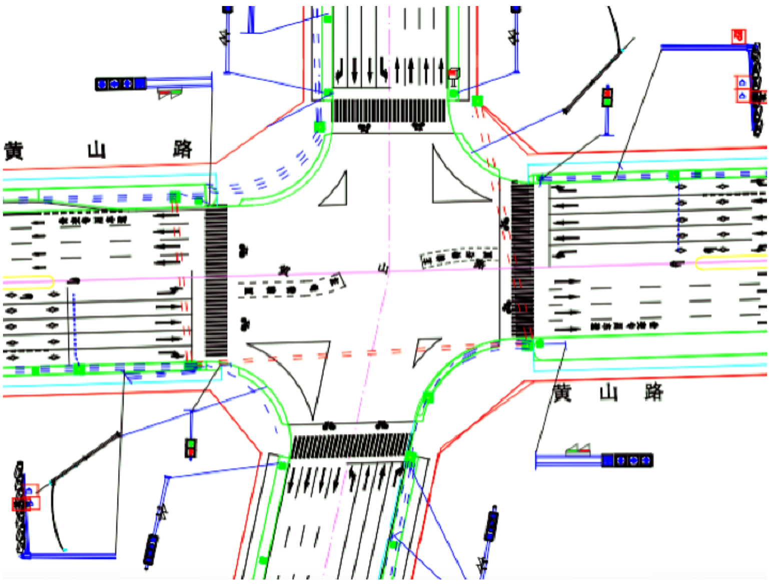


Figure 5.1. The layout of the intersection

To quantitatively analyze how the changes of the parameters due to the emerging autonomous technologies affect the urban road network, signalized intersection scenarios were simulated based on the different traffic demand conditions both in VISSIM as mentioned in the previous section. The simulation time was set to 3600 seconds. The sensitivity analysis of the different gap acceptance was carried out in the simulations. As commonly used indicators, mean speed and average travel time were selected as evaluation indexes of traffic efficiency. In the vehicle network performance evaluation results of VISSIM, the mean speed is defined as total travel distance divided by total travel time (km/h). In VISSIM, the average travel time can be calculated from total travel time divided by the total number of vehicles in the network. Fig. 5.2 shows the mean speed and the average travel time measurements under different gap acceptances in VISSIM.

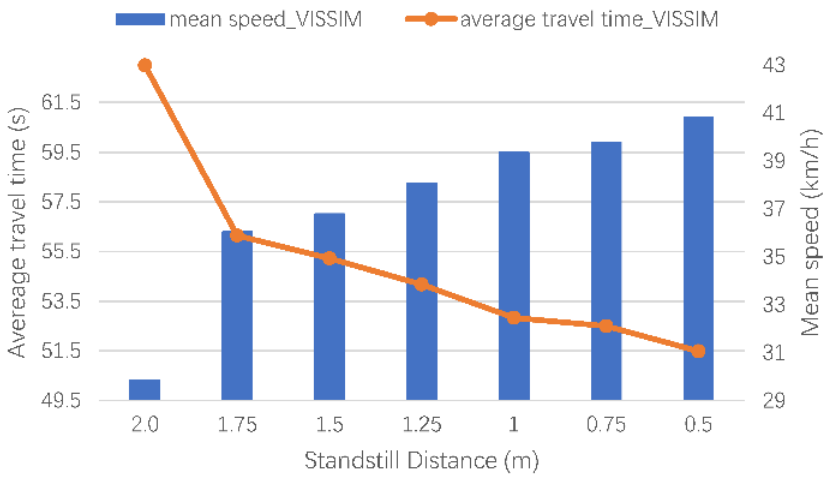


Figure 5.2. Variations of mean speed and average travel time

VISSIM shows a clear sensitivity to gap changes in the oversaturated traffic situation, especially when the gap changes from 2.0 meters to 1.75 meters. Both mean speed and average travel time show significant fluctuation. Based on the sensitivity analysis of the given test intersection, it has been demonstrated that the current simulation practice of traffic engineering needs change due to the emerging presence of highly automated cars and soon the advent of fully autonomous vehicles on public roads.

5.2 Online Calibration of Microscopic Road Traffic Simulator Using Genetic Algorithm

Traffic simulation refers to the interaction of models describing the characteristics and behavior of each vehicle unit in the transportation system via computer technology. Due to microscopic traffic simulation has benefits on the low simulation cost, simulation without any risk, and fast simulation run time, it has been intensively applied for designing and operation of traffic management systems [Csiszár et al. 2020]. To provide a credible microscopic traffic simulation, calibration must be performed to guarantee that the established model truly reflects the real-world traffic situation. This process is generally implemented by comparing the field measurements to the corresponding simulation output [Hale et al. 2015]. Numerous studies have yielded results in terms of providing credible methods such as trial-and-error methods, Genetic Algorithm (GA), Simultaneous Perturbation Stochastic Approximation (SPSA) [Chu et al. 2003; Ma et al. 2007]. Another part of the research focuses on the calibration process, a representative one is the 7-step calibration framework proposed by Hellinga [1988] which many subsequent studies were based on, e.g. Hourdakis et al. [2003]; Park and Schneeberger [2003]. Unlike the research mentioned above using historical static data during the calibration process, the method proposed in the sequel solely applies the previous simulation step output as initial data of the next step to achieve the online calibration of the microscopic road traffic simulator, i.e. in this way real-time traffic modeling can be realized based on online traffic sensor measurements.

5.2.1 Implementation of the Online Calibration

Microscopic traffic simulator calibration is performed by selecting one or more parameters and then repeatedly comparing the measured data with the simulation output until the preset error range is reached. This process can be seen as a complex optimization problem with huge search space. There is no specific functional relationship between the fitness function and the parameter to be calibrated in such kind of optimization problem, therefore, computing based artificial intelligence, genetic algorithm was applied. The main advantage of GA is that it is able to search solutions under multiple criteria, which increases the probability of finding a global solution rather than a local optimal solution [Kyu-Ok and Rilett 2001].

Before using GA, the parameters to be calibrated need to be coded to obtain the individual used in the algorithm, and the problem to be solved is transformed into the fitness function. Then GA will find the individual with the smallest value of the fitness function, and then decode, get the solution to the original problem. When calculating GA, starting from a randomly generated population which is a group of individuals that corresponding to a feasible solution to the original optimization, GA will compare the fitness value of each individual in each generation, and select individuals with the smallest fitness value. Then, with the help of genetic operators, individuals are selected to crossover and mutate to produce new populations. According to the survival of the fittest law, individuals who are more adapted to the environment will be evolved. In past researches, various fitness functions were used to minimize the discrepancies between field measurement and simulation output, such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Global Relative Error (GRE) [Ma and Abdulhai 2002], and GEH statistic (GEH formula gets its name from Geoffrey E. Havers) [Paz et al. 2015]. In this research, the L_∞-norm of relative error is used to form the fitness function in the calibration process:

$$\min_{Q(k)} \sum_{i=1}^{n} \left\| \frac{\bar{F}_i^{Measured}(k) - \bar{F}_i^{Simulated}(Q(k))}{\bar{F}_i^{Measured}(k)} \right\|_\infty \tag{5.2}$$

where:

- $\bar{F}_i^{Measured}(k)$ is the average traffic volume of edge i from the ground truth at step k;
- $\bar{F}_i^{Simulated}(Q(k))$ represents the average traffic volume of edge i produced by the calibrated simulator in the previous simulation time window k;
- $Q(k)$ represents the applied calibration parameter.

During the microscopic simulation, driver behavior, traffic flow characteristics are described by numerous independent microscopic parameters, and the different setting of these parameters affect the simulation output a lot. Using the default model parameter settings will cause the simulated output such as the vehicle number, lane occupancy, traffic density have large errors compared with the measured values in the field. In order to eliminate the influence of these errors on the calibration process, simulated measurements based on real traffic data are used as the ground truth which represents the field measures. In this study, "Edge-Based" measurements of SUMO, which simulate induction loop detector measures are used as ground truth. The simulation output in SUMO for the same edge can be aggregated to generate the performance measures, that is "Average traffic volume (veh/h)".

The microscopic traffic simulator online calibration based on GA follows the subsequent steps:

- Define the research objectives and the overall framework;
- Data collection and preprocessing;
- Create microscopic simulation model;
- Model error review and correction;
- Online calibration using GA;
- Results analysis;
- Final report and technical documentation.

Microscopic traffic simulation usually requires several inputs: road static geometric data (road length, the number of lanes, etc.); traffic control data (signal control schema, right of way rules, prohibiting left turn, etc.); dynamic model demand data (flow, turning rate, etc.); calibration data (capacity, travel time, average speed, etc.).

Microscopic traffic simulation modeling is usually divided into several parts, first establishing a static road network and then setting traffic control rules. On this basis, fuse the traffic demand and other network operation data. After the model was established, error checking and the correction needs to be implemented to make sure that the established model truly reflects the field traffic situation.

The main steps of the model calibration include selecting a few finite key parameters and importing them into the model for a large number of simulations. By comparing the simulated data with the measured data, the optimal values of the selected parameter are finally filtered out.

During the analysis, model calibration work is obtained through a large number of iterations. The representative parameter values are summarized and analyzed, so that the feedback of model calibration work on the practicality of model parameters itself can be completed, which has positive practical significance. Fig. 5.3 shows the whole calibration framework.

The whole calibration process was implemented through the Distributed Evolutionary Algorithms in Python (DEAP) [Rainville et al. 2012]. The core of GA is three operators, namely mutation, crossover and selection, each with its own characteristics, which can be used to generate new individuals in different ways. The general rule of the mutation operator is that it involves only one individual, where a part of its genes will remain unchanged whether the rest genes accept the change (mutation). Two kinds of mutation were proposed in this research (Gaussian mutation and Uniform mutation). The Gaussian mutation generates a random number obeying the normal distribution with a mean μ and a variance σ to replace the real number in the original gene. The Uniform mutation replaces the original gene value in the individual with a random probability that matches a uniformly distributed random number within a certain range. The rate of mutation determines the magnitude of the individual change that will result in the mutated individual that constitutes the next generation of individuals. To avoid random search, the small mutation rate was suggested, while the mutation rate that is too small may hardly change individuals, leading to very slow convergence.

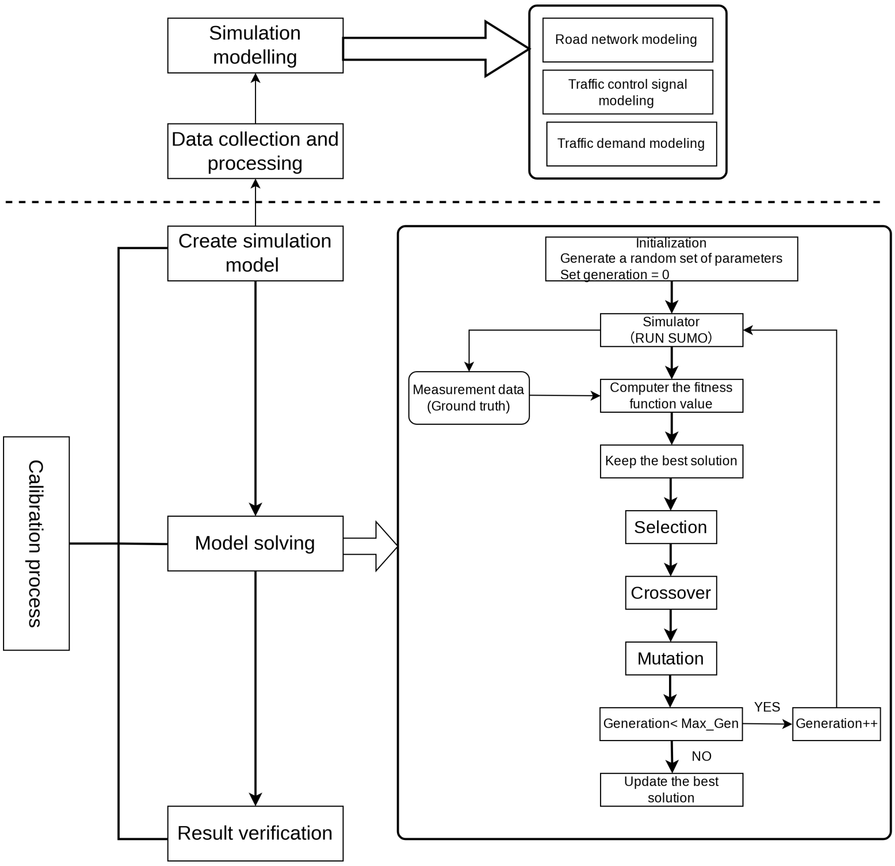


Figure 5.3. Online calibration framework

The general rule of crossover operators is that they will combine the genes of two or more individuals. The crossover operation can be performed in several ways (two-point crossover, binary crossover, uniform crossover), here executes a blend crossover. This operation is a linear combination of two parents x_i and y_i. The operation is initialized by choosing a uniform random real number from the interval $[min(x_i, y_i) - \alpha|(y_i - x_i)|, max(x_i, y_i) + \alpha|(y_i - x_i)|]$, where α is the positive real parameter, $\alpha = 0.1$ is selected in this research.

In the evolution process, individuals that are more adaptable to the environment will have more opportunities to inherit to the next generation, the selection operator used to imitate this process. There are several selection strategies: Roulette Wheel selection, Stochastic Universal Sampling, Tournament selection and Boltzmann selection. Due to the high efficiency and easy implementation of the tournament selection algorithm, it is the most popular selection strategy in genetic algorithms. The strategy is also very intuitive, extract n individuals from the entire population and let them compete, then extract the best individuals among them. Since the first selection is made without considering the fitness values, weak individuals can survive until the next generation, which is good for genetic variability, high values of tournament size can compromise variability, where low value can approximate to the random selection. The number of individuals participating in the tournament becomes the tournament size. Table 5.1 shows the applied genetic operators.

5.2.2 Case Study

To evaluate the proposed calibration algorithm, a network based on an open database (OpenITS) was generated in SUMO traffic simulator. Related traffic data was obtained considering a one hour interval of the morning peak, from 7:00 to 8:00 on March 26, 2015. The field traffic flow

Table 5.1. The genetic operators

genetic operators	method	parameter
mutation	Gaussian mutation	μ=0, σ=5
	Uniform mutation	lower bound=1, upper bound= 200
crossover	blend crossover	$\alpha = 0.1$
selection	tournament selection	tournament size=3

data were aggregated into 300 seconds count. The reliability of microscopic simulation is based on the accuracy of input data, so the investigation of field traffic data should be detailed. In the research the open data of an intersection of Ningxi Road and Xingye Road was used, the main road in the Xiangzhou District of Zhuhai City (China) as an example.

5.2.3 Calibration Result Verification and Analysis

In multiple simulation steps, it was possible to converge the vicinity of the above solution within 30 generations. The change of the fitness function value can be observed in Fig. 5.4.

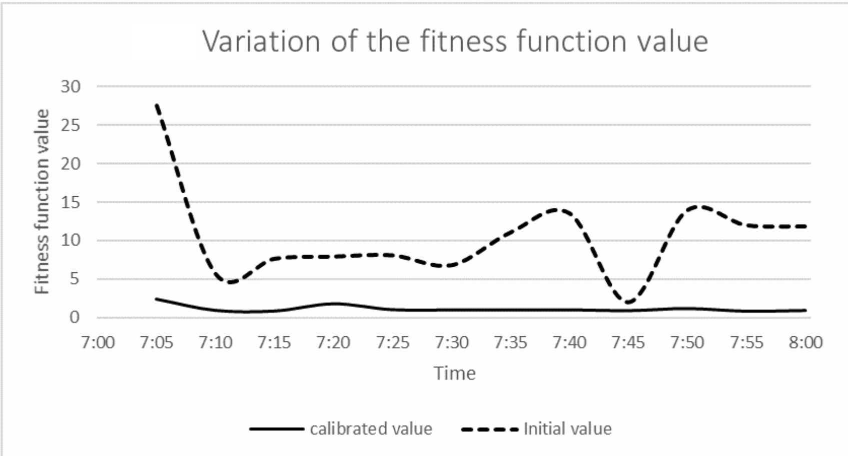


Figure 5.4. The variation of fitness function value

The average traffic volume, (i.e. flow) of the simulation time interval on each edge in the network was measured. Fig. 5.5 depicts the traffic flow from four directions. Edge based measurements of the simulation output based on the real traffic counts is taken as the default value (represented by "Ground truth"), and the result of simulation calibrated output (represented by "Calibration output") are compared. The Mean Absolute Percentage Error (MAPE) was used as validation criteria for the proposed calibration framework, the results are shown in Table 5.2.

Table 5.2. The calibration results

	North	East	South	West
Total traffic volume of ground truth (veh)	3154	9024	13756	10777
Total traffic volume of calibration output (veh)	3250	9972	13131	11888
MAPE (%)	3.03	10.51	4.55	10.31

The smallest difference of total traffic volume between ground truth and calibrated output occurred in the North direction of the intersection which was 3.03% and the mean absolute

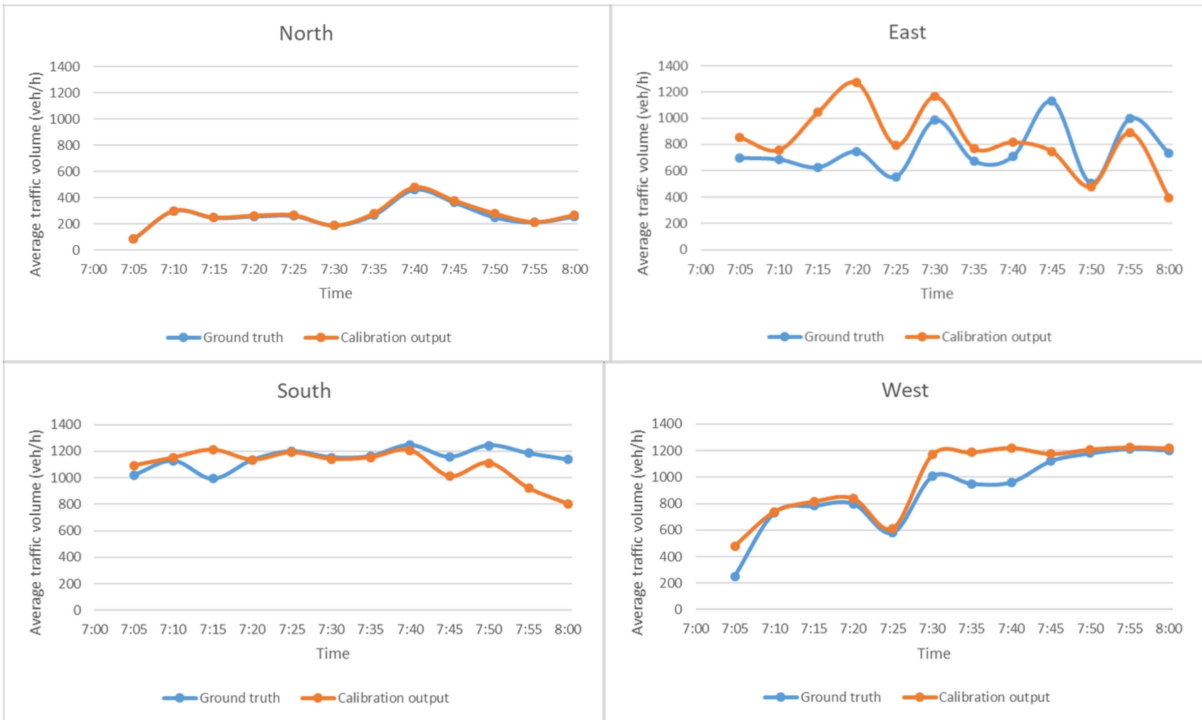


Figure 5.5. The comparison of the traffic volume

percentage error in each direction of the intersection was less than 11%. As Fig. 5.5 shows, the discrepancy in the East direction was a bit bigger than other directions, which probably caused by the non controlled edge on the network. Vehicles operating on the edges belonging to the East direction showed more stochastic behavior due to the existence of the right turn lane (which is not controlled by traffic signal).

5.3 Co-Simulation for CAV Testing with Traffic Simulation

Autonomous driving and related intelligent infrastructure developments open immense possibilities for scientific and technological advance. Beyond the liability issues [Bartolini et al. 2017], the main condition of practical deployment is, however, the proper and reliable functioning which is guaranteed by continuous testing and validation in engineering practice [Szalay et al. 2018].

An important possibility for AV development is the Vehicle-in-the-Loop (ViL) testing. There exist several traffic simulation software capable to mimic realistic dynamics. This opens the way to insert the self-driving test car into the virtually generated traffic system. The option of adding autonomous vehicle models to the traffic simulator already exists (e.g. PTV VISSIM). However, the usual approach today is limited to fully virtual testing, i.e. no real control of autonomous vehicle functions and behavior implemented (only inserting virtual autonomous vehicles into traffic simulators).

One of the greatest advantages of this ViL testing method is cost-effectiveness. Obviously, there is no need to build traffic systems, only the test vehicle must have the software and hardware to process the data of virtual traffic. It is also important to emphasize the safety of simulations, since during the experiments the physical safety of the participants or the proper functioning of a real transport system are not endangered.

The testing and validation structure for autonomous vehicles can be defined as a 5 layer pyramid depicted by Fig. 5.6 (source: Szalay [2016]). It is emphasized that the first 3 layers might efficiently use the opportunities of ViL testing, i.e. the phases of Simulation, Laboratory Testing as well as Proving Ground Testing will benefit from the advantages of microscopic

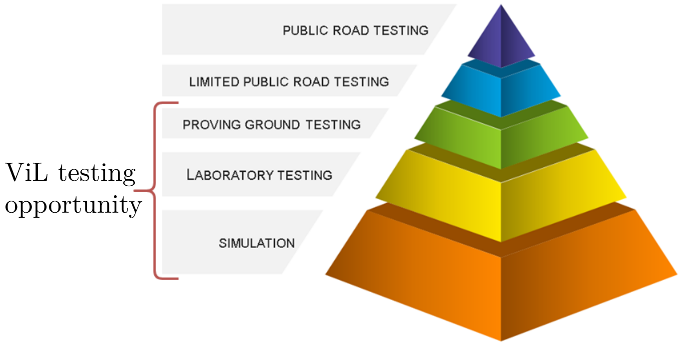


Figure 5.6. Autonomous vehicle testing and validation layers

traffic simulations. In the sequel, the problem of interfacing traffic simulation is introduced in the context of ViL testing of autonomous vehicle technologies.

5.3.1 Integrated VISSIM-MATLAB Environment

One of the proposed simulation environment is based on PTV VISSIM, a microscopic traffic simulation tool. The goal of the microscopic modeling approach is the accurate description of the traffic dynamics. Thus, the simulated traffic network may be analyzed in detail. The simulator uses the so-called psycho-physical driver behavior model developed originally by Wiedemann [1974]. PTV VISSIM is widely used for diverse problems by traffic engineers in practice as well as by researchers for developments related to road traffic.

PTV VISSIM offers a user friendly graphical interface (GUI) through of which one can design the geometry of any type of road networks and set up simulations in a simple way. However, for several problems the GUI is not satisfying. This is the case, for example, when the user aims to access and manipulate VISSIM objects during the simulation dynamically. For this end, an additional interface is offered based on the COM (Component Object Model) technology enabling interprocess communication between software Box [1998]. The VISSIM COM interface defines a hierarchical model in which the functions and parameters of the simulator originally provided by the GUI can be manipulated by programming. It can be programmed in any type of languages which is able to handle COM objects (e.g. C++, Visual Basic, Java, etc.). Through VISSIM COM the user is able to manipulate the attributes of most of the internal objects dynamically. The first step of a COM based simulation is to create the COM client. Then, one can realize the parts or even the whole process of the simulation.

As further possibility Dynamic Link Library (DLL) interface programming is also available for specific parts of the simulator. The APIs (Application Package Interfaces) for this option is provided by the VISSIM software manufacturer written in C++ language. This allows flexibility for the user as one may freely create own developments for the simulations. On the other hand, the simulation run is much faster by using self-developed DLLs for some specific parts compared to the the same simulation controlled by COM interface programming. VISSIM DLL interface programming is available for driver model, emission model, signal control and toll pricing.

VISSIM GUI provides several possibilities to choose for simulation of traffic controllers. In case of static control, the logic can be easily defined by using the Fixed time module. Fixed time signal controller is available for editing directly through the VISSIM GUI. In addition, one can design phase-based and stage-based signal plans. User may design actuated control with the VAP (Vehicle Actuated Programming) module allowing to create arbitrary logic. The module is programmable by the VAP language or flowchart editor (VISVAP) provided by VISSIM. The option of External signal controller may be chosen as well through of which user-defined signal control can be used. In this case, however, user has to apply the Signal Control API. Hence, by using the object oriented C++ language more complex logics can be resulted compared to VAP. Beside the previous options VISSIM offers further control modules which are commercial products adapting to the market demands (e.g. SCATS, SCOOT, SIEMENS VA, LISA+).

To create complex signal control logic one can use VISSIM COM interface exploiting the advantages of the current programming language. By using the attributes of the SignalGroup interface of VISSIM COM, one can realize arbitrary signal control. For example, attribute STATE is able to set any signal state (red, red-amber, green, etc.) on the signal heads at any time.

If one wishes to design logic through VISSIM COM interface Fixed time controller type should be chosen. In this case, Fixed time module can be used also for adaptive control as at the end of each control interval the states of all signal groups can be redefined based on the user-defined logic. Except the Fixed time module, in case of the other control types (VAP, External, etc.), it makes no sense to manipulate the signal control via VISSIM COM if they are designed and used in a standard way.

Another possibility to create own control logic is the programming of the VISSIM API which provides an advanced application. One may choose the External type controller which allows to connect external control to the simulator through the Signal Control API. By using this method, separate DLL file must be defined for each Signal Controller which is called at each controller time step (1 sec by default) during the simulation. When the simulator contacts the DLLs the current signalization states and detector data are passed to the corresponding DLLs. Then, by using the acquired data, DLL files calculate the new desired signal states which will be passed back to the simulation. Depending on the settings, either the new signal states are applied immediately, or transition signal states (amber, red-amber) are used first automatically.

According to the above described opportunities, VISSIM COM (Component Object Model) interface and VISSIM API (Application Package Interface) programming can be efficiently used for customized ViL testing. At the same time, it may worth making calculation by other software concerning some specific parts of the simulation. In case of optimization procedures, for example, the user does not necessarily need to know the optimization algorithm in detail to use it right. Thus, by using specific mathematical software, lots of energy and time can be saved compared to the realization by programming. To exploit other software's functionality for VISSIM the main precondition is the online availability and communication during the simulation. Therefore, as further possibility, the use of MATLAB is proposed to assist VISSIM simulation. MATLAB can be controlled through COM interface similarly to VISSIM. Therefore, MATLAB may be used as a programmable mathematical subroutine library by programming the MATLAB Engine API [MathWorks 2010].

5.3.2 SUMO TraCI Interface for Co-Simulation

SUMO is an open-source microscopic continuous traffic flow simulation software developed by the German National Aerospace Center in 2001 [Krajzewicz et al. 2012]. It comes with a road network editor, which can add roads through interactive editing, modify the connection relationship of lanes, edit signal control schemes. The road network from VISSIM, OpenStreetMap, and OpenDrive can also be imported into SUMO through a separate conversion program. One can specify the route of each vehicle by editing the route file or using parameters to generate randomly. It also provides a visualization terminal based on OpenGL to display traffic simulation results in real-time. Recently, SUMO has also applied to the simulation of autonomous driving, providing random and complex dynamic environments. SUMO is embedded with a variety of car-following models; the default one is the Krauss model.

SUMO offers XML based configuration through of which one can design the geometry of any type of road networks and set up simulations in a simple way. However, for several problems this option is not satisfying. This is the case, for example, when the user aims to access and manipulate SUMO objects during the simulation in a dynamic way. To this end, an additional interface called Sumo TraCI (Traffic Control Interface) [Wegener et al. 2008] is offered based on TCP communication which can be programmed in several languages (e.g. Python, Matlab, Java, etc.). SUMO TraCI allows to control and modify a simulation process online. It also enables to retrieve information from any of the elements of the simulation network, such as vehicles, road links, junctions, traffic lights, etc.

A SUMO TraCI based application example for co-simulation realizing ViL testing is introduced in details in Appendix H.

5.4 Design of a Novel Road Traffic Control System for ZalaZONE Proving Ground

In this part the basic concepts for felxible road traffic control are introduced, specifically developed for ZalaZONE Proving Ground where 7 junctions at the Smart City Zone and 3 intersections at the University Test Track will be signalized. Obviously, the methodology can be adopted for any other test tracks or even for real-world traffic control system.

5.4.1 New Challenges of Traffic Light Controllers

Until now all realizations of traffic lights have been based on the fact that traffic signals are perceived by human drivers exclusively. Therefore, all relevant standards prescribe the technical requirements according to the capability of human perception, e.g. traffic lights' radiation angle [CEN 2015] or the position and number of traffic signal heads at the road crossing.

With the presence of automated cars the time has arrived to basically reconfigure and rethink the classical approach concerning the production and operation of traffic light controllers. The goal of this technological revolution is the cooperation between the traffic controller and the vehicles, i.e. V2I or I2V based technologies. This can be realized in one-way or two-way communication:

1. Traffic light controller provides messages to road vehicles which process the received information for their own purposes.

2. Road vehicles communicate information to traffic light controller.

3. The communication is bidirectional between the traffic controller and the vehicle.

In relation with the wireless technology, the technical specification for Signal Phase and Time and Map Data (SPaT/MAP) [Group 2015] must be considered in future traffic controller design. SPaT/MAP offers a potential channel for detailed information exchange between traffic systems and road users. Based on SPaT data the vehicles (or drivers) can be informed about the current status and change of the traffic signal ahead as well as about the next signal stage change. It also provides information about approaching traffic to optimize the signal system. MAP data describes the physical geometry of one or more intersections. In connection with SPaT/MAP the ISO/TS 19091:2017 norm [CEN 2017] is also important to mention as it defines the message, data structures, and data elements to support exchanges between the roadside equipment and vehicles.

The guidance of the SAE International is also worth mentioning [SAE International 2016]. Under the code SAE J2735_202007 the Dedicated Short Range Communications (DSRC) Message Set Dictionary was published. The aim of this document is to provide a message set, and its data frames and data elements, specifically for use by applications intended to utilize the 5.9 GHz DSRC for wireless access in vehicular environments.

5.4.2 Traffic Control System Design for Test Track

The aim of the planned road traffic control is to enable a flexible system such that traffic signal heads (vehicle, bicycle, pedestrian and auxiliary signals) can be controlled even separately and freely during vehicle tests. The control of all traffic light controllers of the 7 junctions at the ZalaZONE Smart City Zone and that of the 3 traffic light intersections at the University Test Track shall be made available by means of a central control software running in a cloud system. The basic concept of the system is illustrated in Fig. 5.7 where each traffic lights as well as traffic signal heads can be arbitrary controlled. A more specified architecture is shown by Fig. 5.8.

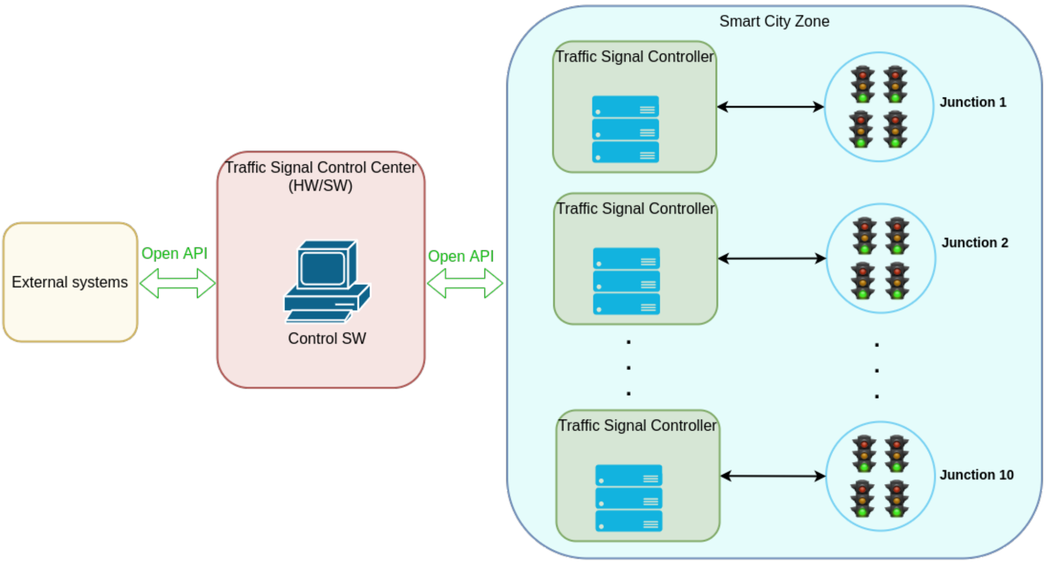


Figure 5.7. Overview of the ZalaZONE traffic light control system

The central software controls the traffic light system via an open source API. The control center shall be made available to external systems via an open source API too.

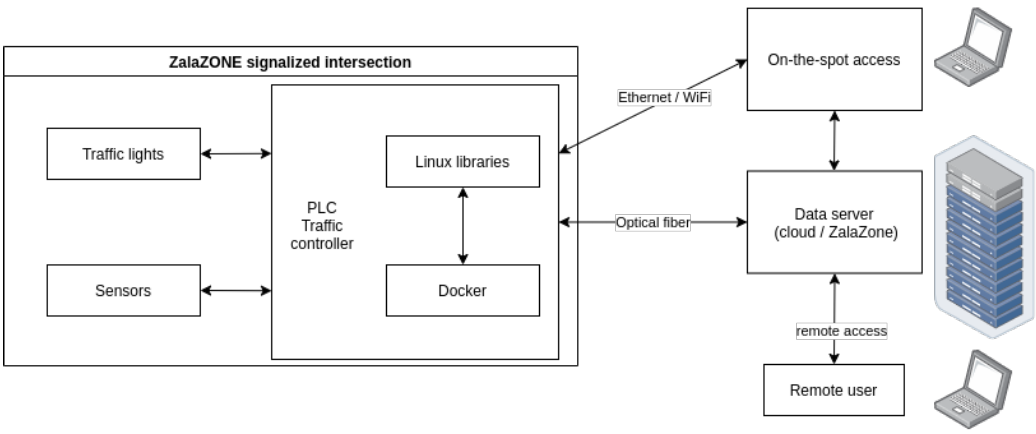


Figure 5.8. System architecture of the traffic control

The main requirements for the control system is defined as follows. The traffic light control system shall be freely programmable. All safety systems common in road traffic management systems (and required by standard) shall be deactivatable (when deactivated, there is no inter-green time matrix, no green conflict monitoring). The central control software shall also ensure that the signal heads are accessible at all times for verification: the software shall continuously check that the predetermined signal phases (even if intentionally irregular for testing purposes) are displayed on the light points and that the LED bulbs are not broken down. The control center has three control modes:

1. Signal Control Script:
   Operation according to a predefined sequence in a script file (a case of this is the conventional fixed-time program protected by intergreen time matrix).

2. Signal Control API:
   Control implemented by commands from an arbitrary program (e.g. Matlab or Python script) via open API.

3. Signal Control GUI:
   Control can be realized via a GUI. In practice, it means an arbitrary modification of the currently running program.

The hierarchy between the 3 control modes follows the following sequence. A Signal Control Script based control (1) can be overwritten by logic (2) via the Signal Control API or modified by control (3) via the GUI at any time. Additionally, a Signal Control API based control (2) can only be overwritten by an intervention through the Signal Control GUI (3).

In the designed system the following access levels are defined for users. Access level ”admin” denotes access to every function and the development environment. Access level ”tester” means access to every relevant function. Access level ” researcher” is the access to every function. Access level ”demo/viewer” means access to limited functions.

Along with the traffic control system design a simulation environment is also prepared for the digital twin realization, see Appendix I.

5.5 Contributions

Scientific contributions have been elaborated in the field of automated vehicle testing and automated traffic system testing. Novel methods have been developed to support co-simulation opportunities with real-world or virtual objects. Additionally, a framework is elaborated which enables to generate surrounding traffic around a given test vehicle.

Thesis 4

I have developed methods including microscopic traffic simulation to support the testing of vehicle and traffic developments related to automated functions, where the traffic simulation can be interfaced with other simulators in a flexible and synchronized way. In this system I have elaborated a genetic algorithm based calibration method which can be used for the real-time tuning of traffic simulation in case of real-time traffic measurement. I have also designed a special traffic control system for traffic lights, specifically for testing and proving activities of automated traffic and vehicles.

Thesis 4.1

Based on sensitivity analysis I have demonstrated that the common practice of microscopic traffic simulation needs thorough revision and modification when it is applied with the presence of autonomous vehicles in order to get realistic results. To show the sensitivity of microscopic simulation to automated vehicle's behavior I applied validated traffic simulator (PTV VISSIM).

Thesis 4.2

I have elaborated the online calibration of microscopic traffic simulation based on genetic algorithm. Practically, a digital twin of the real-world road traffic can be mapped, which can be useful in automated vehicle testing or proving processes. The L_∞-norm of relative errors was used to form the fitness function in the genetic algorithm based calibration process:

$$\min_{Q(k)} \sum_{i=1}^{n} \left\| \frac{\bar{F}_i^{Measured}(k) - \bar{F}_i^{Simulated}(Q(k))}{\bar{F}_i^{Measured}(k)} \right\|_\infty \tag{5.3}$$

where $\bar{F}_i^{Measured}(k)$ is the average traffic volume of edge i, $\bar{F}_i^{Simulated}(Q(k))$ is the average traffic volume of edge i produced by the simulator, and $Q(k)$ is the parameter to be calibrated (traffic inflow).

Thesis 4.3

A special framework has been developed in which road traffic simulators (such as PTV VISSIM or SUMO) can be co-simulated with real-world and/or virtual objects resulting in a mixed-reality system. Via the developed framework it is possible to carry out flexible Vehicle-in-the-Loop testing of automated vehicles such that the surrounding traffic of the test vehicle is virtually created in a synchronized way based on validated traffic simulation.

Thesis 4.4

I designed a special traffic control system applicable on automotive proving ground to realize fully flexible and customized testing with automated vehicles and intelligent infrastructure. The system is designed to provide several control levels for traffic lights based on the concept of cloud based remote control.

Related journal and conference papers

- T. Tettamanti and I. Varga. Development of road traffic control by using integrated VISSIM-MATLAB simulation environment. *Periodica Polytechnica ser. Civil Engineering*, 56(1):43–49, 2012. doi: 10.3311/pp.ci.2012-1.05 **(Q2)**;
- T. Tettamanti, A. Csikós, I. Varga, and A. Eleőd. Iterative calibration of Vissim simulator based on genetic algorithm. *Acta Technica Jaurinensis*, 8(2):145–152, 2015b. doi: 10.331 1/PPtr.7685;
- T. Tettamanti, Á.Z. Milacski, A. Lőrincz, and I. Varga. Iterative calibration method for microscopic road traffic simulators. *Periodica Polytechnica ser. Transp. Eng.*, 43(2):87–91, 2015c. doi: 10.3311/PPtr.7685 **(Q2)**;
- Zs. Szalay, D. Esztergár-Kiss, T. Tettamanti, P. Gáspár, and I. Varga. Recar: Hungarian research center for autonomous road vehicles is on the way. *ERCIM News, In: Special Theme: Autonomous Vehicles*, 109:27–29, 2016;
- C. Bartolini, T. Tettamanti, and I. Varga. Critical features of autonomous road transport from the perspective of technological regulation and law. *Transportation Research Procedia*, 27:791 – 798, 2017. doi: 10.1016/j.trpro.2017.12.002. 20th EURO Working Group on Transportation Meeting, EWGT 2017, 4-6. September 2017, Budapest, Hungary;
- T. Tettamanti, M. Szalai, S. Vass, and V. Tihanyi. Vehicle-In-the-Loop test environment for autonomous driving with microscopic traffic simulation. In *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pages 1–6, 2018b. doi: 10.110 9/ICVES.2018.8519486;
- Zs. Szalay, T. Tettamanti, D. Esztergár-Kiss, I. Varga, and C. Bartolini. Development of a test track for driverless cars: Vehicle design, track configuration, and liability considerations. *Periodica Polytechnica ser. Transportation Engineering*, 46(1):29–35, 2018. doi: 10.3311/PPtr.10753 **(Q2)**;
- M.T. Horváth, Q. Lu, T. Tettamanti, Á. Török, and Zs. Szalay. Vehicle-In-The-Loop (VIL) and Scenario-In-The-Loop (SCIL) automotive simulation concepts from the perspectives of traffic simulation and traffic control. *Transport and Telecommunication Journal*, 2(20): 153–161, 2019a. doi: 10.2478/ttj-2019-0014 **(Q3)**;
- M.T. Horváth, T. Tettamanti, B. Varga, and Zs. Szalay. The Scenario-in-the-Loop (SciL) automotive simulation concept and its realisation principles for traffic control. In *hEART 2019 - 8th Symposium of the European Association for Research in Transportation.* European Association for Research in Transportation, 2019b. Paper 62;
- M. Szalai, B. Varga, T. Tettamanti, and V. Tihanyi. Mixed reality test environment for autonomous cars using Unity 3D and SUMO. In *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 73–78, 2020;

- X. Fang, T. Tettamanti, and A.C. Piazzi. Online calibration of microscopic road traffic simulator. In *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 275–280, 2020. doi: 10.1109/SAMI48414.2020.9108744;
- T. Ormándi, B. Varga, and T. Tettamanti. Autonóm járművek komplex virtuális tesztkörnyezetének fejlesztése. In *Innováció és Fenntartható Felszíni Közlekedés Konferencia (IFFK), Budapest, Hungary, 2020.10.28-2020.10.30*, 2020;
- V. Potó, J.M. Lógó, T. Tettamanti, Á. Barsi, and N. Krausz. Térképi formátumok értékelése az önvezetés szempontjából. In *Az elmélet és a gyakorlat találkozása a térinformatikában XI.: Theory meets practice in GIS*, pages 207–215, 2020;
- S. Duleba, T. Tettamanti, Á. Nyerges, and Z. Szalay. Ranking the key areas for autonomous proving ground development using pareto analytic hierarchy process. *IEEE Access*, 9: 51214–51230, 2021. doi: 10.1109/ACCESS.2021.3064448 **(Q1)**;
- X. Fang and T. Tettamanti. Change in microscopic traffic simulation practice with respect to the emerging automated driving technology. *Periodica Polytechnica ser. Civil Engineering*, 2021b **(Q3)**;
- K. Gangel, Z. Hamar, A. Háry, Á. Horváth, G. Jandó, B. Könyves, D. Panker, K. Pintér, M. Pataki, M. Szalai, Zs. Szalay, T. Tettamanti, V. Tihanyi, B. Tóth, B. Varga, and Zs.J. Viharos. Modelling the ZalaZONE Proving Ground: a benchmark of State-of-the-art Automotive Simulators PreScan, IPG CarMaker, and VTD Vires. *Acta Technica Jaurinensis*, 14(4):488–507, Nov. 2021. doi: 10.14513/actatechjaur.00606;
- B. Varga, T. Tettamanti, and Zs. Szalay. System architecture for Scenario-In-The-Loop automotive testing. *Transport and Telecommunication Journal*, 22(2):141–151, 2021. doi: doi:10.2478/ttj-2021-0011 **(Q3)**;
- T. Tettamanti M. Szalai. Kevert valóság fejlesztési környezet autonóm járművek számára. *Közlekedéstud. Szemle*, LXXI(3):17–28, 2021. doi: 10.24228/KTSZ.2021.3.2. ISSN 0023-4362;
- V. Tihanyi, T. Tettamanti, M. Csonthó, A. Eichberger, D. Ficzere, K. Gangel, L.B. Hörmann, M.A. Klaffenböck, C. Knauder, P. Luley, Z.F. Magosi, G. Magyar, H. Németh, J. Reckenzaun, V. Remeli, A. Rövid, M. Ruether, S. Solmaz, Z. Somogyi, G. Soós, D. Szántay, T.A. Tomaschek, P. Varga, Zs. Vincze, C. Wellershaus, and Zs. Szalay. Motorway measurement campaign to support R&D activities in the field of automated driving technologies. *Sensors*, 21(6), 2021. ISSN 1424-8220. doi: 10.3390/s21062169 **(Q2)**;
- H. Li, V.P. Makkapati, D. Nalic, A. Eichberger, X. Fang, and T. Tettamanti. A real-time co-simulation framework for virtual test and validation on a high dynamic vehicle test bed. In *2021 IEEE Intelligent Vehicles Symposium (IV21)*. IEEE, 2021;
- X. Fang and T. Tettamanti. Traffic congestion phenomena when motorway meets urban road network. In *2021 IEEE 25th International Conference on Intelligent Engineering Systems (INES)*, pages 000025–000030, 2021a. doi: 10.1109/INES52918.2021.9512920;
- T. Ormándi, B. Varga, and T. Tettamanti. Estimating vehicle suspension characteristics for digital twin creation with genetic algorithm. *Periodica Polytechnica Transportation Engineering*, 49(3):231–241, 2021b. doi: 10.3311/PPtr.18576;
- T. Ormándi, B. Varga, and T. Tettamanti. Distributed intersection control based on cooperative awareness messages. In *5th Conference on Control and Fault Tolerant Systems (SysTol)*, pages 323–328, 2021a. doi: 10.1109/SysTol52990.2021.9595376.

Related books and technical manual

- T. Tettamanti, T. Luspay, and I. Varga. *Road Traffic Modeling and Simulation.* Akadémiai Kiadó, 2019a;
- Zs. Bede, A. Csikós, M.T. Horváth, T. Tettamanti, and I. Varga. *Közúti forgalommodellezési gyakorlatok, 4. kiadás 3. és 4. fejezetek a Vissim 9 és 10-es verzió-hoz aktualizálva.* BME Közlekedés- és Járműirányítási Tanszék, 2020. TAMOP-4.1.1.C-12/1/KONV-2012-0002;
- T. Tettamanti and M.T. Horváth. *A practical manual for Vissim-COM programming in Matlab and Python – 5th edition for Vissim version 2020 and 2021.* Budapest University of Technology and Economics, Dept. for Control of Transportation and Vehicle Systems, 5th edition, 2021.

Chapter 6

Conclusions

It is a fact that automated vehicles will replace conventional human driven vehicles in the next few decades. Although there is a plenty of open questions, the emerging autonomous driving technology will definitely bring a massive transformation in the road transport sector soon. Due to the high complexity of transport systems, efficient traffic estimation, traffic control and vehicle/traffic simulation techniques are critical to deal with this disruptive change.

In the dissertation, advanced methods and methodologies were presented as potential candidates for practical applications in a new era of urban road traffic engineering with the appearance of automated vehicles. My research results are concluded in four distinct theses. The main results can be summarized as follows. I have introduced advanced and cost-effective traffic estimation methods which can be directly applied for traffic system operation purposes. I have analyzed the impact of automated driving to the conventional traffic system, and based on the simulation results I proposed an update to the traditional macroscopic fundamental diagram. I have developed dynamic control schemes for private cars and public buses as well as I have elaborated the concept for the distributed road traffic controller. I have worked out methods that enable the real-time application of microscopic traffic simulation for automotive tests through digital twin and co-simulation techniques. Finally, I designed a novel concept for flexible traffic signal control system which is applicable on automotive test track.

Throughout my research, I have always kept in mind that theory should meet practice. Accordingly, the solutions presented in the dissertation can be used both as practical methodologies and as concrete practical applications. I have also prepared the achieved results as algorithm implementations using a combination of validated mathematical and traffic simulation software tools. These algorithms were verified through real-life examples to highlight their usability. The practical applicability is also demonstrated by two patents of mine and my developments directly related to the innovation activities at the ZalaZONE Automotive Proving Ground.

Appendix A

Kalman Filter

Kalman [1960] published a novel solution to the linear filtering problem. The recursive solution algorithm provides an efficient estimation of dynamical systems with noisy inputs. The method is introduced below for the case of a general discrete LTV system.

Let us consider the state and measurement difference equations of a discrete LTV system:

$$x(k+1) = A(k)x(k) + B(k)u(k) + v(k), \tag{A.1}$$

$$y(k) = C(k)x(k) + z(k). \tag{A.2}$$

$x(k) \in \mathbb{R}^n$ denotes the state variable of the system, $u(k) \in \mathbb{R}^m$ the deterministic control input, and $y(k) \in \mathbb{R}^p$ the measurements. The system is given with known and constant system matrices; $A(k) \in \mathbb{R}^{n\times n}$, $B(k) \in \mathbb{R}^{n\times m}$, and $C(k) \in \mathbb{R}^{p\times n}$. State noise $v(k)$ and measurement noise $z(k)$ are both stochastic signals having zero mean Gaussian distribution, i.e. $E\{v(k)\} = 0$ and $E\{z(k)\} = 0$. The noise covariances and are assumed to be known:

$$R_v(k) = E\left\{v(k)v(k)^T\right\}, \tag{A.3}$$

$$R_z(k) = E\left\{z(k)z(k)^T\right\}. \tag{A.4}$$

The covariance matrices may change at each time step. They express the measure of the disturbances in Eq.(A.1) and (A.2). It is also assumed that there is no correlation between the state and measurement noise, i.e. $E\left\{v(k)z(k)^T\right\} = 0$ and $E\left\{z(k)v(k)^T\right\} = 0$, respectively.

Denote $\hat{x}^-(k)$ the *a priori* and $\hat{x}(k)$ the *a posteriori* state estimation. Accordingly, the estimate errors can be defined:

$$e^-(k) = x(k) - \hat{x}^-(k), \tag{A.5}$$

$$e(k) = x(k) - \hat{x}(k). \tag{A.6}$$

Moreover, the *a priori* and *a posteriori* estimate error covariances can be given as

$$P^-(k) = E\left\{e^-(k)e^-(k)^T\right\}, \tag{A.7}$$

$$P(k) = E\left\{e(k)e(k)^T\right\}. \tag{A.8}$$

The Kalman Filter can be formulated in a single linear equation as follows:

$$\hat{x}(k) = F(k)\hat{x}(k) + G(k)y(k) + H(k)u(k), \tag{A.9}$$

where $\hat{x}(k)$ indicates the statistically optimal estimate of the underlying system state $x(k)$. $F(k)$, $G(k)$ and $H(k)$ denote the system matrices. Basically, Kalman Filtering represents a minimum mean square error estimation vie the minimization of $P(k)$ (under the assumption that $E\{e(k)\} = 0$ for all k).

$F(k)$, $G(k)$ and $H(k)$ can be given in compact forms [Lantos 2001, 2003]:

$$F(k) = (I - G(k)C(k))A(k), \tag{A.10}$$
$$H(k) = (I - G(k)C(k))B(k). \tag{A.11}$$

Therefore, only $G(k)$ has to be determined. For this reason, the calculation of *a priori* state $\hat{x}^-(k)$ is given at time step k:

$$\hat{x}^-(k) = A(k-1)\hat{x}(k-1) + B(k-1)u(k-1). \tag{A.12}$$

The *a priori* estimate error covariance is defined as:

$$\hat{P}^-(k) = A(k)P(k)A(k)^T + R_v(k). \tag{A.13}$$

Moreover, the *a posteriori* state is calculable as follows:

$$\hat{x}(k) = \hat{x}^-(k) + G(k)(y(k) - C(k)\hat{x}^-(k)), \tag{A.14}$$

where $G(k)$ denotes the Kalman gain which intends to minimize $P(k)$. The Kalman gain is given as follows:

$$G(k) = \hat{P}^-(k)C(k)^T(C(k)\hat{P}^-(k)C(k)^T + R_z(k))^{-1}. \tag{A.15}$$

Finally, the *a posteriori* estimate error covariance is formulated as:

$$P(k) = (I - K(k)C(k))\hat{P}^-(k). \tag{A.16}$$

According to the above, the linear Kalman Filtering can also be summarized in the well-know two-phase algorithm:

1. Prediction (time update between measurements):

$$\hat{x}^-(k) = A(k-1)\hat{x}(k-1) + B(k-1)u(k-1), \tag{A.17}$$
$$\hat{P}^-(k) = A(k)P(k)A(k)^T + R_v(k). \tag{A.18}$$

2. Correction (measurement update):

$$G(k) = \hat{P}^-(k)C(k)^T(C(k)\hat{P}^-(k)C(k)^T + R_z(k))^{-1}, \tag{A.19}$$
$$\hat{x}(k) = \hat{x}^-(k) + G(k)(y(k) - C(k)\hat{x}^-(k)), \tag{A.20}$$
$$P(k) = (I - G(k)C(k))\hat{P}^-(k). \tag{A.21}$$

Algorithm 2: The Kalman Filter algorithm

Appendix B

Constrained Kalman Filter

As an extension of the standard linear Kalman Filter, constrained filtering can be realized in order to obtain better estimation results.

Assume that the modeled system satisfies the following constraints:

$$A_{eq}\, x(k) = b_{eq}, \tag{B.1}$$

$$A_{in}\, x(k) \leq b_{in}, \tag{B.2}$$

where A_{eq} and A_{in} are known matrices as well as b_{eq} and b_{in} are known vectors. In this case, estimated states also need to satisfy these conditions:

$$A_{eq}\, \hat{x}(k) = b_{eq}, \tag{B.3}$$

$$A_{in}\, \hat{x}(k) \leq b_{in}. \tag{B.4}$$

Compliance with these constraints can be reached by projecting the state to lie in the constrained space at each estimation interval [Gupta and Hauser 2007]. This means that the unconstrained filter runs in a normal way, but at each iteration the updated state estimate is forced to lie in the constrained space. In this approach, the analytic solution is no longer available for filtering. Thus, numerical optimization is needed to be applied. The projection is carried out via the following constrained optimization problem [Gupta and Hauser 2007; Simon 2010]:

$$\tilde{x}(k) = argmin_x\, (x - \hat{x}(k))^T\, W\, (x - \hat{x}(k)), \tag{B.5}$$

$$s.t.\ (B.1)\ and\ (B.2),$$

where $\tilde{x}$ is the projected state estimate and W is a weighing matrix.

W can be chosen as an identity matrix (hereinafter referred to as cKF-I). The result is then the least square estimate subject to the constraints, which means that estimates necessarily get closer to the real state values. If noises are assumed to be white and W is set to $P^{-1}(k)$ in each interval (hereinafter referred to as cKF-P), the result is the maximum probability estimate of the state subject to state constraints [Simon 2010].

Appendix C

Moving Horizon Estimation

MHE is an effective alternative for Kalman Filtering as it can also be applied to estimate the states of a dynamic system. However, this method can manage constraints on the states and noises, and noises do not need to be described as white noise [Findeisen 1997].

MHE uses the results of N previous steps to estimate the k^{th} state, i.e. every estimation is based on the examination of steps $(k-N)$ to k. The basis of the method is the same measurement and state equation as that of the Kalman Filter. The MHE minimizes the measurement and state noises to execute the state estimation.

The estimated states are obtained through minimizing the following $J(x)$ cost function:

$$\begin{aligned} J(x) = \sum_{j=k-N+1}^{k-1} [\hat{v}(j)^T Q^{-1} \hat{v}(j)] \ + \sum_{j=k-N+1}^{k} [\hat{z}(j)^T R^{-1} \hat{z}(j)] \\ +[\hat{x}(k-N+1) - \bar{x}(k-N+1)]^T P^{-1} [\hat{x}(k-N+1) - \bar{x}(k-N+1)]. \end{aligned} \tag{C.1}$$

Index j advances through the steps of the horizon. The first addend of $J(x)$ contains the state noise and the inverse of its covariance matrix, whereas the second addend involves the measurement noise and the inverse of its covariance matrix. $\hat{x}(k-N+1)$ in the third addend means the earliest estimation of the k^{th} step, $\bar{x}(k-N+1)$ is the state estimate of the previous step, and P^{-1} is the inverse of the state error covariance matrix [Tettamanti et al. 2016b], [Kulcsár et al. 2005]. To simplify the cost function, the third addend (also known as "arrival cost") can be omitted [Haugen 2018].

The MHE can manage dynamic and static constraints. For the former, the state and measurement equations are used in the k^{th} step (where A, B, and C can be defined as time varying matrices $A(k)$, $B(k)$, and $C(k)$):

$$\hat{x}(k+1) = A(k)\,\hat{x}(k) + B(k)\,u(k) + \hat{v}(k), \tag{C.2}$$

$$y(k) = C(k)\,\hat{x}(k) + \hat{z}(k). \tag{C.3}$$

Static constraints can be set up on the estimated states, and on the measurement and state noises:

$$x_{min} \leq \hat{x}(j) \leq x_{max} \ , \tag{C.4}$$

$$v_{min} \leq \hat{v}(j) \leq v_{max} \ , \tag{C.5}$$

$$z_{min} \leq \hat{z}(j) \leq z_{max} \ . \tag{C.6}$$

The MHE algorithm in the k^{th} step is detailed as follows.

1. Perform measurements providing $y(k)$.
2. Perform optimization using $y(k)$, subject to dynamic and static constraints:

$$\min_{\hat{x}(j),\hat{v}(j),\hat{z}(j),j=k-N+1,\ldots,k-1} J(k)\ . \tag{C.7}$$

3. Move the horizon one step further, then go to Step 1:

$$k := k+1\ .$$

The results of the optimization in the k^{th} step are the estimated vectors $\hat{x}(j)$, $\hat{v}(j)$, and $\hat{z}(j)$ for each j^{th} step of the horizon (where N is the length of the horizon, and $j = k-N+1,\ \ldots,\ k-1$).

Matrix A_{eq} and vector b_{eq} define the equality constraints. The inequality constraint is defined (using matrix A_{in} and vector b_{in}) so that every turning rate is non-negative. The structure of the constraint matrices and vectors are similar to those described in the case of the constrained Kalman Filter, however, their dimension depends on the length of the horizon (N).

The cost function in Eq. (C.1) contains the measurement and state noise vectors $\hat{z}$ and $\hat{v}$. These are unknown as default, however, they can be expressed from the system equations Eq. (C.2) and Eq. (C.3) as follows (where $A = 1$ and $B = 0$ in the k^{th} estimation step and j^{th} horizon step):

$$\hat{v}(j) = \hat{x}(j+1) - \hat{x}(j)\ , \tag{C.8}$$

$$\hat{z}(j) = y(j) - C(j)\,\hat{x}(j)\ . \tag{C.9}$$

The third addend of cost function (C.1), the arrival cost is assumed to be 0, as its inclusion did not affect the results significantly in the testing phase. In this way, substituting Eq. (C.8) and (C.9) into Eq. (C.1), the cost function is recast as follows:

$$\begin{aligned} J(x) = &\sum_{j=k-N+1}^{k-1} \left[(\hat{x}(j+1) - \hat{x}(j))^T Q^{-1} (\hat{x}(j+1) - \hat{x}(j))\right] + \\ &\sum_{j=k-N+1}^{k} \left[(y(j) - C(j)\,\hat{x}(j))^T R^{-1} (y(j) - C(j)\,\hat{x}(j))\right]\ . \end{aligned} \tag{C.10}$$

Appendix D

Biproportional Procedure

The Biproportional Procedure (BP) is an iterative algorithm [Ben-Akiva et al. 1985], where the variation of two coefficients (a and b) causes the variation of turning flows in each iteration. Two sets of input data are necessary for this procedure. A preliminary origin-destination matrix (t) and the traffic flows on each leg of the roundabout (O_i entering and D_j exiting counts in case of entrance i and exit j). t_{ij} is the traffic volume from i to j, and there exist n_O entrances and n_D exits. The accuracy of the BP estimation depends largely on the accuracy of prior matrix t [Dixon and Rilett 2005].

The BP procedure aims to estimate the elements of the current OD-matrix T, based on the current flows on each leg and prior matrix t. Therefore, the resulting matrix of the procedure contains traffic volumes, which then can be converted into turning rates. This assists the comparison of estimation procedures.

The estimated T has to satisfy the following constraints:

$$O_i = \sum_{j=1}^{n_D} T_{ij}, \tag{D.1}$$

$$D_j = \sum_{i=1}^{n_O} T_{ij}. \tag{D.2}$$

To meet the constraints in Eq. (D.1) and Eq. (D.2), iterations are executed. Each iteration alters the proportions a and b. These proportions from the previous iteration are marked as a^* and b^*. Estimated matrix T has a minimal difference from prior matrix t, whilst satisfying the constraints [Dixon et al. 2007].

The initial conditions for the BP procedure are that a_i^*, b_j, and b_j^* are set to 1, while T_{ij} is set equal to t_{ij} for all turning movements. For a stopping criterion, a sufficiently small value of ϵ needs to be reached by the changes in a_i and b_j.

The steps of the algorithm are detailed as follows.

1. Calculation of a_i:

$$a_i = \left(\frac{O_i}{\sum_{j=1}^{n_D} T_{ij}} \right) a_i^*. \tag{D.3}$$

2. Calculation of T_{ij}:

$$T_{ij} = t_{ij}\, a_i\, b_j. \tag{D.4}$$

3. Calculation of b_j:

$$b_j = \left(\frac{D_j}{\sum_{i=1}^{n_O} T_{ij}} \right) b_j^*. \tag{D.5}$$

4. Calculation of T_{ij} using Eq. (D.4).

5. End of iteration. If the changes in a_i and b_j are greater than the previously defined ϵ, the iteration starts over from Step 1. If the changes are less than or equal to ϵ, the last estimated T_{ij} is the result of the current interval.

The algorithm above depicts only one measurement period. While implementing the BP procedure, turning flows need to be estimated in each interval. After the stopping criterion is met, all elements of T are rounded to the nearest integer. T then becomes the prior matrix for the next period as the volumes of O_i and D_j are updated as well.

An advantage of the BP procedure is its relatively low computational requirements. Also, OD-matrices are estimated based on 8 cross-sectional counts instead of 16 turning movement observations. Another benefit that derives merely from the characteristics of the algorithm is that if U-turns are assumed to be zero in the prior matrix, the estimated matrices also have zeros in the main diagonal. A disadvantage of the procedure is its heavy dependence on the accuracy of the prior matrix.

Appendix E

Uncertainty Definition and H-infinity Filter Design for Traffic Estimation

Considering a macroscopic approach (individual vehicle dynamics are omitted), for link z the number of vehicles can be modeled based on the vehicle-conservation law during $[kT, (k+1)T]$ where T denotes the sample time and $k = 0,\ 1,\ 2,\ ...$ is the discrete time index:

$$n_z(k+1) = n_z(k) + T\left[\sum_{w \in I_M} \alpha_{w,z} q_w(k) - q_z(k)\right]. \tag{E.1}$$

The parameters in Eq. (E.1) are as follows: n_z is the number of vehicles on link z (in PCE); I_M denotes the set of incoming links w at junction M, i.e. $w \in I_M$; $\alpha_{w,z} \in [0,1]$ is the turning rate from link w to link z; q_w denotes the traffic flow from link w (PCE/T); q_z is the traffic outflow from link z (PCE/T).

A crucial point of Eq. (E.1) is the dynamics of link outflows. A possible approach to describe traffic outflow in a given network is described by the theory of urban fundamental diagram. The theory is called macroscopic fundamental diagram (MFD). By using the analogy of the MFD concept, the outflows $q_{w,z}$ and q_z can be defined by restricting the traffic network to link level. This practically means that each link has a dedicated MFD model. MFD assumes the following fundamental relationship:

$$q = \rho \cdot v, \tag{E.2}$$

where ρ denotes the traffic density and v is the space mean speed on a link. There are several formulas available in the literature for v. Here, one of the basic relationships is used for describing the speed of link z (called Pipes-Munjal model [Pipes 1967], which is practically a modified version of Greenshields' model):

$$v_z(\rho) = v_z^{free}\left[1 - \left(\frac{\rho_z}{\rho_z^{jam}}\right)^a\right], \tag{E.3}$$

where v_z^{free} represents the free-flow speed (i.e. no congestion), ρ_z^{jam} is the jam density (practically a 'bumper-to-bumper' case within the road link) and a is an empirical parameter. As traffic density is defined as

$$\rho_z = \frac{n_z}{l_z}, \tag{E.4}$$

(l_z is the link length) Eq. (E.3) can be recast as follows:

$$v_z(n_z) = v_z^{free}\left[1 - \left(\frac{n_z}{n_z^{jam}}\right)^a\right]. \tag{E.5}$$

By substituting Eq. (E.5) into Eq. (E.2), the link-based traffic flow is derived:

$$q_z = \rho_z v_z = \frac{n_z}{l_z} v_z^{free} \left[1 - \left(\frac{n_z}{n_z^{jam}} \right)^a \right]. \tag{E.6}$$

Note that flow q_w is also calculated by the formula of Eq. (E.6) concerning link w.

The two-fluid model [Herman and Prigogine 1979] considers as the whole traffic flow was composed by two flows: the flow of moving vehicles and the flow of vehicles stopped in traffic lanes (e.g. at red signal, in traffic jams, for freight delivery etc). The model defines the fraction of stopped vehicles as f^s, which can represent the ratio of the time while a floating car circulating in a network is stopped divided by its whole travel time:

$$f^s = \frac{T^s}{T}. \tag{E.7}$$

The two-fluid model states that f^s can be given in term of concentration:

$$f^s = \left(\frac{\rho}{\rho^{jam}} \right)^p, \tag{E.8}$$

where ρ^{jam} denotes the jam density and parameter p is the measure of quality of the traffic network. Substituting Eq. (E.4) into Eq. (E.8), f^s can be rewritten as:

$$f^s = \frac{T^s}{T} = \left(\frac{n}{n^{jam}} \right)^p. \tag{E.9}$$

The two-fluid model is usually applied to characterize a whole traffic network (town or districts). Nevertheless, the two-fluid approach is also valid for smaller networks. Therefore, a link-based two-fluid model can be given concerning link z as follows:

$$f_z^s = \frac{T_z^s}{T_z} = \left(\frac{n_z}{n_z^{jam}} \right)^p, \tag{E.10}$$

where T_z^s is the average stop time of the floating cars going through link z and T_z is the average travel time of vehicles on link z. Since f^s provides us information on queue lengths on links, it gives a more specific description of the traffic state on links than average travel time or speed would.

In road traffic technology the most common used sensor types are magnetic sensors and inductive loop-detectors. The time-occupancy parameter of these is calculated as follows:

$$o^t = \frac{\sum t^{occ}}{T}, \tag{E.11}$$

where $\sum t^{occ}$ denotes the sum of all occupancy times while the detector is covered by vehicles during sample time T.

Papageorgiou and Vigos [2008] derives the relationship between time-occupancy measurements of cross-sectional traffic detectors and the road link's space-occupancy. Space-occupancy is defined as the ratio of the sum of all vehicle lengths and the link length:

$$o_z^s = \frac{\sum l^{veh}}{l_z}. \tag{E.12}$$

Moreover, by considering a unit vehicle length l^{PCE}:

$$o_z^s = \frac{n_z \cdot l^{PCE}}{l_z}. \tag{E.13}$$

Time and space-occupancy values are quite similar [Zhang and Rakha 2005], therefore the slight difference between them can be modeled by an appropriate noise term ζ:

$$o_z^t = o_z^s + \zeta = \frac{n_z \cdot l^{PCE}}{l_z} + \zeta. \tag{E.14}$$

The discrete time state space representation of a nonlinear dynamics (without control input in this case) can be given by the following stochastic difference equation:

$$x(k+1) = f\left(x(k), \nu(k)\right), \tag{E.15}$$

with the measurement equation:

$$y(k) = g\left(x(k), \zeta(k)\right), \tag{E.16}$$

where $\nu(k)$ and $\zeta(k)$ represent the process and measurement noise respectively.

State vector is composed as follows:

$$x(k) = \begin{pmatrix} n_1(k) \\ n_2(k) \\ n_3(k) \\ \vdots \\ n_n(k) \end{pmatrix}, \tag{E.17}$$

where n_z denotes the number of vehicles on link z ($z = 1, 2, ..., n$).

Based on Eq. (E.1) the dynamics of each link z in Eq. (E.15) is given as

$$n_z(k+1) = n_z(k) + T\left[\sum_{w \in I_M} \alpha_{w,z} q_w(k) - q_z(k)\right] + \nu_z(k), \tag{E.18}$$

augmented by $\nu_z(k)$ as a noise term in the system.

Applying Eq. (E.6) for traffic flow dynamics, Eq. (E.18) finally becomes:

$$n_z(k+1) = n_z(k) + T\left[\sum_{w \in I_M} \alpha_{w,z} \frac{n_w(k)}{l_w} v_w^{free}\left[1 - \left(\frac{n_w(k)}{n_w^{jam}}\right)^a\right] - \frac{n_z(k)}{l_z} v_z^{free}\left[1 - \left(\frac{n_z(k)}{n_z^{jam}}\right)^a\right]\right] + \nu_z(k) \tag{E.19}$$

Sample time T can be long, even 15 minutes, therefore, the effect of signal controllers are taken into consideration as an average value which means that it is not necessary to know the signal programs.

According to the state space representation form, the measurement equation (E.16) must be defined as well. Using the models provided in the previous sections, the following measurements can be defined in the system:

- o_z^t is the time-occupancy on link z, measured by traffic detectors as given by Eq. (E.11).
- $f_z^s = \frac{T_z^s}{T_z}$ from Eq. (E.10) is detected as floating car data (FCD) for a single vehicle. Hence, the mean of all floating car measurements during the sample time can be calculated as

  $$\bar{f}_z^s = \frac{\sum_{i=1}^{num} f_{i,z}^s}{num}, \tag{E.20}$$

  where num denotes the number of cars measured on link z. As p is a constant parameter, Eq. (E.10) can be rearranged:

  $$\left(\bar{f}_z^s\right)^{1/p} = \frac{n_z}{n_z^{jam}}. \tag{E.21}$$

  Therefore $\left(\bar{f}^s\right)^{1/p}$ is considered as a measured value.

Finally, the discrete time measurement equation is given as follows:

$$\underbrace{\begin{pmatrix} o_1^t(k) \\ o_2^t(k) \\ \vdots \\ o_n^t(k) \\ \left(\bar{f}_1^s\right)^{1/p}(k) \\ \left(\bar{f}_2^s\right)^{1/p}(k) \\ \vdots \\ \left(\bar{f}_n^s\right)^{1/p}(k) \end{pmatrix}}_{y(k)} = \underbrace{\begin{pmatrix} \frac{l^{PCE}}{l_1} & & & \\ & \frac{l^{PCE}}{l_2} & & \\ & & \ddots & \\ & & & \frac{l^{PCE}}{l_n} \\ \frac{1}{n_1^{jam}} & & & \\ & \frac{1}{n_2^{jam}} & & \\ & & \ddots & \\ & & & \frac{1}{n_n^{jam}} \end{pmatrix}}_{C} \underbrace{\begin{pmatrix} n_1(k) \\ n_2(k) \\ n_3(k) \\ \vdots \\ n_n(k) \end{pmatrix}}_{x(k)} + \underbrace{\begin{pmatrix} \zeta_1^{det}(k) \\ \zeta_2^{det}(k) \\ \zeta_3^{det}(k) \\ \vdots \\ \zeta_n^{det}(k) \\ \zeta_1^{FCD}(k) \\ \zeta_2^{FCD}(k) \\ \zeta_3^{FCD}(k) \\ \vdots \\ \zeta_n^{FCD}(k) \end{pmatrix}}_{\zeta(k)}. \tag{E.22}$$

To better deal with the nonlinear dynamics given, the linearization technique via Taylor series [Stengel 1986] can be used for Eqs. (E.15)-(E.16), i.e. the real state x and measurement y vectors are approximated:

$$x(k+1) \approx f(\hat{x}(k),0) + \frac{\partial f(\hat{x}(k),0)}{\partial x}\left(x(k) - \hat{x}(k)\right) + \frac{\partial f(\hat{x}(k),0)}{\partial \nu}\nu(k), \tag{E.23}$$

$$y(k) \approx g(\hat{x}(k),0) + \frac{\partial g(\hat{x}(k),0)}{\partial x}\left(x(k) - \hat{x}(k)\right) + \frac{\partial g(\hat{x}(k),0)}{\partial \zeta}\zeta(k), \tag{E.24}$$

where $\hat{x}(k)$ denotes the estimate of the state at discrete time step k.

Practically, the linearization means the calculation of Jacobian matrices of partial derivatives of functions (E.15)-(E.16):

$$A(k) = \tfrac{\partial f(\hat{x}(k),0)}{\partial x}, \tag{E.25}$$
$$B_\nu(k) = \tfrac{\partial f(\hat{x}(k),0)}{\partial \nu}, \tag{E.26}$$
$$C(k) = \tfrac{\partial g(\hat{x}(k),0)}{\partial x}, \tag{E.27}$$
$$C_\zeta(k) = \tfrac{\partial g(\hat{x}(k),0)}{\partial \zeta}. \tag{E.28}$$

By using the simplified notation of (E.25)-(E.28) for Eqs.(E.23)-(E.24), the following formulas are obtained:

$$x(k+1) \approx \tilde{x}(k) + A(k)\left(x(k){-}\hat{x}(k)\right) + B_\nu(k)\nu(k), \tag{E.29}$$
$$y(k) \approx \tilde{y}(k) + C(k)\left(x(k){-}\hat{x}(k)\right) + C_\zeta(k)\zeta(k), \tag{E.30}$$

where $\tilde{x}(k)$ and $\tilde{y}(k)$ are the approximated state and measurement variables.

Accordingly, the linearized matrices must be determined. The formula described by (E.25) is meant as differentiation by each element of state vector x. Therefore, for the state equation (E.19) two basic cases are given:

1. If the differentiation is done by state variable indexed by z (i.e. n_z):

$$\frac{\partial n_z(k+1)}{\partial n_z} = 1 - T\frac{v_z^{free}}{l_z}\left(1 - (a+1)\left(\frac{n_z(k)}{n_z^{jam}}\right)^a\right). \tag{E.31}$$

2. If the differentiation is done by state variable indexed by w (i.e. n_w):

$$\frac{\partial n_z(k+1)}{\partial n_w} = \alpha_{w,z} T\frac{v_w^{free}}{l_w}\left(1 - (a+1)\left(\frac{n_w(k)}{n_w^{jam}}\right)^a\right). \tag{E.32}$$

Jacobian matrix B_ν is resulted as

$$B_\nu(k) = I. \tag{E.33}$$

The Jacobian matrices of the measurement equation (E.22) are given as follows:

$$C = \begin{pmatrix} \frac{l^{PCE}}{l_1} & & & \\ & \frac{l^{PCE}}{l_2} & & \\ & & \ddots & \\ & & & \frac{l^{PCE}}{l_n} \\ \frac{1}{n_1^{jam}} & & & \\ & \frac{1}{n_2^{jam}} & & \\ & & \ddots & \\ & & & \frac{1}{n_n^{jam}} \end{pmatrix}, \tag{E.34}$$

$$C_\zeta(k) = I. \tag{E.35}$$

Since the investigated system of which the noise descriptions (the statistical properties of turning rates) are unknown, Kalman/H_∞ filter is applied to resolve robust state estimation problem, according to the description provided in Simon [2006].

Turning rate $\alpha_{w,z}$ is a quite ambiguous point of the traffic model described in Eq. (E.19). Obviously, one is able to estimate this term based on previous measurements. However, exact reliable values cannot be found for turning rates as they are strongly stochastic variables. Therefore, a robust approach can be applied for state estimation. By following the method of robust Kalman/H_∞ filtering for a linear system [Simon 2006], uncertainties can be encapsulated into the linearized system model (E.29) derived previously:

$$x(k+1) \approx \tilde{x}(k) + (A(k) + \Delta A(k))\,(x(k) - \hat{x}(k)) + B_\nu(k)\nu(k), \tag{E.36}$$

where ΔA denotes the uncertainty matrix concerning the turning rates. The uncertainty matrix is assumed to be of the following structure:

$$\Delta A(k) = M(k)\Gamma(k)E(k), \tag{E.37}$$

where $M(k)$ and $E(k)$ are known real constant matrices of appropriate dimensions, and $\Gamma(k)$ is an unknown real time-varying matrix satisfying the following inequality:

$$\Gamma^T(k)\Gamma(k) \le I. \tag{E.38}$$

Apart from robust state estimation, the designed filter must also be able to fuse data collected from different sensor sources. These sensors can either be installed into the road infrastructure or can be in connection with the movement of vehicles, for example floating car data (FCD), or floating mobile data (FMD). On those links where there is at least one built-in road traffic sensor, data is generated continuously, therefore, the estimation of the Kalman Filter can always be updated, even if FCD is available. On links where no built-in detector is installed, the continuous Kalman Filter update cannot be guaranteed, since measurement data is only generated if there is a vehicle equipped with such device. If there is no measurement data in a period, the intermittent Kalman Filter technique is used, i.e. the state estimate of the previous time-step is simply propagated [Sinopoli et al. 2004].

A minimal example modeling a simple junction (see Fig.E.1) is provided to show how the proposed method can be applied. The state of the network is represented by the number of

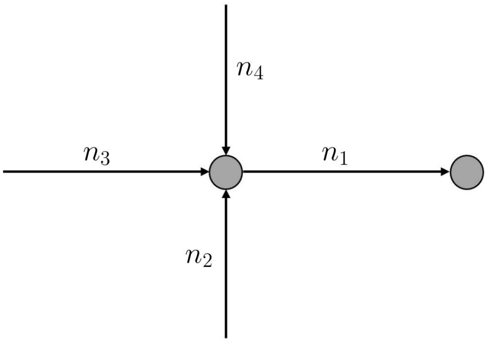


Figure E.1. Example network

vehicles on links while traffic information is collected from on-street traffic detectors and moving vehicles.

The discrete time system model is given as follows:

$$\begin{pmatrix} n_1(k+1) \\ n_2(k+1) \\ n_3(k+1) \\ n_4(k+1) \end{pmatrix} = \begin{pmatrix} n_1(k)+T\sum_{w=2}^{4}\alpha_{w,1}\frac{n_w(k)}{l_w}v_w^{free}\left[1-\left(\frac{n_w(k)}{n_w^{jam}}\right)^a\right]-T\frac{n_1(k)}{l_1}v_1^{free}\left[1-\left(\frac{n_1(k)}{n_1^{jam}}\right)^a\right]+\nu_1(k) \\ n_2(k)+d_2(k)-T\frac{n_2(k)}{l_2}v_2^{free}\left[1-\left(\frac{n_2(k)}{n_2^{jam}}\right)^a\right]+\nu_2(k) \\ n_3(k)+d_3(k)-T\frac{n_3(k)}{l_3}v_3^{free}\left[1-\left(\frac{n_3(k)}{n_3^{jam}}\right)^a\right]+\nu_3(k) \\ n_4(k)+d_4(k)-T\frac{n_4(k)}{l_4}v_4^{free}\left[1-\left(\frac{n_4(k)}{n_4^{jam}}\right)^a\right]+\nu_4(k) \end{pmatrix}, \tag{E.39}$$

where $d_w(k)$ denotes vehicle input demand appearing at the boundary of the traffic network entering to link indexed by $w = 2,\ 3,\ 4$.

Assuming that traffic detector stations are only present on link 2 and 4 the discrete time measurement equation is given as follows:

$$\begin{pmatrix} o_2^t(k) \\ o_4^t(k) \\ \left(\bar{f}_1^s\right)^{1/p}(k) \\ \left(\bar{f}_2^s\right)^{1/p}(k) \\ \left(\bar{f}_3^s\right)^{1/p}(k) \\ \left(\bar{f}_4^s\right)^{1/p}(k) \end{pmatrix} = \begin{pmatrix} & \frac{l^{PCE}}{l_2} & & \\ & & & \frac{l^{PCE}}{l_4} \\ \frac{1}{n_1^{jam}} & & & \\ & \frac{1}{n_2^{jam}} & & \\ & & \frac{1}{n_3^{jam}} & \\ & & & \frac{1}{n_4^{jam}} \end{pmatrix} \begin{pmatrix} n_1(k) \\ n_2(k) \\ n_3(k) \\ n_4(k) \end{pmatrix} + \begin{pmatrix} \zeta_2(k)^{det} \\ \zeta_4(k)^{det} \\ \zeta_1(k)^{FCD} \\ \zeta_2(k)^{FCD} \\ \zeta_3(k)^{FCD} \\ \zeta_4(k)^{FCD} \end{pmatrix}. \tag{E.40}$$

The linearization provides:

$$A(k) = \begin{pmatrix} 1-\frac{Tv_1^{free}}{l_1}\left(1-\frac{(a+1)n_1^a(k)}{(n_1^{jam})^a}\right) & \alpha_{2,1}T\frac{v_2^{free}}{l_2}\left(1-\frac{(a+1)n_2^a(k)}{(n_2^{jam})^a}\right) & \alpha_{3,1}T\frac{v_3^{free}}{l_3}\left(1-\frac{(a+1)n_3^a(k)}{(n_3^{jam})^a}\right) & \alpha_{4,1}T\frac{v_4^{free}}{l_4}\left(1-\frac{(a+1)n_4^a(k)}{(n_4^{jam})^a}\right) \\ & 1-\frac{Tv_2^{free}}{l_2}\left(1-\frac{(a+1)n_2^a(k)}{(n_2^{jam})^a}\right) & & \\ & & 1-\frac{Tv_3^{free}}{l_3}\left(1-\frac{(a+1)n_3^a(k)}{(n_3^{jam})^a}\right) & \\ & & & 1-\frac{Tv_4^{free}}{l_4}\left(1-\frac{(a+1)n_4^a(k)}{(n_4^{jam})^a}\right) \end{pmatrix}. \tag{E.41}$$

$$B_\nu(k) = I. \tag{E.42}$$

The Jacobian matrices of the measurement equation (E.22) are given as follows:

$$C = \begin{pmatrix} & \frac{l^{PCE}}{l_2} & & \\ & & & \frac{l^{PCE}}{l_4} \\ \frac{1}{n_1^{jam}} & & & \\ & \frac{1}{n_2^{jam}} & & \\ & & \frac{1}{n_3^{jam}} & \\ & & & \frac{1}{n_4^{jam}} \end{pmatrix}, \tag{E.43}$$

$$C_\zeta(k) = I. \tag{E.44}$$

The next step is to determine the uncertainty matrix ΔA modeling the ambiguity of the turning rates. According to the formula of (E.37), $M(k)$ and $E(k)$ are defined as follows:

$$M(k) = \begin{pmatrix} 0 & T\frac{v_2^{free}}{l_2}(1-\frac{(a+1)n_2^a(k)}{(n_2^{jam})^a}) & T\frac{v_3^{free}}{l_3}(1-\frac{(a+1)n_3^a(k)}{(n_3^{jam})^a}) & T\frac{v_4^{free}}{l_4}(1-\frac{(a+1)n_4^a(k)}{(n_4^{jam})^a}) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \tag{E.45}$$

$$E(k) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \delta_{2,1}\cdot\alpha_{2,1} & 0 & 0 \\ 0 & 0 & \delta_{3,1}\cdot\alpha_{3,1} & 0 \\ 0 & 0 & 0 & \delta_{4,1}\cdot\alpha_{4,1} \end{pmatrix}, \tag{E.46}$$

where $\delta_{2,1}, \delta_{3,1}$ and $\delta_{4,1}$ are uncertainty factors that weight turning rates. For example, $\delta = 0.1$ expresses that the applied nominal turning rates $\alpha_{w,z}$ of the model might vary by $\pm 10\%$.

The operation of the filter is tested based on simulation data that were generated using PTV Vissim microscopic traffic simulation software. The network shown in Fig. E.1 was implemented into PTV Vissim where the state of traffic was evaluated in 1 minute long periods. Occupancy data were collected from links 2 and 4 and two-fluid data were collected from each link. The filter estimates from these input data the number of vehicles on the links. The exact number of vehicles were also measured and were compared to the estimated number provided by the filter. The expected operation of the filter can be seen in Fig E.2, even if the performance of the model does not reach this accuracy so far.

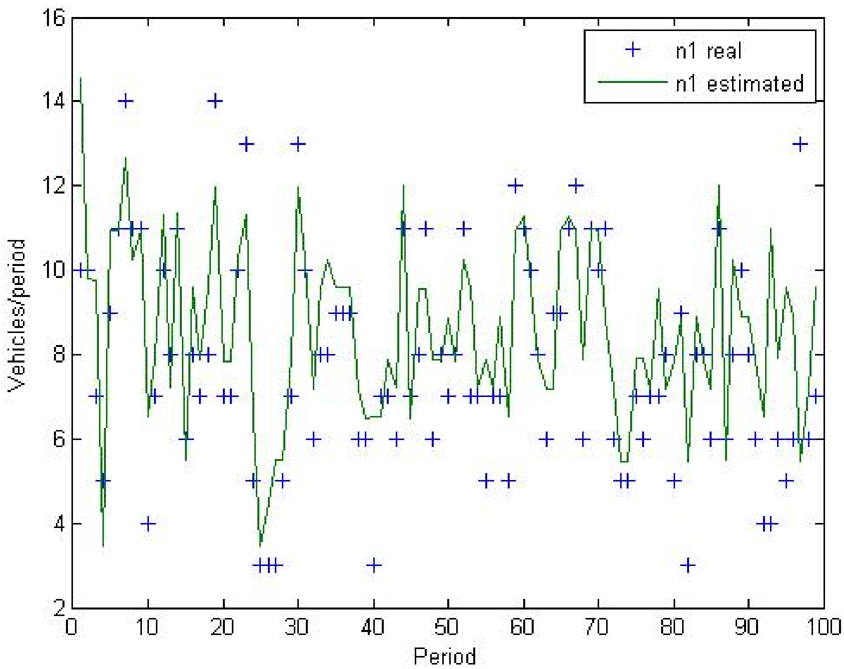


Figure E.2. Expected result of simulation (n1 is the number of vehicles on link 1)

Appendix F

Generalized Additive Model

A generalized additive model (GAM) is structured as

$$g(\mathrm{E}(Y|X_1,\ X_2,\cdots,\ X_m)) = \beta_0 + s_1(X_1) + s_2(X_2) + \cdots + s_m(X_m), \tag{F.1}$$

where Y is the response variable to some predictor variables, X_i. $\mathrm{E}(Y|X_1,\ X_2,\cdots,\ X_m)$ denotes the expected value of Y. g is a link function which relate $\mathrm{E}(Y|X_1,\ X_2,\cdots,\ X_m)$ to X_i. The functions s_i are smooth functions [Hastie and Tibshirani 1990]. GAM is a type of Generalized Linear Model (GLM) in which the linear dependent response variable is linearly dependent on some unknown smooth functions of independent predictor variables. GAM is a free and flexible statistical model, which can be used to solve nonlinear regression problems.

Hastie and Tibshirani [1990] invented GAM by combining the properties of GLM and additive models (AM). Just like in the GLM model, link functions have many commonly used types, which is why the model called "generalized". The smooth function can be parametric or non-parametric, which can be a linear function, spline function or a local regression smooth function. Its non-parametric form makes the model very flexible and reveals the nonlinear effects of the independent variables. However, if the relationship of each predictor variable is fitted with non-parametric fitting, problems such as large amount of calculation and over-fitting may occur. It is easier to explain the relationship of the predictor variables to the parametric form, so semi-parametric generalized additive models appear, the form of which is:

$$g(\mathrm{E}(Y|X_1,\ X_2,\cdots,\ X_m)) = \beta_0 + \boldsymbol{X\beta} + \sum s_i(X_i), \tag{F.2}$$

where, $\boldsymbol{X\beta}$ denotes linear combination. $\boldsymbol{\beta}$ is the unknown parameters. The matrix of predictor variables $\boldsymbol{X}$ works as the coefficients. When all the smooth functions are linear function, the GAM yield to GLM. When the link function is identity function, $g(\mu) = \mu$, and smooth functions are non-parametric, the GAM yield to AM.

Appendix G

Requirements for Road Traffic Signal Controllers

Road traffic controller constitutes a safety critical system which also means that relevant technical and legal rules are clearly determined by international standards as well as by national legislation. In case of new technological concepts for traffic light, all rules must be reviewed and one has to prove their feasibility. Accordingly, the wireless traffic signal controller with distributed system architecture must also ensure the fulfillment of all relevant requirements.

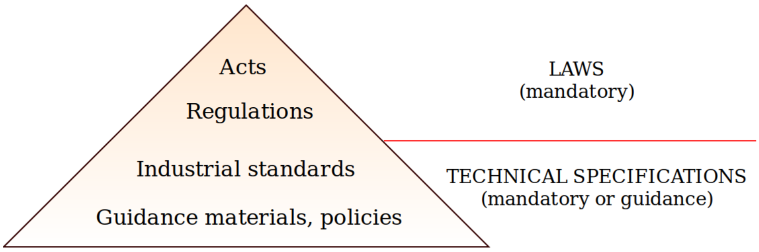


Figure G.1. The hierarchy of the technical legislation (also applicable for traffic signal controller design)

The basic hierarchy of technical legislation is depicted by Fig. G.1. The pyramid is nearly identical in all countries. On the one hand, the strongest legislative measures, the laws, contain the general rules for traffic signal control system and determines the framework of operation on national level, i.e. this is valid everywhere in the country. On the other hand, technical specifications encompass the detailed specific requirements for traffic lights on the basis of the industrial standards and codes. This layer is usually not mandatory *per se*. At the same time, laws frequently refer to specific technical standards making them or the parts of standards obligatory. Guidance notes or policies are typically created by road authorities or operators and are based on industrial standards partly or fully. An important difference compared to the laws is that the technical specifications are not necessarily held mandatory at all times. They are generally required by the road authority or operator for specific procurement or acquisition, e.g. the installation of a traffic signal system on freeways must always fulfill given technical specifications (note that it can be different to that of urban roads). In conclusion, the manufacturing process of traffic lights and installation on the spot (location of the poles and signal heads as well as other structural considerations) are always subject to the valid industrial standards given by several national and local obligations.

The European organization for public standards (CEN: Comité Européen de Normalisation) works for harmonized standardization creating and maintaining the European Norms (ENs). Accordingly, the EU countries fully adopt ENs or integrate them into the national standards and technical specifications. Three basic European standards hold for road traffic signalization systems specifically:

1. EN 12675:2018 Traffic signal controllers. Functional safety requirements. [CEN 2018a]

2. EN 12368:2015 Traffic control equipment. Signal heads. [CEN 2015]

3. EN 50556:2018 Road traffic signal systems. [CEN 2018b]

In the followings, only those parts of the above standards are introduced which are relevant to determine the functional safety requirements for traffic signal controllers. Moreover, only the major faults are investigated which are potentially hazardous to traffic (minor faults are defined as events causing no hazardous situation). As a basic requirement for traffic light, in the case of any major fault the system shall switch to a specific failure mode, i.e. a fail-safe functioning is ensured at all times. This failure mode is defined by the standard as „a non operational state of the traffic signal controller in which the normal operation mode is replaced with a flashing yellow or a signals off condition". The major faults can be classified based on standard EN 12675 [CEN 2018a] as follows.

- Conflicting green signals: the simultaneous display of green lights allowing conflicting traffic movements.

- Failure to display a red signal to traffic: the intended red signal is not displayed.

- Unwanted signal: unintended signal causing ambiguous traffic situation.

- Failure to display the correct signal sequence: the order and appearance of signals, displayed to traffic, differ form that are prescribed in national requirements.

- Failure in correct signal timing: the correct timing of any signal group fails.

- Failure in safety timings: critical error when any safety time setting (intergreen time or minimum green time) fails causing hazardous traffic situation.

According to the listed major faults above, it is indispensable that the wireless and distributed traffic signal controller shall fulfill all critical requirements, i.e. it must realize the same fail-safe operation as ensured by the traditional central traffic controllers.

In relation with the wireless technology, the technical specification for Signal Phase and Time and Map Data (SPaT/MAP) [Group 2015] must be also emphasized in future traffic controller design. SPaT/MAP defines standards for V2I and I2V information exchange. In relation with SPaT/MAP the ISO/TS 19091:2017 norm [CEN 2017] is also important as it defines the message, data structures, and data elements to support exchanges between the roadside equipment and vehicles.

Finally, for the applications using the 5.9 GHz DSRC in vehicular environments the guidance of [SAE International 2016] is also important: J2735_202007 specifies the Message Set Dictionary for DSRC.

Appendix H

Example for Autonomous Vehicle Testing Using Co-Simulation

Firstly, it is important to distinguish the innovation concerning the vehicle functions and the development of the entire transportation system. In the first case, the focus is on the car itself and its close environment. In the second case, however, one aims to ameliorate the performance of the entire traffic system (vehicle, infrastructure and users). Of course, the automotive industry runs intensive vehicle developments by applying very detailed vehicle model and vehicle dynamics simulations. However, these programs have limited capacity to model traffic situations. From the other side the analysis and design of the whole transport system is performed by the traffic engineering profession, typically carried out with micro- or macroscopic traffic simulation software. These software are already capable of detailed modeling of the whole traffic network, but they operate with simplified vehicle dynamics, e.g. acceleration function can be defined but detailed engine dynamics or powertrain are not included. It is observable that the transition between these two simulation methods is currently limited. It is possible to interface these programs but they are not integrated.

To overcome the above problem, similarly to the ”Hardware-In-the-Loop” method, the so-called ”Vehicle-in-the-Loop” (ViL) test simulation framework is designed. The essence of this is to create an environment in which a realistic traffic simulation software (with microscopic traffic model) can simulate one or more real (autonomous) test vehicles in real time while other vehicles are implemented as virtual traffic. In this system it is possible to test real-world cars together with their autonomous functions and capabilities by displaying real vehicle dynamics in the simulator, while the traffic around is virtually generated. In the ViL test environment, arbitrary circumstances can be realized. One can create vehicles, traffic control objects, pedestrians, accident, or simulate situations with poor visibility where vehicle camera is ineffective and are forced to rely on V2V communication. As shown by Fig. H.1, beyond the simulation of the virtual environment around a real vehicle, also communication with virtual systems (other vehicles or the traffic management system) can be realized.

All this makes it possible to carry out cost efficient and safe tests, e.g. on a plain test site without objects.

In order to build a simulation environment, it is necessary to define the main goals. The interface to be created is determined according to the requirements of these simulation targets. The data and control parameters between the test vehicle and the traffic simulation suites are accordingly to be defined. Some of these are trivial, such as the actual vehicle velocity or the vehicle position, but other parameters will depend on the tasks planned to be accomplished. The tasks that can be created in the ViL test environment are classified in the following main categories:

- keeping and changing vehicle speeds according to environmental impact;

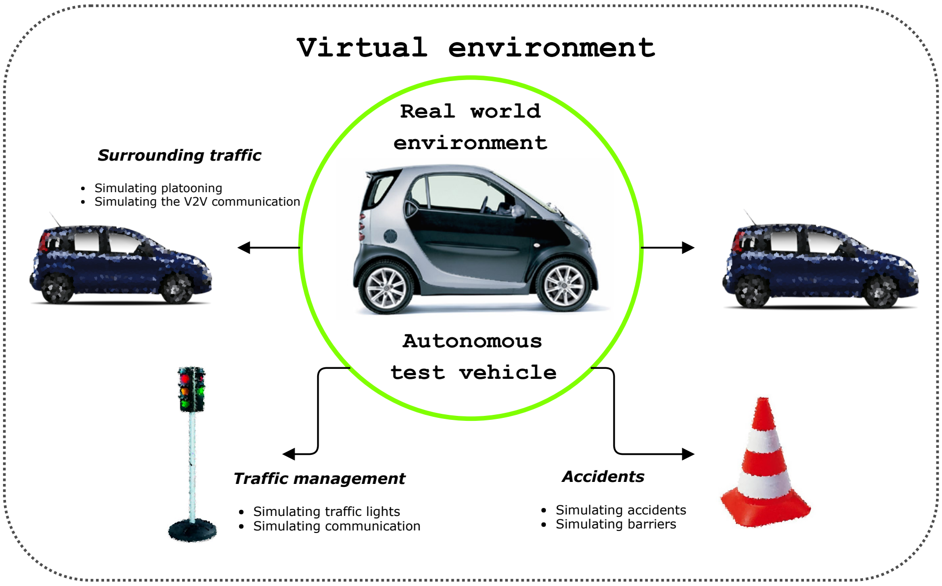


Figure H.1. Simulation possibilities

- test vehicle operation at traffic light;
- emergency brake simulation by virtual interference;
- keeping small headway distance compared to the virtual vehicle ahead;
- platooning situation with two or more real autonomous vehicles.

Of course, each category covers further possible sub-tasks.

In the sequel, the main basic building blocks (hardware, software) and the functioning of the whole ViL test environment are introduced.

Road traffic modeling is commonly used in traffic engineering practice to assist design and validation of newly developed control strategies. Users may choose among many different commercial or open source road traffic simulators. Each of them has advantages or drawbacks depending on the individual demands of the user.

For the purpose of autonomous vehicle tests SUMO software was chosen due to the flexible programming possibilities. Using CAN based messaging for creating connection between the test vehicle and the simulation software is an obvious solution as it is the most frequently used communication bus networks in automotive industry. In most cases, any parameter related to the vehicle can be found on the CAN bus network. This technology enables communication between the elements of the vehicle, e.g. sensors, ECUs. Thanks to the well-defined CAN network, one can easily write databases that can be interpreted and processed in a Matlab environment by using the standard database of CAN (dbc file). The messages contained in the dbc files can be selected separately easily and handled as a distinct parameter using the Matlab's internal CAN communication module. In this way receiving and transmitting of signals to the network can be realized straightforwardly. In order to be able to read these messages from the CAN bus, the Vector CANCaseXL was applied. This device with multi-channel communication acts as a node on the network that receives and sends all messages with any identifier. Vector CANCaseXL can be attached to the car's CAN bus easily.

The Faculty of Transportation Engineering and Vehicle Engineering of BME (Budapest University of Technology and Economics) owns a Smart Fortwo city car. This test car was modified by the university researchers to make it fully capable for "by wire" operation with fail-safe architecture [Tihanyi and Szalay 2017]. The system architecture of the Smart is shown by Fig. H.2. All execution actuators were modified or at least interrupted using a central Autobox device for realizing the control. (AutoBox is an efficient tool for using dSPACE real-time system to carry out in-vehicle control experiments such as tests for autonomous functionalities.) Throttle had originally a potentiometer based control with analog signal that was interrupted and lead through the device. Steering was solved by additional BLDC motor actuator driven by an external servo drive with torque control fed also from Autobox device. For the gear control the original gear button's parallel interface was substituted by a self developed ECU that can communicate with the central controller via CAN bus. Brake system required high modifications, a linear actuator was built in in order to pull the brake pedal via bowden. Brake system also

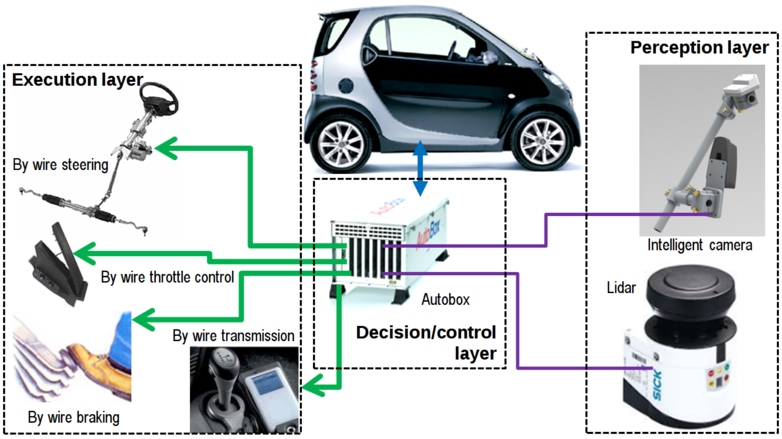


Figure H.2. The system architecture of the test vehicle for autonomous driving tests

includes a spring actuator in case of fail or emergency situations. The vehicle was also equipped with environment sensors (LIDAR and camera) with the capability of lane detection and object detections, however for the ViL tests described in the sequel the LIDAR sensors are not used intentionally. At the same time, LIDAR CAN messages are reproduced by external computer (by Matlab in a simple laptop) to input them for the central controller of the Smart. The vehicle for the time has several automated functions such as lane keeping, lane change and automatic cruise control. Moreover, a high level autonomous function, the traffic jam assist, is also realized combining the existing fundamental driver assistance components.

The whole ViL test environment for autonomous car is depicted by Fig. H.3. The environment can be divided into two main parts:

1. the test vehicle with own computer logic for autonomous functions and its physical environment (where it moves);

2. the computer realizing the ViL test environment (simulation software).

The elements of the system devices are connected via CAN communication channels, i.e. Vector CANCaseXL controls the messages specified for the CAN bus. CAN messages are processed by Matlab and then transferred to the SUMO traffic simulator via the TraCI interface. Thus, a complete communication channel is created between the Smart vehicle (central controller) and the SUMO traffic simulator.

As a real-world demonstration of the developed ViL simulation environment one of the autonomous functions of the Smart, the traffic jam assist, was tested. The test environment is depicted in Fig. H.4. The area that serves as a car park in the university campus of BME was modeled in the traffic simulator. The test scenario simulated the motion of two cars on a specific route (Fig. H.5), i.e.

1. the virtual car (blue) is going ahead and

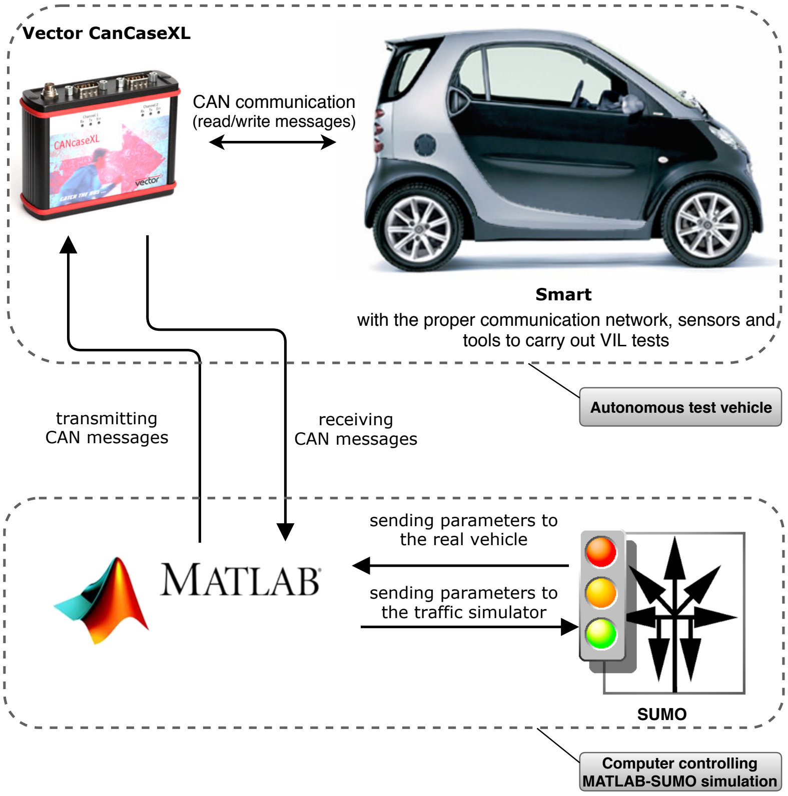


Figure H.3. The whole test environment for autonomous car

2. the real autonomous vehicle (green) follows it.

During the experiment, the acceleration and braking maneuvers of the real vehicle was directly fitted into the traffic simulator while the motion of the virtual car was simulated by SUMO.

At the beginning of the test, the two vehicles started from different positions (Fig. H.5). The virtual vehicle moved at constant speed, slower than the speed of the real car. The real vehicle accelerated up to 10 km/h and kept this speed constant as long as there was no obstacle in front it. The braking maneuver occurred when the real vehicle reached the minimum safety distance (2 meters) allowed to be compared to the car ahead. This distance value was transferred from the SUMO simulator to the controller via the CAN communication network. Practically, the CAN messages of the LIDAR and camera signals were overwritten by the traffic simulator's data. When the distance dropped below the safety distance the traffic jam assist logic stopped the real vehicle.

Obviously, the effectiveness of the test depends on the proper online transfer of speed and distance values. On the one hand, the accuracy of the speed transmission provides the precise locomotion of the vehicle in the virtual reality. On the other hand, the distance value coming from the simulator means the input signal to the real car's controller. Hence, the success of the whole test is influenced by the reliable data transmission. For later evaluation all parameters sent and received during the simulation were logged. Fig. H.6 shows the accuracy of the speed transfer. It can be seen that, as the result of communication at 5 Hz, a slight delay is present between the received CAN signal and the current speed value in SUMO. However, this delay is negligible, so that it did not affect the effectiveness of this test. Moreover, one can apply higher frequency of communication between the real car and the simulator (this is a question of computational resource). Fig. H.7 shows the change of the distance between the two vehicles and the speed variation of the real autonomous test car. It is observable (according to the scenario) that after the start of the real vehicle the headway distance to the virtual car begins

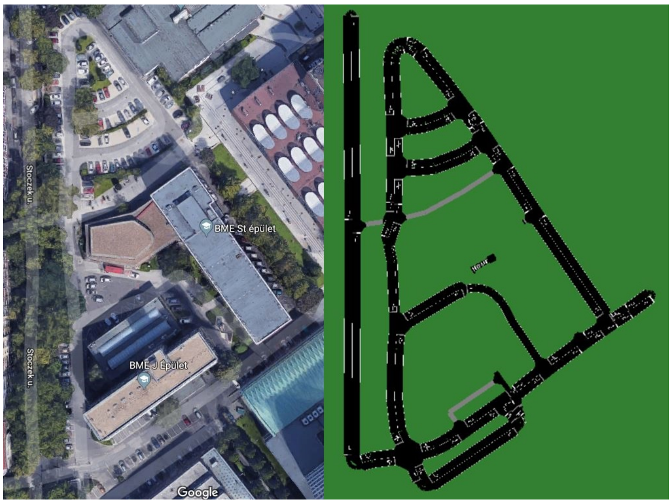


Figure H.4. The location of the test (the campus of Budapest University of Technology and Economics, GPS: 47.478488, 19.056098)

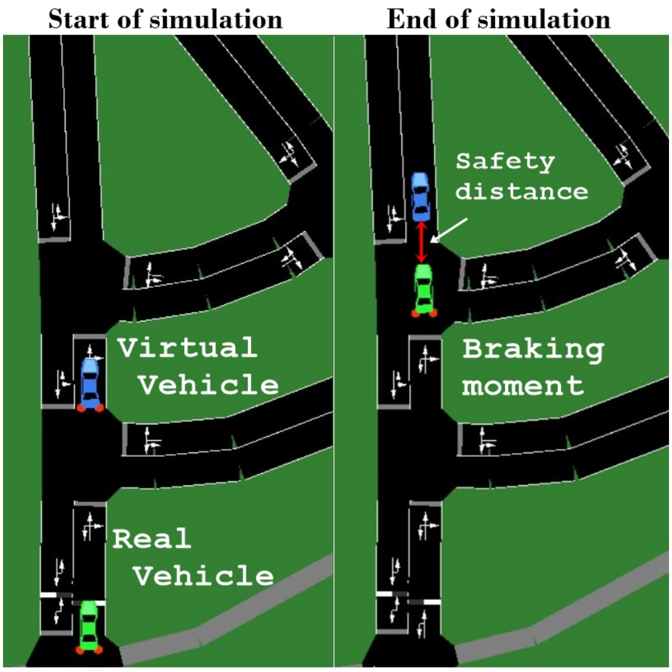


Figure H.5. The real-world test scenario to decrease. Simultaneously the real car holds a constant speed until the safety distance limit is attained. Then the test car breaks down immediately.

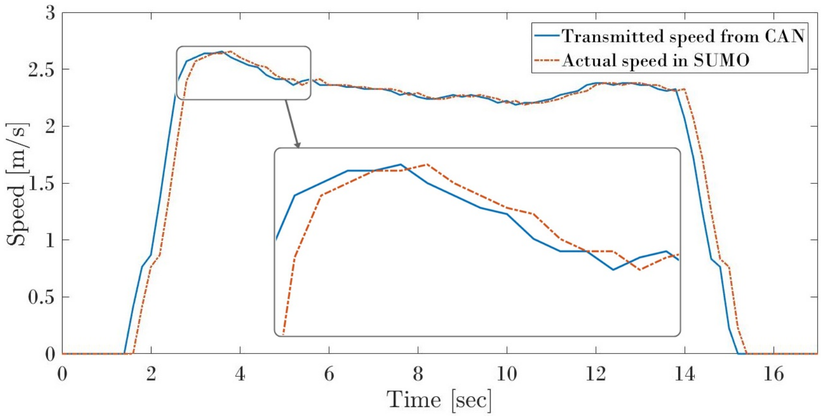


Figure H.6. The speed transmission between the vehicle and the simulator

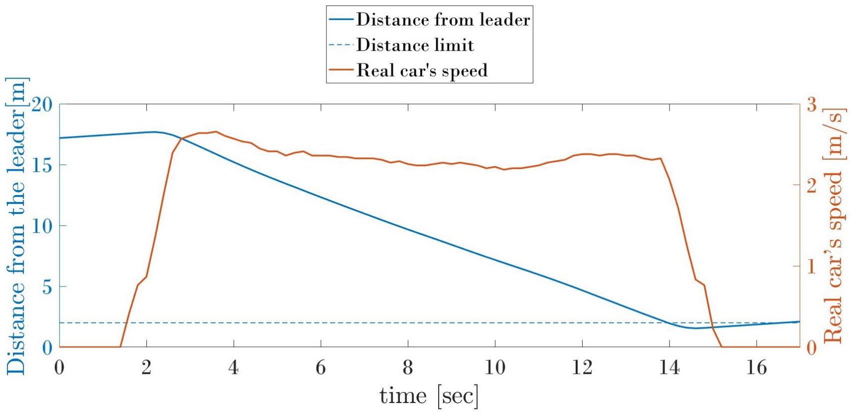


Figure H.7. The braking moment in the simulation

The tools used during the experiment and the test site are shown by Fig. H.8. Concerning the test vehicle, not all autonomous functions of the Smart were used. The self-driving controller of the test car handled the throttle and the break pedals as well as the gearbox, but steering was controlled manually. Simulations were performed on a PC with Intel Core i5-6300HQ 2.3-GHz CPU (4 cores) and 12 GB of RAM.

Figure H.8. The test environment during the experiment

Appendix I

Simulation Environment for the Digital Twin of the Traffic Control System

For the realization of future digital twin of the Smart City Zone at ZalaZONE Proving Ground, two simulation environments have been carried out in SUMO and PTV VISSIM, respectively (see Figs. I.1 and I.2). The requirements for the digital twin realization are itemized below:

- display running simulation;
- display simulation time;
- display traffic signal plans;
- display current traffic signal plan in real time;
- display information about the traffic network, junctions, etc.;
- adjust simulation time;
- adjust traffic flow;
- adjust routes;
- adjust vehicle categories/category ratios,
- adjust traffic signal plans:
  - switch signals on/off;
  - switch between programs;
  - stretch/shrink signal phases, cycle lengths;
  - switch between fixed and actuated programs;

The complete simulation environment is illustrated by Fig. I.3.

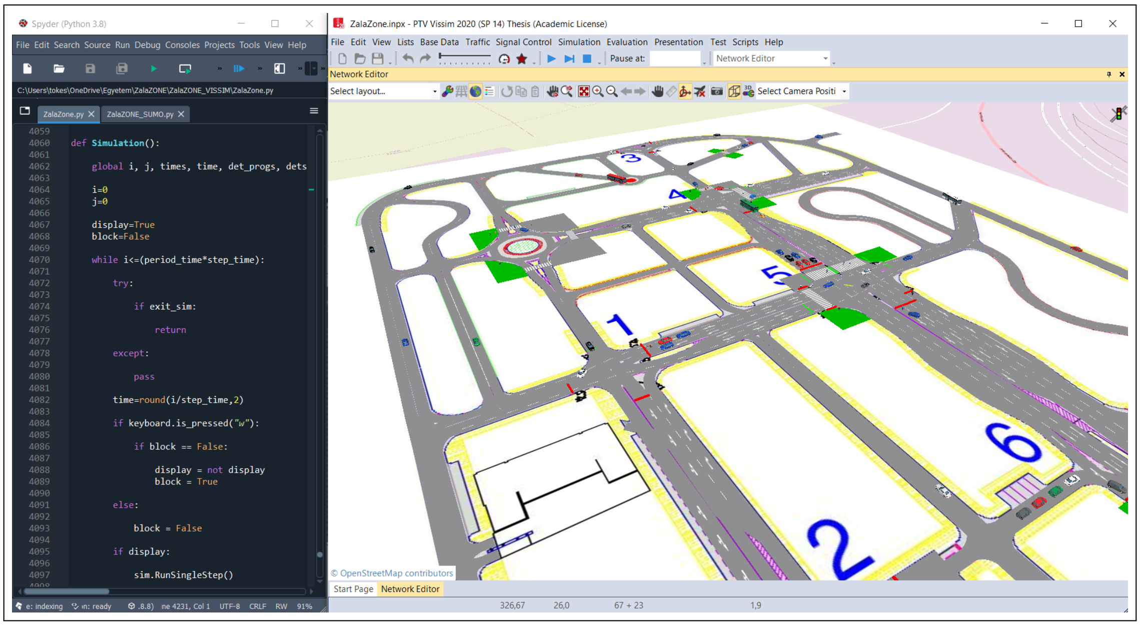


Figure I.1. PTV VISSIM control program and simulation

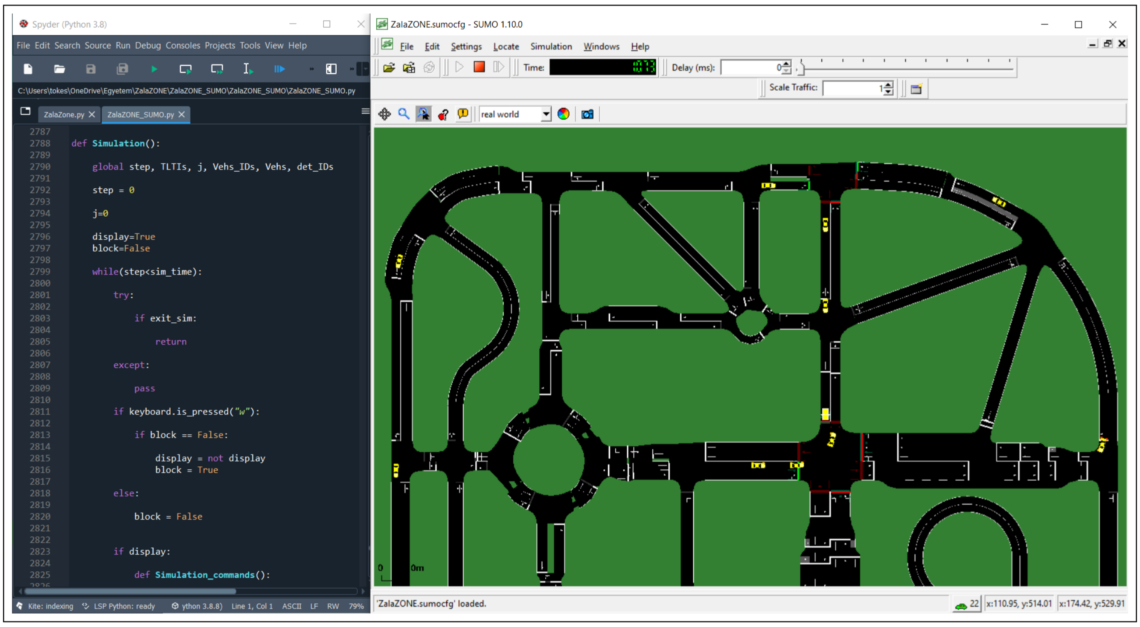


Figure I.2. SUMO control program and simulation

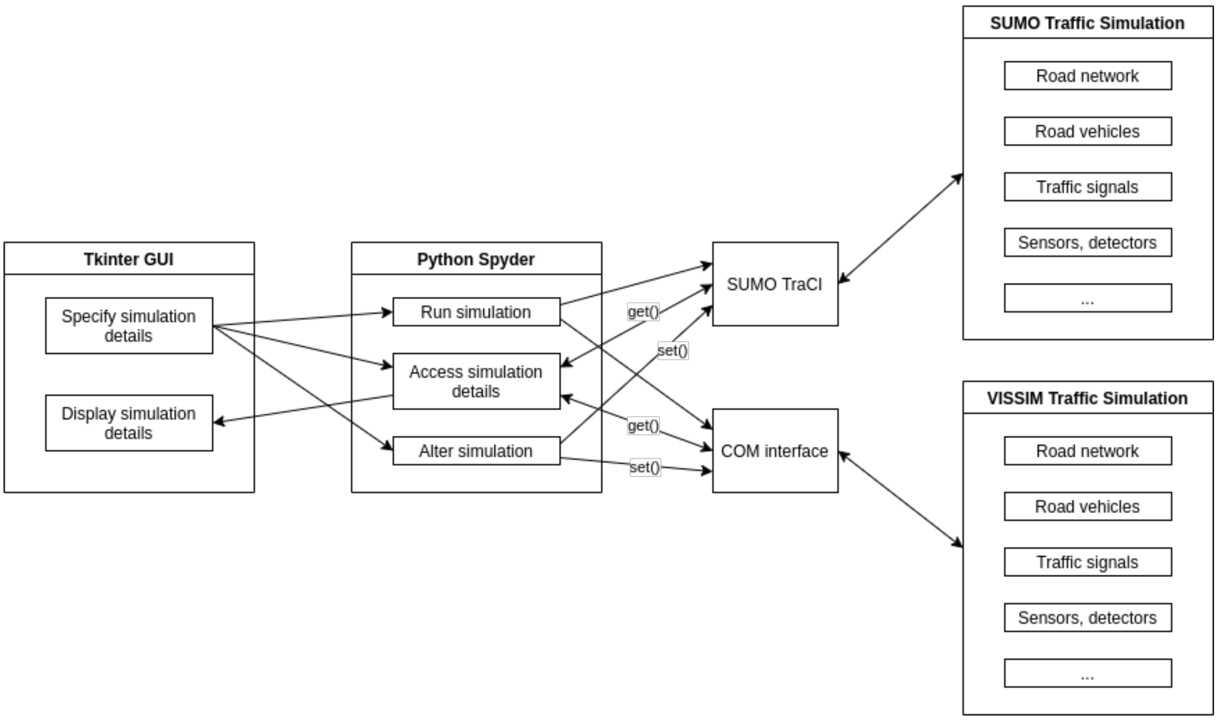


Figure I.3. Simulation environment for the digital twin of the ZalaZONE traffic control center

Publications of the Author

C. Bartolini, T. Tettamanti, and I. Varga. Critical features of autonomous road transport from the perspective of technological regulation and law. *Transportation Research Procedia*, 27:791 – 798, 2017. doi: 10.1016/j.trpro.2017.12.002. 20th EURO Working Group on Transportation Meeting, EWGT 2017, 4-6. September 2017, Budapest, Hungary.

Zs. Bede, A. Csikós, M.T. Horváth, T. Tettamanti, and I. Varga. *Közúti forgalommodellezési gyakorlatok, 4. kiadás 3. és 4. fejezetek a Vissim 9 és 10-es verzió-hoz aktualizálva.* BME Közlekedés- és Járműirányítási Tanszék, 2020. TAMOP-4.1.1.C-12/1/KONV-2012-0002.

A. Csikós, T. Tettamanti, and I. Varga. Modeling of emission in urban traffic networks. Technical Report Research Report no. SCL-001/2014, Systems and Control Laboratory, Research Institute for Computer Sciences and Automation, Hungarian Academy of Sciences, 2014.

A. Csikós, T. Tettamanti, and I. Varga. Nonlinear gating control for urban road traffic network using the network fundamental diagram. *Journal of Advanced Transportation*, 49(5):597–615, 2015a. ISSN 2042-3195. doi: 10.1002/atr.1291.

A. Csikós, Zs. J. Viharos, K.B. Kis, T. Tettamanti, and I. Varga. Traffic speed prediction method for urban networks - an ANN approach. In *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pages 102–108, June 2015b. doi: 10.1109/MTITS.2015.7223243.

Cs. Csiszár, D. Földes, and T. Tettamanti. Mobilitási szolgáltatások komplex automatizálási szintjei. *Közlekedéstud. Szemle*, LXIX(4):33–48, 2019. doi: 10.24228/KTSZ.2019.4.3.

S. Duleba, T. Tettamanti, Á. Nyerges, and Z. Szalay. Ranking the key areas for autonomous proving ground development using pareto analytic hierarchy process. *IEEE Access*, 9:51214–51230, 2021. doi: 10.1109/ACCESS.2021.3064448.

D. Esztergár-Kiss, Y. Shulha, A. Aba, and T. Tettamanti. Promoting sustainable mode choice for commuting supported by persuasive strategies. *Sustainable Cities and Society*, 74:103264, 2021. ISSN 2210-6707. doi: h10.1016/j.scs.2021.103264.

X. Fang and T. Tettamanti. Traffic congestion phenomena when motorway meets urban road network. In *2021 IEEE 25th International Conference on Intelligent Engineering Systems (INES)*, pages 000025–000030, 2021a. doi: 10.1109/INES52918.2021.9512920.

X. Fang and T. Tettamanti. Change in microscopic traffic simulation practice with respect to the emerging automated driving technology. *Periodica Polytechnica ser. Civil Engineering*, 2021b.

X. Fang, T. Tettamanti, and A.C. Piazzi. Online calibration of microscopic road traffic simulator. In *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 275–280, 2020. doi: 10.1109/SAMI48414.2020.9108744.

D. Földes, Cs. Csiszár, and T. Tettamanti. Automation levels of mobility services. *Journal of Transportation Engineering, Part A: Systems*, 147(5):04021021, 2021. doi: 10.1061/JTEPBS.0000519.

K. Gangel, Z. Hamar, A. Háry, Á. Horváth, G. Jandó, B. Könyves, D. Panker, K. Pintér, M. Pataki, M. Szalai, Zs. Szalay, T. Tettamanti, V. Tihanyi, B. Tóth, B. Varga, and Zs.J. Viharos. Modelling the ZalaZONE Proving Ground: a benchmark of State-of-the-art Automotive Simulators PreScan, IPG CarMaker, and VTD Vires. *Acta Technica Jaurinensis*, 14 (4):488–507, Nov. 2021. doi: 10.14513/actatechjaur.00606.

M. Gressai. and T. Tettamanti. Turning rate estimation in roundabouts: Analysis and validation of different estimation methods. In *Proceedings of the 7th International Conference on Vehicle Technology and Intelligent Transport Systems - VEHITS*, pages 65–71. INSTICC, SciTePress, 2021. ISBN 978-989-758-513-5. doi: 10.5220/0010405700650071.

M. Gressai, B. Varga, T. Tettamanti, and I. Varga. Investigating the impacts of urban speed limit reduction through microscopic traffic simulation. *Communications in Transportation Research*, 1:100018, 2021. ISSN 2772-4247. doi: 10.1016/j.commtr.2021.100018.

M.T. Horváth and T. Tettamanti. Real-time queue length estimation applying shockwave theory at urban signalized intersections. *Periodica Polytechnica Civil Engineering*, 65(4):1153–1161, 2021a. doi: 10.3311/PPci.17022.

M.T. Horváth and T. Tettamanti. Robust vehicle count estimation on urban signalized links. *Measurement*, 181:109581, 2021b. ISSN 0263-2241. doi: 10.1016/j.measurement.2021.109581.

M.T. Horváth, T. Tettamanti, and I. Varga. Az autonóm járműforgalom modellezhetősége mikroszkopikus forgalomszimulációs szoftverben. *Közlekedéstud. Szemle*, LXVIII(2):34–44, 2018. doi: 10.24228/KTS.2018.2.3.

M.T. Horváth, Q. Lu, T. Tettamanti, Á. Török, and Zs. Szalay. Vehicle-In-The-Loop (VIL) and Scenario-In-The-Loop (SCIL) automotive simulation concepts from the perspectives of traffic simulation and traffic control. *Transport and Telecommunication Journal*, 2(20):153–161, 2019a. doi: 10.2478/ttj-2019-0014.

M.T. Horváth, T. Tettamanti, B. Varga, and Zs. Szalay. The Scenario-in-the-Loop (SciL) automotive simulation concept and its realisation principles for traffic control. In *hEART 2019 - 8th Symposium of the European Association for Research in Transportation.* European Association for Research in Transportation, 2019b. Paper 62.

M.T. Horváth, T. Tettamanti, and I. Varga. (Smart CPS) Multiobjective dynamic routing with predefined stops for automated vehicles. *International Journal of Computer Integrated Manufacturing*, 32(4-5):396–405, 2019c. doi: 10.1080/0951192X.2018.1535197.

R. Hujber, T. Tettamanti, and I. Varga. Intelligent road traffic light system with distributed control, 2019. Registration number: 5034, NSZO: G08G 1/095 , Case Number: U1800160/10.

A. Kovács, Á. Leelőssy, T. Tettamanti, D. Esztergár-Kiss, R. Mészáros, and I. Lagzi. Coupling traffic originated urban air pollution estimation with an atmospheric chemistry model. *Urban Climate*, 37:100868, 2021. ISSN 2212-0955. doi: 10.1016/j.uclim.2021.100868.

B. Kővári, T. Tettamanti, and T. Bécsi. Deep reinforcement learning based approach for traffic signal control. *Transportation Research Procedia*, 2021. 24th EURO Working Group on Transportation Meeting, EWGT 2021, 8-10 September 2021, Aveiro, Portugal.

H. Lengyel, T. Tettamanti, and Zs. Szalay. Conflicts of automated driving with conventional traffic infrastructure. *IEEE Access*, 8:163280–163297, 2020.

H. Li, V.P. Makkapati, D. Nalic, A. Eichberger, X. Fang, and T. Tettamanti. A real-time co-simulation framework for virtual test and validation on a high dynamic vehicle test bed. In *2021 IEEE Intelligent Vehicles Symposium (IV21)*. IEEE, 2021.

Q. Lu and T. Tettamanti. Impacts of autonomous vehicles on the urban fundamental diagram. In *5th International Conference on Road and Rail Infrastructure, CETRA 2018,*, pages 1265–1271, 17-19. May 2018.

Q. Lu and T. Tettamanti. Traffic control scheme for social optimum traffic assignment with dynamic route pricing for automated vehicles. *Periodica Polytechnica Transportation Engineering*, 49(3):301–307, 2021a. doi: 10.3311/PPtr.18608.

Q. Lu and T. Tettamanti. Impacts of connected and automated vehicles on freeway with increased speed limit. *International Journal of Simulation Modelling (IJSIMM)*, 20(3), 2021b. doi: 10.2507/IJSIMM20-3-556.

Q. Lu, T. Tettamanti, and D. Hörcher. Implications of user and system optimum based traffic control considering autonomous fleets. In *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, pages 1–5, 2019. doi: 10.1109/ICCVE45908.2019.8965210.

Q. Lu, T. Tettamanti, D. Hörcher, and I. Varga. The impact of autonomous vehicles on urban traffic network capacity: an experimental analysis by microscopic traffic simulation. *Transportation Letters*, 12(8):540–549, 2020. doi: 10.1080/19427867.2019.1662561.

T. Luspay, T. Tettamanti, and I. Varga. *Forgalomirányítás, Közúti járműforgalom modellezése és irányítása.* Typotex Kiadó, 2011. ISBN 978-963-279-665-9.

T. Tettamanti M. Szalai. Kevert valóság fejlesztési környezet autonóm járművek számára. *Közlekedéstud. Szemle*, LXXI(3):17–28, 2021. doi: 10.24228/KTSZ.2021.3.2. ISSN 0023-4362.

B. Maximcsuk, Q. Lu, and T. Tettamanti. Determining maximum achievable flows of autonomous vehicles based on macroscopic fundamental diagrams. *Perner's Contacts, Special Issue: "36th International Colloquium on Advanced Manufacturing and Repair Technologies in Vehicle Industry"*, pages 192–199, 2019.

Nokia Solutions and Networks OY, H. Demeter, N. Vékony, T. Tettamanti, I. Varga, and Á. Ludvig. Determining travel information (method and system for real-time travel time calculation in road traffic network using radio signaling data), 2014. Invention Publication No.: WO 2014/023339 A1.

T. Ormándi, B. Varga, and T. Tettamanti. Autonóm járművek komplex virtuális tesztkörnyezetének fejlesztése. In *Innováció és Fenntartható Felszíni Közlekedés Konferencia (IFFK), Budapest, Hungary, 2020.10.28-2020.10.30*, 2020.

T. Ormándi, B. Varga, and T. Tettamanti. Distributed intersection control based on cooperative awareness messages. In *5th Conference on Control and Fault Tolerant Systems (SysTol)*, pages 323–328, 2021a. doi: 10.1109/SysTol52990.2021.9595376.

T. Ormándi, B. Varga, and T. Tettamanti. Estimating vehicle suspension characteristics for digital twin creation with genetic algorithm. *Periodica Polytechnica Transportation Engineering*, 49(3):231–241, 2021b. doi: 10.3311/PPtr.18576.

A.C. Piazzi and T. Tettamanti. LSTM Approach for spatial extension of traffic sensor points in urban road network. In *hEART 2019 - 8th Symposium of the European Association for Research in Transportation*. European Association for Research in Transportation, 2019a. Paper 81.

A.C. Piazzi and T. Tettamanti. Deep learning approach for spatial extension of traffic sensor points in urban road network. In *2019 IEEE 13th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pages 81–86, 2019b. doi: 10.1109/SACI46893.2019.9111522.

J. Polgár, T. Tettamanti, and I. Varga. Passenger number dependent traffic control in signalized intersections. *Periodica Polytechnica ser. Civil Engineering*, 57(2):201–210, 2013. doi: 10.3311/PPci.7175.

J. Polgár, T. Tettamanti, and I. Varga. Utasszám alapú forgalomirányítás városi jelzőlámpás csomópontban. *Közlekedéstud. Szemle*, 61(6):30–37, 2011a.

J. Polgár, T. Tettamanti, and I. Varga. Autóbusz előnybiztosító logika megvalósítása forgalomfüggő irányítású csomópontban. *Városi Közlekedés*, 51:174–181, 2011b.

V. Potó, J.M. Lógó, T. Tettamanti, Á. Barsi, and N. Krausz. Térképi formátumok értékelése az önvezetés szempontjából. In *Az elmélet és a gyakorlat találkozása a térinformatikában XI.: Theory meets practice in GIS*, pages 207–215, 2020.

M. Szalai, B. Varga, T. Tettamanti, and V. Tihanyi. Mixed reality test environment for autonomous cars using Unity 3D and SUMO. In *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 73–78, 2020.

Zs. Szalay, D. Esztergár-Kiss, T. Tettamanti, P. Gáspár, and I. Varga. Recar: Hungarian research center for autonomous road vehicles is on the way. *ERCIM News, In: Special Theme: Autonomous Vehicles*, 109:27–29, 2016.

Zs. Szalay, T. Tettamanti, D. Esztergár-Kiss, I. Varga, and C. Bartolini. Development of a test track for driverless cars: Vehicle design, track configuration, and liability considerations. *Periodica Polytechnica ser. Transportation Engineering*, 46(1):29–35, 2018. doi: 10.3311/PPtr.10753.

G. Tamaskovics, T. Tettamanti, and I. Varga. Az intelligens jelzőfej koncepciója: vezeték nélküli, elosztott rendszerű jelzőlámpás forgalomirányítás. *Közlekedéstud. Szemle*, 6(LXVI):45–54, 2016.

T. Tettamanti. *Advanced Methods for Measurement and Control in Urban Road Traffic Networks*. PhD thesis, Budapest University of Technology and Economics, Dept. of Control for Transportation and Vehicle Systems, Budapest, 2013.

T. Tettamanti. Innovatív közlekedési kutatás - korszerű mérési és irányítási módszerek városi közúti közlekedési hálózatban. *Közlekedéstud. Szemle*, LXIV(1):41–48, 2014.

T. Tettamanti. Wireless traffic signal controller with distributed control system architecture. *Periodica Polytechnica ser. Civil Engineering*, 63(3):918–925, 2019. doi: 10.3311/PPci.13974.

T. Tettamanti. Vezeték nélküli, napelemes jelzőlámpa. *Élet és Tudomány*, LXXV(39), 2020.

T. Tettamanti. Advanced methods for turning rate estimation in roundabouts. *Measurement*, 181:109676, 2021. ISSN 0263-2241. doi: 10.1016/j.measurement.2021.109676.

T. Tettamanti and M.T. Horváth. *A practical manual for Vissim-COM programming in Matlab and Python – 5th edition for Vissim version 2020 and 2021.* Budapest University of Technology and Economics, Dept. for Control of Transportation and Vehicle Systems, 5th edition, 2021.

T. Tettamanti and I. Varga. Elosztott közúti forgalomirányító rendszer. *Városi Közlekedés*, XLIX(6):338–341, 2009a.

T. Tettamanti and I. Varga. Városi forgalomirányító rendszer prediktív szabályozással. *Városi Közlekedés*, XLIX(3):131–135, 2009b.

T. Tettamanti and I. Varga. Traffic control designing using model predictive control in a high congestion traffic area. *Periodica Polytechnica ser. Transp. Eng.*, 37(1-2):3–8, 2009c. doi: 10.3311/pp.tr.2009-1-2.01.

T. Tettamanti and I. Varga. Distributed traffic control system based on model predictive control. *Periodica Polytechnica ser. Civil Eng.*, 54(1):3–9, 2010. doi: 10.3311/pp.ci.2010-1.01.

T. Tettamanti and I. Varga. Robusztus városi forgalomirányítás. *Városi Közlekedés*, LI(1-2): 80–84, 2011.

T. Tettamanti and I. Varga. Development of road traffic control by using integrated VISSIM-MATLAB simulation environment. *Periodica Polytechnica ser. Civil Engineering*, 56(1):43–49, 2012. doi: 10.3311/pp.ci.2012-1.05.

T. Tettamanti and I. Varga. Mobile phone location area based traffic flow estimation in urban road traffic. *Columbia International Publishing Advances in Civil and Environmental Engineering*, 1(1):1–15, 2014.

T. Tettamanti and I. Varga. A jövő intelligens járművei és az infokommunikáció hatása. *Híradástechnika*, LXXI:59–63, 2016. „Smart City a célkeresztben" különszám.

T. Tettamanti and I. Varga. Az autonóm járművek forgalmi hatásai: a jármű- és forgalomirányítás kihívásai. *Közlekedéstud. Szemle*, LXIX(1):35–41, 2019. doi: 10.24228/KTSZ.2019.1.4.

T. Tettamanti, I. Varga, and T. Péni. MPC in urban traffic management. In *Model predictive control*, pages 251–268. Open Access Books, IntechOpen, 2010. doi: 10.5772/9922.

T. Tettamanti, T. Bécsi, and I. Varga. A közúti forgalom becslésére felhasználható mérési adatok és együttes alkalmazhatóságuk. *Közlekedéstud. Szemle*, LXIV(3):29–42, 2014a.

T. Tettamanti, M.T. Horváth, and I. Varga. Road traffic measurement and related data fusion methodology for traffic estimation. *Transport and Telecommunication*, 15(4):269–279, 2014b. doi: 10.2478/ttj-2014-0023.

T. Tettamanti, A. Csikós, and I. Varga. Macroscopic modeling and control of emission in urban road traffic networks. *Transport*, 30(2):152, 161 2015a. doi: 10.3846/16484142.2015.1046137.

T. Tettamanti, A. Csikós, I. Varga, and A. Eleőd. Iterative calibration of Vissim simulator based on genetic algorithm. *Acta Technica Jaurinensis*, 8(2):145–152, 2015b. doi: 10.3311/PPtr.7685.

T. Tettamanti, Á.Z. Milacski, A. Lőrincz, and I. Varga. Iterative calibration method for microscopic road traffic simulators. *Periodica Polytechnica ser. Transp. Eng.*, 43(2):87–91, 2015c. doi: 10.3311/PPtr.7685.

T. Tettamanti, M.T. Horváth, and I. Varga. Közúti eljutási idő becslésének lehetősége adatfúziós technikával városi úthálózaton. *Közlekedéstud. Szemle*, LXVI(3):46–56, 2016a.

T. Tettamanti, I. Varga, and A. Csikós. *Közúti mérések, Eszközök és módszerek a közúti járműforgalom megfigyelésére.* Typotex Kiadó, 2016b. ISBN 978-963-279-916-2.

T. Tettamanti, I. Varga, and Zs. Szalay. Impacts of autonomous cars from a traffic engineering perspective. *Periodica Polytechnica ser. Transp. Eng.*, 44(4):244–250, 2016c. doi: 10.3311/PPtr.9464.

T. Tettamanti, M.T. Horváth, and I. Varga. Nonlinear traffic modeling for urban road network and related robust state estimation. In *Proceedings of the 9th European Nonlinear Dynamics Conference*, page ID 247, 2017a. ISBN 978-963-12-9168-1.

T. Tettamanti, A. Mohammadi, H. Asadi, and I. Varga. A two-level urban traffic control for autonomous vehicles to improve network-wide performance. *Transportation Research Procedia*, 27:913 – 920, 2017b. doi: 10.1016/j.trpro.2017.12.160. 20th EURO Working Group on Transportation Meeting, EWGT 2017, 4-6 September 2017, Budapest, Hungary.

T. Tettamanti, A. Csikós, K.B. Kis, Zs.J. Viharos, and I. Varga. Pattern recognition based speed forecasting methodology for urban traffic network. *Transport*, 33(4):959–970, 2018a. doi: 10.3846/16484142.2017.1352027.

T. Tettamanti, M. Szalai, S. Vass, and V. Tihanyi. Vehicle-In-the-Loop test environment for autonomous driving with microscopic traffic simulation. In *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pages 1–6, 2018b. doi: 10.1109/ICVES.2018.8519486.

T. Tettamanti, T. Luspay, and I. Varga. *Road Traffic Modeling and Simulation.* Akadémiai Kiadó, 2019a.

T. Tettamanti, Á. Török, and I. Varga. Dynamic road pricing for optimal traffic flow management by using non-linear model predictive control. *IET Intelligent Transport Systems*, 13:1139–1147(8), 2019b. doi: 10.1049/iet-its.2018.5362.

V. Tihanyi, T. Tettamanti, M. Csonthó, A. Eichberger, D. Ficzere, K. Gangel, L.B. Hörmann, M.A. Klaffenböck, C. Knauder, P. Luley, Z.F. Magosi, G. Magyar, H. Németh, J. Reckenzaun, V. Remeli, A. Rövid, M. Ruether, S. Solmaz, Z. Somogyi, G. Soós, D. Szántay, T.A. Tomaschek, P. Varga, Zs. Vincze, C. Wellershaus, and Zs. Szalay. Motorway measurement campaign to support R&D activities in the field of automated driving technologies. *Sensors*, 21(6), 2021. ISSN 1424-8220. doi: 10.3390/s21062169.

R.P. Tóth, M. Szalai, and T. Tettamanti. A HU-GO elektronikus útdíjrendszerből származó adatok forgalombecslési és forgalomirányítási célú felhasználási lehetőségei. *Közlekedéstudományi Szemle*, LXX(6):208–214, 2020. doi: 10.24228/KTSZ.2020.6.1.

B. Varga and T. Tettamanti. Városi járműforgalom térbeli becslése kernel módszerek segítségével. *Közlekedéstud. Szemle*, 5(LXXI):37–43, 2021. doi: 10.24228/KTSZ.2021.5.2.

B. Varga, T. Tettamanti, and B. Kulcsár. Multiobjective control to mitigate bus bunching and improve schedule reliability of public transport. In *Swedish transportation research conference*, page Paper 14. KTH, 2017.

B. Varga, T. Tettamanti, and B. Kulcsár. Optimally combined headway and timetable reliable public transport system. *Transportation Research Part C: Emerging Technologies*, 92:1 – 26, 2018a. doi: 10.1016/j.trc.2018.04.016.

B. Varga, T. Tettamanti, and B. Kulcsár. Optimal headway merging for balanced public transport service in urban networks. *IFAC-PapersOnLine*, 51(9):416 – 421, 2018b. doi: 10.1016/j.ifacol.2018.07.068. 15th IFAC Symposium on Control in Transportation Systems CTS 2018.

B. Varga, T. Tettamanti, and B. Kulcsár. Energy-aware predictive control for electrified bus networks. *Applied Energy*, 252:113477, 2019. doi: 10.1016/j.apenergy.2019.113477.

B. Varga, T. Péni, B. Kulcsár, and T. Tettamanti. Network-level optimal control for public bus operation. In *21st IFAC World Congress*, 12-17 July 2020, Berlin, Germany, 2020a.

B. Varga, M. Szalai, Á. Fehér, Sz. Aradi, and T. Tettamanti. Mixed-reality automotive testing with SENSORIS. *Periodica Polytechnica ser. Transportation Engineering*, 2020b.

B. Varga, T. Tettamanti, B. Kulcsár, and X. Qu. Public transport trajectory planning with probabilistic guarantees. *Transportation Research Part B: Methodological*, 139:81 – 101, 2020c. doi: 10.1016/j.trb.2020.06.005.

B. Varga, T. Tettamanti, and Zs. Szalay. System architecture for Scenario-In-The-Loop automotive testing. *Transport and Telecommunication Journal*, 22(2):141–151, 2021. doi: doi:10.2478/ttj-2021-0011.

I. Varga, B. Kulcsár, T. Luspay, and T. Tettamanti. Korszerű szabályozások a közúti forgalomirányításban. *A Jövő Járműve*, 1-2:34–36, 2008.

References

H. Abdulsattar, M.R.K. Siam, and H. Wang. Characterisation of the impacts of autonomous driving on highway capacity in a mixed traffic environment: an agent-based approach. *IET Intelligent Transport Systems*, 14(9):1132–1141, 2020. doi: https://doi.org/10.1049/iet-its.2019.0285.

K. Aboudolas, M. Papageorgiou, and E. Kosmatopoulos. Store-and-forward based methods for the signal control problem in large-scale congested urban road networks. *Transportation Research Part C: Emerging Technologies*, 17:163–174, 2009. doi: 10.1016/j.trc.2008.10.002.

K. Aboudolas, M. Papageorgiou, A. Kouvelas, and E. Kosmatopoulos. A rolling-horizon quadratic-programming approach to the signal control problem in large-scale congested urban road networks. *Transportation Research Part C: Emerging Technologies*, 18(5):680 – 694, 2010. ISSN 0968-090X. doi: 10.1016/j.trc.2009.06.003. Applications of Advanced Technologies in Transportation: Selected papers from the 10th {AATT} Conference.

M. Alonso Raposo, M. Grosso, A. Mourtzouchou, J. Krause, A. Duboz, and B. Ciuffo. Economic implications of a connected and automated mobility in Europe. *Research in Transportation Economics*, 2021. ISSN 0739-8859. doi: https://doi.org/10.1016/j.retrec.2021.101072. paper nr. 101072.

M.S. Apáthy. Practical route planning algorithm. *Periodica Polytechnica Transportation Engineering*, 45(3):133–140, 2017. doi: 10.3311/PPtr.9916.

Atkins Ltd. Research on the impacts of connected and autonomous vehicles (cavs) on traffic flow. Technical report, Department for Transport, 2016.

A.-L. Barabási. *Bursts: the hidden patterns behind everything we do*. Dutton Adult, 2010. ISBN 0525951601.

M. Barth, K. Boriboonsomsin, and G. Wu. *Vehicle Automation and Its Potential Impacts on Energy and Emissions*, pages 103–112. Springer International Publishing, Cham, 2014. ISBN 978-3-319-05990-7. doi: 10.1007/978-3-319-05990-7_10.

M. Ben-Akiva, P.P. Macke, and P. Hsu. Alternative methods to estimate route-level trip tables and expand on-board surveys. *Transportation Research Record*, 1985.

M. Berrazouane, K. Tong, S. Solmaz, M. Kiers, and J. Erhart. Analysis and initial observations on varying penetration rates of automated vehicles in mixed traffic flow utilizing sumo. In *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, pages 1–7, 2019. doi: 10.1109/ICCVE45908.2019.8965065.

L. Bo and Z. Fusheng. Traffic signal control system based on wireless technology. In *2013 Third International Conference on Intelligent System Design and Engineering Applications*, pages 1578–1580, 2013. doi: 10.1109/ISDEA.2012.379.

D. Box. *Essential COM.* Addison-Wesley, 1998. ISBN 0-201-63446-5.

S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory.* SIAM Studies in Applied Mathematics. SIAM, 1994.

M. Brackstone and M. McDonald. Car-following: a historical review. *Transportation Research Part F: Traffic Psychology and Behaviour*, 2(4):181–196, 1999. ISSN 1369-8478. doi: https://doi.org/10.1016/S1369-8478(00)00005-X.

A. Brown, J. Gonder, and B. Repac. *An Analysis of Possible Energy Impacts of Automated Vehicles*, pages 137–153. Springer International Publishing, Cham, 2014. ISBN 978-3-319-05990-7. doi: 10.1007/978-3-319-05990-7_13.

I. Budinska. On ethical and legal issues of using drones. In N.A. Aspragathos, P.N. Koustoumpardis, and V.C. Moulianitis, editors, *Advances in Service and Industrial Robotics*, pages 710–717, Cham, 2019. Springer International Publishing. ISBN 978-3-030-00232-9.

G. Böker and J. Lunze. Stability and performance of switching Kalman filters. *International Journal of Control*, 75(16-17):1269–1281, 2002.

J. Calfee and C. Winston. The value of automobile travel time: implications for congestion policy. *Journal of Public Economics*, 69(1):83–102, July 1998.

M. Carey and M. Bowers. A review of properties of flow–density functions. *Transport Reviews*, 32(1):49–73, jan 2012. doi: 10.1080/01441647.2011.608135.

M.J. Cassidy, K. Jang, and C. F. Daganzo. Macroscopic fundamental diagrams for freeway networks: Theory and observation. *Transportation Research Record*, 2260(1):8–15, 2011. doi: 10.3141/2260-02.

CEN. Traffic control equipment. Signal heads. EN 12368:2015, 2015.

CEN. Intelligent transport systems - Cooperative ITS - Using V2I and I2V communications for applications related to signalized intersections. ISO/TS 19091:2017, 2017.

CEN. Traffic signal controllers. Functional safety requirement. EN 12675:2018, 2018a.

CEN. Road traffic signal systems. EN 50556:2018, 2018b.

G. Cheng, V. Peddinti, D. Povey, V. Manohar, S. Khudanpur, and Y. Yan. An exploration of dropout with LSTMs. In *Proc. Interspeech*, 2017.

L. Chu, H.X. Liu, J.-S. Oh, and W. Recker. A calibration procedure for microscopic traffic simulation. In *Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems*, volume 2, pages 1574–1579 vol.2, 2003. doi: 10.1109/ITSC.2003.1252749.

L. Chu, J-S. Oh, and W. Recker. Adaptive Kalman Filter based freeway travel time estimation. In *Transportation Research Board 84th Annual Meeting*, Washington DC, 2005.

B. Coifman. New methodology for smoothing freeway loop detector data: introduction to digital filtering. *Transportation Research Record: Journal of the Transportation Research Board*, (1554):142–152, 1996.

Corporate Partnership Board. Automated and autonomous driving, regulation under uncertainty. Technical report, International Transport Forum, 2015.

Cs. Csiszár, D. Földes, and Y.He. *Reshaped Urban Mobility*, chapter Sustainability in Urban Planning and Design, pages 251–268. InTechOpen, Rijeka, 2020. doi: 10.5772/intechopen.89211.

J. de Dios Ortúzar and L.G. Willumsen. *Modelling Transport*. Wiley, 2001. ISBN: 978-0471861102.

G. De Luca and M. Gallo. Artificial neural networks for forecasting user flows in transportation networks: Literature review, limits, potentialities and open challenges. In *Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2017 5th IEEE International Conference on*, pages 919–923. IEEE, 2017.

L.B. de Oliveira and E. Camponogara. Multi-agent model predictive control of signaling split in urban traffic networks. *Transportation Research Part C: Emerging Technologies*, 18(1): 120–139, 2010. doi: 10.1016/j.trc.2009.04.022. Information/Communication Technologies and Travel Behaviour; Agents in Traffic and Transportation.

C. Diakaki, M. Papageorgiou, and T. McLean. Application and evaluation of the integrated traffic-responsive urban corridor control strategy in-tuc in glasgow. In *CD-ROM of the 78th Annual Meeting of the Transportation Research Board*, number 990310, Washington, D.C., U.S.A., 1999.

I. Diakonikolas and M. Yannakakis. Small approximate pareto sets for biobjective shortest paths and other problems. *SIAM Journal on Computing*, 39(4):1340–1371, 2009. doi: 10.1137/080724514.

E.W. Dijkstra. Note on two problems in connexion with graphs (spanning tree, shortest path). *Numerical Mathematics*, 1:269–271, 1959. doi: 10.1007/BF01386390.

Y. Disser, M. Müller-Hannemann, and M. Schnee. Multi-criteria shortest paths in time-dependent train networks. In *In Proceedings of the 7th international conference on Experimental algorithms (WEA'08)*, pages 347–361, Berlin, Heidelberg, 2007. Springer-Verlag. doi: 10.1007/978-3-540-68552-4_26.

M.P. Dixon and L.R. Rilett. Population origin–destination estimation using automatic vehicle identification and volume data. *Journal of Transportation Engineering*, 131(2):75–82, 2005.

M.P. Dixon, A. Abdel-Rahim, M. Kyte, P. Rust, H. Cooley, and L. Rodegerdts. Field evaluation of roundabout turning movement estimation procedures. *Journal of Transportation engineering*, 133(2):138–146, 2007.

L.N.N. Do, N. Taherifar, and H.L. Vu. Survey of neural network-based models for short-term traffic state prediction. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(1):e1285, 2019.

L. Du, L. Han, and S. Chen. Coordinated online in-vehicle routing balancing user optimality and system optimality through information perturbation. *Transportation Research Part B: Methodological*, 79:121–133, 2015. ISSN 0191-2615. doi: https://doi.org/10.1016/j.trb.2015.05.020.

M. Duckham and L. Kulik. 'simplest' paths: Automated route selection for navigation. *Lecture Notes in Computer Science*, 2825:169–185, 2003. doi: 10.1007/978-3-540-39923-0_12.

S.M. Eisenman and G. List. A technique for data collection and estimation of turning movements at roundabouts. In *Proceedings of the 84th Annual Conference of the Transportation Research Board*, 2005.

N.-E. El Faouzi, H. Leung, and A. Kurian. Data fusion in intelligent transportation systems: Progress and challenges - a survey. *Information Fusion*, 12(1):4–10, January 2011. ISSN 1566-2535. doi: 10.1016/j.inffus.2010.06.001.

P.K. Findeisen. *Moving horizon state estimation of discrete time systems*. PhD thesis, University of Wisconsin-Madison, 1997.

B. Friedrich. The effect of autonomous vehicles on traffic. In *Autonomous Driving*, pages 317–334. Springer, Berlin, Heidelberg, 2016. doi: 10.1007/978-3-662-48847-8_16.

L. Fu, D. Sun, and L.R. Rilett. Heuristic shortest path algorithms for transportation applications: State of the art. *Computers and Operations Research*, 33(11):3324–3343, 2006. doi: 10.1016/j.cor.2005.03.027.

M. Gallo, F. Simonelli, G. De Luca, and C. Della Porta. An artificial neural network approach for spatially extending road traffic monitoring measures. In *Environmental, Energy, and Structural Monitoring Systems (EESMS), 2016 IEEE Workshop on*, pages 1–5. IEEE, 2016.

P. Gáspár, Zs. Szalay, and Sz. Aradi. *Highly automated vehicle systems*. BME MOGI, 2014. ISBN: 978-963-313-173-2.

V.V. Gayah, X. Gao, and A.S. Nagle. On the impacts of locally adaptive signal control on urban network stability and the macroscopic fundamental diagram. *Transportation Research Part B: Methodological*, 70:255–268, dec 2014. doi: 10.1016/j.trb.2014.09.010.

D.C. Gazis and R.B. Potts. The oversaturated intersection. *In: Proceedings of the Second International Symposium on Traffic Theory, London, UK*, pages 221–237, 1963.

S.K. Gehrig and F.J. Stein. Dead reckoning and cartography using stereo vision for an autonomous car. In *Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on*, volume 3, pages 1507–1512. IEEE, 1999.

M.A. Goodrich and E.R. Boer. Model-based human-centered task automation: a case study in acc system design. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 33(3):325–336, 2003. doi: 10.1109/TSMCA.2003.817040.

Amsterdam Group. Amsterdam group, signal phase and time and map data, version 1.1. Technical report, 2015.

L. Grüne and J. Pannek. *Nonlinear Model Predictive Control, Theory and Algorithms*. Springer, 2011.

N. Gupta and R. Hauser. Kalman filtering with equality and inequality state constraints. *arXiv preprint arXiv:0709.2791*, 2007.

D.K. Hale, C. Antoniou, M. Brackstone, D. Michalaka, A.T. Moreno, and K. Parikh. Optimization-based assisted calibration of traffic simulation models. *Transportation Research Part C: Emerging Technologies*, 55:100–115, 2015. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2015.01.018. Engineering and Applied Sciences Optimization (OPT-i) - Professor Matthew G. Karlaftis Memorial Issue.

T.J. Hastie and R.J. Tibshirani. *Generalized additive models*, volume 43. CRC press, 1990.

F.A. Haugen. A brief introduction to optimization methods. Technical report, University of South-Eastern Norway, 2018.

B.R. Hellinga. Requirements for the calibration of traffic simulation models. In *Proceedings of the Canadian Society for Civil Engineering*, volume 4, pages 211–222, 1988.

R. Herman and I. Prigogine. A two-fluid approach to town traffic. *Science*, 204(4389):148–151, 1979. doi: 10.1126/science.204.4389.148.

J. Herrera and A. Bayen. Traffic flow reconstruction using mobile sensors and loop detector data. In *Transportation Research Board 87th Annual Meeting*, Washington DC, United States, 12 2007.

B. Higgs, M. Abbas, and A. Medina. Analysis of the wiedemann car following model over different speeds using naturalistic data. In *3rd International Conference on Road Safety and Simulation*, pages 1–22, 09 2011.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

P.K. Houpt, M. Athans, D.G. Orlhac, and W.J. Mitchell. Traffic surveillance data processing in urban freeway corridors using kalman filter techniques. Technical report, United States. Dept. of Transportation. Research and Special Programs . . . , 1978.

J. Hourdakis, P.G. Michalopoulos, and J. Kottommannil. Practical procedure for calibrating microscopic traffic simulation models. *Transportation Research Record*, 1852(1):130–139, 2003. doi: 10.3141/1852-17.

H.S. Houthakker. Revealed preference and the utility function. *Economica*, 17(66):159–174, 1950. ISSN 00130427, 14680335.

R. Iannella, M. Gallo, G. De Luca, and F. Simonelli. Regression methods for spatially extending traffic data. *INTERNATIONAL JOURNAL OF ENGINEERING, TECHNOLOGY MANAGEMENT & APPLIED SCIENCES*, 5:5–14, 2017.

K. Jadaan, H. Khreis, and Á. Török. Exposure to traffic-related air pollution and the onset of childhood asthma: A review of the literature and the assement methods used. *Periodica Polytechnica Transportation Engineering*, 46(1):21–28, 2018. doi: 10.3311/PPtr.10113.

K. Jerath and S.N. Brennan. Analytical prediction of self-organized traffic jams as a function of increasing acc penetration. *IEEE Transactions on Intelligent Transportation Systems*, 13 (4):1782–1791, 2012. doi: 10.1109/tits.2012.2217742.

S. Jones and B.H. Philips. Cooperative adaptive cruise control: Critical human factors issues and research questions. In *7th International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design*, 2013. doi: 10.17077/drivingassessment.1477.

R.E. Kalman. A new approach to linear filtering and prediction. *Journal of Basic Engineering (ASME)*, 82(D):35–45, 1960.

A. Kesting, M. Treiber, M. Schänhof, and D. Helbing. Adaptive cruise control design for active congestion avoidance. *Transportation Research Part C: Emerging Technologies*, 16(6):668 – 683, 2008. ISSN 0968-090X. doi: 10.1016/j.trc.2007.12.004.

M. Keyvan-Ekbatani, M. Papageorgiou, and I. Papamichail. Perimeter traffic control via remote feedback gating. *Procedia-Social and Behavioral Sciences*, 111:645–653, 2014. doi: 10.1016/j.sbspro.2014.01.098.

A.R. Khan, M.F. Jamlos, N. Osman, M.I. Ishak, F. Dzaharudin, Y.K. Yeow, and K.A. Khairi. DSRC Technology in Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) IoT System for Intelligent Transportation System (ITS): A Review. In *Recent Trends in Mechatronics Towards Industry 4.0*, pages 97–106, Singapore, 2022. Springer Singapore. ISBN 978-981-33-4597-3.

F.S. Koppelman. Non-linear utility functions in models of travel choice behavior. *Transportation*, 10(2):127–146, Jun 1981. ISSN 1572-9435. doi: 10.1007/BF00165262.

D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3-4):128–138, December 2012.

S. Krauß. *Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics.* PhD thesis, Universitat zu Köln, 1998.

N. Kudarauskas. Analysis of emergency braking of a vehicle. *Transport*, 22(3):154–159, 2007.

B. Kulcsár, I. Varga, and J. Bokor. Constrained split rate estimation by moving horizon. In *16th IFAC World Congress Prague*, volume 16, Czech Republic, 2004. doi: 10.3182/20050703-6-CZ-1902.02036.

B. Kulcsár, T. Bécsi, and I. Varga. Estimation of dynamic origin destination matrix of traffic systems. *Periodica Polytechnica ser. Transp. Eng*, 33(1-2):3–14, 2005.

K. Kyu-Ok and L.R. Rilett. Genetic-algorithm based approach for calibrating microscopic simulation models. In *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*, pages 698–704, 2001. doi: 10.1109/ITSC.2001.948745.

I. Lana, J. Del Ser, M. Velez, and E.I. Vlahogianni. Road traffic forecasting: Recent advances and new challenges. *IEEE Intelligent Transportation Systems Magazine*, 10(2):93–109, 2018. doi: 10.1109/MITS.2018.2806634.

B. Lantos. *Irányítási rendszerek elmélete és tervezése I-II.* Akadémiai Kiadó, 2001, 2003.

M.G. Lay. *Handbook of Road Technology.* Spon Press, Abingdon, U.K., 2009.

L.J. LeBlanc and M. Abdulaal. A comparison of user-optimum versus system-optimum traffic assignment in transportation network design. *Transportation Research Part B: Methodological*, 18(2):115–121, 1984. ISSN 0191-2615. doi: https://doi.org/10.1016/0191-2615(84)90025-0.

S. Lin, B. De Schutter, Y. Xi, and J. Hellendoorn. Fast model predictive control for urban road networks via MILP. *IEEE Transactions on Intelligent Transportation Systems*, 12:846–856, 2011.

J. Long, J. Chen, W.Y. Szeto, and Q. Shi. Link-based system optimum dynamic traffic assignment problems with environmental objectives. *Transportation Research Part D: Transport and Environment*, 60:56–75, 2018. ISSN 1361-9209. doi: https://doi.org/10.1016/j.trd.2016.06.003. Special Issue on Traffic Modeling for Low-Emission Transport.

P.A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems.* IEEE, 2018.

J. Ma, H. Dong, and H. M. Zhang. Calibration of microsimulation with heuristic optimization methods. *Transportation Research Record*, 1999(1):208–217, 2007. doi: 10.3141/1999-22.

T. Ma and B. Abdulhai. Genetic algorithm-based optimization approach and generic tool for calibrating traffic microscopic simulation parameters. *Transportation Research Record*, 1800 (1):6–15, 2002. doi: 10.3141/1800-02.

Y. Ma and Y. Gao. Passenger transportation structure optimization model based on user optimum. *Procedia Engineering*, 137:202–209, 2016. ISSN 1877-7058. doi: https://doi.org/10.1016/j.proeng.2016.01.251. Green Intelligent Transportation System and Safety.

J.M. Maciejowski. *Predictive control: with constraints.* Prentice Hall, Harlow, UK, 2002. ISBN 0-201-39823-0.

M. Makridis, K. Mattas, B. Ciuffo, M. Alonso, T. Toledo, and C. Thiel. Connected and automated vehicles on a freeway scenario. effect on traffic congestion and network capacity. 04 2018. doi: 10.5281/zenodo.1483132.

MathWorks. *MATLAB 7 External Interfaces.* The MathWorks, Inc, USA, 2010.

D. Metz. Peak car in the big city: Reducing london's transport greenhouse gas emissions. *Case Studies on Transport Policy*, 3(4):367–371, 2015. ISSN 2213-624X. doi: https://doi.org/10.1016/j.cstp.2015.05.001.

H.B. Mitchell. *Multi-sensor data fusion: An introduction.* New York, Springer, 2007.

G.F. Newell. A simplified theory of kinematic waves in highway traffic. *Transportation Research Part B: Methodological*, 27(4):281 – 313, 1993. ISSN 0191-2615. doi: https://doi.org/10.1016/0191-2615(93)90038-C.

D. Ni, II J.D. Leonard, A. Guin, and B.M. Williams. Systematic approach for validating traffic simulation models. *Transportation Research Record*, 1876(1):20–31, 2004. doi: 10.3141/1876-03.

A. Olia, S. Razavi, B. Abdulhai, and H. Abdelgawad. Traffic capacity implications of automated vehicles mixed with regular vehicles. *Journal of Intelligent Transportation Systems*, 22(3): 244–262, 2018.

J. Oskarbski, K. Birr, M. Miszewski, and K. Zarski. Estimating the average speed of public transport vehicles based on traffic control system data. In *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pages 287–293, Budapest, Hungary, 2015. doi: 10.1109/MTITS.2015.7223269.

M. Papageorgiou. Overview of road traffic control strategies. *IFAC Proceedings Volumes*, 37 (19):29 – 40, 2004. ISSN 1474-6670. doi: https://doi.org/10.1016/S1474-6670(17)30657-2. 4th IFAC Workshop DECOM-TT 2004: Automatic Systems for Building the Infrastructure in Developing Countries, Bansko, Bulgaria, October 3-5, 2004.

M. Papageorgiou and G. Vigos. Relating time-occupancy measurements to space-occupancy and link vehicle-count. *Transportation Research Part C*, 16(1):1–17, 2008. doi: doi:10.1016/j.trc.2007.06.001.

M. Papageorgiou, J. M. Blosseville, and H. Hadj-Salem. Modelling and real-time control of traffic flow on the southern part of Boulevard Périphérique in Paris: Part I: Modelling. *Transportation Research A*, 24(5):345–359, 1990.

B. Park and J.D. Schneeberger. Microscopic simulation model calibration and validation: Case study of vissim simulation model for a coordinated actuated signal system. *Transportation Research Record*, 1856(1):185–192, 2003. doi: 10.3141/1856-20.

D. Park, S. You, J. Rho, H. Cho, and K. Lee. Investigating optimal aggregation interval sizes of loop detector data for freeway travel-time estimation and prediction. *Canadian Journal of Civil Engineering*, 36(4):580–591, 2009.

A. Paz, V. Molano, E. Martinez, C. Gaviria, and C. Arteaga. Calibration of traffic flow models using a memetic algorithm. *Transportation Research Part C: Emerging Technologies*, 55:432–443, 2015. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2015.03.001. Engineering and Applied Sciences Optimization (OPT-i) - Professor Matthew G. Karlaftis Memorial Issue.

J.L. Peterson. *Petri Net Theory and the Modeling of Systems.* Prentice Hall PTR, USA, 1981. ISBN 0136619835.

Louis A. Pipes. Car following models and the fundamental diagram of road traffic. *Transportation Research*, 1(1):21–29, 1967. ISSN 0041-1647. doi: https://doi.org/10.1016/0041-1647(67)90092-5.

O. Qing. *Fusing Heterogeneous Traffic Data: Parsimonious Approaches using Data-Data Consistency.* PhD thesis, Delft University of Technology, 2011.

F-M. De Rainville, F-A. Fortin, M-A. Gardner, M. Parizeau, and C. Gagné. Deap: A python framework for evolutionary algorithms. In *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '12, page 85–92, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450311786. doi: 10.1145/2330784.2330799.

N. Raju and H. Farah. Evolution of traffic microsimulation and its use for modeling connected and automated vehicles. *Journal of Advanced Transportation*, 2021:2444363, 2021. ISSN 0197-6729. doi: 10.1155/2021/2444363.

A.J. Robinson and F. Fallside. *The utility driven dynamic error propagation network.* University of Cambridge Department of Engineering, 1987.

SAE International. Dedicated Short Range Communications (DSRC) Message Set Dictionary, SAE J2735_2020007, 2016.

SAE International. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles, 04 2021.

G. Salvo, L. Caruso, and A. Scordo. Urban traffic analysis through an uav. *Procedia - Social and Behavioral Sciences*, 111:1083 – 1091, 2014. ISSN 1877-0428. doi: https://doi.org/10.1016/j.sbspro.2014.01.143. Transportation: Can we do more with less resources? – 16th Meeting of the Euro Working Group on Transportation – Porto 2013.

S.E. Shladover. Highway capacity increases from automated driving. *California PATH Program*, 2012.

Siemens AG. Siemens Sitraffic SST5 Outstation for the control of motorway traffic management system. *Technical specification*, page 5, 2019.

D. Simon. Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. *IET Control Theory & Applications*, 4(8):1303–1318, 2010.

Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches.* John Wiley & Sons, 2006.

B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M.I. Jordan, and S.S. Sastry. Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49(9): 1453–1464, 2004. doi: 10.1109/TAC.2004.834121.

J. Snoek, H. Larochelle, and R.P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

R. F. Stengel. *Optimal Control and Estimation*. Dover books on advanced mathematics. Dover Publications, 1986. ISBN 9780486682006.

S. Storandt. *Algorithms for vehicle navigation*. PhD thesis, Universität Stuttgart, Stuttgart, Germany, 2012.

Swarco Group. Intersection of Things. *Drive On, The Corporate Magazine of the Swarco Group*, pages 4–5, 2018.

Zs. Szalay. Structure and architecture problems of autonomous road vehicle testing and validation. In *15th Mini Conference on Vehicle System Dynamics, Identification and Anomalies (VSDIA)*, Hungary, Budapest, 7-9. November 2016.

M.W. Szeto and D.C. Gazis. Application of Kalman filtering to the surveillance and control of traffic systems. *Transportation Science*, 6:419–439, 1972.

M. Taiebat, A.L. Brown, H.R. Safford, S. Qu, and M. Xu. A review on energy, environmental, and sustainability implications of connected and automated vehicles. *Environmental Science & Technology*, 52(20):11449–11465, 2018. doi: 10.1021/acs.est.8b00127. PMID: 30192527.

A. Talebpour and H.S. Mahmassani. Influence of connected and autonomous vehicles on traffic flow stability and throughput. *Transportation Research Part C: Emerging Technologies*, 71: 143 – 163, 2016. ISSN 0968-090X. doi: 10.1016/j.trc.2016.07.007.

A. Talebpour, H.S. Mahmassani, and A. Elfar. Investigating the effects of reserved lanes for autonomous vehicles on congestion and travel time reliability. *Transportation Research Record*, 2622(1):1–12, 2017. doi: 10.3141/2622-01.

C.J. Taylor, R. Kennedy, Y. Yang, Delaware V. Regional Planning Commission, et al. Automated video-based traffic count analysis. Technical report, University of Pennsylvania, 2016.

K. Thatsanavipas, N. Ponganunchoke, S. Mitatha, and C. Vongchumyen. Wireless traffic light controller. *Procedia Engineering*, 8:190–194, 2011. ISSN 1877-7058. doi: https://doi.org/10.1016/j.proeng.2011.03.035. The 2nd International Science, Social Science, Engineering and Energy Conference 2010 (I-SEEC 2010).

J. Thomas, H.-L. Hwang, B. West, and S. Huff. Predicting light-duty vehicle fuel economy as a function of highway speed. *SAE Int. J. Passeng. Cars - Mech. Syst.*, 6:859–875, 04 2013. doi: 10.4271/2013-01-1113.

P. Tientrakool, Y.-C. Ho, and N.F. Maxemchuk. Highway capacity benefits from using vehicle-to-vehicle communication and sensors for collision avoidance. In *Vehicular Technology Conference (VTC Fall), 2011 IEEE*, pages 1–5. IEEE, 2011. doi: {10.1109/vetecf.2011.6093130}.

V. Tihanyi and Zs. Szalay. Autonomous vehicle platform for demonstration purposes. In *34th International Colloquium on Advanced Manufacturing and Repairing Technologies in Vehicle Industry*, pages 145–148, Visegrád, Hungary, May 2017.

M. Treiber and D. Helbing. Reconstructing the spatio-temporal traffic dynamics from stationary detector data. *Cooperative Transportation Dynamics*, 1(3):3–1, 2002.

B. Van Arem, C.J.G. Van Driel, and R. Visser. The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Transactions on Intelligent Transportation Systems*, 7(4): 429–436, 2006. doi: 10.1109/tits.2006.884615.

G. Vigos, M. Papageorgiou, and Y. Wang. Real-time estimation of vehicle-count within signalized links. *Transportation Research Part C: Emerging Technologies*, 16(1):18 – 35, 2008. ISSN 0968-090X. doi: 10.1016/j.trc.2007.06.002.

A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux. Traci: An interface for coupling road traffic and network simulators. In *Proceedings of the 11th Communications and Networking Simulation Symposium*, CNS '08, pages 155–163, New York, NY, USA, 2008. ACM. ISBN 1-56555-318-7. doi: 10.1145/1400713.1400740.

P.J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

R. Wiedemann. Simulation des Straßenverkehrsflusses. *Schriftenreihe des Instituts für Verkehrswesen der Universität Karlsruhe*, 8, 1974.

J.C. Williams, H.S. Mahmassani, S. Iani, and R. Herman. Urban traffic network flow models. *Transportation Research Record*, 1112:78–88, 1987.

H. Yang. System optimum, stochastic user equilibrium, and optimal link tolls. *Transportation Science*, 33(4):354–360, 1999. doi: 10.1287/trsc.33.4.354.

L. Ye and T. Yamamoto. Impact of dedicated lanes for connected and autonomous vehicle on traffic flow throughput. *Physica A: Statistical Mechanics and its Applications*, 512:588–597, 2018. ISSN 0378-4371. doi: https://doi.org/10.1016/j.physa.2018.08.083.

J.Y. Yen. An algorithm for finding shortest routes from all source nodes to a given destination in general networks. *Quarterly of applied mathematics*, 27(4):526–530, 1970. doi: 10.1090/qam/253822.

W. Zhang and H. Rakha. Estimating traffic stream space-mean speed and reliability from dual and single loop detectors. *Transportation Research Record*, 1925:38–47, 2005. doi: 10.3141/1925-05.

C.-L. Zhao and L. Leclercq. Graphical solution for system optimum dynamic traffic assignment with day-based incentive routing strategies. *Transportation Research Part B: Methodological*, 117(PA):87–100, 2018. doi: 10.1016/j.trb.2018.08.018.

S. Zhu, G. Jiang, and H.K. Lo. Capturing value of reliability through road pricing in congested traffic under uncertainty. *Transportation Research Procedia*, 23:664–678, 2017. ISSN 2352-1465. doi: https://doi.org/10.1016/j.trpro.2017.05.037. Papers Selected for the 22nd International Symposium on Transportation and Traffic Theory Chicago, Illinois, USA, 24-26 July, 2017.