Risk in Traffic Systems

Von der Fakultät für Mathematik und Physik
der Gottfried Wilhelm Leibniz Universität Hannover

zur Erlangung des Grades
Doktor der Naturwissenschaften
Dr. rer. nat.

genehmigte Dissertation von

M.Sc. Marcel Kleiber

2023

Referent: Prof. Dr. Stefan Weber
Korreferent: Prof. Dr. Zachary Feinstein
Korreferent: Prof. Dr. Bernhard Friedrich
Korreferent: Prof. Dr. Thomas Knispel
Tag der Promotion: 3. Juli 2023

Risk in Traffic Systems

M.Sc. Marcel Kleiber
Leibniz Universität Hannover

Abstract

Safe and efficient traffic systems constitute a cornerstone of modern life. This thesis provides a comprehensive treatment of the topic of risk in traffic systems. Based on an interdisciplinary research approach, novel models and methodologies are developed to investigate the impact of individual driving style, technological innovation, and traffic system design on safety and efficiency. For this purpose, perspectives and techniques from traffic modeling and insurance mathematics are combined.

First, we develop a microscopic traffic model that can describe the occurrence of traffic accidents due to random misperception, a type of error that is relevant for both human drivers and sensors of autonomous vehicles. The model allows us to characterize the real-world tradeoff between safety and efficiency in case studies.

Second, we generalize the microscopic traffic model to study the important scenario of an unsignalized urban intersection, a particularly accident-prone area. We apply the concept of random misperception to model the occurrence of accidents and discuss the numerical solution of the random ordinary differential equations involved using state-of-the-art methods. In case studies, we analyze the impact of driving styles on different types of conflicts; we also consider the important case of heterogeneous traffic participants where human drivers and autonomous vehicles coexist.

Third, we devise a methodology that makes microscopic traffic models accessible for a statistical study of traffic accidents and corresponding financial losses. The approach enables comprehensive risk management for traffic systems: We study the impact of changes in the design of vehicles and transport systems on functionality and road safety, and price insurance contracts that cover residual risks.

Fourth, we complement the above microscopic approaches with a macroscopic perspective: We investigate stochastic cell transmission models of traffic networks. The performance of traffic systems is evaluated based on preference functionals and acceptable designs. The numerical implementation combines simulation, Gaussian process regression, and a stochastic exploration procedure.

Keywords: Acceptable design, accidents, active learning, autonomous vehicles, car-following models, cell transmission models, digital twins, Gaussian process regression, insurance premiums, machine learning, microscopic traffic models, MODIS, perception errors, random ordinary differential equations, SUMO, traffic flow.

Contents

1 Introduction

Mobility provides a basis for economic activity and social interaction. Efficient traffic systems are indispensable for the exchange of goods, the daily commute to work, or even travel. They lay the foundations for prospering economies and growing welfare. Yet, mobility has a price in terms of safety. Be it conscious or not, entering a vehicle is always an individual decision to take a risk: Traffic accidents, ranging from minor rear-end collisions to fatal crashes, occur every year all over the world. The resulting losses are significant and potentially tragic:

- The global economic cost of traffic crashes is estimated to be $1.8 trillion from 2015–2030 (cf. Chen et al. (2019)).
- Around 1.3 million people die in traffic accidents worldwide each year (cf. World Health Organization (2019)).

These figures underscore the need for improvement. The key to tackling this lies in the interplay between technological innovation, traffic rules, and driving behavior.

In the future, autonomous vehicles are expected to replace human drivers as the main source of errors that lead to traffic accidents. Broadly, the idea is that these vehicles will use computer vision and machine learning to perceive their environment and make better decisions in less time. Autonomous vehicles are supposed to increase both safety and efficiency. It is a disruptive technology that allows us to rethink and redesign traffic systems (e.g., Bertoncello & Wee (2015)).

Nevertheless, traffic accidents will never be completely avoided – the risks can only be mitigated. A key question here is: How can we examine the risks associated with future technologies? Classically, one would use statistical methods to evaluate historical data and quantify potential impacts. However, this has limited applicability because historical data on future transportation systems are, by definition, unavailable. In this thesis, this fundamental challenge is addressed as follows: We develop models and generate artificial data through simulation. Counterfactual case studies allow us to envision and evaluate developments in a controlled simulation environment.

More generally, the optimal design and operation of vehicles and traffic systems is an important subject of study. Challenging problems arise in both academia and practice, even in the theoretical absence of accidents. Technological improvements and adapted traffic rules can mitigate accident risks and improve the tradeoff between safety and efficiency; traffic systems can be designed to yield acceptable outcomes. This is an engineer's

view of risk in traffic systems. A comprehensive risk management approach for traffic systems adds an actuarial perspective to the traditional engineering perspective: When residual risks are identified and appropriately quantified, insurance contracts enable the financial transfer of risks to insurance companies. The computation of adequate premiums that reflect technology and driving behavior requires a thorough understanding and precise quantification of losses.

In line with this general motivation, this thesis develops models and corresponding methodology for studying risks in traffic systems. In particular, the following specific topics are addressed:

- *Models of Traffic Accidents.* We develop microscopic traffic models to better understand the formation of accidents in relation to driving styles and errors. Our approaches allow us to characterize the real-world tradeoff between safety and efficiency for potential future transportation systems.

- *Risk Management of Traffic Systems.* A comprehensive view of the risk management of traffic systems considers not only efficiency, but also the financial losses caused by accidents. We develop a methodology to make microscopic traffic models accessible for the quantification of such financial losses. This allows us to study the influences of driving style and technology not only on efficiency, but also on the distribution of total losses and the prices of corresponding insurance contracts.

- *Design of Traffic Systems.* We investigate general cell transmission models that capture the motion of traffic participants at a high level of aggregation. These macroscopic models allow to study the performance of large-scale traffic systems depending on their design. To control and compare different traffic systems in the face of randomness and risk, we suggest the notion of acceptable design as a normative classification and categorization.

All contributions presented in this thesis are the result of an interdisciplinary research approach. The ubiquitous combination of engineering and insurance perspectives makes the study of risk in traffic systems an important research question in the context of operations research. A stylized representation of the underlying approach can be found in Figure 1.1; it is an ongoing process that thrives on taking the different perspectives and connecting them. Here, a pluralism of models, both microscopic and macroscopic, allows different aspects to be illuminated: While microscopic traffic models focus on the behavior of individual road users, macroscopic models enable investigations at the system level.

The following chapters of this thesis are self-contained and point out specific references to related research. The contributions can be summarized as follows:

Contributions of Chapter 2. In Chapter 2, we extend an existing microscopic traffic model by the concept of random misperception: This refers to randomly occurring

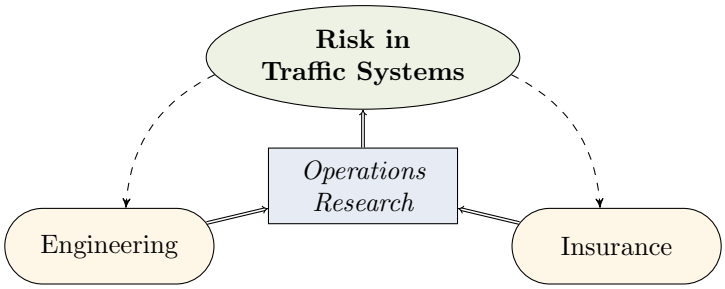


Figure 1.1: Stylized research approach.

perceptional errors which are relevant to both human drivers and autonomous vehicles. We model errors dynamically by stochastic processes and investigate their impact on the safety and the efficiency of traffic systems through Monte Carlo simulations.

(i) This chapter yields insights into the occurrence of accidents and their effects on traffic flow. Particular emphasis is put on the interplay between safety gaps and margins of perceptional errors.

(ii) We characterize the tradeoff between safety and efficiency in simple scenarios (one-lane road and t-junction).

Contributions of Chapter 3. In the third chapter, we develop a more general microscopic traffic model of an unsignalized intersection to which we apply the concept of random misperception. Besides, we extend the previous work of Chapter 2 in multiple directions:

(i) We provide a methodological analysis of a general class of random differential equations arising in the context of model development. For their numerical treatment, we use a state-of-the-art technique and explain necessary adjustments.

(ii) We investigate a traffic scenario where different types of traffic accidents are jointly present, namely, rear-end collisions and frontal crashes.

(iii) We also analyze the important case of heterogeneous traffic participants in which human drivers and autonomous vehicles coexist.

The case studies presented in Chapter 2 and 3 are based on the microscopic traffic simulator MODIS. Appendix A contains a documentation of this software.

Contributions of Chapter 4. The objective of this chapter is to develop a methodology to make microscopic models of traffic systems accessible for a statistical study of accidents and corresponding financial losses.

(i) We develop a powerful methodological framework to generate accident data based on microscopic traffic models in analogy to the concept of digital twins.

(ii) We illustrate the potential of our approach in comprehensive case studies. Specifically, we construct an implementation based on the state-of-the-art open-source traffic simulator SUMO.

(iii) Structurally, we characterize the total loss distribution approximately as a mean-variance mixture. This also yields alternative valuation procedures.

(iv) Based on Stein's method, we can improve the valuation with a correction term, derived from the results of El Karoui & Jiao (2009).

Detailed tables supplementing the case studies in Chapter 4 can be found in Appendix B.

Contributions of Chapter 5. In the final chapter of this thesis, we complement the previous microscopic approaches with a macroscopic perspective. We investigate stochastic cell transmission models that capture the motion of traffic participants at a high level of aggregation and study the performance of traffic systems depending on their design.

(i) We provide a rigorous framework for cell transmission models in general traffic networks. Traffic participants traveling in different directions interact with each other locally. Traffic volumes and conditions can vary stochastically.

(ii) To classify and categorize traffic systems, we propose the notion of acceptable design inspired by preference functionals and systemic risk measures.

(iii) The numerical estimation of acceptable designs combines Monte Carlo simulation, Gaussian process regression, and a stochastic exploration procedure in the parameter space. The performance of this algorithm is demonstrated through case studies.

2 Modeling Traffic Accidents Caused by Random Misperception

The original version of this chapter[1] was previously published in the proceedings of the *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2568–2574, see Berkhahn et al. (2018).

2.1 Introduction

Classically, the problem of determining the probability of traffic accidents has been of statistical nature: On the basis of empirical data, the probabilities of accidents are estimated by the corresponding relative frequencies. However, the automotive industry is undergoing a massive disruption with the appearance of autonomously driving vehicles (cf., e.g., Bertoncello & Wee (2015)). Traffic, as we know it, will change; in particular, in terms of efficiency and safety – but little to no data is available, yet! As long as operations are protected from large-scale cyber attacks, existing studies (e.g., Blanco et al. (2016)) indicate that the number of accidents will significantly be reduced when vehicles are controlled by computers. To overcome the lack of empirical accident data for future transportation systems, we propose a simulation based approach that yields insight into the occurrence of accidents and their effects on traffic flow.

Both human drivers and control systems of autonomous vehicles need to process large amounts of information about their environment. In most theoretical traffic models, decisions are based on exact information – in reality, errors may occur when positions and velocities of other vehicles are determined. The size of these errors depends on external conditions (e.g., weather) and on the driving style of a human operator or control algorithm. Another factor are potential malfunctions of systems. This chapter presents a stylized model for potential errors and investigates the impact on accidents and traffic flow. The key idea is to focus on *random misperception* as an omnipresent cause for accidents. Particular emphasis is put on the interplay between safety gaps and margins of perceptional errors. On a methodological level, the model facilitates an understanding of risks that are associated to beneficial future developments. Ultimately, our approach and techniques may form a basis for management decisions on the design of safety measures for autonomous driving systems.

[1] The case studies presented in this chapter are based on the microscopic traffic simulator MODIS. Appendix A contains a documentation of this software.

We choose the Intelligent Driver Model (IDM) (cf. Treiber, Hennecke & Helbing (2000)) as the underlying model for describing the movement of vehicles on lanes. Additionally, we incorporate adjustments allowing for driving errors that may lead to accidents. The IDM sets the acceleration of a vehicle based on the distance to its preceding vehicle and the difference of their velocities, i.e., the approaching rate. As originally proposed, this model is accident-free since the maximal deceleration is unbounded and, consequently, vehicles may execute unrealistic emergency braking maneuvers when they encounter dangerous situations. We modify the IDM at this point and also include the possibility of random misperception. The consequences of these changes are investigated in the context of two scenarios.

In Scenario A ("One-Lane Road Segment"), traffic is considered on a segment of a one-lane road on which vehicles drive in a consecutive order. We include two adjustments to the IDM: First, instead of assuming that the input variables (distance and approaching rate) are known with absolute precision, we include stochastic deviations in order to model random misperception; both distance and approaching rate may be over- or underestimated. Second, we limit the deceleration when braking, i.e., negative acceleration is bounded from below. With these two components, the model admits accidents. Whenever an accident occurs, the road is blocked and a traffic jam emerges. We assume that the collided vehicles are removed from the road after a random time and then traffic resumes. In this scenario, we focus on the occurrence of rear-end collisions. By means of Monte Carlo simulations, we study the tradeoff between safety and efficiency in terms of the number of accidents and traffic flow.

Scenario B ("Left-Turning on T-Junction") is an extension that builds on the first scenario. We consider a more complex element of a road system: a simplified t-junction. We capture this by considering two one-lane road segments which intersect. On each lane, the movement of vehicles is modeled as before; moreover, a conflict detection and reaction is implemented for vehicles which turn left. Turning vehicles extrapolate trajectories of conflicting vehicles on the basis of several observations. If the analysis of these trajectories suggests a collision, the turning vehicle will decelerate to allow conflicting vehicles to pass. We implement random misperception in the conflict detection and reaction behavior and use Monte Carlo simulations to analyze traffic at t-junctions, focusing again on the number of accidents and traffic flow.

The chapter is organized as follows: Section 2.2 reviews mathematical prerequisites. Section 2.3 presents the traffic model. Section 2.4 describes numerical case studies and analyzes the tradeoff between safety and efficiency. Section 2.4 concludes and discusses further research.

Literature. The analysis of the tradeoff between safety and efficiency of autonomous vehicles is a novel area of research. Most closely related to our approach is Segata & Cigno (2013) who analyze emergency braking scenarios on the basis of a deterministic

IDM. This paper also observes the necessity to bound deceleration in order to observe accidents. The focus lies on the impact of inter-vehicle communication on safety, and the computation terminates whenever an accident occurs. Similar ideas can also be found in Nekovee & Bie (2013). Stochastic extensions of the IDM are introduced in Treiber, Kesting & Helbing (2006). Random misperceptions provide a rationale for the empirical behavior of human drivers that is characterized by fluctuating accelerations. Such an approach is also studied in Treiber & Kesting (2017), Laval, Toth & Zhou (2014), and Lehmann (2000). These papers add white noise to the acceleration terms of car-following models. Accidents are, however, not investigated.

2.2 Mathematical Foundations

In this section, we review random ordinary differential equations and Ornstein-Uhlenbeck processes. These are ingredients to our traffic model with random misperception.

2.2.1 Random Ordinary Differential Equations

The classical IDM is described by ordinary differential equations (ODEs). Random misperception leads to a stochastic analogue of the equations, *random ordinary differential equations* (RODEs). We briefly describe RODEs and how to solve them; a comprehensive presentation of RODEs can be found in Han & Kloeden (2017).

Definition 2.2.1 (Random Ordinary Differential Equation). *Let $(\varepsilon_t)_{t\geq 0}$ be a stochastic process on some probability space $(\Omega, \mathcal{F}, P)$ with values in $\mathbb{R}^m$ and continuous paths. Suppose that $f\colon \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^d$ is continuous. A* random ordinary differential equation *in $\mathbb{R}^d$ for some function $y\colon [0,\infty) \to \mathbb{R}^d$ is given by*

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(y, \varepsilon_t).$$

For each scenario $\omega \in \Omega$, a RODE defines a non-autonomous ordinary differential equation via

$$\frac{\mathrm{d}y}{\mathrm{d}t} = F_\omega(t, y) := f(y, \varepsilon_t(\omega)).$$

Given $y(0) = y_0 \in \mathbb{R}$, this is a standard initial value problem and classical ODE-theory (e.g., Theorem of Picard-Lindelöf) applies when characterizing existence and uniqueness of solutions. Pathwise RODEs are ODEs which also allows to use standard numerical methods for ODEs in order to solve RODEs. This approach can be applied whenever sufficiently many realizations of the paths of the underlying stochastic process (ε_t) are available.

2.2.2 Ornstein-Uhlenbeck Process

In the extended IDM we model misperceptions as random deviations from the true values. This can be captured by a mean-reverting process in continuous time. A well-known Gaussian process of this type is the Ornstein-Uhlenbeck process.

Definition 2.2.2 (Ornstein-Uhlenbeck Process). *Let $\beta \in \mathbb{R}$ and $\alpha, \sigma > 0$. A stochastic process $(\varepsilon_t)_{t\geq 0}$ is called an* Ornstein-Uhlenbeck process, *if $\varepsilon_0 = a \in \mathbb{R}$ and $(\varepsilon_t)_{t\geq 0}$ solves the following stochastic differential equation:*

$$d\varepsilon_t \;=\; \alpha(\beta - \varepsilon_t)dt \;+\; \sigma dW_t,$$

where $(W_t)_{t\geq 0}$ denotes a one-dimensional standard Brownian motion.

Following Glasserman (2003), an Ornstein-Uhlenbeck process can iteratively be simulated exactly on an equidistant time grid $0 = t_0 < t_1 < \cdots < t_N$ with $t_{i+1} - t_i = \Delta t > 0$ for all $i \in \{0, 1, \ldots, N-1\}$ by

$$\varepsilon_{t_{i+1}} = h\varepsilon_{t_i} + \beta\,(1-h) + \sigma\sqrt{\frac{1-h^2}{2\alpha}}Z_{i+1},$$

where $h := e^{-\alpha\Delta t}$ and (Z_i) is a sequence of i.i.d. standard normal random variables. In Figure 2.1, we show typical simulated paths of the Ornstein-Uhlenbeck process for different values of σ. The parameter σ is the *volatility* of the process and captures both its tendency to fluctuate as well as the size of the infinitesimal random innovations.

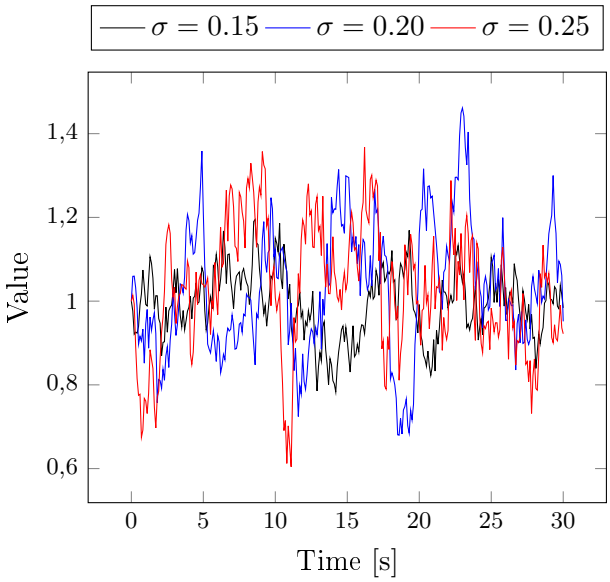


Figure 2.1: Simulated paths of an Ornstein-Uhlenbeck process (ε_t) for different values of σ with $\alpha = 1$, $\varepsilon_0 = 1$, $\beta = 1$.

2.3 The Traffic Model

The traffic models we consider are car-following models: cars react to preceding vehicles in order to maintain a minimum safety distance and to avoid crashes. Among car-following models, the IDM has attracted a lot of attention.

In this section, we extend the IDM and define the novel Intelligent Driver Model with Random Misperception: first by bounding the maximal deceleration, and second by introducing random misperception. As a consequence, accidents may occur.

2.3.1 Movement of Vehicles

We denote by $\mathcal{M} := \{1, 2, \dots\}$ the collection of vehicles. Each vehicle $i \in \mathcal{M}$ drives on a one-lane road modeled by a one-dimensional line $[0, L]$ of length $L > 0$; it enters the road at a time $t_0^i \geq 0$. The time sequence is increasing, i.e., $t_0^1 < t_0^2 < \dots$.

The velocity of each vehicle is determined according to the *Intelligent Driver Model with Random Misperception* (IDMrm): Let $(\varepsilon_t^{i,1})_{t\geq 0}, (\varepsilon_t^{i,2})_{t\geq 0}, (\varepsilon_t^{i,3})_{t\geq 0}$, $i \in \mathcal{M}$, be independent stochastic processes with continuous paths. The IDMrm sets the velocity $v^i(t)$ and the position $x^i(t)$ of vehicle $i \in \mathcal{M}$ at time $t \geq 0$ according to the following initial value problem composed of a system of coupled random ordinary differential equations

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t} x^i(t) = \max\{v^i(t), 0\}, \\ \frac{\mathrm{d}}{\mathrm{d}t} v^i(t) = \max\left\{ a_{\max}^i \cdot \left(1 - \left(\frac{\varepsilon_t^{i,1} v^i(t)}{v_{\mathrm{d}}^i}\right)^{\delta} - \left(\frac{s^*(\varepsilon_t^{i,1} v^i(t), \Delta_{\mathrm{per}} v^i(t))}{\Delta_{\mathrm{per}} x^i(t)}\right)^2\right), a_{\min}^i \right\}, \\ x^i(t_0^i) = 0,\ v^i(t_0^i) = v_0^i,\ t \geq t_0^i,\ i \in \mathcal{M} \end{cases}$$

where $s^*(s_1, s_2) = s_0 + s_1 T + \frac{s_1 s_2}{2\sqrt{a_{\max}^i b}}$ and

$$\begin{aligned} \Delta_{\mathrm{per}} v^i(t) &= \varepsilon_t^{i,1} v^i(t) - \varepsilon_t^{i,2} v^{i-1}(t), \\ \Delta_{\mathrm{per}} x^i(t) &= \varepsilon_t^{i,3} \Delta x^i(t) = \varepsilon_t^{i,3} (x^{i-1}(t) - x^i(t) - l_{i-1}), \end{aligned}$$

where l_i is the length of vehicle $i \in \mathcal{M}$. Moreover, $a_{\max}^i > 0$ is the maximal acceleration, and $a_{\min}^i < 0$ the minimal acceleration (i.e., maximal deceleration) of the i-th vehicle. The other parameters originate from the classic IDM model, and we refer to Treiber, Hennecke & Helbing (2000) for a detailed explanation. For the first vehicle $i = 1$, we set the interaction term $s^*(\varepsilon_t^{1,1} v^t(t), \Delta_{\mathrm{per}} v^1(t)) \cdot (\Delta_{\mathrm{per}} x^1(t))^{-1} := 0$ as there is no preceding vehicle.

2.3.2 Accidents

The stochastic processes $(\varepsilon_t^{i,1}), (\varepsilon_t^{i,2})$ and $(\varepsilon_t^{i,3})$, $i \in \mathcal{M}$, may be interpreted as different sources of errors. The classic IDM determines the velocity on the basis of the distance to the preceding vehicle and the approaching rate. In contrast, the IDMrm assumes

that all these quantities are subject to perceptional errors. The perceived quantities are inputs to the calculation of the acceleration of each vehicle. Vehicle i uses for this computation instead of the true velocities $(v^i(t), v^{i-1}(t))$ of itself and the preceding vehicle the distorted values $(\varepsilon_t^{i,1} v^i(t), \varepsilon_t^{i,2} v^{i-1}(t))$; in addition, instead of the true distance to the preceding car $\Delta x^i(t)$ the randomly distorted value $\varepsilon_t^{i,3} \Delta x^i(t)$ is the third input to the calculation. There are no errors, as long as $\varepsilon_t^{i,1} = \varepsilon_t^{i,2} = \varepsilon_t^{i,3} = 1$. Our model is sufficiently flexible to admit many stochastic error processes. In our numerical case studies, we will assume that $(\varepsilon_t^{i,1}), (\varepsilon_t^{i,2})$ and $(\varepsilon_t^{i,3})$, $i \in \mathcal{M}$, are independent Ornstein-Uhlenbeck processes that randomly fluctuate around 1. This can be interpreted as noisy perception of the true values. Misperception can cause accidents.

An accident occurs when vehicles collide. Up to this point, their movement is described by the RODEs above. However, we assume that this is not the case anymore after a collision. If an accident occurs, collided vehicles will remain at their position for some time. Then they will be removed from the system. In the following, we will make this precise.

For $i \in \mathcal{M}$, let $A^i(t)$ denote the area of the road which is occupied by vehicle i at time $t > 0$. In the one-dimensional case this corresponds to the interval $A^i(t) = [x^i(t) - l_i/2, x^i(t) + l_i/2]$ where l_i denotes the vehicle's length and $x^i(t)$ is the position of the vehicle's midpoint. Formally, an accident occurs, if

$$\exists\, i, j \in \mathcal{M},\ i \neq j,\ \exists\, t > 0 \colon A^i(t) \cap A^j(t) \neq \emptyset.$$

Now, if two vehicles collide, their velocities are immediately set to 0. Depending on the traffic constellation, further vehicles may crash into an existing collision or perform a safe emergency braking maneuver. We assume that at the time of the first collision, an exponentially distributed waiting time $t_{\text{removal}} \sim \text{Exp}(\gamma)$, $\gamma > 0$, is triggered; as this time has passed, all vehicles that collided disappear from the model. The expected waiting time until vehicles are removed is $\mathbb{E}(t_{\text{removal}}) = \gamma^{-1} > 0$. We note that other accidents may occur at different locations in the system; the removal time at different locations is triggered independently in each case.

In summary, if an accident occurs, the one-lane road is blocked and a traffic jam emerges. Later – after a random time t_{removal} – vehicles that collided are removed from the road; remaining vehicles will continue their journey, and the traffic jam dissipates.

2.4 Case Studies

In this section, our approach will be illustrated in the context of two traffic scenarios: In the first scenario, a one-dimensional road segment is considered. The vehicles enter at the beginning of the road segment, the origin, and disappear at its other end. We analyze the evolution of traffic over a fixed period of time and focus on safety and efficiency. The second scenario describes a more complex situation: a t-junction composed of two

intersecting one-lane road segments.

Our traffic model is capable of capturing heterogeneous vehicles. Note that each vehicle is endowed with its own set of parameters and associated stochastic processes. This allows to model individual driving behavior and corresponding error patterns. In this chapter, we focus on a simplified version of the model with homogeneous traffic participants, highlighting the effects of varying parameters. Misperception is captured by the processes $(\varepsilon_t^{i,1}), (\varepsilon_t^{i,2}), \dots, (\varepsilon_t^{i,5})$, $i \in \mathcal{M}$, which we assume to be Ornstein-Uhlenbeck processes fluctuating around 1.

We fix a terminal time $T_{\text{sim}} > 0$ for the traffic simulation. Vehicles are consecutively enumerated by $1, 2, 3, \dots$ and enter each lane-segment at its origin paying attention to existing traffic. The exact procedure will be described below, but we already stress at this point that due to the randomness of traffic flow also the collection of vehicles $\mathcal{M}$ that are generated until terminal time T_{sim} is random. We simulate the traffic system and compute statistics characterizing safety and efficiency from $m \in \mathbb{N}$ independent simulation runs.

Measure of Efficiency. As a measure of efficiency for the traffic system we choose traffic *flow* per time unit, measured at position d (in our simulations, we choose d as the end of the road):

$$Q = \frac{\operatorname{card}\{j \in \mathcal{M} : \exists\, t \le T_{\text{sim}} \colon x^j(t) = d\}}{T_{\text{sim}}}.$$

Here, *card* denotes cardinality. In the following, we denote sample averages that we compute from our simulation runs by a circumflex. For example, the sample average of the flow is $\hat{Q}$.

Measure of Safety. A measure of traffic safety is the number of accidents per time unit. The term *accident* refers to an event where at least two vehicles collide. If more vehicles crash into an existing collision, this does not create a new accident according to our convention.

Recall that the area occupied by vehicle $i \in \mathcal{M}$ at time $t \ge 0$ is denoted by $A^i(t)$; additionally, for $M \subseteq \mathcal{M}$ we define $A^M(t) := \bigcup_{i \in M} A^i(t)$. The number of accidents per time unit, denoted by f_{acc}, is given by

$$f_{\text{acc}} = \frac{1}{T_{\text{sim}}} \cdot \operatorname{card}\big\{\emptyset \neq M \subseteq \mathcal{M} \colon \exists\, t \le T_{\text{sim}}\ \forall\, i \in M \colon A^i(t) \cap A^{M\setminus\{i\}}(t) \neq \emptyset \\ \text{and } \forall\, t \le T_{\text{sim}} \colon A^M(t) \cap A^{M^c}(t) = \emptyset\big\},$$

where M^c denotes the complement of M. The first condition ensures that all vehicles in M collide, the second that all vehicles involved in the accident are identified. The corresponding sample average is denoted by $\hat{f}_{\text{acc}}$.

2.4.1 One-Lane Road Segment

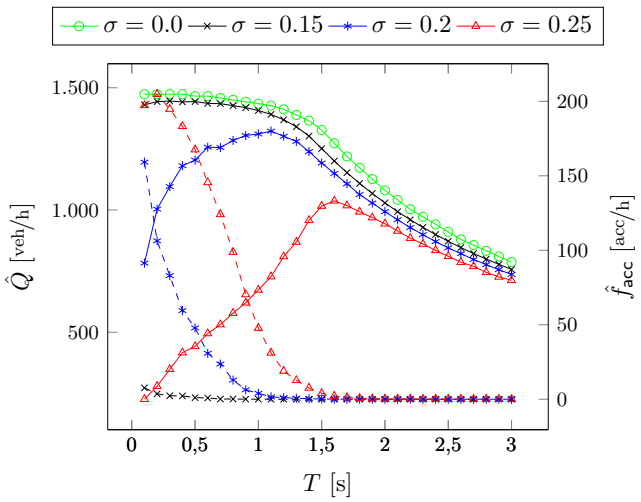


Figure 2.2: Averaged flow and number of accidents for varying T and fixed σ with $1,000$ independent simulations for each parameter combination: Dashed lines correspond to number of accidents, solid lines to flows.

This scenario consists of a segment of a one-lane road, a one-dimensional line $[0, L]$ of length $L = 2,000\,\mathrm{m}$. Vehicles are generated at the origin and are removed when they reach the end of the road. Their generation is defined by the following algorithm: Vehicles are created deterministically with a constant demand (here, $1,500\,\mathrm{veh/h}$), if there is enough space available at the beginning of the road. More precisely, a vehicle $i \in \mathcal{M}$ may be generated according to the desired demand, if there is no other vehicle in the first $7.5\,\mathrm{m}$ of the road (which equals the vehicle length plus an additional safety margin of $1.5\,\mathrm{m}$); otherwise, the generation of the new vehicle is delayed until this condition is satisfied. The initial velocity of any new vehicle matches the velocity of the preceding vehicle. In summary, the initial generation of vehicles avoids artificial accidents; instead, accidents may be caused by random misperception at a later point in time somewhere on the lane.

Table 2.1: Parameter choice for the scenarios.

Scenario	$a_{\max}$	v_{d}	δ	$a_{\min}$	s_0	T	b	l	α	β	σ	γ
A	2.0	15	4	−3.5	1.2	·	1.67	6	1	1	·	1/60
B	2.0	10	4	−3.5	2.0	1.5	1.67	6	1	1	·	1/300

The remaining parameters used for our simulation are given in Table 2.1. Traffic is simulated for a duration of $T_{\mathrm{sim}} = 600\,\mathrm{s}$ according to IDMrm. The error processes $(\varepsilon_t^{i,1}), (\varepsilon_t^{i,2}), (\varepsilon_t^{i,3})$ are independent and identically distributed copies of an Ornstein-Uhlenbeck process with $\alpha = \beta = 1$ and different values of σ; in order to guarantee that we may observe sufficiently many accidents in our small-scale example, we choose rela-

tively high volatilities. Smaller volatilities require longer simulations and larger roads to observe a sufficient number of accidents which increases the computational effort. This does not alter the methodology, and the corresponding case studies may be analyzed in the future.

In the analysis of the model we focus on the effect of a varying error volatility σ and a varying time headway T. The volatility σ is a measure for the size of the random misperception in the model. The time headway T is a parameter in the IDMrm that influences the safety distances. In the absence of random misperception, the bigger the time headway, the greater is the distance between vehicles and the lower the traffic flow. If errors are present, a larger time headway will decrease the number of accidents. Since a large number of traffic accidents may also decrease flow, we expect that the dependence of flow on time headway is not always monotone anymore.

We analyze the behavior of the system after the first vehicle has reached the end of the road. Both flow and the number of accidents are random. We display their sample averages (approximating their expectations) in Figure 2.2 for varying T and different fixed values of σ. The case $\sigma = 0$ corresponds to no misperception with no accidents. As a consequence, minimizing T leads to the maximal flow – almost equal to the demand of $1,500\,^{\text{veh}}/_{\text{h}}$. With increasing σ, we observe decreasing flow. The rational is that accidents lead to traffic congestion which decreases flow. If we fix σ but vary the time headway T, we can find a T that maximizes the flow. This shows the interplay between safety and efficiency: Accidents decrease the traffic flow. A larger time headway T decreases the number of accidents, but, if there are only few accidents, also decreases flow. Thus, for small T, flow increases with increasing T due to a decreasing number of accidents, but for large T flow decreases with increasing T.

This tradeoff can also be observed in Figure 2.3. The figure displays the sample averages of both flow and the number of accidents for varying σ and different fixed values of T. Of course, with increasing σ, $\hat{f}_{\text{acc}}$ increases and $\hat{Q}$ decreases. The key point is to observe that the flow curves intersect! At a certain level of misperception (with too many accidents occurring), it becomes more efficient to select a larger time headway T that decreases the number of accidents and increases efficiency.

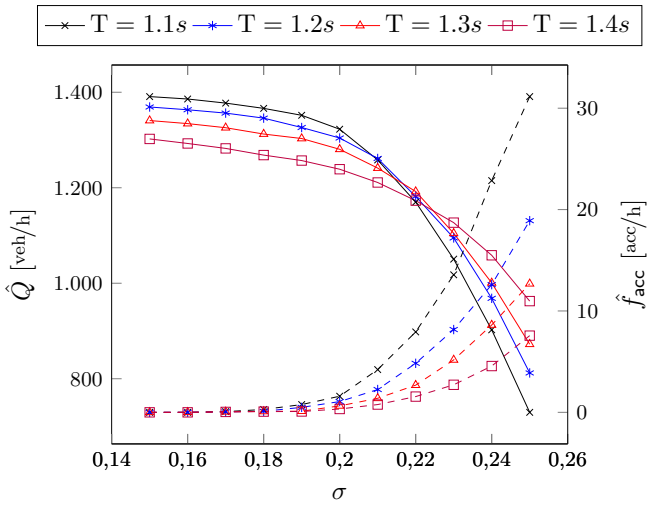


Figure 2.3: Averaged flow and number of accidents for varying σ and fixed T with $1,000$ independent simulations for each parameter combination: Dashed lines correspond to number of accidents, solid lines to flows.

2.4.2 Left-Turning on T-Junction

In Scenario B, we analyze traffic at a t-junction with vehicles that turn left. We focus on the simplified setting shown in Figure 2.4: Vehicles on the bottom lane always turn left, following the green path – while vehicles on the top lane never turn, following the black path. On an abstract level, this scenario can be decomposed into two one-lane road segments with the additional property that these segments intersect. By creating an intersection, we introduce conflicts: vehicles can collide at the t-junction. We assume that left-turning vehicles give way to oncoming vehicles on the upper lane. For example, they might need to stop at a certain point (denoted by x_{stop}; cf. Figure 2.4) in order to let the conflicting vehicles pass the junction (i.e., reach x_{exit}; cf. Figure 2.4). We assume that vehicles on the upper lane insist on their right of way and are oblivious to oncoming traffic, i.e., they do not react.

In this case study, the horizontal lane has a length of 300 m with the junction placed at 97.5 m. Green and black path intersect at 94.5 m and x_{stop} is located at 85.0 m. We generate vehicles similarly to Scenario A, but instead of assuming a fixed rate we create them with an exponentially distributed headway with mean 500 veh/h on the upper lane and mean 200 veh/h on the lower lane. This stochastic generation of vehicles yields interesting dynamics with vehicles on the lower lane reacting to the upper lane: vehicles need to wait for emerging gaps in order to turn; gaps occur at random times.

We implement IDM on the upper lane with the same adjustments as for IDMrm, i.e., we implement IDMrm with perfect perception. We do the same on the lower lane, but introduce as an additional feature conflict detection and reaction. Random misperception could be introduced at different points of the model, but we focus in this chapter only on errors in the conflict detection and reaction which may create accidents in the context

of left-turning maneuvers.

We heuristically describe the implementation of the conflict detection and reaction for a turning vehicle i and one oncoming vehicle j. Vehicle i follows the green path according to the implemented car-following model. Additionally, as a turning vehicle it can detect conflicts. Turning vehicles are aware of oncoming vehicles and extrapolate the trajectories of potentially conflicting vehicles based on observations of their movement. They also extrapolate their own trajectories. Vehicle i checks if braking is necessary in order to turn safely. For this purpose, a distance $\hat{d}^{ij}(t)$ of vehicle i to the approaching vehicle j is estimated. The situation is classified as a conflict, if the estimate $\hat{d}^{ij}(t)$ is smaller than a given safety threshold d_s at some point in time. We refer to Pascucci et al. (2015) for a detailed description of conflict detection. If no conflict arises, the vehicle turns. Otherwise, it decelerates according to the following algorithm. To simplify the notation, we omit dependency on t. First, we compute a naive deceleration a_naive such that vehicle i stops at x_stop with a constant (negative) acceleration given by $a_\mathrm{naive} = -(v^i)^2/\left(2\left(x_\mathrm{stop} - x^i\right)\right)$. For this, the vehicle needs a time of $t_\mathrm{naive} = -v^i/a_\mathrm{naive}$. We let $\hat{t}^j$ be the time vehicle j needs to reach x_exit which is computed by numerically inverting the extrapolated trajectory. If $\hat{t}^j > t_\mathrm{naive}$, vehicle i sets a_naive as its acceleration, stops at x_stop and waits until it can turn safely. If $\hat{t}^j \leq t_\mathrm{naive}$ (i.e., vehicle i does not need to stop since vehicle j will have passed the junction by that time), we choose the following acceleration

$$a^{i,j}_\mathrm{smooth} = \left(\frac{x_\mathrm{stop} - x^i}{\hat{t}^j} - v^i\right)\frac{2}{\hat{t}^j}$$

which is determined such that vehicle i reaches x_stop when vehicle j arrives at x_exit. Stopping is not necessary in this situation. These computations are carried out for all possibly conflicting vehicles, and the minimal resulting acceleration is chosen, while also taking into account possible car-following behavior.

Random misperception is implemented as follows: For the conflict detection, vehicle i extrapolates trajectories and computes an estimated distance $\hat{d}^{ij}(t)$ from the conflicting vehicle. Only this distance measure is the quantity that triggers i's reaction as described above. The extrapolation of i's own trajectory is based on its velocity $v^i(t)$. We assume that both $v^i(t)$ and $\hat{d}^{ij}(t)$ are subject to misperception: We implement two independent and identically distributed Ornstein-Uhlenbeck processes $(\varepsilon^{i,4}_t)_{t\geq 0}$ and $(\varepsilon^{i,5}_t)_{t\geq 0}$ (as in Scenario A) to distort these values, i.e., replacing them with $\varepsilon^{i,4}_t v^i(t)$ and $\varepsilon^{i,5}_t \hat{d}^{ij}(t)$ where $\hat{d}^{ij}(t)$ is already computed on the basis of the misperceived velocity $\varepsilon^{i,4}_t v^i(t)$. As a consequence, both the conflict detection and the conflict reaction may be erroneous.

We begin with our measurement when the first vehicles reach the ends of both lanes. As before, we simulate traffic for 600 s and implement an exponentially distributed removal time t_removal for vehicles that collided. We evaluate the dependency of flow and number of accidents on safety distance d_s and volatility σ of the Ornstein-Uhlenbeck processes.

In Figure 2.5 we fix $d_\mathrm{s} = 1.5\,\mathrm{m}$ and present the effect of increasing volatility on num-

ber of accidents and traffic flow for both lanes. The figure displays similar phenomena as Figure 2.3, confirming our expectation that random misperception causes accidents. Compared to Scenario A, we observe fewer accidents which is due to more conservative parameter choices in the implementation of the turning maneuver (cf. Table 2.1). However, one can still see that with higher volatility more accidents occur, causing a decline of traffic flow.

Next, we investigate the impact of d_s for fixed volatilities σ. We analyze the traffic flow on the lower lane (see Figure 2.6) and on the upper lane (see Figure 2.7) separately. The case $\sigma = 0$ corresponds to no misperception; accidents may still occur in this case due to extrapolation errors for d_s too small. For different values of σ, traffic on the lower lane exhibits a similar behavior as in Scenario A, see Figure 2.2: If d_s is too small, many accidents occur such that traffic flow is impaired. With increasing d_s, the number of accidents decreases. This initially improves the traffic flow, but if d_s becomes too large, traffic flow again decreases. In this situation, vehicles cannot easily find gaps in the oncoming traffic that permit turns. On the upper lane, in contrast, flow is strictly increasing with increasing d_s. This is not surprising, since the traffic flow on the upper lane is only distorted, if turning vehicles cause accidents.

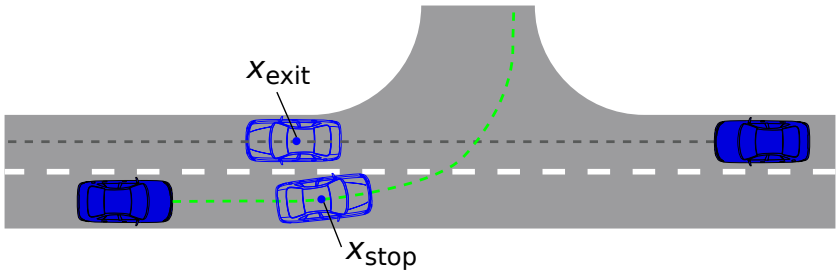


Figure 2.4: Simplified left-turning scenario on t-junction.

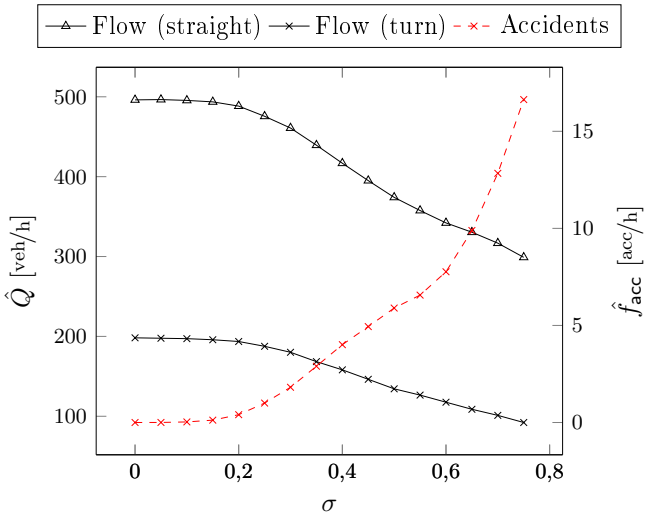


Figure 2.5: Averaged traffic flow (turn and straight) and number of accidents for varying σ and $d_{\mathrm{s}} = 1.5\,\mathrm{m}$ with $10,000$ independent simulations.

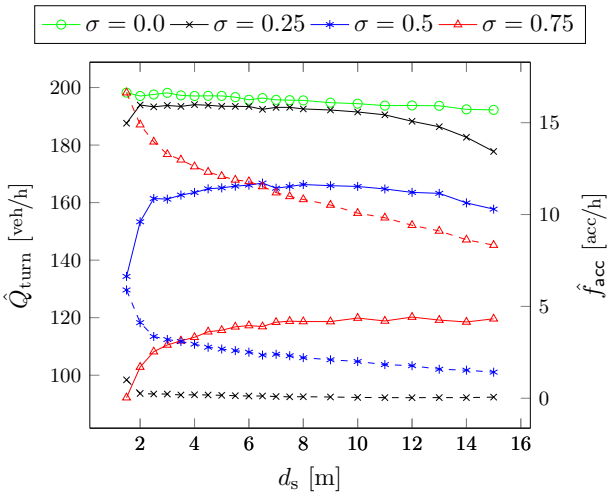


Figure 2.6: Averaged turn flow (lower lane) and number of accidents for varying d_{s} and fixed σ with $10,000$ independent simulations for each parameter combination: Dashed lines correspond to number of accidents, solid lines to flows.

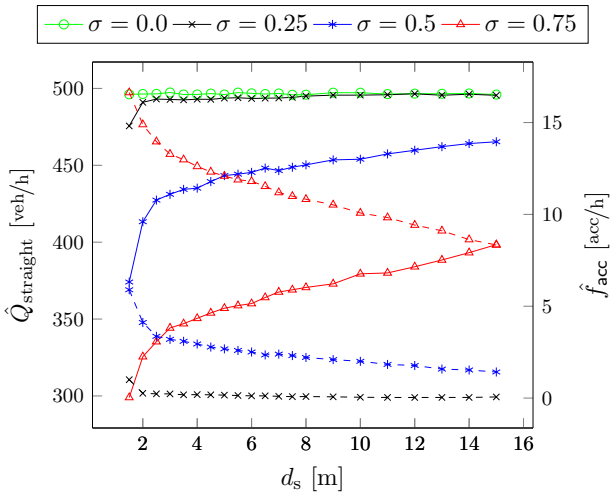


Figure 2.7: Averaged straight flow (upper lane) and number of accidents for varying d_{s} and fixed σ with 10,000 independent simulations for each parameter combination: Dashed lines correspond to number of accidents, solid lines to flows.

2.5 Conclusion & Future Research

We introduced a traffic model that admits accidents. The accidents are caused by random misperception, a type of error that affects both human drivers and autonomous vehicles. The simulation model admits a characterization of the tradeoff between safety and efficiency of traffic systems. While empirical data on the traffic systems of the future are not available yet, our causal stochastic model produces simulated data that provide guidance to the design and risk management of future traffic systems.

In our case studies, we studied homogeneous traffic participants and one particular error pattern, modeled by independent Ornstein-Uhlenbeck processes. However, our approach can also capture heterogeneity, i.e., multiple driving styles and error types. In particular, the model can be used to analyze effects of systems that include both human drivers and autonomous vehicles. Such traffic systems will be relevant in the near future. Moreover, one could try to generalize the model to include the effects of V2V-communication that might be subject to random errors.

We have demonstrated that optimality in terms of traffic flow does typically not imply that accidents are absent; accidents cause harm to the society. Future research should define and include the cost of accidents to the analysis. This requires a model of the severity of accidents, an issue that was neglected in the current chapter. From a computational point of view, the efficiency of the simulation might be improved by applying well-designed variance reduction techniques. Since accidents are rare events, variance reduction techniques for rare-event simulation, such as importance sampling, might be promising.

Misperception, 21st International Conference on Intelligent Transportation Systems (ITSC), 2018.

3 Traffic Dynamics at Intersections Subject to Random Misperception

The original version of this chapter[1] was previously published in *IEEE Transactions on Intelligent Transportation Systems* 23(5), pp. 4501–4511, see Berkhahn et al. (2022).

3.1 Introduction

The self-organization of traffic is a highly complex phenomenon. Traffic flow is distorted by accidents which are often triggered by errors in perception or judgement of traffic participants. It seems plausible that in a future world of autonomous vehicles improved technology will substantially reduce, but not completely eliminate traffic accidents (cf., e.g., Bertoncello & Wee (2015), Blanco et al. (2016)). In particular, accidents will still occur whenever autonomous vehicles and human drivers coexist. For these future scenarios, a sufficient amount of real world statistical data on traffic systems with different proportions of autonomous vehicles is not yet available. To overcome this lack of information, we propose a stochastic model that generates artificial data on both traffic flow and accidents. In this setting, we study the tradeoff between safety and efficiency as a function of the driving style of the individual vehicles. In the absence of real data, the sound design of future traffic systems requires such a strategy. Simulations that generate artificial data are a prerequisite for the anticipation of both future capabilities and risks associated with autonomous vehicles and their algorithms, advanced driver-assistance systems, and human drivers.

Vehicles in traffic systems are constantly in conflict with each other; autonomous vehicles and human drivers have to observe their environment, predict its future behavior and react accordingly in order to avoid accidents. Thereby, they control the distance to preceding vehicles, or – when turning or overtaking – they give way to other vehicles in order to avoid collisions. This requires the extrapolation of trajectories of potentially conflicting vehicles, the estimation of the size of safety gaps and decisions about when to stop and to wait, and when to proceed. These issues jointly appear at intersections turning them into a particularly risky location in traffic systems; Dresner and Stone Dresner & Stone (2008) state that "vehicle collisions at intersections account for anywhere between 25% and 45% of all collisions".

[1] The case studies presented in this chapter are based on the microscopic traffic simulator MODIS. Appendix A contains a documentation of this software.

In this chapter, we focus on the traffic dynamics at intersections. We propose a model and present case studies for unsignalized intersections and discuss possible extensions for signalized ones. Intersections are modeled as multiple intersecting one-lane roads. On each of these roads, the basic movement of the vehicles is described by a microscopic car-following model. Cars need to control their distance to other vehicles in order to avoid rear-end collisions. At an intersection, additional conflicts between turning vehicles arise. We implement a conflict detection, fix a priority regime (right has right-of-way) and assume that vehicles will wait for emerging gaps if they have to give way. In reality, the three components – car-following, conflict detection, and conflict reaction – may be subject to errors of human drivers, possibly assisted by suitable technology. We model perceptional errors by stochastic processes which randomly fluctuate around the correct quantities.

The stylized model provides a conceptual framework for understanding the causal relationship between perceptional errors, (parametrized) driving style, accidents, and traffic flow. We study these features for different penetration rates of error-free autonomous vehicles in the traffic system. Our model captures the occurrence of two possible collision types: rear-end collisions resulting from low headways and frontal crashes in the context of turning maneuvers.

We provide a methodological basis and explain how traffic at intersections can be modeled by a system of coupled random ordinary differential equations.

This chapter extends previous work (see Berkhahn et al. (2018)) in multiple directions:

- Berkhahn et al. (2018) primarily focuses on the Intelligent Driver Model with random misperception in the context of one-lane roads and heuristically discusses extensions to t-junctions. Rear-end collisions and collisions at t-junctions were analyzed separately. Now, we present a general and rigorous framework comprising both cases.

- This chapter provides a comprehensive methodological analysis of a general class of random differential equations modeling both conflict detection and potential misperception. We use a state-of-the-art simulation technique and explain necessary adjustments in the context of the suggested model.

- In numerical case studies, we analyze the tradeoff between safety and efficiency.

- Besides scenarios with homogeneous traffic participants, we also analyze the heterogeneous case in which human drivers and autonomous vehicles coexist.

The chapter is organized as follows: Section 3.2 reviews related literature, Section 3.3 introduces our model of intersections, Section 3.4 explains the simulation methodology which we apply in various case studies. Simulation results are presented and discussed in Section 3.5. Section 3.6 concludes.

3.2 Literature Review

Literature on traffic modeling is vast. Our approach is based on a stochastic extension of microscopic traffic models which describe the movement of each vehicle individually. In particular, we adapt the Intelligent Driver Model (IDM), originally proposed by Treiber, Hennecke & Helbing (2000). It belongs to the class of car-following models (also called follow-the-leader models). Random misperception may also be implemented in other car-following models, e.g., the Optimal Velocity Model (cf. Bando et al. (1994) and Bando et al. (1995)).

Several papers develop stochastic extensions of car-following models: Random fluctuations to the Optimal Velocity Model are implemented in Lehmann (2000); Laval, Toth & Zhou (2014) proposes and analyzes a stochastic "desired acceleration model"; Treiber, Kesting & Helbing (2006) and Treiber & Kesting (2018) include stochastic processes within IDM. All these studies use randomness to explain naturally occurring fluctuations in traffic flow. In contrast, this chapter focuses on stochastic processes that model perceptional errors which might trigger accidents, and it thereby rigorously extends our preliminary analysis in Berkhahn et al. (2018). In the context of (deterministic) emergency braking scenarios, accidents are also analyzed in Segata & Cigno (2013); similar ideas are discussed in Nekovee & Bie (2013).

The stochastic character of perception and other cognitive processes of drivers is studied in Hamdar et al. (2008); Tversky and Kahneman's prospect theory is used as a framework for decision making in the face of risk. Closely related to the present chapter is Mitra et al. (2018) where perception errors of autonomous vehicles are studied. Based on real data of Geiger, Lenz & Urtasun (2012), errors are calibrated using methods from time series analysis. The calibrated error models are incorporated into a commercial traffic simulation software, and the effects of errors are studied in test cases. The approach in Mitra et al. (2018) is complementary to ours. While we study the impact of errors on the number of accidents and traffic efficiency on an aggregate level, Mitra et al. (2018) does not capture the global impact of errors via variables such as traffic flow or the number of accidents, but focuses on its microscopic implications. Also, presumably due to substantial computational costs, the authors do not provide a statistical analysis of the consequences of the implemented errors – only four test trajectories in a braking scenario are presented, where an autonomous vehicle approaches a pedestrian. While Mitra et al. (2018) constructs a model that reflects many details of the collision dynamics of individual vehicles, our parsimonious model is sufficiently simple to study implications on the level of the whole traffic system.

The preceding papers mainly focus on unidirectional traffic, modeling vehicles on one-lane roads without intersections. Other models were developed for conflicting streams of traffic, e.g., unsignalized intersections or overtaking. Early contributions developed the idea of a *gap-acceptance function*: a certain time is needed to perform a potentially

conflicting maneuver, defining a critical gap. If two successive conflicting vehicles exceed this critical gap, the maneuver is accepted, otherwise rejected. Gap-acceptance functions and assumptions on arrival times of vehicles at an intersection admit an analysis of delay times and capacities (c.f., e.g., Tanner (1962) or Hawkes (1968)). Typically, heterogeneous human drivers obey different critical gaps. To reflect this issue, probability distributions are estimated from empirical data (e.g., Daganzo (1981), Mahmassani & Sheffi (1981)). Various refinements have been suggested; for example, Pollatschek, Polus & Livneh (2002) includes risk assessments when entering an intersection; impatience is reflected by increasing the risk tolerance, the longer a vehicle waits.

Besides queuing theoretic approaches, some microscopic traffic models for intersections include the concept of gap-acceptance. A model based on cellular automata is suggested in Esser & Schreckenberg (1997): a vehicle enters the intersection, if and only if enough cells of the conflicting stream of traffic are vacant. In the open source project SUMO, intersections are realized by comparing time slots in which potentially conflicting vehicles occupy the intersection (cf. Erdmann & Krajzewicz (2014)). An application of the gap-acceptance paradigm for lane-changing maneuvers was developed in Kesting, Treiber & Helbing (2007). In comparison, our model continuously detects potentially conflicting vehicles via trajectory extrapolation; potential conflicts trigger adjustments of the velocity of vehicles.

In the context of autonomous vehicles, models have been developed to demonstrate how traffic efficiency can be increased due to novel communication technologies. The benefits of inter-vehicle communication or communication with a central controller are studied in the context of *autonomous intersection management*, cf. Dresner & Stone (2008). The paper discusses incident mitigation techniques, but does not incorporate the possibility of endogenously occurring accidents. Auction and reservation based strategies for intersection management are, e.g., analyzed in Carlino, Boyles & Stone (2013). Sophisticated intersection management is not considered in this chapter; a comprehensive analysis of autonomous intersection managements in the face of risk and uncertainty would be an interesting topic for future research.

3.3 The Traffic Model

In this chapter, we model traffic on intersecting lanes and incorporate the possibility of accidents caused by perceptional errors. We assume that all vehicles move on prespecified paths, attempting to reach a target velocity. A vehicle accelerates, if its velocity is too low, unless conflicts with other vehicles are detected. We model two types of conflicts, namely an insufficient distance to the directly preceding vehicle, and vehicles crossing at intersections. In order to avoid collisions vehicles decelerate; the exact procedure is described below. Our model of an uncontrolled four-way intersection is illustrated in Figure 3.1.

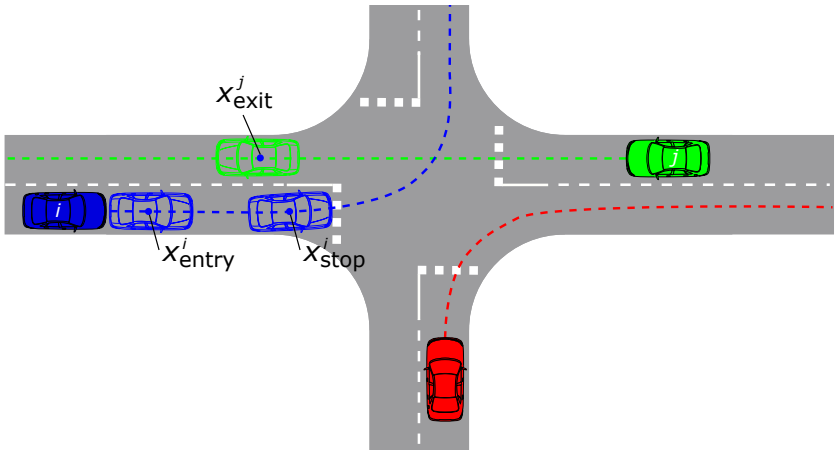


Figure 3.1: Left-turning at an intersection.

We first explain how we model a priority regime that mimics existing traffic regulations. Second, we describe a car-following model governing the movement on individual lanes. Third, we present a methodology for conflict detection. Fourth, we explain how vehicles adjust their speed. Our model incorporates errors due to *random misperception*. Estimates of distances and velocities are input quantities to the car-following model; conflict detection and reactions of vehicles depend on these variables. Our model assumes that the estimates of human drivers are subject to randomly fluctuating errors that are captured by suitable stochastic processes.

We begin with formal notation. The set $\mathcal{M} = \{1, 2, 3, \dots\}$ consists of all considered vehicles. We associate each vehicle $i \in \mathcal{M}$ with three stochastic processes with continuous paths, denoted by $(\varepsilon_t^{i,1})_{t\geq 0}, (\varepsilon_t^{i,2})_{t\geq 0}, (\varepsilon_t^{i,3})_{t\geq 0}$ that fluctuate around the value 1. The processes are multipliers that distort the true values of velocities and distances and thereby capture random misperception. Throughout the chapter, for each vehicle i, the first process $(\varepsilon_t^{i,1})$ refers to the misperception of vehicle i's own velocity; the second process $(\varepsilon_t^{i,2})$ models errors in the estimation of the velocity of other vehicles; the third process $(\varepsilon_t^{i,3})$ captures estimation errors of relevant distances. Further assumptions on the structure of these processes will be described in Section 3.5.

3.3.1 Priority Regime

The dynamics of uncontrolled intersections mimics German traffic regulations; of course, the approach could be amended to capture other countries. In Germany, "vehicles coming from the right have the right of way" unless specified otherwise.

While often applicable, this simple rule does not always produce a solution: If vehicles come from all directions at the same time, traffic may be deadlocked. In these situations, the following additional traffic rule applies Bundesministerium für Verkehr und digitale Infrastruktur (2017, Section 11 Special traffic situations, (3)):

> Moreover, anyone who, according to traffic rules, may proceed or otherwise

> has the right of way must relinquish this priority if the traffic situation so requires; a person not having the right of way may proceed only if the person having the right of way has signaled to them to do so.

In our model, we check if there is a cycle in the chain of priority. If this is the case, all waiting vehicles i observe independent exponentially distributed waiting times $t^i_{\text{solve}} \sim \text{Exp}(\lambda),\ \lambda > 0$, with expectation $\mathbb{E}(t^i_{\text{solve}}) = \lambda^{-1}$; the vehicle whose clock rings first will give up its priority.

Remark 3.3.1. *Of course, the behavior stipulated by traffic regulations is not efficient. For autonomous vehicles, one could envision control algorithms that lead to both safer and more efficient outcomes. Research on this topic runs under the keyword* autonomous intersection management *(cf. also Section 3.2).*

3.3.2 Car-Following Model

The paths of vehicles, also called trajectories, lie on one-dimensional curves that describe the geometry of the traffic system; this is illustrated in Figure 3.1. In our model, the paths of vehicles are prespecified and fixed; speed can be adjusted. Vehicles with the same trajectory follow each other. Their behavior is modeled by the *Intelligent Driver Model with Random Misperception* (IDMrm) as developed by our research group in Berkhahn et al. (2018). IDMrm is a stochastic extension of the classical IDM with a bound on maximal deceleration in which perceptional errors are incorporated. On each of the one-dimensional curves that capture potential paths of vehicles, we fix an origin. For any vehicle i that moves along this curve we denote by $x^i(t)$ the distance of the vehicle's position at time t to the origin along the section of the curve, i.e., the arc length of the corresponding segment of its trajectory; the time derivative $v^i(t)$ of $x^i(t)$ is the velocity of vehicle i at time t.

Vehicles are controlled on the basis of measurements of distances and velocities. But these measurements are subject to errors. Distortions are captured by multiplicative factors $(\varepsilon^{i,1}_t), (\varepsilon^{i,2}_t)$, and $(\varepsilon^{i,3}_t)$. On its one-dimensional trajectory, each vehicle computes its acceleration based on the perceived values of its own velocity $\varepsilon^{i,1}_t v^i(t)$, its perceived distance to the preceding vehicle $\Delta_{\text{per}} x^i(t)$ and its perceived approaching rate $\Delta_{\text{per}} v^i(t)$. The identity of the preceding vehicle may change, since vehicles can turn at the intersection, and we denote this vehicle by $i_{\text{pre}}(t)$. Letting $\Delta x^i(t)$ be the exact distance to the preceding vehicle along the path, we may formally define the perceived quantities:

$$\Delta_{\text{per}} v^i(t) := \varepsilon^{i,1}_t v^i(t) - \varepsilon^{i,2}_t v^{i_{\text{pre}}(t)}(t), \tag{3.3.1}$$

$$\Delta_{\text{per}} x^i(t) := \varepsilon^{i,3}_t \Delta x^i(t) \tag{3.3.2}$$

For a vehicle i on a one-dimensional line its acceleration is computed as

$$a^i_{\mathrm{IDMrm}}(t) := a^i_{\max}\left(1 - \left(\frac{\varepsilon^{i,1}_t v^i(t)}{v^i_{\mathrm{d}}}\right)^{\delta} - \left(\frac{s^*(\varepsilon^{i,1}_t v^i(t), \Delta_{\mathrm{per}} v^i(t))}{\Delta_{\mathrm{per}} x^i(t)}\right)^2\right),$$

where $s^*(s_1, s_2) = s_0 + s_1 T + \frac{s_1 s_2}{2\sqrt{a^i_{\max} b}}$; we set $s^*(\varepsilon^{i,1}_t v^i(t), \Delta_{\mathrm{per}} v^i(t)) \cdot (\Delta_{\mathrm{per}} x^i(t))^{-1} := 0$ if there is no preceding vehicle. The quantity $a^i_{\max} > 0$ is the maximal acceleration of the i-th vehicle, and $v^i_{\mathrm{d}} > 0$ denotes its desired velocity. The other parameters originate from the classic IDM model, and we refer to Treiber, Hennecke & Helbing (2000) for a detailed explanation.

Remark 3.3.2. *In* (3.3.1) *&* (3.3.2)*, misperception of human drivers is modeled by multiplicative errors. Multiplicative errors are relative errors, and their advantage is that their size scales with the magnitude of the true values. Alternatively, additive errors could be chosen. The simulation method described in Section 3.4 could easily be adapted.*

3.3.3 Conflict Detection at Intersections

The control of individual vehicles and traffic flow depends on priority regimes. For each vehicle i, we denote by $\mathcal{M}^i_{\mathrm{rel}}(t) \subseteq \mathcal{M}$ the family of vehicles to which it has to give way. These are vehicles approaching the intersection which are coming from the right; this includes oncoming vehicles when vehicle i is turning left.

Vehicles always stay on their prespecified paths, signaling their turning intentions correctly. At time t, trajectories of other vehicles are extrapolated into the future for a fixed time horizon of length t^* on the basis of potentially distorted estimates of distances and velocities (see Pascucci et al. (2015) for more details on trajectory extrapolation). Using the extrapolated trajectories, one computes an estimate of vehicle i's distance to another vehicle j at future time u which is denoted by $\hat{d}^{ij}(u)$; vehicles' paths may be located on different one-dimensional curves, and for this reason we measure $\hat{d}^{ij}(u)$ as the usual Euclidean distance in the two-dimensional plane into which the trajectories are embedded. Note that $\hat{d}^{ij}(u)$ implicitly depends on t, but we suppress this dependence in the notation, since it will always be clear from the context. As in the context of car-following, we assume that distances to other vehicles are misperceived. Analogously, we assume that vehicle i perceives its own distance to vehicle j as $\varepsilon^{i,3}_t \hat{d}^{ij}(u)$, i.e., the estimate $\hat{d}^{ij}(u)$ is distorted by the multiplier $\varepsilon^{i,3}_t$.

If $j \in \mathcal{M}^i_{\mathrm{rel}}(t)$, vehicle i detects a conflict at time t, if $\varepsilon^{i,3}_t \hat{d}^{ij}(u) < d_{\mathbf{s}}$ for a safety threshold $d_{\mathbf{s}} \geq 0$ and $t \leq u \leq t + t^*$, i.e., the (distorted) extrapolated distance between the two vehicles i and j falls below the safety threshold at a future time horizon. In addition, if vehicle i is in the area of the intersection and detects a conflict with another vehicle j that has the right of way, vehicle i keeps the conflict in mind until vehicle j leaves the area of the intersection. In order to make this precise, we introduce fixed

locations x^i_{stop}, x^i_{entry}, and x^j_{exit}:

- x^i_{stop} is the position on i's trajectory where vehicle i should stop in order to let conflicting vehicles j pass;
- x^i_{entry} is the position such that vehicle i, driving with desired speed v^i_d, is able to come to a complete stop at x^i_{stop} using its maximal deceleration;
- vehicle j has passed the intersection if it has reached x^j_{exit};
- we say i is in the area of the intersection at time t if $x^i_{\text{entry}} < x^i(t) < x^i_{\text{exit}}$.

In summary, we define the set $\mathcal{M}^i_{\text{conflict}}(t)$ of conflicting vehicles by

$$\begin{aligned}\mathcal{M}^i_{\text{conflict}}(t) := &\left\{ j \in \mathcal{M}^i_{\text{rel}}(t) \mid \exists\, u \in [t, t+t^*] \colon \varepsilon^{i,3}_t \hat{d}^{ij}(u) < d_{\mathbf{s}} \right\} \\ &\cup \left(\bigcup_{u<t} \left\{ j \in \mathcal{M}^i_{\text{conflict}}(u) \mid x^i_{\text{entry}} < x^i(u), x^j(t) < x^j_{\text{exit}} \right\} \right).\end{aligned}$$

3.3.4 Conflict Reaction

If the set of conflicting vehicles $\mathcal{M}^i_{\text{conflict}}(t)$ is nonempty, vehicle i reacts to this situation. We distinguish two cases: stopping, or decelerating when stopping is unnecessary.

- Complete stop: If vehicle i is in position $x^i(t)$ with velocity $v^i(t)$, the constant (negative) acceleration to stop at x^i_{stop} equals

$$a^i_{\text{stop}}(t) := -\frac{(v^i(t))^2}{2(x^i_{\text{stop}} - x^i(t))}.$$

  The duration of this maneuver is $t^i_{\text{stop}}(t) := -v^i(t)/a^i_{\text{stop}}(t)$.

- Deceleration: Stopping is not always necessary. Consider a vehicle i and a conflicting vehicle j. We assume that vehicle i bases its acceleration on a simplified prediction of vehicle j by assuming that j's velocity is fixed. The time it would take for vehicle j to leave the intersection with fixed speed $v^j(t)$ is $t^j_{\text{exit}}(t) := (x^j_{\text{exit}} - x^j(t))/v^j(t)$. If $t^i_{\text{stop}}(t) > t^j_{\text{exit}}(t)$, vehicle i does not intend to stop, but only to slow down. The constant deceleration such that vehicle i arrives at x^i_{stop} at the predicted time equals

$$a^{ij}_{\text{break}}(t) := \left(\frac{x^i_{\text{stop}} - x^i(t)}{t^j_{\text{exit}}(t)} - v^i(t) \right) \frac{2}{t^j_{\text{exit}}(t)}.$$

Conflict reaction to vehicles $j \in \mathcal{M}^i_{\text{conflict}}(t)$ is modeled by bounding the acceleration from above by

$$a^{ij}_{\text{conflict}}(t) := \begin{cases} a^{ij}_{\text{break}}(t), & \text{if } t^i_{\text{stop}}(t) > t^j_{\text{exit}}(t), \\ a^i_{\text{stop}}(t), & \text{if } t^i_{\text{stop}}(t) \le t^j_{\text{exit}}(t). \end{cases}$$

We do, however, not assume that these quantities depend on the correct distances or velocities, but on their perceived values and replace the arguments of the functions accordingly, i.e., $v^i(t)$ by $\varepsilon_t^{i,1}v^i(t)$, $x^i_{\text{stop}} - x^i(t)$ by $\varepsilon_t^{i,3}(x^i_{\text{stop}} - x^i(t))$, $x^j_{\text{exit}} - x^j(t)$ by $\varepsilon_t^{i,3}(x^j_{\text{exit}} - x^j(t)$, and $v^j(t)$ by $\varepsilon_t^{i,2}v^j(t)$, $j \in \mathcal{M} \setminus \{i\}$.

3.3.5 Intersection Dynamics Subject to Random Misperception

The motion of the vehicles can be expressed as a system of coupled random ordinary differential equations:

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}x^i(t) = \max\{v^i(t), 0\}, \\ \frac{\mathrm{d}}{\mathrm{d}t}v^i(t) = \max\left\{a^i_{\min}, \min\left\{a^i_{\text{IDMrm}}(t),\ \min_{j\in\mathcal{M}^i_{\text{conflict}}(t)} a^{ij}_{\text{conflict}}(t)\right\}\right\}, \\ x^i(t^i_0) = 0,\ v^i(t^i_0) = v^i_0,\ t \geq t^i_0,\ i \in \mathcal{M} \end{cases} \tag{3.3.3}$$

Velocities are bounded from below by 0. The minimal acceleration of vehicle i is set to $a^i_{\min}$; this is both realistic and necessary, if accidents are admissible. The acceleration of a vehicle i is the minimum of $a^i_{\text{IDMrm}}(t)$ (to control the distance to the preceding vehicle) and $\min_{j\in\mathcal{M}^i_{\text{conflict}}(t)} a^{ij}_{\text{conflict}}(t)$ (to solve all conflicts in the intersection simultaneously). Each vehicle i enters the system on its path with an initial velocity $v^i_0 \geq 0$ at time t^i_0 and is removed from the system once it reaches the end of its path.

Remark 3.3.3. *The described conflict reaction is closely related to microscopic gap-acceptance models. Instead of critical gaps in time, we measure the distance to conflicting vehicles and adjust the velocity accordingly. Gap-acceptance models reflect heterogeneity via probability distributions. A similar approach could be applied to the safety threshold in our model and the adjustment of the velocities.*

Remark 3.3.4. *In this chapter, we study uncontrolled bi-directional two-lane intersections. Our approach can be generalized to other priority regimes:*

- *Prioritized roads can be modeled by adjusting $\mathcal{M}^i_{\text{rel}}(t)$, i.e., the set of vehicles to which one needs to give way.*
- *To model a signalized intersection, one could include another time-dependent acceleration term $a^i_{\text{signal}}(t)$ forcing vehicle i to decelerate on red. The resulting acceleration would be $\min\{a^i_{\text{IDMrm}}(t), a^i_{\text{signal}}(t), \min_{j\in\mathcal{M}^i_{\text{conflict}}(t)} a^{ij}_{\text{conflict}}(t)\}$. Also, other relevant conflicts and other types of misperception could be integrated in a more comprehensive model.*

Remark 3.3.5. *Mathematically, the model in* (3.3.3) *is a system of coupled random ordinary differential equations. Random ordinary differential equations (RODEs) are ordinary differential equations whose right-hand side depends on some stochastic process. Pathwise, these are non-autonomous classical ordinary differential equations and can be*

solved by deterministic calculus. Local existence of a weak solution is guaranteed by Theorem 3.7.1 (cf. Section 3.7), i.e., we find trajectories $x^i(t)$ which satisfy (3.3.3) *for Lebesgue almost all times. However, many classical numerical methods are inappropriate due to the roughness of the paths of the stochastic processes. Suitable schemes will be explained in the next section.*

Computing $a^{ij}_{\text{conflict}}(t)$ is expensive. This effort can be reduced by virtue of the following simple lemma: If vehicle i anyway intends to stop due to some conflicting vehicle j, it does not need to analyze other conflicting vehicles anymore. The proof is trivial.

Lemma 3.3.6. *Let $i \in \mathcal{M}$ and $j \in \mathcal{M}^i_{\text{conflict}}(t)$. The following statements hold:*

(i) $a^i_{\text{stop}}(t) \le a^{ij}_{\text{break}}(t)$,

(ii) If there exists $j^* \in \mathcal{M}^i_{\text{conflict}}(t)$ *such that* $a^{ij^*}_{\text{conflict}}(t) = a^i_{\text{stop}}(t)$, *then*

$$\min_{j \in \mathcal{M}^i_{\text{conflict}}(t)} a^{ij}_{\text{conflict}}(t) = a^i_{\text{stop}}(t).$$

Random misperception may trigger accidents, i.e., collisions of vehicles. In order to capture this, we denote by $A^i(t)$ the area that is occupied by vehicle $i \in \mathcal{M}$ at time t; this is modeled by an ellipse in the two-dimensional plane. A *collision* occurs if $A^i(t) \cap A^j(t) \neq \emptyset$ for $i, j \in \mathcal{M}$ and $t \ge 0$. In this case, we set the velocity of the vehicles i and j to 0 and adjust the dynamics of the traffic system (3.3.3) accordingly. Moreover, we trigger an exponentially distributed waiting time

$$t_{\text{removal}} \sim \text{Exp}(\gamma), \quad \gamma > 0,$$

with expectation $\mathbb{E}(t_{\text{removal}}) = 1/\gamma$. Meanwhile, other vehicles may crash into the existing collision; however, after t_{removal} has passed, *all* vehicles that are involved in this particular accident are removed from the model. Of course, simultaneously other accidents may occur at other places.

3.3.6 Calibration

The model includes three dimensions: car-following, conflict detection and reaction, and misperception. The aim of the model is to provide an experimental lab that allows to envision future traffic systems. In particular, future advanced driver-assistance systems will modify the behavior of human drivers, and we will also study the coexistence of human drivers and autonomous vehicles. This implies that the model cannot fully be calibrated to statistical data. In fact, the purpose of the model is to generate artificial data of novel traffic systems that do not yet exist. However, calibration needs to be discussed in the context of suitable benchmarks. Our model will deviate from these benchmarks, and it can be compared to them.

- **Car-following models** can be calibrated from either aggregate or disaggregate traffic data (see, e.g., Treiber, Hennecke & Helbing (2000), Kesting & Treiber (2008), Ossen & Hoogendoorn (2008), Punzo, Ciuffo & Montanino (2012), Sharma, Zheng & Bhaskar (2019)). These may also serve as benchmark models of autonomous vehicles; however, their behavior will deviate from current traffic data due to the increased capabilities of the vehicles and their flexible and partially unknown future design.
- **Conflict detection and reaction** is captured by a stylized model in this chapter. A benchmark can be estimated on the basis of similar methodologies previously suggested for gap-acceptance models (see, e.g., Daganzo (1981), Mahmassani & Sheffi (1981), Pollatschek, Polus & Livneh (2002)).
- **Perception errors** of human drivers could be estimated on the basis of statistical data (see, e.g., Evans & Rothery (1974), Taieb-Maimon & Shinar (2001)). In the future, the size of these errors may be further reduced by improved technology, e.g., advanced driver-assistance systems.

3.4 Simulation Method

On a pathwise level, RODEs are non-autonomous ordinary differential equations; classical first order methods from deterministic calculus can be applied to solve them. RODEs depend, however, on stochastic processes which typically possess paths of unbounded variation that are nowhere differentiable. Typical examples are (fractional) Brownian motion and related processes such as the Ornstein-Uhlenbeck process that we will consider in this chapter. Due to their insufficient smoothness, many classical numerical methods are not appropriate; the reason is that standard arguments for the error analysis of numerical schemes are not applicable anymore, since these are often based on Taylor expansions requiring sufficient regularity (we refer to Han & Kloeden (2017) for a more detailed discussion of this issue).

These challenges are addressed by simulation schemes that are specifically taylored for RODEs. To approximate the solutions, we employ the γ-RODE-Taylor scheme (cf. Jentzen & Kloeden (2009)). This method requires that there exists $\theta^i = (\theta^{i,1}, \theta^{i,2}, \theta^{i,3})^\top \in (0,1]^3$ such that each component process $(\varepsilon_t^{i,k})_{t\geq 0}$ is Hölder continuous for all exponents $\eta^{i,k}$ satisfying $0 < \eta^{i,k} < \theta^{i,k}$, $k = 1,2,3$ (cf. Assumption 3.1 in Jentzen & Kloeden (2009)).

We consider a time discretization $\mathbb{T} = \{t_0, t_1, ...\}$; for the individual time points and for all $i \in \mathcal{M}$ we determine an approximate solution $\begin{pmatrix} x_k^i \\ v_k^i \end{pmatrix}_{k=0,1,2,...}$ to our system (3.3.3).

Consider the iteration interval $[t_k, t_{k+1}]$. In order to compute for a fixed $i \in \mathcal{M}$ the update $\begin{pmatrix} x_{k+1}^i \\ v_{k+1}^i \end{pmatrix}$ we treat $\begin{pmatrix} x_k^j \\ v_k^j \end{pmatrix}$ for $j \neq i$ as fixed exogenous input values. We

compute $\mathcal{M}^i_{\text{conflict}}(t_k)$ and $i_{\text{pre}}(t_k)$, and fix these values for the iteration interval $[t_k, t_{k+1}]$. Under these assumptions, the right-hand side of the evolution equation for i, as given in (3.3.3), can be rewritten in terms of a function $f\colon \mathbb{R}^3 \times \mathbb{R}^2 \to \mathbb{R}^2$ with arguments $(\varepsilon_t^{i,1}, \varepsilon_t^{i,2}, \varepsilon_t^{i,3})^\top \in \mathbb{R}^3$ and $(x^i(t), v^i(t))^\top \in \mathbb{R}^2$. We replace f by a suitable infinitely differentiable approximation that we again denote by f.

In the case studies in the next section, we consider error processes $(\varepsilon_t^{i,1})_{t\geq 0}$, $(\varepsilon_t^{i,2})_{t\geq 0}$, $(\varepsilon_t^{i,3})_{t\geq 0}$ with $\theta^i = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})^\top$ (cf. Section 3.5.1). Setting $\gamma = 1$, we obtain the pathwise γ-RODE-Taylor scheme

$$\Phi_1(z,t,h) := z \;+\; h \cdot f(\varepsilon_t^i, z) \;+\; \frac{h}{n} \sum_{k=1}^{3} \partial_{w_k} f(\varepsilon_t^i, z) \sum_{j=1}^{n-1} \Delta\varepsilon_{t,\tau_j}^{i,k},$$

where $\tau_j = t + \frac{j}{n} \cdot h$, $\Delta\varepsilon_{t,\tau_j}^{i,k} = \varepsilon_{\tau_j}^{i,k} - \varepsilon_t^{i,k}$ and $n = \left\lceil h^{1-\frac{2}{1-2\xi}} \right\rceil$ for $\xi > 0$ small. Here, $\lceil\cdot\rceil$ is a Gauss-bracket, and $\partial_{w_1}, \ldots, \partial_{w_3}$ denote the partial derivatives with respect to the three error components of f. The derivatives of f are approximated by difference quotients. The stepwise order of convergence equals $\gamma = 1$. The approximation of the solution of (3.3.3) for vehicle $i \in \mathcal{M}$ at time t_{k+1} is given by

$$\begin{pmatrix} x_{k+1}^i \\ v_{k+1}^i \end{pmatrix} = \Phi_1 \left(\begin{pmatrix} x_k^i \\ v_k^i \end{pmatrix}, t_k, \Delta t_{k+1} \right)$$

with $\Delta t_k := t_k - t_{k-1}$.

3.5 Case Study

Performance Measures. We evaluate our model in terms of risk and efficiency. We study

- the *number of accidents*, the *number of collided vehicles*, and the *number of collided vehicles per accident* as quantitative measures of the riskiness of the system, and
- *network traffic flow* as a measure of system efficiency.

The length of the simulation period is denoted by T_{sim}.

Network Traffic Flow. We assign to each vehicle $i \in \mathcal{M}$ a final destination dest_i on its path. Network traffic flow Q is measured by the number of vehicles $i \in \mathcal{M}$ per time unit that reach their destinations:

$$Q = \frac{\text{card}\{j \in \mathcal{M} \colon \exists\, t \leq T_{\text{sim}} \colon x^j(t) = \text{dest}_j\}}{T_{\text{sim}}},$$

where card denotes the cardinality. The corresponding sample mean is denoted by $\hat{Q}$.

Number of Accidents. A first proxy for the safety of traffic systems is given by the number of accidents per time unit

$$f_{\text{acc}} = T_{\text{sim}}^{-1} \cdot \text{card}\big\{\emptyset \neq M \subset \mathcal{M} \colon \exists\, t \leq T_{\text{sim}}\ \forall\, i \in M \colon A^i(t) \cap A^{M\setminus\{i\}}(t) \neq \emptyset \\ \text{and } \forall\, t \leq T_{\text{sim}} \colon A^M(t) \cap A^{M^c}(t) = \emptyset\big\},$$

where $M^c := \mathcal{M} \setminus M$ and $A^M(t) := \bigcup_{i\in M} A^i(t)$. The corresponding sample mean is denoted by $\hat{f}_{\text{acc}}$. An accident is the event that multiple cars are jointly involved in collisions. A collision occurs, if the associated areas of two vehicles intersect.

Number of Collided Vehicles. The number of vehicles per time unit that are involved in accidents is given by

$$f_{\text{veh}} = T_{\text{sim}}^{-1} \cdot \text{card}\big\{i \in \mathcal{M} \colon \exists\, t \leq T_{\text{sim}}\ \exists\, j \in \mathcal{M} \setminus \{i\} \colon A^i(t) \cap A^j(t) \neq \emptyset\big\}.$$

The corresponding sample mean is denoted by $\hat{f}_{\text{veh}}$.

Number of Collided Vehicles per Accident. A measure for the average severity of an accident is the number of collided vehicles divided by the number of accidents:

$$g_{\text{veh/acc}} = \frac{f_{\text{veh}}}{f_{\text{acc}}}.$$

Its sample mean is denoted by $\hat{g}_{\text{veh/acc}}$.

3.5.1 Misperception Model

Perception of the environment is subject to errors. In our multiplicative error model, mean-reverting processes are capable of capturing noisy deviations; mean-reverting processes are stochastic processes that randomly fluctuate around fixed values. In contrast, permanent malfunctions can, for example, be modeled by jump processes such as continuous time Markov chains. Both aspects may also be combined in a joint model.

In this chapter, we focus only on the first dimension of misperception, i.e., random noise that distorts perceived quantities around their true values. An important example of a mean-reverting stochastic process that fluctuates around a constant level is the Ornstein-Uhlenbeck process. This process was also used in Berkhahn et al. (2018) to model perceptional errors. We refer to Mitra et al. (2018) for potentially more realistic, but less tractable alternative approaches.

Definition 3.5.1 (Ornstein-Uhlenbeck Process). *Let $\beta \in \mathbb{R}$ and $\alpha, \sigma > 0$. A stochastic process $(\varepsilon_t)_{t\geq 0}$ is called an* Ornstein-Uhlenbeck process, *if $\varepsilon_0 = a \in \mathbb{R}$ and $(\varepsilon_t)_{t\geq 0}$ solves the following stochastic differential equation:*

$$d\varepsilon_t = \alpha(\beta - \varepsilon_t)dt + \sigma dW_t,$$

where $(W_t)_{t\geq 0}$ *denotes a one-dimensional standard Brownian motion.*

Typical simulated paths are shown in Figure 3.2. For more details regarding simulation and interpretation, we refer to Berkhahn et al. (2018) and Glasserman (2003).

In our case studies, we analyze two scenarios – a homogeneous and a heterogeneous scenario:

(i) In the homogeneous case, we assume that all $(\varepsilon_t^{i,1})$, $(\varepsilon_t^{i,2})$ and $(\varepsilon_t^{i,3})$ are independent and identically distributed Ornstein-Uhlenbeck processes with parameters $a = \beta = \alpha = 1$ and varying values of the parameter σ which controls the volatility of the process.

(ii) In the heterogeneous case, we assume that vehicles with and without misperception coexist. For the latter, we assume $\varepsilon_t^{i,1} = \varepsilon_t^{i,2} = \varepsilon_t^{i,3} \equiv 1$, capturing perfect perception.

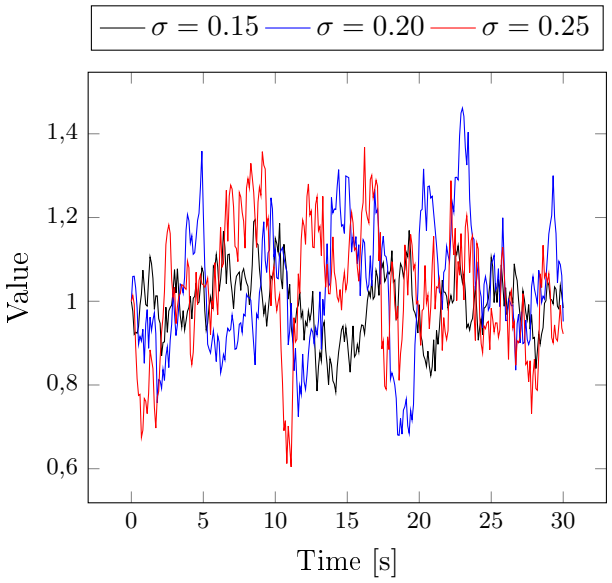


Figure 3.2: Simulated paths of an Ornstein-Uhlenbeck process (ε_t) for different values of σ with $\alpha = 1$, $\varepsilon_0 = 1$, $\beta = 1$.

3.5.2 Scenario Description

The intersection consists of two two-lane roads of length 210 m, each having a width of 10 m. For this case study, we generate vehicles in the following way: The intersection can be approached from four directions, i.e., vehicles are generated at four origins. We create vehicles with an exponentially distributed headway such that the expected rate of vehicles per time unit at each source is 150 $^{\text{veh}}/_{\text{h}}$. Larger gaps may randomly emerge between vehicles – allowing vehicles to turn that wait at the intersection. If substantial

traffic jams occur, our simulation delays the generation of new vehicles such that traffic flow cannot completely break down.

When a vehicle is generated, it chooses with probability 1/3 one of the following paths: turn right, go straight, turn left. Its initial velocity is set to the velocity of the preceding vehicle. If there is no preceding vehicle, it starts with its desired velocity.

We simulate the traffic system for a duration of $T_{\text{sim}} = 600\,\text{s}$. To reach a representative, potentially stationary state of the Markovian model, we implement a burn-in period of 100 s. Data for the computation of the relevant statistics are recorded afterwards. The model is simulated on an equidistant time grid with $\Delta t_k \equiv 0.1\,\text{s}$. We sample repeatedly and compute averages of Q, f_{acc}, and f_{veh} from the empirical distributions.

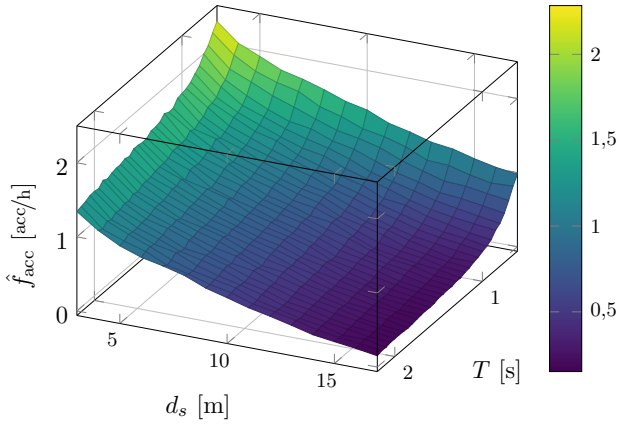
Remark 3.5.2. *A generation rate of* $150\,\text{veh/h}$ *per source corresponds to a scenario of low to moderate traffic volume. In the error-free case, the generation rate could be increased. Real-world traffic regulations require the installation of traffic signals if traffic volumes are slightly larger: According to German regulations (cf. Reinhold Baier et al. (2006)), a traffic light should be installed if total traffic flow exceeds* $800\,\text{veh/h}$*; US regulations (cf. U. S. Department of Transportation, Federal Highway Administration (2013)) specify that the sum of traffic flow on the major lanes should not exceed* $500\,\text{veh/h}$ *and the maximum of flow on the minor lanes should be below* $150\,\text{veh/h}$*. In our case study, all sources have the same generation rate of maximally* $150\,\text{veh/h}$ *without any distinction between major and minor lanes.*

3.5.3 Simulation Results

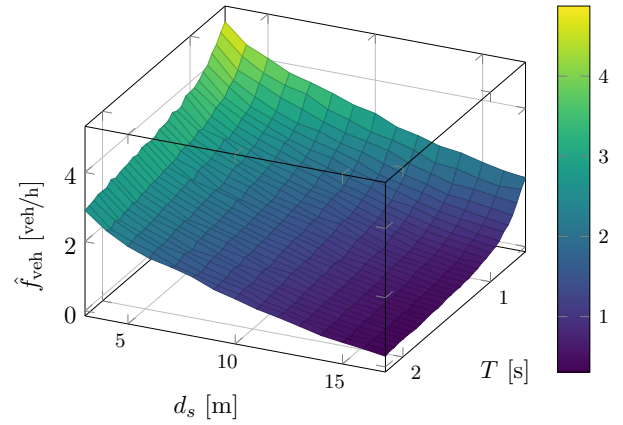
We independently simulate the four-way intersection 20,000 times. The parameters underlying the simulation are displayed in Section 3.8. We separately analyze the homogeneous and the heterogeneous case and study the effect of two important control parameters:

- We vary the time headway T of the vehicles (T is a parameter of the car-following model IDMrm), and
- the safety distance $d_{\mathbf{s}}$ which controls whether an oncoming vehicle is classified as conflicting or not.

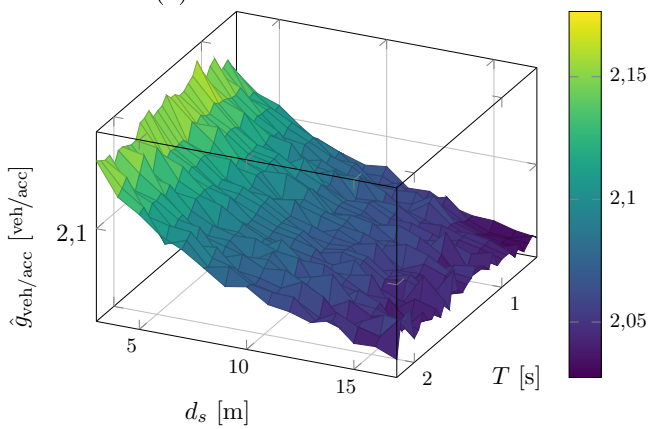
Homogeneous Traffic. We assume that all vehicles are subject to misperception as described in Section 3.5.1. This occurs if only human drivers are present. Figure 3.3 displays the quantitative measures of the risk of the system: Figure 3.3a shows the number of accidents, Figure 3.3b the number of collided vehicles. Of course, safety increases when time headway and safety threshold are increased. Both graphs are quite similar in shape, since for all parameters accidents involve on average about 2 to 2.2 vehicles, see Figure 3.3c. Due to low velocities ($10\,\text{m/s}$) and a moderate rate at which vehicles are generated, vehicles are able to react to most collisions.



(a) Number of accidents

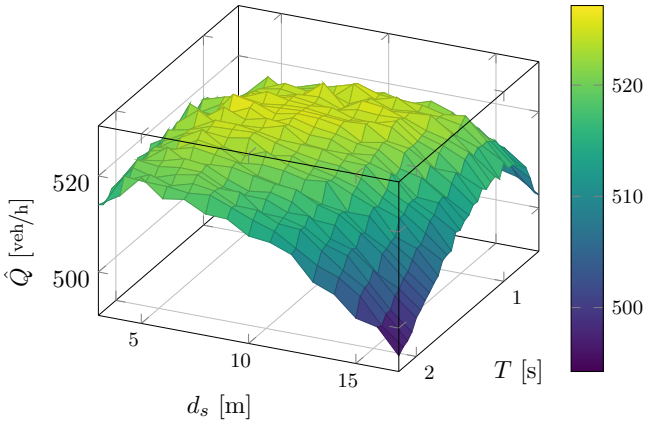


(b) Number of collided vehicles

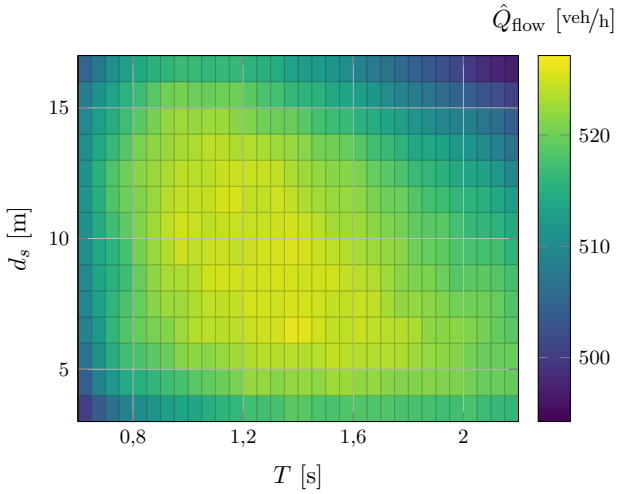


(c) Number of collided vehicles per accident

Figure 3.3: Number of accidents, number of collided vehicles and number of collided vehicles per accident for $\sigma = 0.2$ and varying d_s and T with 20,000 independent simulations for each parameter combination.



(a) Surface plot



(b) Top view

Figure 3.4: Average traffic flow for $\sigma = 0.2$ and varying $d_{\mathbf{s}}$ and T with $20,000$ independent simulations for each parameter combination.

The efficiency of the traffic system is shown in Figure 3.4: Figure 3.4a shows a surface plot, Figure 3.4b a top view of the same graph. The function is strictly concave and has a unique maximum. This captures the tradeoff between risk and efficiency: If $d_{\mathbf{s}}$ or T are small, the number of accidents is high and the intersection is blocked. Hence, traffic flow is low. With increasing $d_{\mathbf{s}}$ and T, the number of accidents decreases and traffic flow increases. However, traffic becomes inefficient, if $d_{\mathbf{s}}$ or T are large, i.e., if vehicles drive too carefully; in this case, traffic flow decreases. Accidents do still occur in the most efficient traffic flow scenario. The findings generalize our preliminary results in Berkhahn et al. (2018).

Our model admits two types of collisions: rear-end collisions and collisions at the intersection. Figure 3.5 depicts the fraction of rear-end accidents among all accidents. Within the considered parameter range, T has a stronger influence, but also $d_{\mathbf{s}}$ has an impact by reducing the number of collisions at the intersections. The dependence of Q

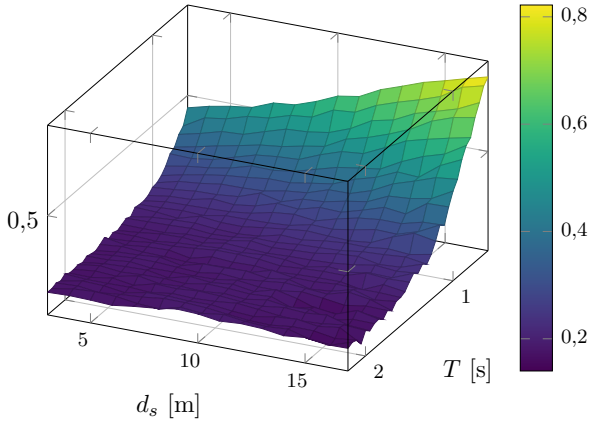


Figure 3.5: Fraction of rear-end accidents for $\sigma = 0.2$ and varying $d_{\mathbf{s}}$ and T with $20,000$ independent simulations for each parameter combination.

and f_{acc} on σ is shown in Figure 3.6: Of course, with increasing volatility more accidents occur and traffic flow decreases. These effects are superlinear.

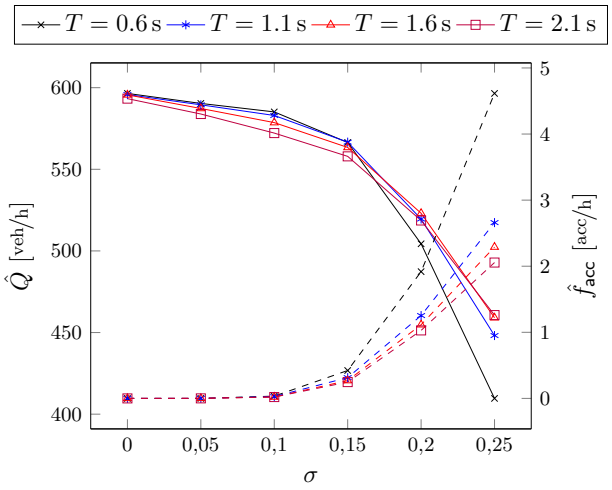
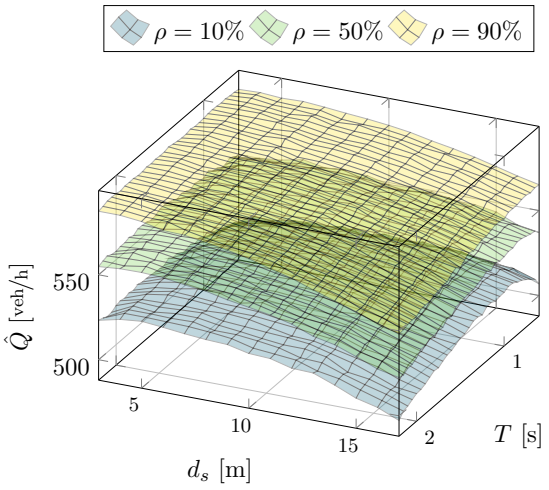


Figure 3.6: Average network traffic flow and number of accidents for $d_{\mathbf{s}} = 5\,\text{m}$, fixed T, and varying σ with $20,000$ independent simulations for each parameter combination. Dashed lines correspond to number of accidents, solid lines to flows.

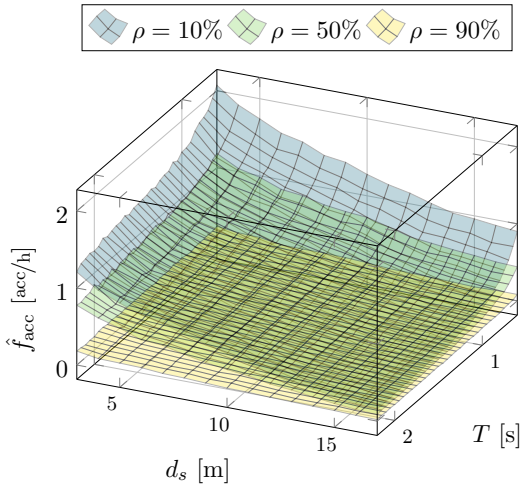
Heterogeneous Traffic. We now consider the coexistence of vehicles with perfect perfection and misperception, analyzing their impact on traffic efficiency and accidents. Perfect perception might be associated with autonomous vehicles, while random misperception occurs in human drivers, possibly using advanced driver-assistance systems that tame the size of errors. The parameter $\rho \in [0, 1]$ denotes the penetration rate of vehicles with perfect perception. Sampling independent Bernoulli-distributed random variables with parameter ρ, we randomly pick the type of each vehicle.

Figure 3.7 shows the average traffic flow and number of accidents. As in the homoge-

neous case, we vary the time headway T and the safety distance $d_{\mathbf{s}}$. Here, we consider penetration rates $\rho \in \{10\%,\ 50\%,\ 90\%\}$. Increasing the penetration rate of error-free vehicles, reduces the number of accidents and, consequently, increases traffic flow. Figure 3.8 studies this behavior in more detail for selected values of T and $d_{\mathbf{s}}$. While these qualitative features are expected, key to our model are exact quantitative characterizations of tradeoffs for each given set of model parameters.



(a) Traffic flow



(b) Number of accidents

Figure 3.7: Average network traffic flow and number of accidents for different penetration rates ρ, $\sigma = 0.2$, and varying $d_{\mathbf{s}}$ and T with $20,000$ independent simulations for each parameter combination.

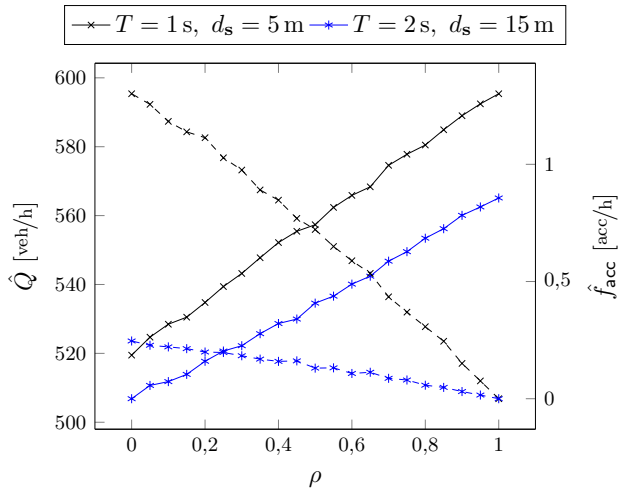


Figure 3.8: Average network traffic flow and number of accidents for $\sigma = 0.2$, fixed T and $d_\mathbf{s}$, and varying ρ with 20,000 independent simulations for each parameter combination. Dashed lines correspond to number of accidents, solid lines to flows.

3.6 Conclusion

This chapter studies safety and efficiency of future traffic systems. We develop a rigorous microscopic model for traffic at intersections. Random misperception of human drivers may trigger accidents. The system is captured by random ordinary differential equations (RODEs) that require specific numerical schemes for their efficient simulation. The proposed setup is general and can be extended to more complex traffic scenarios.

Accidents are a consequence of perceptional errors of human drivers whose probability and size may depend on advanced driver-assistance systems. Our case study clearly illustrates the tradeoff between risk and efficiency. If too many accidents occur, traffic breaks down; but if the safety margins are very large, the system becomes inefficient. Besides delivering these expected qualitative results, the model provides a quantitative simulation lab of future traffic systems. Our model captures both homogeneous vehicles and the coexistence of autonomous vehicles and human drivers. Our techniques can easily be modified to allow for more than two types of drivers.

The proposed approach can also be implemented in more comprehensive models, yet computational costs increase. Other road types, such as roundabouts and multi-lane roads, could be modeled accordingly. In the context of lane-changing, misperception will also imply the occurrence of accidents. The model is capable of characterizing improved or even optimal designs of autonomous vehicles if human drivers that make errors are present and coexist with autonomous vehicles. Another interesting issue are errors of autonomous vehicles, e.g., misclassification of objects; but this important subject requires a more comprehensive model than ours which includes also other participants besides cars.

Finally, future research should, on the one hand, derive surrogate models from the microscopic traffic system. On the other hand, a more detailed analysis of accidents that includes incurred losses may provide additional guidance for the design of traffic systems with autonomous vehicles and for suitable risk management solutions.

3.7 Appendix: Theoretical Existence Result

General existence results for ordinary differential equations can be found in the literature, e.g., the Theorem of Constantin Carathéodory, see Carathéodory (1918) or Han & Kloeden (2017, Chapter 2.1).

Let $B_r(x_0) \subseteq \mathbb{R}^d$ be the open ball with radius $r > 0$ centered in $x_0 \in \mathbb{R}^d$.

Theorem 3.7.1 (Carathéodory's Existence Theorem). *Let $f\colon [0,T]\times B_r(x_0) \to \mathbb{R}^d$ such that*

(i) $f(t,x)$ is continuous in x for almost every $t \in [t_0,T]$,

(ii) $f(t,x)$ is Lebesgue measurable in t for all $x \in B_r(x_0)$,

(iii) $|f(t,x)| \leq M(t)$ for all $x \in \mathbb{R}$ and almost every $t \in [t_0,T]$ for some absolutely continuous function $M(t)$.

Then there exists an absolutely continuous function $x^*\colon [t_0, t_0+\delta] \to \mathbb{R}^d$ with $x^*(t_0) = x_0$ which solves the initial value problem

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(t,x), \qquad x(t_0) = x_0, \qquad x \in \mathbb{R}^d,$$

for Lebesgue almost all $t \in [t_0, t_0+\delta]$.

The three conditions are also referred to as *Carathéodory conditions*.

3.8 Appendix: Choice of Parameters

The parameters for our simulations are displayed in Table 3.1.

Table 3.1: Parameter choice for the scenario.

$a_{\max}$	v_{d}	δ	$a_{\min}$	s_0	T	b	l
2.0	10.0	4	−3.5	2.0	·	1.67	6
γ	α	β	σ	d_s	t^*	Δt	λ
1/300	1	1	0.2	·	10	0.1	1/3

tions Subject to Random Misperception, IEEE Transactions on Intelligent Transportation Systems 23(5), May 2022.

4 Microscopic Traffic Models, Accidents, and Insurance Losses

The original version of this chapter has been submitted for publication, see Kim, Kleiber & Weber (2022).

4.1 Introduction

Every year, traffic accidents cause substantial damage, both property damage and injuries and deaths. For example, nearly 43,000 people died in road traffic accidents in the USA in 2021 (NHTSA (2022)). The frequency and severity of these accidents depends on the driving behavior of vehicles, on the one hand, and on the characteristics of traffic systems themselves, on the other. Improvements in road safety are achieved, for example, by reducing serious injuries in accidents through the design of vehicles, by car body design, airbags, seatbelts, etc. However, the frequency and type of accidents can also be influenced by modifying the transportation system itself and by changes in driving behavior. From a higher-level perspective, at least two dimensions are central, and we will examine them in this chapter:

(i) *Engineering.* From an engineering perspective, the focus is on the good design of vehicles and traffic systems, combining functionality and safety. Instruments in this respect include traffic rules and their implementation, the layout of streets, and innovation in vehicle technology. Improvements of this type may reduce the number of accidents and their severity, but cannot completely prevent accidents.

(ii) *Insurance.* Residual risks remain, and accidents cannot be completely prevented. However, at least in financial terms, the associated losses can be covered by insurance contracts. The role of actuaries is to develop adequate contract structures, calculate correct premiums, and implement quantitative risk management in insurance firms. These tasks require the modeling and analysis of probability distributions of accident frequencies, corresponding damages, and insurance losses.

The objective of this chapter is to develop a methodology to enable microscopic models of transportation systems to be accessible for a statistical study of traffic accidents. Our approach is intended to permit an understanding not only of historical losses, but also of incidents that may occur in altered, potential future systems. Through this, it is possible,

from both an engineering and insurance perspective, to assess changes in the design of vehicles (e.g., the driving behavior of autonomous vehicles) and transport systems in terms of their impact on functionality and road safety.

To understand current traffic events or structural relationships in the past, historical data can be used. Historical data can also be applied to test whether a model framework is appropriate in principle to describe traffic systems realistically. These data also constitute the essential basis for the specific pricing of insurance contracts in practice.

But how can we examine risks associated with new technologies and with novel future strategies for traffic systems? Consider autonomous vehicles, for example: due to their altered driving behavior, these will reshape existing traffic patterns, and in turn, accident occurrences and associated losses. Insurance companies presumably will have to adapt their business models as well; in the future, premiums for auto insurance may depend upon the driving configuration of the vehicle rather than the risk profile of the driver.

In order to investigate future developments, we are suggesting to devise simulation tools in analogy to digital twins of real transport systems, which allow counterfactual case studies of possible future transport systems. The digital twin paradigm refers to the triad of a "physical entity, a virtual counterpart, and the data connections in between" (Jones et al. (2020)). In our application, the physical entity is the (future) real-world transportation system for which data on losses are not yet available. Its virtual counterpart is the model we are building. Counterfactual case studies can be used to generate data, evaluate future driving technologies and their impact on accident losses. Based on the results, newly developed concepts (e.g., modified traffic rules, novel insurance coverage and their insurance premiums, etc.) can be adapted in the real world. The concept of the digital twin makes it possible to experiment with technologies and policies, and their effects on accident damage without having to implement risky tests in reality.

Methodologically, this chapter combines existing microscopic traffic models with probabilistic tools from actuarial science and quantitative risk management to study accident damage and insurance losses in the context of simulations. In particular, we use the well-established traffic simulator SUMO (Lopez et al. (2018)) to realistically model traffic systems. We extend this to include random accidents and corresponding losses. The losses are modeled as random variables whose distributions depend on microscopic data. Since insurance contracts typically cover annual periods, we set up a model for aggregate losses over a one-year time horizon. We also show that aggregate losses can be approximated by a mean-variance mixture of Gaussian distributions. This provides an alternative perspective on the distribution of the aggregate loss and a second method of evaluation besides crude Monte Carlo sampling. For certain insurance contracts, we improve the accuracy of the approximation-based valuation by using a correction term. This was originally developed by El Karoui & Jiao (2009) for the efficient pricing of complex financial instruments, there in the context of a classical Gaussian approximation.

Our digital twin approach enables a comprehensive analysis of risk in transportation

systems: We study the impact of fleet sizes and their driving configurations on system efficiency and insurance prices. System efficiency is measured using traditional traffic statistics based on local traffic counts such as traffic flow, average speed, and density. Insurance claims are examined in terms of their probability distributions and selected statistical functionals.

The main contributions of this chapter are:

(i) We develop a powerful methodological framework to generate accident data based on microscopic traffic models in analogy to the concept of digital twins.

(ii) Specifically, we construct an implementation based on the state-of-the-art open-source traffic simulator SUMO and illustrate the potential of the approach in comprehensive case studies.

(iii) Structurally, we characterize the total loss distribution approximatively as a mean-variance mixture. This also yields alternative valuation procedures.

(iv) Based on Stein's method, we obtain a correction term in the valuation, derived from the results of El Karoui & Jiao (2009), which enables surprisingly accurate pricing of insurance contracts.

4.1.1 Outline

The chapter is organized as follows. Section 4.1.2 discusses related contributions in the literature. Section 4.2 presents the microscopic traffic model that captures also accidents. Section 4.3 discusses the evaluation of the losses. Case studies are presented in Section 4.4. Section 4.5 concludes and discusses further research challenges. The supplementary material in Section 4.6 contains details on the implemented sampling procedure; additional simulation results, not presented in Section 4.4, are documented in Appendix B.

4.1.2 Literature

This chapter combines microscopic traffic models with probabilistic tools from actuarial science and quantitative risk management to study risks in traffic systems. The literature can be predominantly classified along the two dimensions described in the introduction, the engineering perspective and the actuarial perspective.

The Engineering Perspective. An important field of operations research is the analysis and optimization of road traffic systems (see, e.g., Gazis (2002)) with respect to their efficiency. Traffic models are indispensable tools for this purpose: Macroscopic models are based on the functional relationships between macroscopic features such as traffic flow, traffic density, and average speed. These models allow the study of issues such

as the efficient routing of vehicles under different constraints (see, e.g., Acemoglu et al. (2018), Colini-Baldeschi et al. (2020)). Stochastics can be used to extend such risk considerations in terms of uncertain travel times (e.g., Nikolova & Stier-Moses (2014)). In the context of various applications of transportation systems, tailored stochastic models provide suitable analytical tools; the literature is extensive and includes, for example, the efficient routing of ambulances (Maxwell et al. (2010)) or the allocation of capacity in bike-sharing systems (Freund, Henderson & Shmoys (2022)).

To model transportation systems at a level of higher granularity, microscopic traffic models are used (see, e.g., Helbing (2001)). These models typically determine the acceleration behavior of individual vehicles. Their simulation, i.e., the computation of trajectories from accelerations, is computationally more demanding. There are established software solutions that facilitate the application of microscopic models. In this work, we use SUMO (see the Section 4.2.3 for an overview). Examples of competing microscopic traffic simulators include VISSIM (Fellendorf & Vortisch (2010)) and Aimsun (Casas et al. (2010)). While SUMO is open source software, these competitors are commercial.

Leveraging the simulator Aimsun, Osorio & Bierlaire (2013) address questions regarding optimal operation of traffic networks using microscopic traffic models; the authors develop a stochastic optimization framework based on coupling the Aimsun simulator with a metamodel to optimize traffic efficiency – here for signal plans in a city. Osorio & Nanduri (2015) extend the microscopic traffic model for fuel consumption to determine energy-efficient traffic management strategies.

In addition to model building (if data are available), whether macroscopic or microscopic, calibration can be a challenge. Flötteröd, Bierlaire & Nagel (2011) propose a Bayesian approach to calibrating travel demand. Zhang, Osorio & Flötteröd (2017) discuss the calibration of large-scale traffic simulators; Osorio & Punzo (2019) focus on the calibration of car-following models for the simulation of a traffic network.

In addition to the traditional focus on efficiency, there is another strand of literature that evaluates the safety of transportation systems. Up to now, mainly historical data have been used to examine accident frequency and severity. For reviews, we refer to Lord & Mannering (2010) and Tsoi & Gabler (2015). Statistical modeling approaches permit inference when sufficient data are available on the level of the granularity of the analysis. For example, Yu et al. (2019) estimate the impact of microscopic traffic variables on crash risks. Ortelli, Lapparent & Bierlaire (2021) analyze the impact of public traffic policies on crash severity.

In the absence of data, physical models of traffic can be used to generate artificial data. In our research group, we have shown how perceptual errors can be added to microscopic traffic models to endogenously model the occurrence of accidents (cf. Berkhahn et al. (2018) and Berkhahn et al. (2022)) – a topic that is particularly relevant for sensors of autonomous vehicles. The models allow characterizing the trade-off between safety and

efficiency of transportation systems.

To deploy autonomous vehicles in the real world, lengthy and expensive testing phases are required. Acceleration strategies are being developed to shorten these times (e.g., Zhao et al. (2018)). These approaches rely on importance sampling techniques to overcome the rare-event nature of safety-critical situations. Arief, Glynn & Zhao (2018) develop simulation-based testing methodologies in order to analyze autonomous vehicles in relevant scenarios that are constructed using collected data. Norden, O'Kelly & Sinha (2019) create a framework for the black-box assessment of the safety of autonomous vehicles. They apply their framework on a commercial autonomous vehicle system. Our work focuses on aggregate losses over relatively long time horizons. By considering one-year losses via a conditional loss modeling approach, we bypass the problem of simulating rare events.

The Actuarial Perspective. The ambitious goal of achieving maximum efficiency and complete safety through engineering design cannot be realized in reality; accidents can never be completely excluded, even if residual risks can be kept very small. Insurance is an instrument to deal with infrequent losses. They make financial transfer payments in the event of claims. We refer to McNeil, Frey & Embrechts (2015) and Wüthrich (2013) for overviews of mathematical and statistical methods in quantitative risk management and non-life insurance.

An important task of actuaries is pricing; the premiums of motor insurance contracts are based on historical claims data collected by insurance companies. Insurance premiums are calculated based on individual characteristics of the driver (age, driving experience, etc.) and the vehicle (type, location, etc.). These tariffs are often complemented by bonus-malus schemes (see, e.g., Denuit et al. (2007)) to incentivize more careful driving and prevent insurance fraud.

Novel pricing approaches use telematics technology (see, e.g., Husnjak et al. (2015) for an overview). This involves collecting GPS data from vehicles, which can be analyzed and classified. Machine learning techniques are suitable to process these large amounts of data. We refer to Gao, Meng & Wüthrich (2022) for a methodological overview. Verbelen, Antonio & Claeskens (2018) discuss telematics pricing and use generalized additive models to interpret the impact of telematics variables on accident frequency. More recently, Henckaerts & Antonio (2022) develop a usage-based auto insurance product in which a base premium is dynamically updated based on newly available telematics data, depending on the policyholder's driving behavior.

Our approach can be understood as complementary to telematics pricing: Instead of analyzing driving data to determine the driving behavior of individuals, we specify the behavior of vehicles as a *driving configuration* and subsequently generate driving data and insurance claims. Our approach is in particular suitable, if novel autonomous technologies are studied. To our knowledge, there is no other work that develops a

microfounded model of traffic accidents that can be leveraged to study insurance pricing.

4.2 Model Components

Our microfounded simulation model for investigating accident losses is based on two components:

(i) At its core is a deterministic microscopic traffic model that realistically characterizes the motion of vehicles in a traffic system typically represented by a system of ordinary differential equations.

(ii) This microscopic traffic model is extended to include the possibility of random accidents. At random accident times, local traffic data are observed which characterize the probability distribution of the occurring losses. In our specific implementation of this general conceptual approach, the SUMO microscopic traffic simulator is applied to realistically represent and simulate the underlying traffic scenarios.

In this section, we introduce the notation and construct microscopic traffic models that integrate random accident occurrences and losses. We also explain in more detail how SUMO is used.

4.2.1 Microscopic Traffic Networks

We consider a road network that is typically embedded into a two-dimensional area $A \subseteq \mathbb{R}^2$. The network may consist of roads, junctions, roundabouts, intersections, highways, etc. on which vehicles move. The collection of all vehicles in the network is denoted by $\mathcal{M}$. Each vehicle $i \in \mathcal{M}$ is assigned an origin-destination pair $(O^i, D^i) \in A$.

We consider a fixed time horizon $T > 0$. Vehicles move over time from their origin to their destination on a (potentially changing) path. We denote by $x^i(t)$ the position of vehicle i at time $t \in [0,T]$, by $v^i(t) = \frac{\mathrm{d}}{\mathrm{d}t}x^i(t)$ their velocity, and by $a^i(t) = \frac{\mathrm{d}}{\mathrm{d}t}v^i(t)$ their acceleration. We make the implicit assumption that vehicles are located in O^i until some release time and remain in D^i once reached. Thus, we let $\mathcal{M}(t) = \big\{i \in \mathcal{M} \colon x^i(t) \notin \{O^i, D^i\}\big\}$ be those vehicles which are currently *inside* the network, i.e., they have left their origin but not reached their destination, yet. If we only consider vehicles which belong to a certain group of vehicles (also called a *fleet*) $\Phi \subseteq \mathcal{M}$, we will write $\mathcal{M}^\Phi(t)$. This could, for example, be a fleet of vehicles with the same driving characteristics.

At the core of many microscopic traffic models are *car-following models*[1]. These determine the acceleration behavior of an individual vehicle i along its path on the basis

[1] Prominent examples include the Intelligent Driver Model (Treiber, Hennecke & Helbing (2000)), the Optimal Velocity Model (Bando et al. (1994) and Bando et al. (1995)), and the Krauß model (Krauß (1998)).

of information on the positions and velocities of the vehicles, typically in a neighborhood of i, and the properties of the system. Often, only the preceding vehicle on the road is relevant, and the acceleration of i is constructed such that vehicles move forward while maintaining a minimal distance. Through specific choices more complex traffic scenarios (e.g., intersections, overtaking) can still be represented in such a manner. Mathematically, car-following models correspond to systems of coupled ordinary differential equations.

Traffic State. We denote by $\gamma(t) = \big(x^i(t), v^i(t), a^i(t)\big)_{i\in\mathcal{M}}$ the state of the traffic system at time t. It records the position, velocity, and acceleration of any vehicle. The evolution of the traffic system over time is depicted by the (high-dimensional) trajectory $t \mapsto \gamma(t)$.

Macroscopic traffic statistics aggregate these microscopic data. Typical examples include traffic flow (number of vehicles that pass a certain point per time unit), traffic density (number of vehicles per length unit), and average speed. These measures quantify the performance of traffic systems.

Local Traffic Conditions. In order to model the occurrence of accidents depending on local traffic conditions, we partition A in regions. More precisely, we partition A into a finite number of disjoint sets $A_r \subseteq A$ such that $A = \bigcup_{r=1}^{R} A_r$ and $R \in \mathbb{N}$. We call the elements A_r of the partition a *traffic module*.

We let $\mathcal{M}_r(t) = \big\{i \in \mathcal{M}(t) \colon x^i(t) \in A_r\big\} \subseteq \mathcal{M}(t)$ denote those vehicles that are in A_r at time t (with $\mathcal{M}_r^{\Phi}(t)$ defined in analogy). The local traffic state of the module is $\gamma_r(t) = (x^i(t), v^i(t), a^i(t))_{i\in\mathcal{M}_r(t)}$. Key local traffic characteristics (density, flow, speed, etc.) can then be expressed as functions of $\gamma_r(t)$ and its evolution over small time windows.

4.2.2 Microscopic Traffic Model with Accident Losses

So far, the evolution of the traffic system is a deterministic function of time. The advantage of microscopic models is that they enable a detailed simulation of traffic systems. The driving behavior of the vehicles can be varied, likewise their number and paths, road conditions, etc. in order to generate many different scenarios. Such models provide a detailed picture, similar to digital twins of reality, and can be used to analyze potential future traffic systems or to understand the impact of new technologies.

We consider a finite collection of different traffic scenarios $\gamma^k := (\gamma^k(t))_{t\in[0,T]}$ with $k \in \{1, 2, \dots, K\}$ for a short time horizon $T > 0$. The aim is to analyze characteristics of traffic over the long time horizon NT, e.g., one year, for some large $N \in \mathbb{N}$; this is modeled by a finite sequence of traffic scenarios $(k_1, k_2, \dots, k_N) \in \{1, 2, \dots, K\}^N$. The N subintervals of length T are called time buckets. We will be interested in quantities aggregated or averaged over the whole time horizon NT. Examples include the average

traffic flow, the total number of accidents, the aggregate losses due to accidents, etc. These quantities do not depend on the order of the traffic scenarios during this time period, but only on their number of occurrences.

We denote by μ^k the number of occurrences of scenario k divided by N, i.e., the relative frequency of this traffic scenario over the considered time horizon NT. The vector $\mu = (\mu^1, \mu^2, \dots, \mu^K)^\top$ lies in the simplex $\Delta^{K-1} = \{x \in \mathbb{R}_+^K : \sum_{k=1}^K x^k = 1\}$. We assume that μ is not deterministic, but a random variable. This is to account for the fact that the relative frequencies of traffic scenarios fluctuate over different years due to varying weather conditions, random changes in traffic demand, or other factors. From a mathematical point of view, this construction leads to a mixture model with exogenous factor μ.

Accident Occurences. We now introduce our traffic accident model that will permit an analysis of aggregate losses and corresponding insurance contracts. The likelihood of the occurrence of an accident is modeled as a function of the traffic scenario. We consider two specifications, a Binomial and a Poisson model.

(i) *Binomial Model.* Accidents are rare events. For a given traffic scenario k, we assume that the probability p^k of an accident is close to zero. This probability may, of course, depend on the evolution of the traffic scenario, i.e., on the path $t \mapsto \gamma^k(t)$, and we will discuss concrete specifications later. Given a realization of μ, accidents are assumed to be independent across time buckets. This implies that traffic scenario k occurs for $N\mu^k$ time buckets corresponding to a duration of $NT\mu^k$, and the number of accidents C^k during this period has a conditional Binomial distribution with parameters p^k and $N\mu^k$:

$$C^k \mid \mu \sim \mathrm{Bin}(p^k, N\mu^k).$$

(ii) *Poisson Model.* An alternative model assumes that accidents occur at random times with a distribution governed by an intensity λ^k/T that depends on the traffic scenario k. More specifically, the number of accidents C^k during the period governed by scenario k of duration $NT\mu^k$ is conditionally Poisson distributed with parameter $\lambda^k N\mu^k$:

$$C^k \mid \mu \sim \mathrm{Poiss}(\lambda^k N\mu^k).$$

Accident Losses. Loss sizes conditional on the occurrence of accidents are assumed to be independent across traffic scenarios and across time buckets. We assume that the conditional loss distribution with conditional distribution function F^k depends only on the traffic scenario k. We will discuss examples below. Random total losses over the

considered time horizon NT are equal to

$$L = \sum_{k=1}^{K} \sum_{c=1}^{C^k} X_c^k,$$

where the random variables X_c^k, $k = 1, 2, \ldots, K$, $c \in \mathbb{N}$, are independent and $X_c^k \sim F^k$, $c \in \mathbb{N}$, for any k.

Concrete Specifications. Microscopic traffic models are experimental environments that allow to simulate the behavior of systems where no real data are yet available. Traffic planning can be supported by such models, and the impact of new technologies can be tested in a counterfactual analysis. Here, we specify the general principles how accident occurrences and losses can be based on microscopic traffic models. An implementation will in this chapter be based on SUMO, see Section 4.2.3, but could also rely on any other suitable traffic model.

Initially, K traffic scenarios need to be selected as a basis for the model. While running any deterministic traffic scenario k over the time window $[0, T]$, information can be extracted about the traffic states γ_r^k in each module $r \in \{1, 2, \ldots, R\}$. In SUMO typically not complete data on the whole paths are extracted, but only selected information at loop detectors in the network that are part of the implementation.

In reality, the likelihood of accidents typically increases with higher traffic density and higher velocities, ceteris paribus. Also the distribution of losses is influenced by quantities of this type. Examples are described in Section 4.4. This allows a computation of p^k and λ^k as a function of the data. Using the data associated with the modules, we may specify probabilities and intensities for the modules such that $p^k = \sum_{r=1}^{R} p_r^k$ and $\lambda^k = \sum_{r=1}^{R} \lambda_r^k$. In the Binomial model, p_r^k/p^k is the conditional probability that the accident is in module r given that an accident occurs. In the Poisson model, the intensities λ_r^k, $r = 1, 2, \ldots, R$, determine the accident times for each module. The resulting sequence of random times in the whole traffic system possesses the intensity λ^k. Conversely, if we first simulate random times with intensity λ^k and then randomly choose a corresponding module with probability λ_r^k/λ^k in a second step, the random times associated with each module r possess intensity λ_r^k. Both procedures produce a number of accidents C^k that occur during the period governed by scenario k.

The distributions of losses given a single random event will be chosen as follows. For each traffic scenario, we consider a collection of distribution functions $(F^{k,\psi})_{\psi \in \Psi}$ where ψ corresponds to data that may be extracted from the traffic simulation. In order to do so, we uniformly simulate a random time in $[0, T]$ and extract at this time the data from scenario k that determine ψ. The resulting distribution F^k is a mixture of the distributions $(F^{k,\psi})_{\psi \in \Psi}$. The mixing distribution is derived on the basis of the traffic data of scenario k that are generated from our microscopic traffic model.

Insurance Contracts & Statistical Functionals. The microscopic traffic model with accidents will be the basis for the generation of aggregate losses. We study statistical functionals and insurance contracts. The analysis will be based on Monte Carlo simulations, but we also compare approximation techniques that we describe in Section 4.3.

The focus is on functions of aggregate losses L. Letting $h\colon \mathbb{R} \to \mathbb{R}$ be an increasing function, we analyze $h(L)$. In particular, we investigate the following functions corresponding to three types of insurance coverage: $h(x) = x$ (full coverage), $h(x) = \max(x - \theta, 0)$, $\theta \geq 0$ (constant deductible), $h(x) = \min(x, \theta)$, $\theta \geq 0$ (stop loss). In each case, we evaluate various statistical functionals:

(i) *Expectation.* $\mathbb{E}(h(L))$,

(ii) *Variance.* $\mathrm{Var}(h(L)) = \mathbb{E}((h(L))^2)) - \mathbb{E}(h(L))^2$,

(iii) *Skewness.* $\varsigma_{h(L)} = \frac{\mathbb{E}\left[(h(L)-\mathbb{E}(h(L)))^3\right]}{(\mathrm{Var}(h(L)))^{3/2}}$,

(iv) *Value-at-Risk.* $\mathrm{VaR}_p(h(L)) = \inf\{x \in \mathbb{R}\colon P(h(L) \leq x) \geq p\}$,

(v) *Expected Shortfall.* $\mathrm{ES}_p(h(L)) = \frac{1}{1-p}\int_p^1 \mathrm{VaR}_q(h(L))\,\mathrm{d}q$.

These functionals allow also the computation of insurance premiums on the basis of premium principles such as the expectation principle, the variance principle, or the standard deviation principle.

4.2.3 Traffic Scenarios in SUMO

4.2.3.1 A Brief Overview

A state-of-the-art open source software that allows us to generate traffic scenarios is SUMO, "**S**imulation of **U**rban **MO**bility". A reference publication on SUMO is Lopez et al. (2018); in addition a detailed user documentation can be found online[2]. Freely available since 2001, SUMO was originally developed by the German Aerospace Center and extended by an active research community. It allows for a plethora of modeling choices at different levels and has been successfully applied to tackle many important research questions addressing[3], e.g., traffic light optimization, routing, traffic forecasting, and autonomous driving.

In the following, we give a short overview. At its core, SUMO is a software which generates a traffic scenario $\gamma = (\gamma(t))_{t\in[0,T]}$ from a given set of input files:

(i) *Network File.* In SUMO, a traffic network is described by a directed graph whose nodes represent intersections and edges roads. All nodes and edges have attributes including positions, shapes, speed limits, traffic regulation, etc. As an example, the city of Wildau is represented as a SUMO network in Figure 4.1.

[2]See `sumo.dlr.de/docs/index.html`.
[3]See `eclipse.org/sumo/about/`.

(ii) *Route File.* Vehicles are generated on the basis of *traffic demands* between origins and destinations. Routes can either be defined for each vehicle as a *trip* or as recurring *flows* along a specific path. If only origin and destination are provided, the corresponding route is computed[4] when the vehicle enters the system.

The *vehicle type* determines the microscopic characteristics of the vehicle such as the governing car-following model, driving parameters (e.g., maximal speed, maximal acceleration, time headway), size, color, etc. By default, vehicles are passenger cars. Other modes, such as pedestrian, bicycle, or truck can also be selected.

(iii) *Additional Files.* Further components are specified in additional files. An important example are induction loop detectors. These collect time series data on aggregate traffic statistics by counting the vehicles which pass a certain position during a short time interval.

The collection of input files determines the traffic evolution, also called the *SUMO scenario*. The computation can be executed either as a command line application or with a GUI that visualizes the movement of the vehicles through the network over time.

Data Extraction. One particularly appealing extension of SUMO is the "**Tra**ffic **C**ontrol **I**nterface", TraCI (see Wegener et al. (2008)). TraCI provides online access to the microscopic traffic simulation and permits, at each time step, through a comprehensive list of commands[5] to retrieve data and to change the states of objects such as vehicles, roads, traffic lights, etc. Available in standard programming languages[6], TraCI yields easy access to SUMO without the need to modify the underlying code. We use TraCI to extract microscopic data on positions, velocities, and accelerations of randomly selected vehicles.

4.2.3.2 Generation of Traffic Scenarios

To represent traffic in a given area over a longer time horizon (e.g., $NT = 1\,\text{year}$), we generate a diverse collection of traffic scenarios $\gamma^1, \ldots, \gamma^K$ of duration T in SUMO by varying the input files. Traffic over a longer time horizon is represented by a random composition of these traffic scenarios. Our general construction has already been discussed in Section 4.2.2.

SUMO Scenario. SUMO provides tools that facilitate the creation of input files, e.g., the graphical network editor *netedit*[7] that visualizes a SUMO scenario and allows

[4] The problem of allocating traffic demand to routes in a network is referred to as the traffic assignment problem. A standard reference is Patriksson (2015).
[5] A detailed description can be found at `sumo.dlr.de/docs/TraCI.html`.
[6] Our case studies are based on the Python implementation.
[7] See `sumo.dlr.de/docs/Netedit/index.html`.

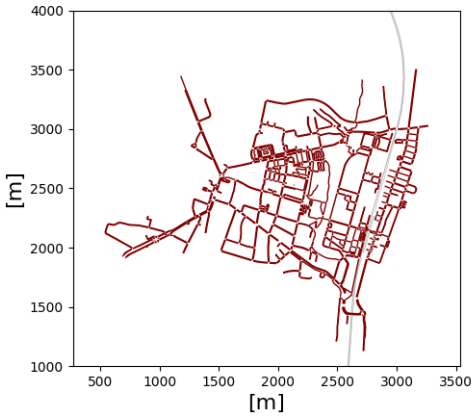


Figure 4.1: SUMO network of Wildau.

to modify its properties. In practice, network files are typically imported from other data sources[8]. For example[9], one can build a real-world traffic network in SUMO from OpenStreetMap data by selecting an area from a map. The route file specifies the trips of the vehicles and the definition of the general vehicle types with their microscopic characteristics. First, there are several options to generate trips in SUMO[10]. These can be obtained from empirical data in the form of traffic counts, imported to SUMO as origin-destination matrices, or modeled via ad-hoc choices, e.g., using *netedit*. Second, each vehicle is associated to a vehicle type specified on the basis of a comprehensive list of attributes[11]. The corresponding values can be set manually in the route file or accessed and modified via *netedit*. In the absence of detailed traffic data for calibration, SUMO offers an *activity-based demand generation*[12] which deduces traffic demand from general assumptions on the structure of the population (inhabitants, households, etc.) in the considered area. The tool *activitygen*[13] automates the process and produces an artificial route file.

SUMO admits a large variety of modeling choices. Tailored to the needs of the modeler, a highly detailed SUMO scenario can be constructed. Our case studies will be based on publicly available SUMO scenarios[14]; these consist of network and route files which are calibrated to real-world cities.

Varying Traffic Conditions. The input files need to be constructed in such a way that they reflect varying traffic conditions over longer time periods. This includes weather

[8] We refer to `sumo.dlr.de/docs/index.html#network_building` for an overview on network generation.
[9] See `sumo.dlr.de/docs/Networks/Import/OpenStreetMap.html`.
[10] We refer to `sumo.dlr.de/docs/Demand/Introduction_to_demand_modelling_in_SUMO.html` for an overview on demand modeling in SUMO.
[11] See `sumo.dlr.de/docs/Definition_of_Vehicles%2C_Vehicle_Types%2C_and_Routes.html`.
[12] See `sumo.dlr.de/docs/Demand/Activity-based_Demand_Generation.html`.
[13] See `sumo.dlr.de/docs/Demand/activitygen.html`.
[14] See `sumo.dlr.de/docs/Data/Scenarios.html`.

conditions, variation of traffic demand, and other factors.

Maze, Agarwal & Burchett (2006) review empirical studies on the impact of adverse *weather conditions* on traffic. These may induce i) lower traffic demand, ii) higher risk of accidents, and iii) modified driving behavior. Based on empirical findings, Phanse, Chaturvedi & Srivastava (2022) implement reduced velocities due to rainfall. In Weber, Driesch & Schramm (2019), the idea of introducing into SUMO a friction parameter per road is discussed. Traffic scenarios under adverse weather conditions can be captured by suitable driving parameters in the route file (e.g., by variation of maximal speed, maximal acceleration, etc.), and this may be combined with a weather-dependent model of the occurrence and the severity of accidents.

Traffic demand is traditionally estimated from traffic counts. We refer to Bera & Rao (2011) for an overview. With increasing data availability, the estimation can be enhanced by floating car data (cf., e.g., Nigro, Cipriani & Giudice (2018)), i.e., data generated from vehicles over time as they are driving. Traffic demand varies over time, but patterns reoccur over longer time horizons (see, e.g., Soriguera (2012)). Demand depends on the considered traffic network. Weekdays differ from days on the weekend; peaks in demand occur at common commute times. Rush hours are spatio-temporal phenomena that can be analyzed in detail (see, e.g., Xia et al. (2018)).

To reflect the heterogeneity of traffic scenarios, two options are available in SUMO: i) A variety of route files is generated that is consistent with the desired modeling granularity. This process can be automatized via an additional program, a *route file generator*, that produces route files with the desired characteristics. ii) Another option is to select a medium time horizon (e.g., $T_{\mathrm{SUMO}} = 24\,\mathrm{h}$) with a corresponding route file that depicts varying traffic demand over time. From the generated SUMO scenario, a selection of small time horizon scenarios (e.g., $T = 1\,\mathrm{min}$) can be efficiently generated by utilizing SUMO's option to save the state of the running simulation at a priori specified times and load these later.

Besides weather and traffic demand, many other factors influence the traffic dynamics. Wagner (2016) discusses the representation of autonomous vehicles in SUMO. Lücken et al. (2019) utilize SUMO to study control transition, i.e., selected safety critical situations where the human driver needs to take over control from an autonomously driving vehicle. Pagany (2020) study the impact of wildlife on traffic, an issue that is also relevant in the context of traffic accidents.

4.3 Evaluation Methods

The accident losses L can be simulated using Monte Carlo methods. The simulations may be used to estimate the value of statistical functionals and to price insurance products. We will briefly describe the Monte Carlo methods. In addition, on the basis of the Binomial model, we construct a Gaussian approximation to $\mathbb{E}(h(L))$ where the function

h corresponds to the three types of insurance coverage that we consider: full coverage, constant deductible, and stop loss. This allows a numerical evaluation similar to Frey, Popp & Weber (2008) and El Karoui & Jiao (2009). The latter paper provides a correction term derived by Stein's method that we will exploit in our application.

4.3.1 Monte Carlo Methods

The Monte Carlo simulation of L requires sampling the number of accidents C^k in either the Binomial or Poisson model and sampling the independent conditional losses X_c^k, $c \in \mathbb{N}$, for each traffic scenario $k = 1, 2, \ldots, K$ from the corresponding distribution F^k. These tasks can be performed separately, but require both a prior evaluation of the microscopic traffic model.

(i) *Prior Evaluation of Traffic Model.* For each traffic scenario k a single run delivers data that are the basis for a computation of, respectively, the accidents probabilities p^k and intensities λ^k as well as the corresponding values p_r^k and λ_r^k on the level of the modules $r = 1, 2, \ldots, R$.

(ii) *Number of Accidents.* Sampling from μ and using the results of the prior evaluation allows to sample the number of accidents C^k for each traffic scenario k in both the Binomial and Poisson model.

(iii) *Conditional Accident Losses.* Based on the precomputed values of the accident probabilities respectively the accident intensities, we simulate for each traffic scenario k the random locations and times of accidents. These data can be stored. For these locations and times, traffic data ψ are extracted from an additional single run of traffic scenario k. Given ψ, the losses are generated according to conditional loss distributions $F^{k,\psi}$ as described in Section 4.2.2. Details of the implementation are explained in Section 4.4.

These Monte Carlo methods can be flexibly applied to all considered functionals. In the special cases of expectation and variance, Wald's equation can simplify the computation, since the losses L are given in the form of a collective model.

4.3.2 Gaussian Approximation

Another way to compute $\mathbb{E}(h(L))$ for the considered types of insurance coverage is a Gaussian approximation, possibly improved by a correction term. A Gaussian approximation can easily be motivated within the Binomial model. For N sufficiently large, the distribution of L given μ is approximately normal, implying that L is a mean-variance-mixture of Gaussian distributions. This is an important structural insight from this approximation.

In this section, we condition on μ, i.e., suppose that μ is fixed and given. The general results for μ random are then a corollary by considering suitable mixtures according to the distribution of μ. The total random losses in the Binomial model can be rewritten as

$$L \;=\; \sum_{k=1}^{K} \sum_{c=1}^{N\mu^k} \mathbf{1}_c^k \cdot X_c^k,$$

where the random variables $\mathbf{1}_c^k$, X_c^k, $k = 1, 2, \ldots, K$, $c \in \mathbb{N}$, are independent, $X_c^k \sim F^k$, and $\mathbf{1}_c^k$ are Bernoulli random variables taking the value 1 with probability p^k and the value 0 otherwise, $c \in \mathbb{N}$, for any k. Setting[15]

$$\begin{aligned} Y_c^k \;&:=\; \mathbf{1}_c^k \cdot X_c^k, \\ m^k \;&:=\; \mathbb{E}(Y_c^k), \\ \left(\sigma^k\right)^2 \;&:=\; \mathbb{E}\left([Y_c^k - m^k]^2\right), \\ \left(\zeta^k\right)^3 \;&:=\; \mathbb{E}\left([Y_c^k - m^k]^3\right), \end{aligned}$$

$k = 1, 2, \ldots, K$, $c \in \mathbb{N}$, a classical normal approximation of L is $\sum_{k=1}^{K} N\mu^k m^k + Z$ with

$$Z \quad \sim \quad \mathcal{N}\left(0, \sum_{k=1}^{K} N\mu^k \left(\sigma^k\right)^2\right).$$

We focus on three types of insurance coverage, $h(x) = x$ (full coverage), $h(x) = \max(x - \theta, 0)$, $\theta \geq 0$ (constant deductible), $h = \min(x, \theta)$, $\theta \geq 0$ (stop loss), and obtain an approximation $\mathbb{E}(h(Z))$ of $\mathbb{E}(h(L))$ in each of these cases.

On the basis of Stein's method[16], El Karoui & Jiao (2009) suggest correction terms in order to improve the approximation, i.e., the approximation $\mathbb{E}\left(h\left(\sum_{k=1}^{K} N\mu^k m^k + Z\right)\right)$ is replaced by the corrected approximation

$$\mathbb{E}\left(h\left(\sum_{k=1}^{K} N\mu^k m^k + Z\right)\right) \;+\; C_h.$$

The correction C_h depends on the degree of smoothness of the derivatives of the function h and thus differs[17] for the three types of coverage. We define

$$d_1 \;=\; \sum_{k=1}^{K} N\mu^k m^k, \quad d_2 \;=\; \sum_{k=1}^{K} N\mu^k \left(\sigma^k\right)^2, \quad d_3 \;=\; \sum_{k=1}^{K} N\mu^k \left(\zeta^k\right)^3,$$

and let $\tilde{h}(x) \;=\; h\left(x + d_1\right)$. We obtain the following correction terms:

[15] The estimation of m^k, $\left(\sigma^k\right)^2$, and $\left(\zeta^k\right)^3$ requires the simulation of the random variables X_c^k, $c \in \mathbb{N}$. The independent terms $\mathbf{1}_c^k$, $c \in \mathbb{N}$, factor out, are idempotent and have known expectation p^k.

[16] For an overview on Stein's method we refer to Chen, Goldstein & Shao (2011) and Ross (2011).

[17] See Theorem 3.1 and Proposition 3.6 in El Karoui & Jiao (2009).

(i) *Full Coverage.* In the case of full coverage, the correction term of El Karoui & Jiao (2009) disappears. In general, if h is some Lipschitz function with bounded third derivative, the correction term equals

$$C_h = \frac{d_3}{2d_2^2} \cdot \mathbb{E}\left(\left\{\frac{Z^2}{3d_2} - 1\right\} Z\tilde{h}(Z)\right).$$

(ii) *Constant Deductible.* $$C_h = \frac{(\theta - d_1)d_3}{6d_2} \cdot \frac{1}{\sqrt{2 \cdot \pi \cdot d_2}} \cdot \exp\left\{-\frac{(\theta - d_1)^2}{2 \cdot d_2}\right\}.$$

(iii) *Stop Loss.* A stop loss $x \mapsto \min(x, \theta)$ can be written as the difference between full coverage $x \mapsto x$ and a constant deductible $x \mapsto \max(x - \theta, 0)$. This implies that the correction term for a constant deductible appears with a negative sign in this case.

The advantage of the (corrected) Gaussian approximation in comparison to pure Monte Carlo is that, once the numbers m^k, $\left(\sigma^k\right)^2$, and $\left(\zeta^k\right)^3$ have been computed for each traffic scenario $k = 1, 2, \ldots, K$, no further data need to be stored or sampled in order to compute $\mathbb{E}(h(L))$. The approximate representation of the distribution of L as a mean-variance-mixture is a considerable simplification.

4.4 Application

We illustrate the application of our microscopic traffic model with accidents on the basis of a publicly available SUMO scenario of a real city.

4.4.1 SUMO Scenario & Accident Data

Wildau is a small German city of approximately 10,000 inhabitants, located around 30 km south-east of the capital Berlin. A SUMO model of the city was developed within a study project by the Technical University of Applied Sciences Wildau and is publicly available[18].

SUMO Scenario. The implemented road network is visualized in Figure 4.1. It is specified using 646 nodes connected by 1,426 edges. The city itself is crossed by the railway; the tracks are represented by the gray line. Vehicles are calibrated from real traffic counts. The original scenario has a duration of $7,010\,\mathrm{s}$. Empty in the beginning, vehicles enter the system with a peak of approximately 240 vehicles that drive simultaneously. In total, 2502 vehicles are generated.

In the following section, we describe in detail how we adjust this SUMO scenario to obtain suitable ingredients for our case studies. This yields a collection of traffic scenarios that allow us to compare the effects of different driving characteristics and fleet sizes on the total loss and related insurance premiums.

[18] See `github.com/DLR-TS/sumo-scenarios/tree/main/Wildau`.

Varying Traffic Conditions. For different collections of model parameters representing different traffic systems we generate adjusted SUMO scenarios. For each choice of parameters we proceed as follows to produce $K = 100$ traffic scenarios of length $T = 60\,\mathrm{s}$. Traffic scenarios $k = 1, 2, \ldots, 50$ correspond to selected time intervals from the SUMO scenario. Traffic scenarios $k = 51, 52, \ldots, 100$ represent higher traffic volumes. They are generated by replacing the original route file by a route file that consists of two copies of the original route file. This simple procedure generates a larger amount of vehicles along the original paths. The traffic scenarios are again selected time intervals from the corresponding SUMO scenario.

To represent the full year, we set $N = 365 \cdot 24 \cdot 60 = 525,600$. We need to specify the random vector $\mu = (\mu^1, \ldots, \mu^K)^\top$ describing the number of occurrences of the individual traffic scenarios divided by N. For the purpose of illustration, we specifically assume that two probability measures ν_g, ν_b are given on $\{1, 2, \ldots, K\}$ which approximately correspond to the relative frequencies of traffic scenarios in two prototypical years $y = g, b$. In addition, we suppose that the type y of the current year is random where both values g and b have probability $1/2$. Given y, we generate $\mu = (\mu^1, \ldots, \mu^K)^\top$ from a multinomial distribution corresponding to ν_y. That is, for all time buckets $n = 1, 2, \ldots, N$, a traffic scenario k is chosen independently from the distribution ν_y on $\{1, 2, \ldots, K\}$. Dividing the number of occurrences of a scenario k by N, one obtains its random relative frequency μ^k for any $k = 1, 2, \ldots, K$. In our case study, the distribution ν_g corresponds to lower traffic densities on average, while ν_b is associated with higher traffic densities, i.e., we set

$$\nu_g := \begin{cases} 1/75, & k = 1, 2, \ldots, 50, \\ 1/150, & k = 51, 52, \ldots, 100 \end{cases}, \quad \nu_b := \begin{cases} 1/150, & k = 1, 2, \ldots, 50, \\ 1/75, & k = 51, 52, \ldots, 100. \end{cases}$$

Accident Data. The German Accident Atlas[19] depicts the locations of all police-reported accidents *involving personal damage* that occurred within one year. In 2020, within the modeled area of Wildau (approximately) 48 accidents were registered. There are also aggregate statistics for Germany for all police-reported accidents. In 2020, approximately 11.8 % of all road accidents involved personal damage[20]. We use an estimate of $\bar{c}_{\text{year}} = 48/11.8\,\% \approx 407$ accidents for calibration purposes.

4.4.2 Model Specification

Our goal is to analyze accident losses for a fleet Φ over the time horizon of one year.

[19] The data are provided by the *Statistische Ämter des Bundes und der Länder* under the "Data licence Germany – attribution – Version 2.0" and can be accessed via `unfallatlas.statistikportal.de/`.

[20] See the online resource: `destatis.de/EN/Themes/Society-Environment/Traffic-Accidents/Tables/accidents-registered-police.html`.

Fleet Definition. In the Wildau scenario, vehicles are defined using repeated flows from origins to destinations. Passenger cars (next to trucks and the train) are defined via 90 different flows. Each passenger car belongs to the same vehicle type with fixed driving characteristics.

To introduce a fleet Φ of vehicles whose driving characteristics we can vary, we define a new vehicle type Φ and construct corresponding new SUMO scenarios. Fixing a fraction $\rho^\Phi \in [0,1]$ of vehicles belonging to Φ, we retain approximately $1-\rho^\Phi$ of the existing flow definitions and modify ρ^Φ of the flow definitions suitably in order to model the fleet. In our case studies, we consider $\rho^\Phi = 10\,\%,\ 50\,\%,\ 90\,\%$.

Driving Configuration. Vehicles in a fleet Φ are of the same type. Various characteristics can be varied in SUMO; we focus on *maximal speed* $v_{\max}$, *maximal acceleration* $a_{\max} > 0$, and *time headway* $\zeta > 0$. The time headway is the distance which is kept to the preceding vehicle measured in time, i.e., a velocity weighted safety distance. We refer to a fixed selection of driving characteristics as a *driving configuration*.

In our case studies, we will vary the driving configuration for all vehicles in the fleet Φ and keep all other vehicles as originally introduced[21]. A driving configuration of vehicles in fleet Φ is denoted by $\xi = (v_{\max}, a_{\max}, \zeta)$. Specifically, we consider[22]:

$$\begin{aligned}
\xi^{1a} &= (5\,\mathrm{m/s},\ 0.8\,\mathrm{m/s^2},\ 3.0\,\mathrm{s}), & \xi^{1b} &= (5\,\mathrm{m/s},\ 2.6\,\mathrm{m/s^2},\ 3.0\,\mathrm{s}),\\
\xi^{2a} &= (10\,\mathrm{m/s},\ 0.8\,\mathrm{m/s^2},\ 2.0\,\mathrm{s}), & \xi^{2b} &= (10\,\mathrm{m/s},\ 2.6\,\mathrm{m/s^2},\ 2.0\,\mathrm{s}),\\
\xi^{3a} &= (15\,\mathrm{m/s},\ 0.8\,\mathrm{m/s^2},\ 1.0\,\mathrm{s}), & \xi^{3b} &= (15\,\mathrm{m/s},\ 2.6\,\mathrm{m/s^2},\ 1.0\,\mathrm{s}).
\end{aligned}$$

The configurations $1, 2, 3$ increase in terms of "aggressiveness" from driving slowly with a large headway to fast with a small headway – with two options a and b for the maximal acceleration.

Accident Occurrence. The best estimate for the total number of accidents in Wildau is $\bar{c}_{\text{year}} \approx 407$. From this, we derive a uniform and a non-uniform accident occurrence model. In both cases, we specify accident probabilities $p^{1,k}$ and $p^{2,k}$ for the binomial model as well as accident intensities $\lambda^{1,k}$ and $\lambda^{2,k}$ for the Poisson model.

(i) *Uniform Accident Occurrence.* Assuming that accidents occur uniformly over the year, we obtain a probability per time bucket of an accident in the system of $p^1 = \bar{c}_{\text{year}}/N \approx 7.7 \cdot 10^{-4}$. This is the accident probability that we allocate to each traffic scenario k. We also suppose that accidents occur uniformly across all

[21] We use the implementation of an Intelligent Driver Model without any speed deviation that does not include further random effects. We refer to `sumo.dlr.de/docs/Simulation/Randomness.html` for any random effects in SUMO.

[22] The specific choices are inspired by the following considerations: The implemented road speed limit in Wildau is 50 km/h which is approximately 13.9 m/s. Vehicles in the original Wildau scenario have a maximal acceleration of 0.8 m/s^2, while SUMO's default value is 2.6 m/s^2. Similarly, SUMO's default time headway is 1.0 s.

vehicles in the system. This implies that the probability that any accident occurs in scenario k within the fleet Φ is

$$p^{\Phi,1,k} \;=\; \rho^{\Phi} \cdot p^{1}, \quad k \in \{1,\dots,K\}.$$

This probability is used in the Binomial model. For the Poisson model we set $\lambda^{\Phi,1,k} = p^{\Phi,1,k}$, $k = 1, 2, \dots, K$, since the intensity approximately equals the probability of an accident per time bucket.

In the case of uniform accident occurrence, we do not consider any spatial variations of the likelihood of accidents due to different traffic conditions. This means that we do not distinguish any modules, i.e., we set $R = 1$.

(ii) *Non-Uniform Accident Occurrence.* In reality, the likelihood of accidents depends on external factors such as weather and local traffic conditions, e.g., the velocity of vehicles and traffic density. The quantities vary spatially and over time.

From SUMO runs, we obtain for each traffic scenario $k = 1, 2, \dots, K$ and each module $r = 1, 2, \dots, R$ pairs $(d_r^k, \bar{v}_r^k)$ on the average density and velocity. These statistics can be computed in SUMO, for example, from data that are obtained at induction loop detectors which are placed within the modules; as a proxy for density, we extract the *occupancy* of the loop detector, i.e., the fraction of time which it is occupied by a vehicle.
For $r = 1, \dots, R$, we choose benchmark values d_r^* and $\bar{v}_r^*$ for the density and velocity and specify occurrence probabilities and intensities that vary spatially and over time:

$$\lambda_r^{\Phi,2,k} \;=\; p_r^{\Phi,2,k} \;:=\; \frac{p^{\Phi,1,k}}{R} \cdot \frac{\bar{v}_r^k}{\bar{v}_r^*} \cdot \frac{d_r^k}{d_r^*} \cdot e^{-(\zeta^{\Phi}-1)}, \quad k \in \{1,\dots,K\}, \quad r \in \{1,\dots,R\}.$$

The last term refers to deviations of the time headway from SUMO's default value of 1.0 s: a larger time headway is associated with less risky driving. We set $p^{\Phi,2,k} = \sum_{r=1}^{R} p_r^{\Phi,2,k}$ and $\lambda^{\Phi,2,k} = \sum_{r=1}^{R} \lambda_r^{\Phi,2,k}$.

In our case studies, we will consider a grid of $R = 4$ modules and compute d_r^k and $\bar{v}_r^k$ as averages over measurements from 10 induction loop detectors that are placed in each module (see also Figure 4.2). We use the scenario averages $d_r^* = \frac{1}{K}\sum_{k=1}^{K} d_r^k$ and $\bar{v}_r^* = \frac{1}{K}\sum_{k=1}^{K} \bar{v}_r^k$. If $\sum_{k=1}^{K} \mathbb{E}(\mu^k)\bar{v}_r^k = \bar{v}_r^*$, $\sum_{k=1}^{K} \mathbb{E}(\mu^k) d_r^k = d_r^*$, and $\zeta^{\Phi} = 1$, then we essentially recover on average the case of uniform accident occurrence.

Accident Losses. The distributions F^k of accident losses associated with a traffic scenario $k = 1, 2, \dots, K$ are constructed on the basis of traffic data that are extracted from the SUMO runs. The general procedure was described in Section 4.2.2; here, we explain the specific implementation that we use in our case studies.

Figure 4.2: Partition of Wildau and placement of induction loop detectors.

The likelihood of accident occurrence was discussed in the previous section. F^k is the conditional distribution of an accident loss in traffic scenario k if an accident occurs. Time in traffic module k is enumerated by $t \in [0, T]$, and we assume that the time τ of the accident conditional on its occurrence is uniformly distributed on $[0, T]$, i.e.,

$$\tau \ \sim \ \mathrm{Unif}[0, T].$$

We choose a module $\mathcal{R}$ at random in which the accident occurs and assume, respectively, that

$$P(\mathcal{R} = r) \ = \ \frac{p_r^{\Phi,\cdot,k}}{p^{\Phi,\cdot,k}}, \qquad P(\mathcal{R} = r) = \frac{\lambda_r^{\Phi,\cdot,k}}{\lambda^{\Phi,\cdot,k}}, \qquad r \in \{1, \ldots, R\}.$$

These ratios depend in the case of non-uniform accident occurrence on the specific fleet, since the properties of the fleet alter the route file that is used to generate the SUMO scenarios; this is true, although the multiplicative terms ρ^Φ appear in both the numerator and denominator and cancel out.

In the chosen module we pick one or more vehicles at random, and extract from the traffic scenario data for these vehicles. In our concrete implementation, we simply choose at time τ a single vehicle I uniformly at random in module $\mathcal{R}$, i.e., its conditional distribution is

$$I \mid \tau, \mathcal{R} \ \sim \ \mathrm{Unif}(\mathcal{M}_{\mathcal{R}}^{\Phi}(\tau)).$$

For the purpose of illustrating our approach, the only data we extract are the velocities v^I of the randomly chosen vehicles that are involved in accidents. We set $\psi = v^I$ and assume that the conditional loss distribution $F^{k,\psi}$ is known[23]. If we denote the distribution of

[23] We assume that $F^{k,0}$ corresponds to a Dirac measure in 0; if $\mathcal{M}_{\mathcal{R}}^{\Phi}(\tau) = \emptyset$, we set $\psi = 0$, resulting in 0 losses.

ψ by $\mathcal{L}^k$ we obtain the distribution F^k as a mixture

$$F^k \;=\; \int F^{k,\psi}\, d\mathcal{L}^k.$$

In our case studies, we will assume that $F^{k,\psi} = F^{\psi}$ for all k; however, the mixing distribution $\mathcal{L}^k$ will dependend on the traffic scenario k. We consider the following examples for F^{ψ}:

(i) *Gamma Distribution.* We define distributions with varying levels of *dispersion*[24]. For a given coefficient of variation $c_v \in \{1/2,\ 1,\ 2\}$, we choose

$$F^{\psi} \;=\; \Gamma\left(\frac{1}{c_v^2},\ \frac{1}{c_v^2\psi^2}\right).$$

The expectation of this distribution is ψ^2 and increases quadratically with ψ, the velocity of the vehicle involved in an accident; this is consistent with the fact that losses scale with kinetic energy. The variance of the distribution F^{ψ} equals $c_v^2\psi^4$, hence the coefficient of variation is indeed c_v.

(ii) *Log-Normal Distribution.* We consider log-normal distributions with expectation ψ^2 and variance $c_v^2\psi^4$, implying that the coefficient of variation is again $c_v \in \{1/2,\ 1,\ 2\}$. This log-normal distribution is obtained as the distribution of $\exp(Z)$ for a normal random variable Z with expectation $\ln(\psi^2/\sqrt{1+c_v^2})$ and variance $\ln(1+c_v^2)$, i.e.,

$$F^{\psi} \;=\; \mathcal{LN}\left(\ln\left(\frac{\psi^2}{\sqrt{1+c_v^2}}\right),\ \ln(1+c_v^2)\right).$$

4.4.3 Case Studies

4.4.3.1 Overview

We illustrate our modeling approach in case studies on multiple levels. A selection of case studies is discussed in detail in Sections 4.4.3.2 & 4.4.3.3. All numerical results for the following choices are documented in tables in Appendix B:

(i) *Fleet Models.* We analyze six driving configurations with three different fleet proportions.

(ii) *Accident Occurrence.* In our traffic system accidents occur uniformly or non-uniformly in space. Their number is given by a Binomial or a Poisson model with parameters depending on traffic condition.

(iii) *Accident Losses.* We study two parametric families of loss distributions with three different choices for the coefficient of variation.

[24] A measure for the dispersion of a random variable X is the coefficient of variation defined by $c_v = \sqrt{\mathrm{Var}(X)}/\mathbb{E}(X)$.

(iv) *Insurance Design.* Insurance losses are a function of the total losses; we distinguish three contract designs.

Denoting aggregate losses by L, we evaluate for each type of insurance coverage h the resulting insurance losses $h(L)$ in terms of their expectation, variance, and skewness, and the monetary risk measures Value-at-Risk and Average Value-at-Risk, also called Expected Shortfall. To analyze the distributions in detail, we provide qq-plots and estimates of cumulative distribution functions and densities. The main tool to access the random variable $h(L)$ is Monte Carlo sampling; we provide a pseudo-code how we obtain samples of L in Algorithm 1 in Section 4.6. In Section 4.4.3.3, we compare this approach to the Normal mean-variance-mixture approximation introduced in Section 4.3.2.

This chapter explores the analysis and management of risks that occur in vehicle fleets in traffic systems. We distinguish to perspectives:

(i) *The Engineering Perspective.* In these case studies, we fix the accident occurrence and accident loss distributions and vary the fleet models, i.e., driving configurations and fleet proportions. We focus on $\mathbb{E}(L)$, $\mathrm{Var}(L)$, and complement these with analyses of the performance of traffic system.

(ii) *The Actuarial Perspective.* In these case studies, we fix the fleet model and vary accident occurrence and accident loss distributions as well as the insurance design. We study the distribution of L and the insurance prices $\mathbb{E}(h(L))$.

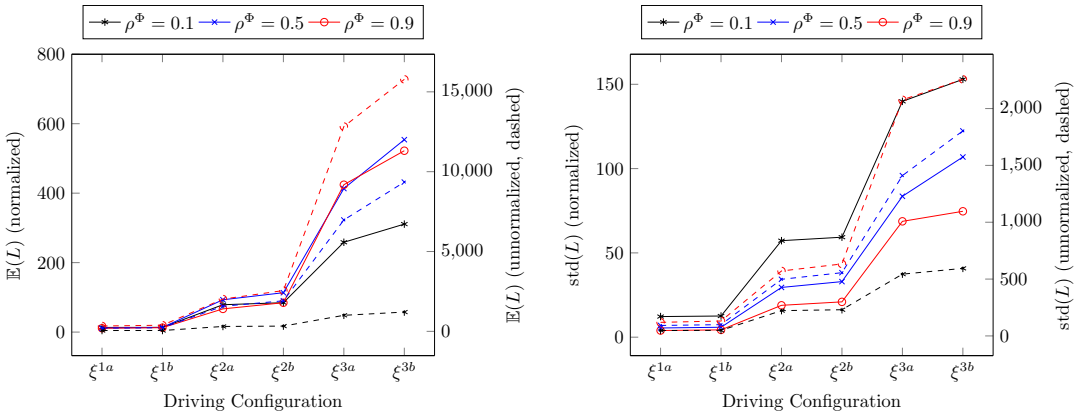
4.4.3.2 The Engineering Perspective

Our micro-modeling approach allows us to study the effects of different traffic-related controls on total losses L; we investigate the effects of fleet size and traffic configuration. Throughout this section, we consider non-uniform accident occurrence in the Binomial model with Gamma distributed accident losses and a coefficient of variation $c_v = 1$.

Losses. We evaluate expected loss $\mathbb{E}(L)$ and standard deviation $\mathrm{std}(L)$ for different fleet models. To compare losses for different fleet sizes, we normalize losses per 100 expected insured vehicles[25]. The model specification was explained in Section 4.4.2 which includes in particular a description of the driving configurations. The results are documented in Figure 4.3. The solid lines are the normalized quantities.

In Figure 4.3a, we see that increasing the aggressiveness of driving increases both the total and normalized expected loss. An impact of the maximal acceleration on losses is only substantial for the most aggressive driving configurations $\xi^{3\cdot}$. Increasing the fleet size increases the expected loss which is primarily due to the fact that we count losses only within the fleet and a higher volume is associated with higher losses. More

[25] For each traffic scenario k, the number of insured vehicles is the number of vehicles belonging to Φ as given in the underlying route file. We refer to Appendix B for more details.



(a) Expectation of the total loss

(b) Standard deviation of the total loss

Figure 4.3: Impact of fleet size and driving configuration on the total loss L (solid lines represent normalized values (left y-axis), dashed lines unnormalized ones (right y-axis)).

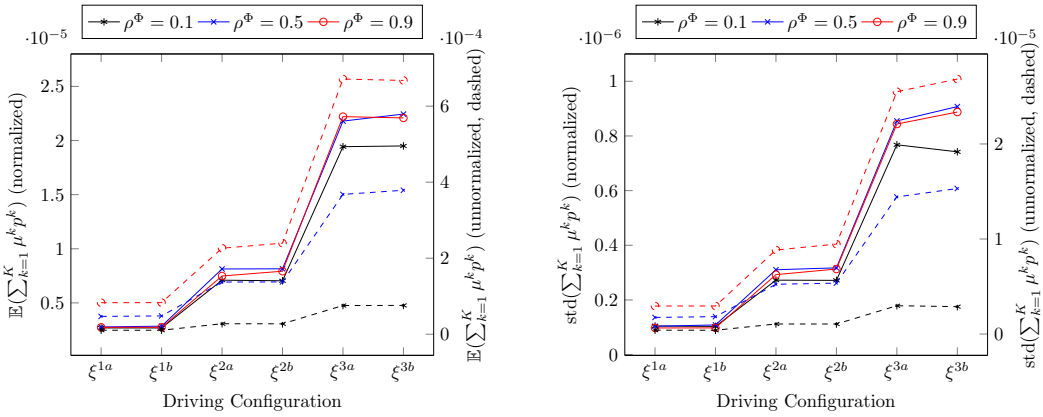
interesting is the normalized case: apparently higher speeds also increase the normalized losses.

In Figure 4.3b, we have the corresponding standard deviations. Increasing the aggressiveness of driving increases the standard deviation of the total loss. The standard deviations of the normalized losses are decreasing in the fleet size. The main reason is that fluctuations normalized for a fixed volume are larger for smaller pools than for larger pools; a rational for this is provided by the law of large numbers and the central limit theorem.

The frequency and severity of accidents are, of course, increasing in the aggressiveness of driving. To demonstrate this, we evaluate the expectation and the standard deviation of the average accident frequency $\sum_{k=1}^{K} \mu^k p^k$ (both normalized and unnormalized) and the average accident severity $\sum_{k=1}^{K} \mu^k X_1^k$, as displayed in Figure 4.4 and Figure 4.5. A larger fleet increases frequency and, in aggressive scenarios, also the expected accident severity. This is, of course, due to the specific choice of the driving behavior of the considered fleets in comparison to the driving behavior of the remaining vehicles and does not necessarily hold for all traffic systems in general.[26]

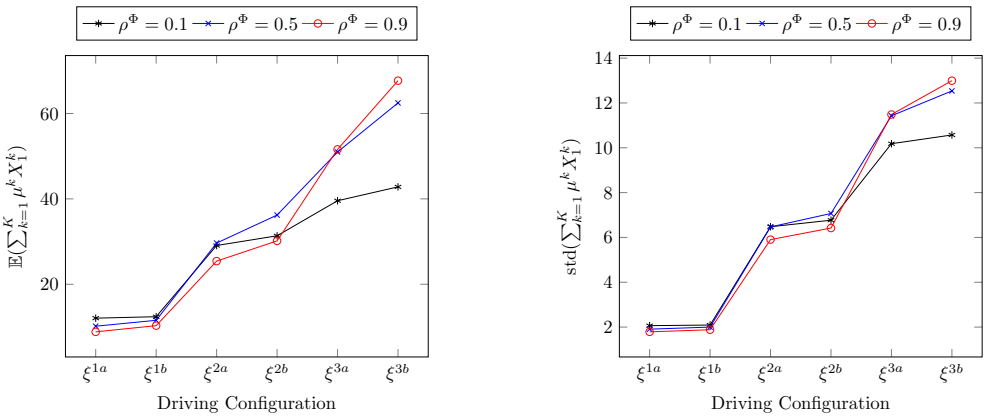
Traffic System Performance. Changing the characteristics of the fleet not only affects the losses. At the same time, this has an impact on the performance of the traffic system. Using the 40 induction loop detectors placed in the traffic system, we evaluate the values of selected traffic statistics (flow, average speed, and occupancy). Denoting them by $\chi_k^1, \ldots, \chi_k^{40}$ for each scenario $k = 1, \ldots, K$, we compute averages $\bar{\chi}_k = 1/40 \sum_{i=1}^{40} \chi_k^i$. Pairing flow-occupancy values and speed-occupancy values yields empirical fundamental diagrams on the urban level (see, e.g., Geroliminis & Daganzo

[26] In this section, we briefly discussed losses for selected special cases. A comprehensive set of detailed tables for all other cases and statistical functionals is provided in Appendix B.



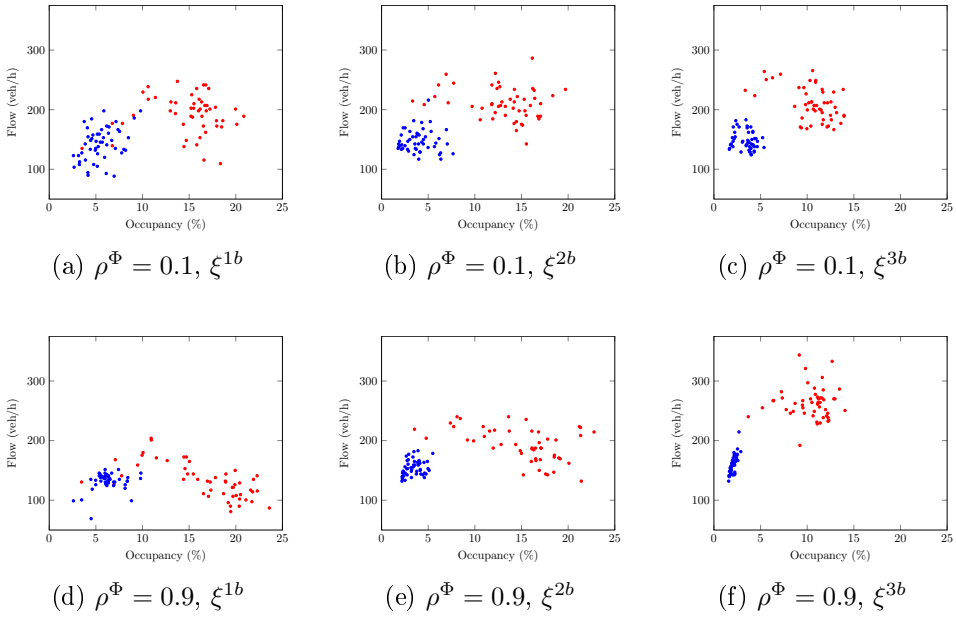
(a) Expectation (b) Standard deviation

Figure 4.4: Impact of fleet size and driving configuration on expectation and standard deviation of average accident frequency (solid lines represent normalized values (left y-axis), dashed lines unnormalized ones (right y-axis)).



(a) Expectation (b) Standard deviation

Figure 4.5: Impact of fleet size and driving configuration on expectation and standard deviation of average accident severity.



(a) $\rho^{\Phi} = 0.1$, ξ^{1b} (b) $\rho^{\Phi} = 0.1$, ξ^{2b} (c) $\rho^{\Phi} = 0.1$, ξ^{3b}

(d) $\rho^{\Phi} = 0.9$, ξ^{1b} (e) $\rho^{\Phi} = 0.9$, ξ^{2b} (f) $\rho^{\Phi} = 0.9$, ξ^{3b}

Figure 4.6: Flow-occupancy fundamental diagrams (high traffic volume scenarios are highlighted in red and low traffic volume scenarios in blue).

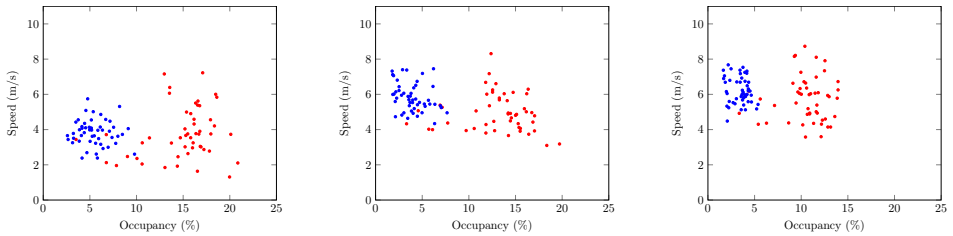
(2008)). For the purpose of data exploration, we draw scatter plots of these pairs for the driving configurations $\xi^{1b}, \xi^{2b}, \xi^{3b}$ and fleet sizes $\rho^{\Phi} = 0.1, 0.9$ in Figure 4.6 and Figure 4.7. We recover the classical u-shape in the flow-occupancy plot. The blue points refer to the low volume traffic scenarios, the red points to the high volume traffic scenarios, as introduced in Section 4.4.1. This is also reflected by the fact that red points correspond to higher occupancy. Aggressiveness in driving decreases overall occupancy, increases speed, and increases flow, if the fleet is large.

To better understand the impact of individual driving configurations and fleet sizes, we study the scenario averages $\mathbb{E}\left(\sum_{k=1}^{K} \mu_k \bar{\chi}_k\right) = \frac{1}{K}\sum_{k=1}^{K} \bar{\chi}_k$. The results are displayed in Figure 4.8. We see that the performance of the traffic system improves with the aggressiveness of driving in the considered case studies; flows and average speeds increase, and the occupancy decreases.

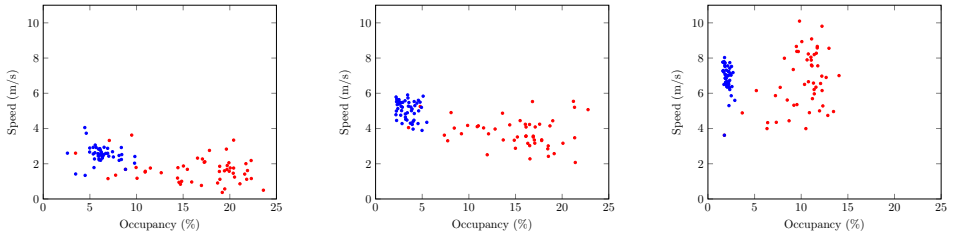
4.4.3.3 The Actuarial Perspective

From an actuarial perspective, it is relevant to understand the *risk* that corresponds to the insurance losses. This requires a more detailed analysis of the *probability distributions*. To do this, we pick a particular fleet model and use probabilistic techniques to evaluate the distribution of $h(L)$. From now on, we consider $\rho^{\Phi} = 0.5$ with driving configuration ξ^{2a} and non-uniform accident occurrence.

Distributional Analysis of Losses. We start our investigations with the total losses L. Table 4.1 shows the evaluation of statistical functionals for different accident losses

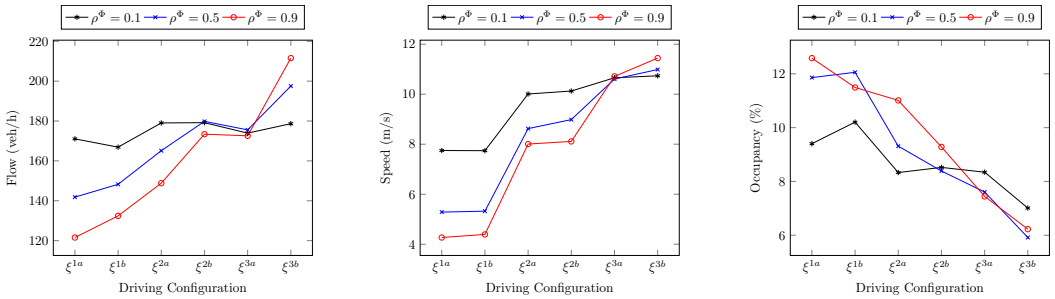


(a) $\rho^{\Phi} = 0.1$, ξ^{1b} (b) $\rho^{\Phi} = 0.1$, ξ^{2b} (c) $\rho^{\Phi} = 0.1$, ξ^{3b}



(d) $\rho^{\Phi} = 0.9$, ξ^{1b} (e) $\rho^{\Phi} = 0.9$, ξ^{2b} (f) $\rho^{\Phi} = 0.9$, ξ^{3b}

Figure 4.7: Speed-occupancy fundamental diagrams (high traffic volume scenarios are highlighted in red and low traffic volume scenarios in blue).



(a) Flow (b) Speed (c) Occupancy

Figure 4.8: Impact of fleet size and driving configuration on traffic system performance.

Table 4.1: Statistical functionals of L for $\rho^{\Phi} = 0.5$ and ζ^{2a}.

	Binomial Model					
	Gamma			Log-Normal		
	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$
$\mathbb{E}(L)$	1577.8	1571.5	1578.4	1581.7	1576.2	1582.7
$\mathrm{Var}(L)$	160179.5	247943.9	626190.6	162067.5	247069.8	614713.4
ς_L	0.333	0.561	0.993	0.377	0.660	2.152
$\mathrm{VaR}_{0.9}(L)$	2111.9	2219.1	2634.2	2112.9	2242.4	2526.6
$\mathrm{ES}_{0.9}(L)$	2331.7	2538.8	3233.3	2342.4	2557.5	3280.3
$\mathrm{VaR}_{0.95}(L)$	2275.9	2471.3	3070.5	2288.4	2468.1	2996.2
$\mathrm{ES}_{0.95}(L)$	2468.9	2748.0	3628.9	2491.4	2772.5	3838.4
$\mathrm{VaR}_{0.99}(L)$	2587.3	2891.3	3999.6	2608.8	2943.8	4228.6
$\mathrm{ES}_{0.99}(L)$	2755.9	3188.6	4490.2	2809.0	3234.4	5424.2

The statistical functionals of the total loss are approximated using 10,000 independent samples of L.

in the case of the Binomial model. These numbers quantify the risk entailed in the total losses. Both the distributional family and the chosen coefficient of variation for the accident loss model have a substantial effect on the risk.

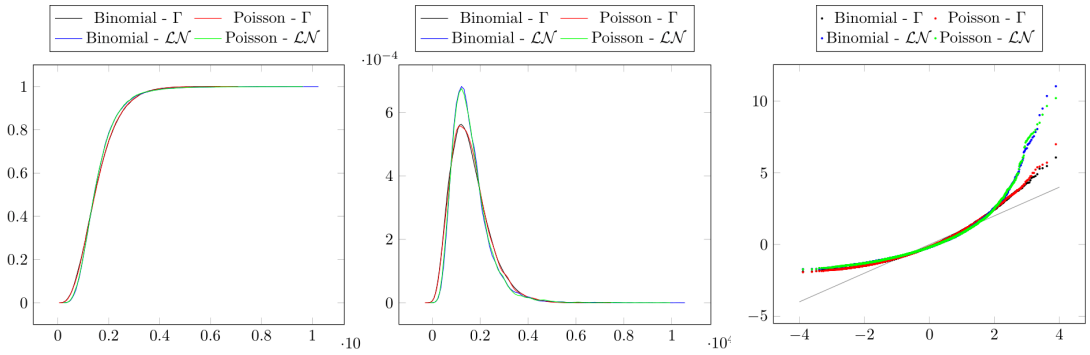
A visual impression of the distributions is provided in Figure 4.9. We compare different accident loss models while fixing the coefficient of variation $c_v = 2$. We plot the empirical distribution functions as estimates of the cumulative distribution function (CDF) and a kernel density estimate of the corresponding densities. Moreover, Figure 4.9c shows qq-plots for the quantiles of standardized[27] values of the losses against quantiles of a standard Normal distribution.

We find that the Binomial and the Poisson model do not differ too much. Yet, log-normal accident losses produce heavier tails than the corresponding Gamma losses. The qq-plots reveal that the total losses are not normally distributed: In particular, the right tails are heavier compared to a Normal distribution while the left tails are lighter. The latter observation simply relates to the fact that the original losses are non-negative while the Normal distribution takes values on the whole real line.

In Figure 4.10, we analyze the impact of the coefficient of variation while fixing the log-normal distribution for the accident losses. We see again that Binomial and Poisson model do not differ substantially. However, the effect of the coefficient of variation is clearly visible: increasing c_v produces heavier right tails. Introducing dispersion to accident losses substantially changes the distribution of the total losses.

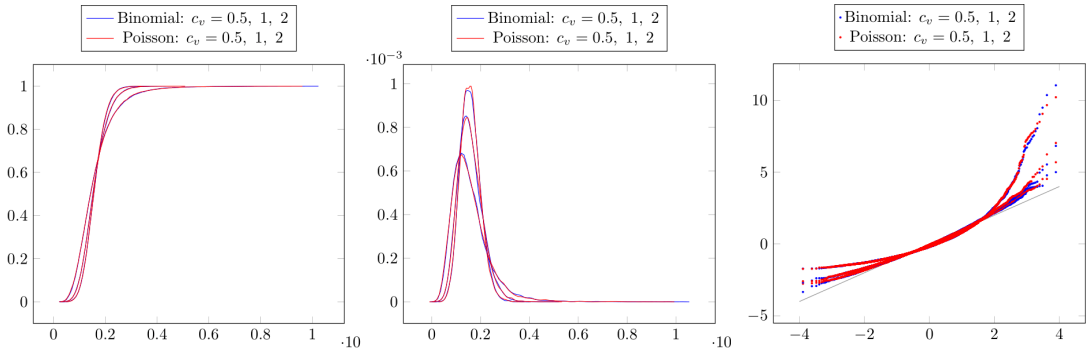
Comparison of Losses and Normal Mean-Variance-Mixture Approximation. In Section 4.3.2, we suggested a Normal mean-variance-mixture approximation for the total loss. To study the quality of this approximation, we generate 10,000 samples from the approximation; in the following, we focus on the case of Gamma distributed accident

[27] Samples are standardized by subtracting their sample average and dividing by their sample standard deviation.



(a) CDF (b) Density (c) QQ-plot

Figure 4.9: Distribution of the total loss for fixed coefficient of variation $c_v = 2$.



(a) CDF (b) Density (c) QQ-plot

Figure 4.10: Distribution of the total loss for log-normal accident losses and varying coefficient of variation.

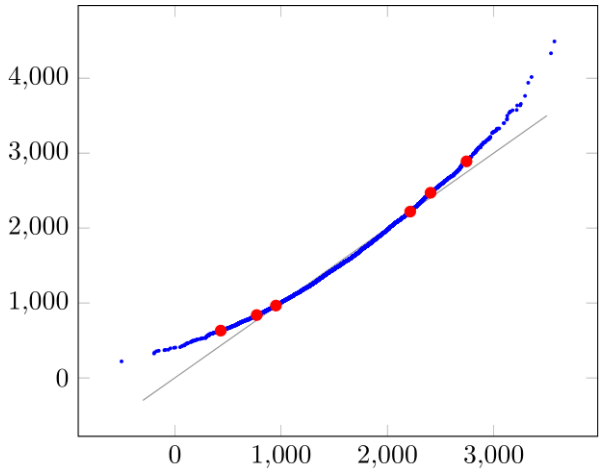


Figure 4.11: QQ-Plot of $10,000$ Monte Carlo Samples (y-axis) vs $10,000$ Samples of Mixture Approximation (x-axis) (the red dots mark the 1%, 5%, 10%, 90%, 95%, 99% quantiles).

losses with coefficient of variation $c_v = 1$.

To sample from the approximation, we rely on the following computations[28] of m^k and $(\sigma^k)^2$. Using $\mathbb{E}(\mathbf{1}_c^k) = p^k$ and $\mathbb{E}(X_c^k) = \mathbb{E}(\mathbb{E}(X_c^k \mid \psi)) = \mathbb{E}(\psi^2) = \int \psi^2 d\mathcal{L}^k$, we obtain for the Gamma losses $m^k = p^k \cdot \int \psi^2 d\mathcal{L}^k$ and $(\sigma^k)^2 = p^k \cdot c_v^4 \cdot \int \psi^4 d\mathcal{L}^k \cdot \left(1 + \frac{1}{c_v^2}\right) \frac{1}{c_v^2} - (m^k)^2$. The involved moments of ψ are approximated using $10,000$ samples from the traffic simulation, for each $k = 1, \ldots, K$. A sample from the Normal mean-variance-mixture approximation is generated by, first, sampling μ and, second (conditional on μ), sampling the normal random variable $\sum_{k=1}^K N\mu^k m^k + Z$ with $Z \sim \mathcal{N}\left(0,\ \sum_{k=1}^K N\mu^k \left(\sigma^k\right)^2\right)$.

Figure 4.11 shows the qq-plot comparing quantiles of the crude Monte Carlo simulation with quantiles of the approximation. This demonstrates the quality of our suggested approximation. It is almost exact between the 5 % and 95 % quantile as the values lie on the halfline. It is still very good for the $1 - 5$ % and $95 - 99$ % quantiles and is only less accurate in the extreme tails where also in the Monte Carlo simulation only few data points are available. These analyses, on the one hand, confirm the postulated structural model insight. On the other hand, they also validate the implementation of our crude Monte Carlo sampling.

Pricing and Evaluation Methods. To conclude our case studies, we study prices for various insurance contracts. We compare $\mathbb{E}(L)$ (full coverage), $\mathbb{E}(\max(L - \theta, 0))$ (constant deductible), and $\mathbb{E}(\min(L, \theta))$ (stop loss) for different values of θ. The results are given in Figure 4.12. We obtain the typical hockey stick profiles satisfying the parity $\mathbb{E}(L) = \mathbb{E}(\max(L - \theta, 0)) + \mathbb{E}(\min(L, \theta))$. We note that other insurance contracts can easily be represented in our framework; also deductibles per accident can be implemented

[28] For the notation, we refer to Sections 4.3.2 & 4.4.2. The computations are valid for any coefficient of variation c_v, but we use only $c_v = 1$ in the numerical case study.

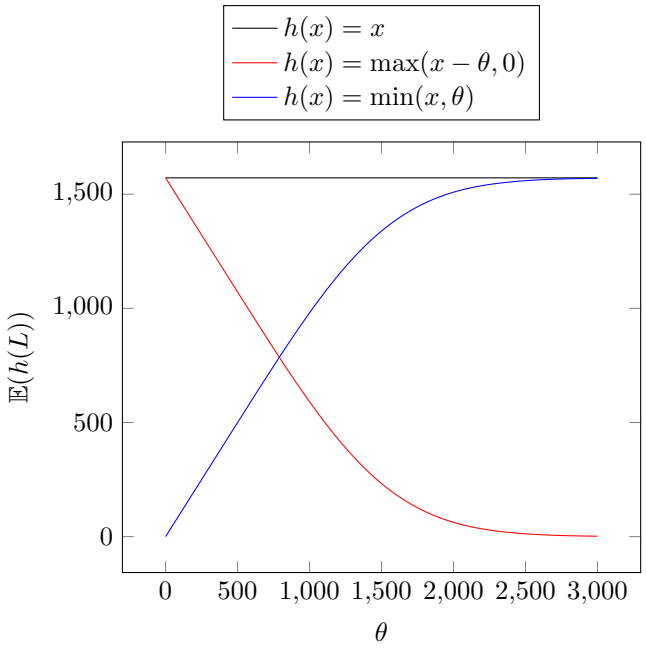


Figure 4.12: Insurance prices.

by changing the accident loss distributions accordingly.

Besides Monte Carlo methods, our Normal mean-variance mixture approximation and the correction suggested in Section 4.3.2 provide alternative techniques to compute the prices $\mathbb{E}(h(L))$ for the different types of coverage h. To compute the correction term C_h, we need to compute[29] $(\zeta^k)^3$. For Gamma distributed losses, we obtain

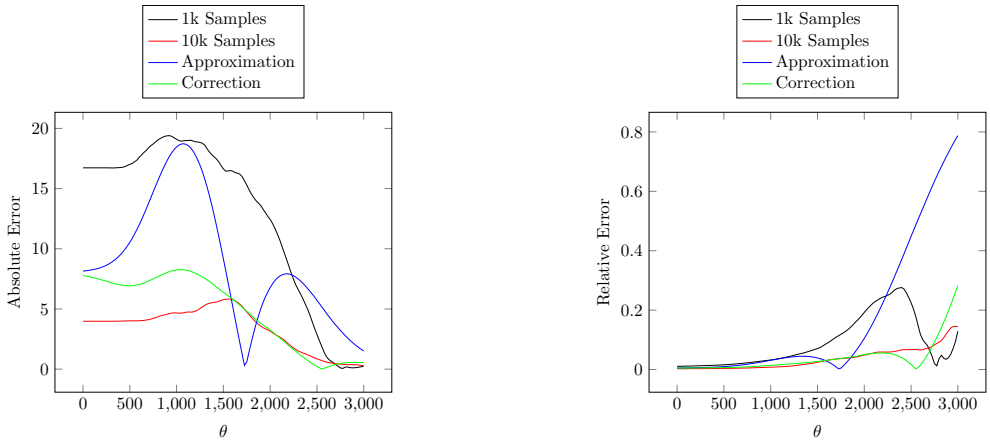
$$(\zeta^k)^3 = p^k \cdot c_v^6 \cdot \int \psi^6 d\mathcal{L}^k \cdot \left(2 + \frac{1}{c_v^2}\right)\left(1 + \frac{1}{c_v^2}\right)\frac{1}{c_v^2} - 3m^k(\sigma^k)^2 - (m^k)^3.$$

Since $\mathbb{E}(h(L)) = \mathbb{E}(\mathbb{E}(h(L) \mid \mu))$, we may generate samples of μ and evaluate the corresponding conditional expectations $\mathbb{E}(h(L) \mid \mu)$; in the Normal mean-variance mixture approximation, these are expectations of functions of normally distributed random variables. We compute these expectations numerically as integrals with respect to Lebesgue measure using a normal density.

We compare the estimation errors of the different approaches in Figure 4.13. We produce $100,000$ samples of L to approximate the "true" value of $\mathbb{E}(\max(L - \theta, 0))$ of coverage with constant deductible, for different values of θ. Independently, we generate $10,000$ samples and consider Monte Carlo approximations based on all $10,000$ samples, and based only on the first $1,000$ samples. We also study the Normal mean-variance mixture approximation with and without correction using $1,000$ samples of μ.

While the absolute error generally decreases in the deductible θ for all methods (apart from some local effects), the relative error increases for larger values of θ which are associated with a lower price of the contract. At the same time, we observe that, in

[29] For the notation we refer to Sections 4.3.2 & 4.4.2.



(a) Absolute error

(b) Relative error

Figure 4.13: Comparison of estimation errors.

terms of the relative error, the Normal mean-variance mixture approximation produces reasonable estimation results (compared to $1,000$ samples) for moderate values of θ. This is in line with our previous observations on the quality of our Normal mean-variance mixture approximation which becomes worse in the extreme tail of L. Interestingly, the estimation error can largely be reduced using the correction; with the correction, the estimation becomes quite good even for large values of θ.

4.5 Conclusion

This chapter developed a methodology to study accident losses based on an established microscopic traffic simulator, here SUMO. An adaption of the digital twin paradigm enabled us to test the impact of fleet sizes and their driving configuration on system efficiency, accident losses, and insurance premiums. We illustrated in counterfactual case studies how accident risk can be successfully analyzed, both from an engineering and an actuarial perspective. It was shown that – on a one-year horizon – total losses can be approximated by a mean-variance mixture of normal distributions. This offered an alternative technique to evaluate the model; the numerical efficiency can be increased adding a correction term that is derived by Stein's method. The proposed methodology can be extended and modified, for example, based on traffic simulators other than SUMO, and utilized to study future traffic systems.

Future research should also address the important issues of calibration and validation. While real data can be used to calibrate models that describe historical and current transportation systems, simulation models that generate artificial data are essential to evaluate new technologies in future transportation systems. An important question is to what extent and how historical data can be methodically used to calibrate and validate such simulation models. For example, real microscopic traffic data, e.g., on accident

patterns, collected by means of telematics technologies, could be applied to optimize models in the future.

4.6 Appendix: Sampling Procedure

We provide a detailed pseudo-code for the procedure to obtain samples from L in Algorithm 1. In our case studies, we use $M' = M = 10,000$ samples of ψ in each scenario k to approximate its distribution $\mathcal{L}^k$. We note that, instead of this bootstrapping approach, one could also pre-sample sufficiently many values.

Algorithm 1 Sampling of losses L.

Phase 1: Prior Evaluation of Traffic Model

for $k = 1, \dots, K$ **do**
  Run SUMO in scenario k.
  Obtain data to calculate p_r^k and λ_r^k for $r = 1, \dots, R$.
  Terminate SUMO.
  Set $p^k = \sum_{r=1}^R p_r^k$ and $\lambda^k = \sum_{r=1}^R \lambda_r^k$.
end for

Phase 2: Pre-Sampling of ψ conditional on scenario $k = 1, \dots, K$.

for $k = 1, \dots, K$ **do**
  for $j = 1, \dots, M'$ **do**
    Sample $\hat{t}_j = \text{Unif}(0, T)$.
    Sample $\hat{r}_j \sim \mathcal{R}$ where $P(\mathcal{R} = r) = p_r^k / p^k$ (or $P(\mathcal{R} = r) = \lambda_r^k / \lambda^k$).
  end for
  Sort $\hat{t}_1, \dots, \hat{t}_{M'}$ by size (again denoted by $\hat{t}_1, \dots, \hat{t}_{M'}$).
  Start SUMO in scenario k.
  for $j = 1, \dots, M'$ **do**
    Continue SUMO until time $\hat{t}_j$.
    Sample $\hat{i}_j \sim \text{Unif}\left(\mathcal{M}_{\hat{r}_j}^{\Phi}\left(\hat{t}_j\right)\right)$.
    Set $\hat{\psi}_j^k = v^{\hat{i}_j}(\hat{t}_j)$.
  end for
  Terminate SUMO.
  Store $\hat{\psi}_1^k, \dots, \hat{\psi}_{M'}^k$.
end for

Phase 3: Sampling of total losses L.

for $j = 1, \dots, M$ **do**
  Sample $\hat{y}^j$ with $P(y = g) = P(y = b) = 1/2$.
  for $n = 1, \dots, N$ **do**
    Sample $\hat{\nu}^{j,n} \sim \nu_{\hat{y}^j}$.
  end for
  for $k = 1, \dots, K$ **do**
    Set $(\hat{\mu}^k)^j = 1/N \sum_{n=1}^N \mathbb{1}\{\hat{\nu}^{j,n} = k\}$.
  end for
  for $k = 1, \dots, K$ **do**
    Sample $\left(\hat{C}^k\right)^j \sim \text{Bin}(p^k, N(\hat{\mu}^k)^j)$ (or $\left(\hat{C}^k\right)^j \sim \text{Poiss}(\lambda^k N(\hat{\mu}^k)^j)$).
    for $c = 1, \dots, \left(\hat{C}^k\right)^j$ **do**
      Sample $\hat{l} \sim \text{Unif}(\{1, \dots, M'\})$.
      Sample $\hat{X}_c^k \sim F^{\hat{\psi}_{\hat{l}}^k}$.
    end for
  end for
  Set $\hat{L}^j = \sum_{k=1}^K \sum_{c=1}^{(\hat{C}^k)^j} \hat{X}_c^k$.
end for

Output: $\hat{L}^1, \dots, \hat{L}^M$.

5 Stochastic Cell Transmission Models of Traffic Networks

The original version of this chapter has been submitted for publication, see Feinstein, Kleiber & Weber (2023).

5.1 Introduction

Cell transmission models enable to capture the motion of traffic participants on a high level of aggregation. This provides computational advantages in comparison to microscopic traffic models that capture the motion of traffic participants in great detail. This gain in computational efficiency is sometimes disadvantageously associated with lower granularity, which complicates the representation of complex traffic modules and interactions of traffic participants. In this chapter, we propose a rigorous framework for cell transmission models that incorporates three important features: a) The cells are identified with the nodes of a graph. We introduce a precise notation for the directions of the traffic participants within each cell. This allows the construction of cell transmission models for general traffic networks. b) Within each cell, road users traveling in one direction interact with road users traveling in other directions. Sending and receiving functions can capture these interactions of traffic flow and density with oncoming traffic flows and densities. c) Traffic volumes and conditions may vary randomly. Our general framework allows the inclusion of probabilistic phenomena.

The proposed models enable the evaluation of traffic systems under a wide range of conditions. They can also be used for traffic planning by testing the effects of changes in design parameters. Comparisons can be made not only for deterministic systems, but also in the face of randomness and risk. We use preference functionals and their level sets for the normative classification and categorization of transportation systems. This approach is also closely related to the construction of measures of systemic risk. In concrete applications, we specify random benchmark flows and compute the collection of parameters associated with traffic systems that are weakly preferred. We call these sets *acceptable designs*. They accurately combine the descriptive, possibly random cell transmission model and the normative evaluation framework.

Although less granular than microscopic traffic models, the extended flexibility compared to classical cell transmission models and the inclusion of randomness increase the computational complexity. In particular, stochastic simulation of the system under dif-

ferent conditions for multiple design parameters is costly. To address this problem, we employ a powerful machine learning technique, Gaussian process regression (GPR). GPR allows an interpolation of system performance between simulated points while providing at the same time measures of uncertainty. Our innovation is the adaptive selection of points to refine the estimation of the acceptable designs. For this purpose, we use the GPR estimation of the boundary of this level set and GPR variance estimates at candidate points from the previous iteration. We also provide error bounds.

The capabilities of our algorithms in the context of generalized cell transmission models are illustrated in two case studies. We study two traffic networks, one with two signalized intersections and another one with variable capacities of highways and speed limits. Acceptable designs are identified and interpreted. From an algorithmic point of view, we compare the squared exponential kernel to Matérn kernels.

Our main contributions are the following:

(i) We provide a rigorous framework for cell transmission models in general traffic networks. Traffic participants traveling in different directions interact with each other locally. Traffic volumes and conditions can vary stochastically.

(ii) To classify and categorize traffic systems, we propose the notion of acceptable design inspired by preference functionals and systemic risk measures.

(iii) The numerical estimation of acceptable designs combines Monte Carlo simulation, Gaussian process regression, and a stochastic exploration procedure in the parameter space. The performance of this algorithm is demonstrated through case studies.

5.1.1 Structure of the Chapter

The chapter is organized as follows: Section 5.1.2 reviews the related literature on cell transmission models, systemic risk measures, and Gaussian process regression. Section 5.2 presents our general framework for cell transmission models of traffic networks. Section 5.3 describes the objective of the machine learning estimation problem: sets of acceptable designs. Our algorithm is discussed in Section 5.4. It is applied in numerical case studies in Section 5.5. Questions for further research are presented in Section 5.6. The appendix contains proofs and auxiliary material.

5.1.2 Literature

We develop a general and rigorous formulation of cell transmission models in a network environment. Considering general transmission and receiving functions and general directions of travel, we can consider many related contributions in the literature as special cases. Inspired by the theory of systemic risk measurements, we introduce as a diagnostic instrument the notion of acceptable designs of a traffic system. This constitutes

a normative instrument of traffic planning, which enables the evaluation and control of traffic systems. To identify the acceptable designs of traffic systems, we develop an active learning method based on Gaussian process regression, a powerful machine learning technique. In the following, we review the relevant literature and compare it with our innovations.

Cell Transmission Models. The classical cell transmission model (CTM) was developed in the seminal work of Daganzo (1994) and Daganzo (1995). It is a deterministic macroscopic traffic model that captures the evolution of traffic flows and densities in discrete time. Daganzo (1994) explains how this model can be viewed as a discrete approximation to the LWR model (Lighthill & Whitham (1955) and Richards (1956)); accordingly, he originally introduces CTM to study homogeneous traffic flows on highways.

Since then, CTM has been revisited countless times in the literature. Important research questions range from estimating traffic densities (Munoz et al. (2003)) to establishing variable speed limits on highways (Hadiuzzaman & Qiu (2013))). The introduction of randomness increases the informativeness and allows a representation of more complex phenomena. Sumalee et al. (2011) develop a version with stochastic demand and supply constraints. Jin & Amin (2019) study highway dynamics under random capacity-reducing incidents modeled by an exogenous Markov chain.

However, the popularity of CTM is also due to the fact that it can be used to represent urban traffic. For example, Long et al. (2008) examines the formation and dissipation of congestion in urban networks. Other papers discuss traffic lights and their optimization (e.g., Pohlmann & Friedrich (2010), Xie et al. (2013), Srivastava, Jin & Lebacque (2015)). We refer to Adacher & Tiriolo (2018) for a detailed review on CTM, especially for urban traffic.

Another strand of literature generalizes CTM for different traffic types. Different traffic users with their different driving characteristics can share the available space. Tuerprasert & Aswakul (2010) and Tiaprasert et al. (2017) partition a cell, Levin & Boyles (2016) add partial densities. Buses can be introduced as moving bottlenecks that reduce capacity (Liu et al. (2015), Tang et al. (2022)).

In this chapter, we explain how cells can represent different types of roads, including highways, roundabouts, signalized intersections. Our setup allows for randomness as a general modeling paradigm. We present a precise formulation of the direction of travel that allows detailed modeling of the interaction of competing traffic flows. Some related conceptional issues are also discussed by Tampère et al. (2011). We note that conflicting flows are common in pedestrian flow modeling (see, e.g., Flötteröd & Lämmel (2015)); Moustaid & Flötteröd (2021) can be viewed as a special case of our model. We also briefly indicate how our model can be generalized for multiple interacting traffic types.

Systemic Risk Measures. The axiomatic theory on the quantification of risk dates back to the seminal paper Artzner et al. (1999). Past contributions have focused primarily on the quantification of financial risk: The central construction is based on a notion of acceptability, i.e., a set of (financial) positions with an acceptable risk. A monetary risk measure quantifies the risk of a financial position in monetary units: It is the minimum amount of cash that must be added to a position to make it acceptable. We refer to Föllmer & Weber (2015) for an overview.

This construction can be generalized to quantify the risk of a system of interacting entities. Systemic risk measures as a precise mathematical notion are introduced by Feinstein, Rudloff & Weber (2017) and Biagini et al. (2018). The theory has proven useful not only for quantifying risk in financial networks (e.g., Weber & Weske (2017)); Cassidy, Feinstein & Nehorai (2016) applies it to measuring the risk of power outages in transmission networks. Salomon et al. (2020) use it to control the resilience of technical systems.

Borrowing from these measures of systemic risk, we introduce the concept of acceptability to assess the efficiency of traffic systems. The results, typically measures of efficiency such as traffic flow, are normatively categorized into acceptable and unacceptable outcomes. The set of acceptable designs of a traffic system then refers to those design parameters (e.g., noise parameters, traffic light configurations, initial densities) that lead to acceptable outcomes. As a conceptual difference, we recognize that systemic risk measures are introduced for financial systems whose risk decreases with the amount of available capital. There is no such a priori monotonic dependence of traffic flows on underlying system parameters.

Gaussian Process Regression. Mathematically, the acceptable designs of a traffic system form a set of real vectors defined in terms of a level set of a function. To estimate this set, we estimate the underlying function. To address the computational cost in the context of stochastic simulation, we develop an active learning approach.

We apply Gaussian process regression (GPR), also called kriging, as a Bayesian inference method to estimate a metamodel from isolated noisy data. The method assumes a Gaussian process as a prior distribution over functions and is updated with observed data to produce an estimate. The popularity of this method is due to its probabilistic foundation, which also allows an evaluation of the uncertainty of the estimate. We refer to Rasmussen & Williams (2005) as a standard reference and Kanagawa et al. (2018) and Swiler et al. (2020) for insightful surveys.

Ankenman, Nelson & Staum (2010) examines GPR as a tool for metamodeling in the context of simulation. Similarly, Binois, Gramacy & Ludkovski (2018) discusses practical aspects. The focus is on approximating the entire underlying function rather than just a particular level set. Most closely related to our active learning framework are Gotovos et al. (2013) and Lyu, Binois & Ludkovski (2021), which also develop iterative procedures

for approximating level sets with GPR. In contrast to previous work, we construct a random search algorithm to determine where the function values will be estimated next. This circumvents the need to rely on complicated optimization methods. In addition, we introduce a general sandwich principle to impose upper bounds on the approximation error of set estimation algorithms. Error bounds on the estimated function, as extensively discussed in Srinivas et al. (2012) and Lederer, Umlauft & Hirche (2019), implicate bounds on the approximation error of level sets.

5.2 Cell Transmission Models for Traffic Networks

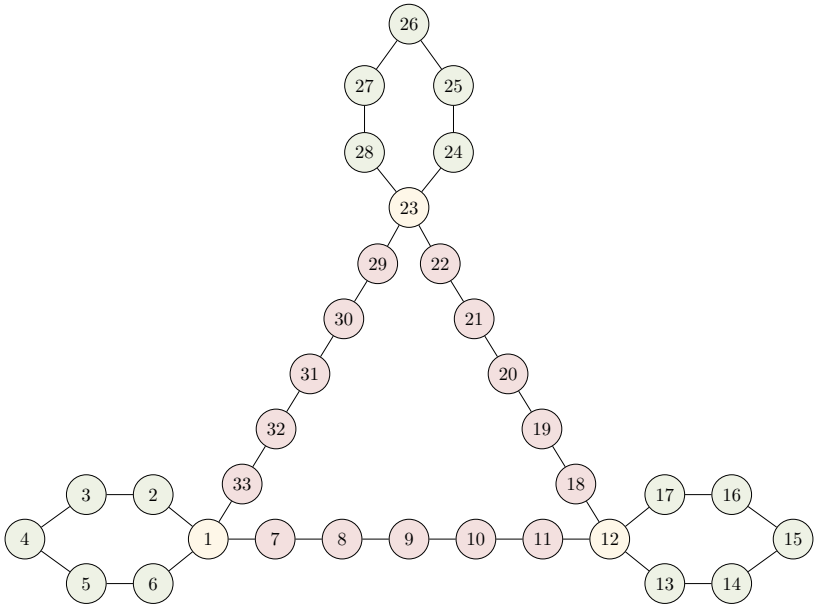


Figure 5.1: Highway network.

5.2.1 A Motivating Example

Cell transmission models capture the dynamic evolution of traffic densities and flows in traffic networks. Figure 5.1 shows a stylized network with 28 cells or nodes. The nodes can be of different types, for example, the red and green nodes in Figure 5.1 could be sections of highways or roads, and the yellow nodes could be intersections or roundabouts. The type of each node determines how traffic participants interact within the node. The total traffic volume in the system may vary (possibly randomly) due to traffic participants entering or leaving the system. For example, the green areas in Figure 5.1 could be sources and sinks of the traffic network.

The model determines how much traffic is transmitted from one node to the next. In generalized cell transmission models on graphs, traffic in the nodes may move in various directions. This is especially true for nodes that are connected to multiple other nodes,

such as nodes 1, 12, and 23. It is thus important to indicate different directions or routes by notation when modeling traffic dynamics. A rigorous framework is described in the next sections.

5.2.2 General Framework

To model general traffic networks, we consider a set of vertices or nodes V denoting all existing traffic spaces; these are connected by a collection of edges $E \subseteq V \times V$. The corresponding graph $G = (V, E)$ is the traffic network under consideration. Vehicles, bicycles, and pedestrians move through the graph, and traffic flows interact with each other at the nodes. Traffic flows and corresponding traffic densities in a node v are distinguished by the preceding location $u \in \mathscr{I}(v)$ and the subsequent destination $w \in \mathcal{O}(v)$; here, the sets $\mathscr{I}(v) = \{v' : (v', v) \in E\}$ and $\mathcal{O}(v) = \{v' : (v, v') \in E\}$ collect the nodes from which v can be reached and which can be reached from v, respectively.

Traffic Dynamics. Traffic is modeled in discrete time, enumerated as $t = 0, 1, 2, \ldots$. For simplicity, we consider only one type of traffic participants, although the formalism can easily be extended to multiple types. For each node v and each route (u, v, w) through v with $u \in \mathscr{I}(v)$, $w \in \mathcal{O}(v)$, $\rho_{(u,v,w)}(t)$ is the traffic density of agents at time t traveling through v along (u, v, w). The flow into v of traffic participants traveling along (u, v, w) through v during the time interval $(t, t+1)$ is denoted by $q^{\text{in}}_{(u,v,w)}(t+1)$; the corresponding flow out of v is $q^{\text{out}}_{(u,v,w)}(t+1)$. We further assume that sources and sinks exist in the system, and denote by $q^{\text{net}}_{(u,v,w)}(t+1)$ the (possibly random) net flow of traffic participants entering or leaving the system in v on the route (u, v, w) during the time interval $(t, t+1)$. Nodes represent traffic spaces such as roads, intersections, roundabouts, shared spaces, and can be of different sizes, labeled l_v for $v \in V$. Densities are updated iteratively for each $v \in V$, $u \in \mathscr{I}(v)$ and $w \in \mathcal{O}(v)$ at each point in time $t+1$:

$$\rho_{(u,v,w)}(t+1) = \rho_{(u,v,w)}(t) + \frac{1}{l_v}\left(q^{\text{in}}_{(u,v,w)}(t+1) - q^{\text{out}}_{(u,v,w)}(t+1) + q^{\text{net}}_{(u,v,w)}(t+1)\right) \quad (5.2.1)$$

To model a specific traffic network, its initial conditions and the dynamic behavior of sources and sinks must be specified exogenously. Traffic flows are constrained on routes (u, v, w) by the characteristics of the corresponding traffic space, the density of agents on that route, and other traffic participants traveling through v; this is modeled by general supply and demand constraints. In addition, we assume that turning fractions are conserved, cf. Tampère et al. (2011); this captures the principle of first-in-first-out for incoming traffic. The solution of the traffic model according to (5.2.1) is defined as the solution of a global optimization problem that maximizes traffic flow. This solution captures a perfect cooperation among traffic participants to achieve the objective and provides a benchmark solution; inspired by Tampère et al. (2011), we introduce interac-

tion rules in Section 5.2.4 that realistically capture non-cooperative behavior. Adequate traffic dynamics is the solution of an optimization problem under these additional constraints.[1]

Sending and Receiving Functions. We begin by discussing general supply and demand constraints. Traffic densities constrain both inflow and outflow; this includes both the density of traffic on a given route (u, v, w) and the counter-densities of routes that pass through traffic space v. The demand constraint is formalized by the sending function S, which captures existing traffic participants that would leave a traffic area in the next step if they could continue without any constraints in the subsequent modules (*"demand for space by traffic participants", "flow that may be sent"*). The supply constraint is formalized by the receiving function R and captures the maximum amount of traffic that can be absorbed from preceding traffic modules (*"supply of space for traffic participants", "flow that may be received"*). The sending function S and the receiving function R are of the following form:[2]

$$S_{(u,v,w)}: \quad \begin{cases} \mathbb{R}_+^{\mathscr{I}(v)\times\mathcal{O}(v)} & \longrightarrow & \mathbb{R}_+ \\ (\rho_{(u',v,w')})_{u'\in\mathscr{I}(v),w'\in\mathcal{O}(v)} & \mapsto & q_{(u,v,w)} \end{cases}$$

$$R_{(u,v,w)}: \quad \begin{cases} \mathbb{R}_+^{\mathscr{I}(v)\times\mathcal{O}(v)} & \longrightarrow & \mathbb{R}_+ \\ (\rho_{(u',v,w')})_{u'\in\mathscr{I}(v),w'\in\mathcal{O}(v)} & \mapsto & q_{(u,v,w)} \end{cases}$$

They bound inflow and outflow, i.e.,

$$\begin{aligned} q^{\text{in}}_{(u,v,w)}(t+1) &\leq R_{(u,v,w)}\left((\rho_{(u',v,w')}(t))_{u'\in\mathscr{I}(v),w'\in\mathcal{O}(v)}\right), \\ q^{\text{out}}_{(u,v,w)}(t+1) &\leq S_{(u,v,w)}\left((\rho_{(u',v,w')}(t))_{u'\in\mathscr{I}(v),w'\in\mathcal{O}(v)}\right). \end{aligned}$$

For the sending function that bounds the outflow, we will always require that

$$S_{(u,v,w)}(\rho_{(u',v,w')}(t))_{u'\in\mathscr{I}(v),w'\in\mathcal{O}(v)} \leq \rho_{(u,v,w)}(t), \tag{5.2.2}$$

i.e., the outflow during the time interval $(t, t+1)$ cannot be greater than the occupation of (u, v, w) with traffic participants.

Sending and receiving functions do not have to be constant in time, but can vary periodically, randomly or depending on circumstances. This can be modeled by a dependency on additional state variables besides the dependency on the traffic densities of the agents on the paths through a node.

[1] The cooperative benchmark does not guarantee the greatest flow over a longer time horizon because myopic optimization focuses on a single time period. When focusing on a longer time horizon, constrained optimization solutions may be superior in some cases, although they are suboptimal for each time window given the same state at the beginning of that period.

[2] Another function with the same domain and range is the fundamental diagram that characterizes the stationary or long-run traffic flow on a given route (u, v, w), if both demand from preceding traffic modules and supply of subsequent modules are unrestricted. It can be computed as the minimum of sending function $S_{(u,v,w)}$ and receiving function $R_{(u,v,w)}$.

Cooperative Driving Benchmark Model. To be able to specify the cooperative driving benchmark model, we denote the fraction of traffic participants on (u, v, w) turning to $y \in \mathcal{O}(w)$ by $f_{(u,v,w)\to y}(t+1) \geq 0$ with $\sum_{y\in\mathcal{O}(w)} f_{(u,v,w)\to y}(t+1) = 1$. This leads to the following identity:

$$q^{\text{in}}_{(u,v,w)}(t+1) = \sum_{x\in\mathscr{I}(u)} f_{(x,u,v)\to w}(t+1) \cdot q^{\text{out}}_{(x,u,v)}(t+1) \tag{5.2.3}$$

The flows $q^{\text{out}}_{(u,v,w)}(t+1)$ are the solutions of the following myopic global optimization problem:

$$\text{argmax} \sum_{v\in V} \sum_{u\in\mathscr{I}(v),\, w\in\mathcal{O}(v)} q^{\text{out}}_{(u,v,w)}(t+1) \tag{5.2.4}$$

s.t. for all $v \in V$, $u \in \mathscr{I}(v)$, $w \in \mathcal{O}(v)$:

- $q^{\text{out}}_{(u,v,w)}(t+1) \geq 0$,
- $q^{\text{in}}_{(u,v,w)}(t+1) = \sum_{x\in\mathscr{I}(u)} f_{(x,u,v)\to w}(t+1) \cdot q^{\text{out}}_{(x,u,v)}(t+1)$,
- $q^{\text{in}}_{(u,v,w)}(t+1) \leq R_{(u,v,w)}\left((\rho_{(u',v,w')}(t))_{u'\in\mathscr{I}(v),w'\in\mathcal{O}(v)}\right)$,
- $q^{\text{out}}_{(u,v,w)}(t+1) \leq S_{(u,v,w)}\left((\rho_{(u',v,w')}(t))_{u'\in\mathscr{I}(v),w'\in\mathcal{O}(v)}\right)$.

Remark 5.2.1. *The global problem can be split into decoupled local problems. Since*

$$\sum_{v\in V} \sum_{u\in\mathscr{I}(v),\, w\in\mathcal{O}(v)} q^{\text{out}}_{(u,v,w)}(t+1) = \sum_{v\in V} \sum_{u\in\mathscr{I}(v)} \sum_{x\in\mathscr{I}(u)} q^{\text{out}}_{(x,u,v)}(t+1),$$

we can instead solve the decoupled problems

$$\text{argmax} \sum_{x\in\mathscr{I}(u)} q^{\text{out}}_{(x,u,v)}(t+1) \tag{5.2.5}$$

for all $v \in V$, $u \in \mathscr{I}(v)$ under the corresponding constraints.[3]

5.2.3 Examples of Traffic Nodes

Our general framework allows to capture various traffic modules, including the cell transmission model of Daganzo (1994), roads, multidirectional pedestrian areas as in Moustaid & Flötteröd (2021), unsignalized and signalized intersections, roundabouts, and many other types of traffic spaces. The model can also be extended to multiple types of traffic participants by introducing additional constraints that link these types together.

To describe a few motivating examples, consider a node labeled # and assume for simplicity that $\mathscr{I}(\#) = \mathcal{O}(\#)$ are adjacent nodes in a plane. Its elements are enumerated

[3] The constraints require that for all $x \in \mathscr{I}(u)$, $w \in \mathcal{O}(v)$: $q^{\text{out}}_{(x,u,v)}(t+1) \geq 0$, $q^{\text{in}}_{(u,v,w)}(t+1) = \sum_{x\in\mathscr{I}(u)} f_{(x,u,v)\to w}(t+1) \cdot q^{\text{out}}_{(x,u,v)}(t+1)$, $q^{\text{in}}_{(u,v,w)}(t+1) \leq R_{(u,v,w)}\left((\rho_{(u',v,w')}(t))_{u'\in\mathscr{I}(v),w'\in\mathcal{O}(v)}\right)$, $q^{\text{out}}_{(x,u,v)}(t+1) \leq S_{(x,u,v)}\left((\rho_{(x',u,v')}(t))_{x'\in\mathscr{I}(u),v'\in\mathcal{O}(u)}\right)$.

counterclockwise by $0, 1, \ldots, n-1$ with $n := \operatorname{card}(\mathscr{I}(\#))$. A convenient approach will be to identify $\mathscr{I}(\#)$ with the additive group $\mathbb{Z}_n$, i.e., to equip $0, 1, \ldots, n-1$ with the operation + modulo n.

Highways. Lanes of highways are separated, thus interaction between different directions is not present. Setting $\mathscr{I}(\#) = \mathbb{Z}_2$, a simple linear model for sending and receiving functions is, with $u \in \mathbb{Z}_2$,

$$\begin{aligned} S_{(u,\#,u+1)}\left(\left(\rho_{(u',\#,w')}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#)}\right) &= \min\left(s_{\#}^{\max}, a\rho_{(u,\#,u+1)}\right), \\ R_{(u,\#,u+1)}\left(\left(\rho_{(u',\#,w')}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#)}\right) &= \max\left(b\left(\frac{\rho_{\#}^{\max}}{2} - c\rho_{(u,\#,u+1)}\right), 0\right), \end{aligned}$$

where $\rho_{\#}^{\max} > 0$ is the maximum density, $s_{\#}^{\max} > 0$ is the maximum flow, $0 < a \leq 1$ is the free-flow speed, $0 < b \leq 1$ is the congestion wave speed, and $c > 0$ is an interaction parameter.

Bidirectional Linear Interfaces. A generalization of the unidirectional situation is adequate for pedestrians with bidirectional interacting flows. We introduce an additional interaction parameter $d > 0$ that reflects the impact of traffic travelling in opposite direction and obtain a simple linear model for sending and receiving functions; their specification is, with $u \in \mathbb{Z}_2$,

$$\begin{aligned} S_{(u,\#,u+1)}\left(\left(\rho_{(u',\#,w')}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#)}\right) &= \min\left(s_{\#}^{\max}, a\rho_{(u,\#,u+1)}\right), \\ R_{(u,\#,u+1)}\left(\left(\rho_{(u',\#,w')}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#)}\right) &= \max\left(b\left(\rho_{\#}^{\max} - c\rho_{(u,\#,u+1)} - d\rho_{(u+1,\#,u)}\right), 0\right). \end{aligned}$$

The models may, of course, include nonlinear relationships if the data or expert knowledge suggest other functional forms. An example can be found in Moustaid & Flötteröd (2021), where the sending function is nonlinear if both the density and the counter-density are subcritical. Their model can be easily transferred to our notation.

Pedestrian Square. Bidirectional linear interfaces can be generalized canonically to multiple directions. A fully symmetric geometry with n entries/exits and can be captured by a node # with $\mathscr{I}(\#) = \mathbb{Z}_n$. Assuming pedestrians do not return to the same entry, constant turning rates $f_{(x,u,\#)\to w}(t) \equiv 1/(n-1)$ for all $u \in \mathbb{Z}_n$, $w \in \mathbb{Z}_n \setminus \{u\}$ are a simple choice. For $u \neq w$, a simple model is

$$\begin{aligned} S_{(u,\#,w)}\left(\left(\rho_{(u',\#,w')}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#)}\right) &= \min\left(s_{\#}^{\max}, a\rho_{(u,\#,w)}\right), \\ R_{(u,\#,w)}\left(\left(\rho_{(u',\#,w')}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#)}\right) &= \max\left(b\left(\rho_{\#}^{\max} - c\rho_{(u,\#,w)}\right.\right. \\ &\qquad \left.\left. - d\sum_{\substack{u'\in\mathscr{I}(\#)\setminus\{u\},\\ w'\in\mathcal{O}(\#)\setminus\{w\}}} \rho_{(u',\#,w')}\right), 0\right). \end{aligned}$$

Roundabouts. In roundabouts the interaction of the various participants is also determined by the overlap of their paths. This leads to somewhat more complicated sending and receiving functions, but the main ideas are similar to those outlined above. We provide a description of the details in Appendix 5.7, including a bidirectional roundabout for pedestrians and an extension of the framework to traffic models with multiple populations.

Intersections. Just as with roundabouts, complex interactions of traffic participants can also be modeled at intersections. As examples, we consider two cases: a highly simplified model of an unsignalized junction and a rather complex model of an intersection with a traffic light.

A possible *simplified model* of an intersection adjusts the pedestrian space. Since cars move slower on crowded intersections, we include an exponential damping factor between 0 and 1 in the definition of the sending function that depends on a parameter $\zeta > 0$. For $u \in \mathscr{I}(v)$ and $u \neq w \in \mathcal{O}(v)$ we set

$$S_{(u,\#,w)}\left(\left(\rho_{(u',\#,w')}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#)}\right) = \min\left(s_{\#}^{\max}, a\rho_{(u,\#,w)}\exp\Big(-\zeta\cdot\sum_{\substack{u'\in\mathscr{I}(\#),\\ w'\in\mathcal{O}(\#)}}\rho_{(u',\#,w')}\Big)\right),$$

$$R_{(u,\#,w)}\left(\left(\rho_{(u',\#,w')}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#)}\right) = \max\left(b\Big(\rho_{\#}^{\max} - c\sum_{\substack{u'\in\mathscr{I}(\#),\\ w'\in\mathcal{O}(\#)}}\rho_{(u',\#,w')}\Big), 0\right).$$

At the other end of the spectrum are models incorporating more details that allow cell transmission models (albeit still much simpler than microscopic models) to reproduce complex interaction patterns. As an example, consider a *signalized intersection* with $\mathscr{I}(\#) = \mathcal{O}(\#) = \mathbb{Z}_4$. Consider $u \in \mathscr{I}(\#)$ and right-hand traffic. The path $(u, v, u+1)$ corresponds to a right turn, while the paths $(u, v, u+2)$ and $(u, v, u+3)$ represent going straight and a left turn, respectively. As additional state variables, we consider for each path the current signal (state 0 – red, or state 1 – green) and the time it has been in this state. To capture the traffic lights, we assume that both the sending functions and receiving functions depend on these states. More specifically, $LA_{(u,\#,u+i)}$, $i = 1, 2, 3$, adjusts the free-flows speed of traffic. In signal state $LS_{(u,\#,w)} = 0$, it is set to 0, but in state $LS_{(u,\#,w)} = 1$ the adjustment $LA_{(u,\#,u+i)}$ depends on the time the signal has been in this state. The traffic light does not immediately show green in state 0 but with a delay, so that $LA_{(u,\#,u+i)}$ is initially 0 and increases with time due to the acceleration of traffic until the maximal free-flow speed is reached. When turning right and driving straight ahead, the adjusted sending functions are

$$S_{(u,\#,u+1)}\left(\left(\rho_{(u',\#,w')}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#)}, LA_{(u,\#,u+1)}\right) = \min\left(s_{\#}^{\max}, LA_{(u,\#,u+1)}a\rho_{(u,\#,u+1)}\right),$$

$$S_{(u,\#,u+2)}\left(\left(\rho_{(u',\#,w')}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#)}, LA_{(u,\#,u+2)}\right) = \min\left(s_{\#}^{\max}, LA_{(u,\#,u+2)}a\rho_{(u,\#,u+2)}\right).$$

When turning left, the sending function may be decreased due to oncoming traffic. This

can be modeled by an exponential term for a parameter $\zeta > 0$, for example:

$$S_{(u,\#,u+3)}\left(\left(\rho_{(u',\#,w')}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(v)}, LA_{(u,\#,u+3)}\right) = \min\left(s_{\#}^{\max}, LA_{(u,\#,u+3)} a \rho_{(u,\#,u+3)} \cdot \exp\left(-\zeta\cdot\left(\rho_{(u+2,\#,u)}+\rho_{(u+2,\#,u+3)}\right)\right)\right).$$

The traffic node # of the signalized intersection also includes the areas in front of the traffic lights. Their capacity is limited by $\rho_{u,\#}^{\max}$, and this is reflected by the receiving functions:

$$R_{(u,v,w)}\left(\left(\rho_{(u',v,w')}\right)_{u'\in\mathscr{I}(v),w'\in\mathcal{O}(v)}\right) = \max\left(b\left(\rho_{u,\#}^{\max} - \sum_{w'\in\mathcal{O}(\#)}\rho_{(u,\#,w')}\right), 0\right).$$

5.2.4 Interaction Rules

The global optimization problem (5.2.4) captures perfect cooperation to achieve maximum (albeit myopic) traffic flow in the traffic system. This is unrealistic for models of real traffic, since individual participants optimize only their own utility, which may conflict with the goals of others. As explained in Tampère et al. (2011), further constraints – *interaction rules* – can mimic the local behavior of agents. We explicitly specify three formal approaches.

Demand Proportional Flows. Sending functions model the demand of road users for movement. An interaction rule could specify that realized flows are proportional to demand. To be more specific, we focus on the decoupled problems (5.2.5) and assume that there exists a constant $\lambda_{(u,v)}(t+1) \in [0,1]$, independent of x, such that

$$q_{(x,u,v)}^{\text{out}}(t+1) = \lambda_{(u,v)}(t+1) S_{(x,u,v),k}\left(\left(\rho_{(x',u,v')}(t)\right)_{x'\in\mathscr{I}(u),v'\in\mathcal{O}(u)}\right) \tag{5.2.6}$$

Maximizing flow in (5.2.5) is now considerably simplified and equivalent to solving for all $v \in V$ and $u \in \mathscr{I}(v)$ the problems $\operatorname{argmax}_{\lambda_{(u,v)}\in[0,1]} \lambda_{(u,v)}$ under the constraints given in Remark 5.2.1. These problems possess the explicit solutions

$$\lambda_{(u,v)}(t+1) = \min\left\{1, \min_{w\in\mathcal{O}(v)}\left\{\frac{R_{(u,v,w)}\left(\left(\rho_{(u',v,w')}(t)\right)_{u'\in\mathscr{I}(v),w'\in\mathcal{O}(v)}\right)}{\sum_{x\in\mathscr{I}(u)} f_{(x,u,v)\to w}(t+1)\cdot S_{(x,u,v)}\left(\left(\rho_{(x',u,v')}(t)\right)_{x'\in\mathscr{I}(u),v'\in\mathcal{O}(u)}\right)}\right\}\right\},$$

yielding the flows

$$q_{(x,u,v)}^{\text{out}}(t+1) = \min\left\{S_{(x,u,v)}\left(\left(\rho_{(x',u,v')}(t)\right)_{x'\in\mathscr{I}(u),v'\in\mathcal{O}(u)}\right), \min_{w\in\mathcal{O}(v)}\left\{\frac{R_{(u,v,w)}\left(\left(\rho_{(u',v,w')}(t)\right)_{u'\in\mathscr{I}(v),w'\in\mathcal{O}(v)}\right)}{\sum_{x\in\mathscr{I}(u)} f_{(x,u,v)\to w}(t+1)}\right\}\right\}.$$

Capacity Proportional Flows. Instead of assuming that realized flows are proportional to demand for the same factor, different directions could have different capacities that determine the proportionality factors. Letting $\sum_{x\in\mathscr{I}(u)} d_{(x,u,v)} = 1$ with $d_{(x,u,v)} \geq 0$, $x \in \mathscr{I}(u)$, one may assume that there exists a constant $\lambda_{(u,v)}(t+1) \in [0,1]$, independent of x, such that

$$q^{\text{out}}_{(x,u,v)}(t+1) = \min\left(\lambda_{(u,v)} d_{(x,u,v)}, 1\right) S_{(x,u,v)}\left((\rho_{(x',u,v')}(t))_{x'\in\mathscr{I}(u),v'\in\mathcal{O}(u)}\right). \tag{5.2.7}$$

The factor $\min\left(\lambda_{(u,v)} d_{(x,u,v)}, 1\right)$ ensures that realized flows cannot become larger than the sending flows. As in the case of demand proportional flows, this leads again to several one-dimensional optimization problems that can be easily solved explicitly:

$$\lambda_{(u,v)}(t+1) = \min_{w\in\mathcal{O}(v)} \left\{ \inf\left\{\lambda \geq 0 \colon \sum_{x\in\mathscr{I}(u)} f_{(x,u,v)}(t+1) \min\left(\lambda d_{(x,u,v)}, 1\right) \cdot \right.\right.$$
$$\left.\left. S_{(x,u,v)}\left((\rho_{(x',u,v')}(t))_{x'\in\mathscr{I}(u),v'\in\mathcal{O}(u)}\right) = R_{(u,v,w)}\left((\rho_{(u',v,w')}(t))_{u'\in\mathscr{I}(v),w'\in\mathcal{O}(v)}\right) \right\}\right\}.$$

The interior minimization is simply the solution of an equation in the single variable λ. The right hand side is continuous and increasing in λ. Due to its piecewise linearity, this problem can be solved by a finite number of iterations.

Priority Rules. Priority rules vary from country to country. A common example is that on an intersection traffic from the right often has priority. This can be implemented as an interaction rule in our model. Again, for $v \in V$ and $u \in \mathscr{I}(v)$, as in problem (5.2.5), we consider the decoupled problems of local optimization of the flow over the next time period. To capture priority rules, we assume that based on the current traffic state, a fixed enumeration of $\mathscr{I}(u) = \{x_{u,1}, \ldots, x_{u,I_u}\}$ is chosen. This order of incoming nodes is fixed for a certain period of time during which local traffic flows are computed hierarchically. Traffic originating from the nodes listed before the others has priority. The duration of the regime must be chosen according to the real situation being modeled and the real time to which each time period in the model corresponds. Formally, we sequentially solve for $i = 1, \ldots, I_u$ the problems argmax $q^{\text{out}}_{(x_{u,i},u,v)}(t+1)$ for the corresponding constraints[4] and obtain the solution

$$q^{\text{out}}_{(x_{u,1},u,v)}(t+1)$$
$$= \min\left(S_{(x_{u,1},u,v)}\left((\rho_{(x',u,v')}(t))_{x'\in\mathscr{I}(u),v'\in\mathcal{O}(u)}\right), \frac{R_{(u,v,w)}\left((\rho_{(u',v,w')}(t))_{u'\in\mathscr{I}(v),w'\in\mathcal{O}(v)}\right)}{f_{(x_{u,1},u,v)\to w}(t+1)}\right),$$
$$q^{\text{out}}_{(x_{u,i+1},u,v)}(t+1) = \min\left(S_{(x_{u,i+1},u,v)}\left((\rho_{(x',u,v')}(t))_{x'\in\mathscr{I}(u),v'\in\mathcal{O}(u)}\right),\right.$$
$$\left.\frac{R_{(u,v,w)}\left((\rho_{(u',v,w')}(t))_{u'\in\mathscr{I}(v),w'\in\mathcal{O}(v)}\right) - \sum_{j=1}^{i} f_{(x_{u,j},u,v)\to w}(t+1)\cdot q^{\text{out}}_{(x_{u,j},u,v)}(t+1)}{f_{(x_{u,i+1},u,v)\to w}(t+1)}\right),$$

[4] The constraints are $q^{\text{out}}_{(x_{u,i},u,v)}(t+1) \geq 0$, $q^{\text{in}}_{(u,v,w)}(t+1) = \sum_{j=1}^{i} f_{(x_{u,j},u,v)\to w}(t+1)\cdot q^{\text{out}}_{(x_{u,j},u,v)}(t+1)$, $q^{\text{in}}_{(u,v,w)}(t+1) \leq R_{(u,v,w)}\left((\rho_{(u',v,w')}(t))_{u'\in\mathscr{I}(v),w'\in\mathcal{O}(v)}\right)$ for all $w \in \mathcal{O}(v)$ and $q^{\text{out}}_{(x_{u,i},u,v)}(t+1) \leq S_{(x_{u,i},u,v)}\left((\rho_{(x',u,v')}(t))_{x'\in\mathscr{I}(u),v'\in\mathcal{O}(u)}\right)$.

for all $i = 1, \ldots, I_u - 1$.

5.3 Acceptable Configurations and Designs

5.3.1 The Question

We will be interested in evaluating the performance of traffic systems under various conditions. Specifically, the influence of measures to regulate traffic should be understood. To this end, we consider a collection of cell transmission models enumerated by a vector $k \in \mathbb{D} \subseteq \mathbb{R}^r$ for some dimension $r \in \mathbb{N}$. The set $\mathbb{D}$ is assumed to be bounded. The components of k specify the characteristics of a traffic system, such as total traffic volume, traffic control parameters, the magnitude of random variations of various variables, and weather conditions. For fixed $k \in \mathbb{D}$, the (possibly random) time evolution of the corresponding cell transmission model is described following the approach from Section 5.2. We will call k a *design parameter*. For each design parameter $k \in \mathbb{D}$, the traffic system can be simulated and the corresponding random variables of interest can be calculated. We assume that we wish to evaluate a random variable Q_k and compare the results across $k \in \mathbb{D}$. The random variable Q_k could model the total network traffic flow or traffic flow per traffic volume over a given time horizon, for example.

5.3.2 Preference Functionals and Acceptable Designs

By $\mathcal{X}$ we denote a normed space of random variables such as L^p, $p \in [1, \infty]$, and assume that $Q_k \in \mathcal{X}$, $k \in \mathbb{D}$. We evaluate the performance of the traffic system by a preference functional $U : \mathcal{X} \to \mathbb{R}$. Typically, U is increasing on $\mathcal{X}$, i.e., if $Q \leq Q'$ almost surely, then $U(Q) \leq U(Q')$. A special case is expected utility with $U(Q) = \mathbb{E}(u(Q))$ for an increasing function $u : \mathbb{R} \to \mathbb{R}$. In the case studies in Section 5.5, we will study

(i) **Expectation:** $u(x) = x$,

(ii) **Polynomial Utility:** $u(x) = -|x - c_p|^\alpha \mathbb{1}\{x \leq c_p\}, \quad c_p \in \mathbb{R},\ \alpha \geq 1$,

(iii) **Expectile Utility:** $u(x) = \alpha(x - c_e)_+ - (1 - \alpha)(x - c_e)_-, \quad c_e \in \mathbb{R},\ \alpha \leq 1/2$ where $x_+ = \max(x, 0)$ and $x_- = \max(-x, 0)$,

(iv) **Square Root Utility:** $u(x) = \sqrt{x}$.

For the practical evaluation of all traffic systems enumerated by design parameters $k \in \mathbb{D}$, categorization by performance is a convenient methodology. This is related to the level sets of the utility functionals. For a utility functional U and a fixed level $\gamma \in \mathbb{R}$, the set of *acceptable designs* is

$$\mathcal{D} = \mathcal{D}_{U,\gamma} = \{k \in \mathbb{D} \colon U(Q_k) \geq \gamma\}.$$

This characterizes the traffic systems with utility of at least γ. In applications, the level γ is often chosen as the utility of a benchmark distribution, i.e., $\gamma := U(Q)$ for a random variable Q with the benchmark distribution.

The acceptable designs are closely related to systemic risk measures as introduced in Feinstein, Rudloff & Weber (2017). In the special case that U and $k \mapsto Q_k$ are increasing and k parametrizes the design parameters in terms of incremental monetary costs, the systemic risk measure

$$R((Q_m)_{m\in\mathbb{R}^r}; k) = \{m \in \mathbb{D}\colon U(Q_{k+m}) \geq \gamma\} = \{m \in \mathbb{D}\colon k + m \in \mathcal{D}\} = \mathcal{D} - k$$

is the collection of vectors of additional investments required for the various features of the traffic system to achieve the acceptable design.

5.4 Learning the Acceptable Design

We are interested in characterizing acceptable designs $\mathcal{D} = \{k \in \mathbb{D}\colon \mathbb{E}(u(Q_k)) \geq \gamma\}$ of traffic systems. The challenge is that Q_k can only be simulated for finitely many $k \in \mathbb{D}$ and needs to be interpolated in between these points. The selection of points for the simulation is also an important issue. In this section, we propose a machine learning algorithm for the accelerated estimation of this set.

We approach the problem by approximating the function[5]

$$\mu\colon \mathbb{D} \to \mathbb{R}, \quad k \mapsto \mathbb{E}(u(Q_k)),$$

based on simulated data[6] $(k, \hat{\mu}_k)_{k\in\mathbb{D}}$. The simulated data are generated by Monte Carlo simulation of Q_k for selected points $k \in \mathbb{D}$. We use an iterative learning algorithm that successively selects finite sets $\mathbb{D}^i$ of points, $i = 1, 2, \ldots$. These sets form an increasing sequence $\mathbb{D}^0 \subseteq \mathbb{D}^1 \subseteq \mathbb{D}^2 \subseteq \ldots$, and new points $\mathbb{D}^i \setminus \mathbb{D}^{i-1}$ are strategically selected. The corresponding values $\mu(k)$ at $k \in \mathbb{D}^i \setminus \mathbb{D}^{i-1}$ are estimated with increasing accuracy. To extend the $(k, \hat{\mu}_k)_{k\in\mathbb{D}^i}$ to the entire design space $\mathbb{D}$, we use Gaussian process regression (GPR), a Bayesian inference method. The corresponding estimator of μ is denoted by $m^i\colon \mathbb{D} \to \mathbb{R}$. A key feature of GPR is that it not only produces an estimator of μ, but also captures the corresponding uncertainty.

The acceptable designs $\mathcal{D}$ are estimated by the plug-in estimators $\hat{\mathcal{D}}^i = \{k \in \mathbb{D}\colon m^i(k) \geq \gamma\}$, $i \in \mathbb{N}$. Gaussian process regression has been used previously for estimating level sets (Lyu, Binois & Ludkovski (2021)). The main innovation of this section is to develop a framework for active learning using sequential statistics in conjunction with GPR. For this purpose, we use a heteroscedastic version of Gaussian process regression.

[5] To control the approximation error, we assume that $\mathbb{D}$ is bounded. This implies that $\mathcal{D} \subseteq \mathbb{D}$ is bounded.

[6] We use the index notation for the data $\hat{\mu}_k$ at isolated points k and retain the notation $\mu(\cdot)$ when referring to a function defined on $\mathbb{D}$.

5.4.1 Monte Carlo Estimation of Function Values

We first describe the Monte Carlo estimation of the simulated data, i.e., the estimation of the function value $\mu(k) = \mathbb{E}(u(Q_k))$ given a fixed design parameter $k \in \mathbb{D}$ in a fixed iteration $i \in \mathbb{N}$. Denote by $(\tau^i)^2 > 0$ a selected *target variance* of estimates $(k, \hat{\mu}_k)_{k\in\mathbb{D}^i}$. Increasing precision is obtained by letting τ^i be decreasing for $i \geq 1$; however, τ^0 is associated to the initialization of the algorithms and therefore typically chosen smaller than τ^1. We combine a heuristic from the central limit theorem with sequential statistics to determine a stopping criterion.

If $\hat{Q}_k^1, \hat{Q}_k^2, \dots$ is a sequence of i.i.d. samples of Q_k, the central limit theorem implies that, for large $n \in \mathbb{N}$, the distribution of the sample mean $\hat{\mu}_k^n = \frac{1}{n}\sum_{j=1}^n u(\hat{Q}_k^j)$ is roughly $\mathcal{N}\left(\mu(k), \frac{(\hat{\sigma}_k^n)^2}{n}\right)$ where the correct variance is replaced by the sample variance $(\hat{\sigma}_k^n)^2$ of our estimates of expected utility.

We stop sampling when the sample noise $(\hat{\sigma}_k^n)^2/n$ falls below the desired target noise $(\tau^i)^2$. Additionally, we impose a minimal number of simulations $n_{\min}$ and a maximal number of simulations $n_{\max} \gg n_{\min}$ so that a known upper bound on computational resources can be guaranteed – but this does not have to be required. To be precise, we end our simulations of Q_k when

$$n = \min\left\{\min\left\{n_{\min} \leq \bar{n} \colon \frac{(\hat{\sigma}_k^{\bar{n}})^2}{\bar{n}} \leq (\tau^i)^2\right\}, n_{\max}\right\}.$$

Finally, we set $\hat{\mu}_k := \hat{\mu}_k^n$ and (with a slight abuse of notation) $\tau_k^2 := \frac{(\hat{\sigma}_k^n)^2}{n}$.

5.4.2 Gaussian Process Regression

We give a brief overview of Gaussian process regression[7], adapted to our framework. A *Gaussian process* defines a probability law over functions from $\mathbb{D}$ to $\mathbb{R}$ such that the finite-dimensional marginal distributions are normal. The distribution is fully characterized by its mean function $m \colon \mathbb{D} \to \mathbb{R}$ and covariance function (also called kernel) $c \colon \mathbb{D} \times \mathbb{D} \to [0, \infty)$. The corresponding law or process is denoted by $\mathcal{GP}(m, c)$. For any finite set[8] $\mathbb{D}^i \subseteq \mathbb{D}$, we know that $M(\mathbb{D}^i) \sim \mathcal{N}\left(m(\mathbb{D}^i), \Sigma(\mathbb{D}^i, \mathbb{D}^i)\right)$ with $M(\mathbb{D}^i) = (M(l))_{l\in\mathbb{D}^i}$, $m(\mathbb{D}^i) = (m(l))_{l\in\mathbb{D}^i}$, and $\Sigma(\mathbb{D}^i, \mathbb{D}^i) = (c(l, l'))_{l,l'\in\mathbb{D}^i}$. The kernel c is also associated to properties of the process such as the existence of regular versions.

Gaussian process regression is a probabilistic procedure for inference on an unknown function from possibly noisy data on the values of the function at some points. GPR is a Bayesian approach, i.e., the function is assumed to be an unknown sample from a prior distribution. The prior is taken to be the law of a Gaussian process. The posterior distribution is then calculated based on the data.

There are numerous approaches for choosing the prior. In this chapter, we will use the

[7] We refer to Rasmussen & Williams (2005) as a standard reference for more details

[8] Note that we retain the index i even though it is not technically required for this section.

standard choice $m \equiv 0$ as the prior mean and consider two commonly used covariance functions[9], namely the squared exponential function and the Matérn kernel. Both are defined in terms of hyperparameters $\sigma_c^2 > 0$ (referred to as the *signal variance*) and $l > 0$ (*characteristic length scale*). The squared exponential function is given by

$$c_{\mathrm{SE}}(k, k') = \sigma_c^2 \cdot \exp\left(-\frac{\|k - k'\|^2}{2l^2}\right), \quad k, k' \in \mathbb{D},$$

and, for $\nu > 0$, the Matérn kernel is defined as

$$c_{\mathrm{M}}(k, k'; \nu) = \sigma_c^2 \cdot \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\|k - k'\|}{l}\right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}\|k - k'\|}{l}\right), \quad k, k' \in \mathbb{D},$$

with K_ν being a modified Bessel function. The squared exponential function has mean square derivatives, yielding smooth behavior of the sample paths. The smoothness properties of the Matérn kernel are controlled by ν: Small choices lead to rougher sample paths, while, in the limit $\nu \to \infty$, the Matérn kernel equals the squared exponential function and can, thus, be understood as a generalization. For more details, we refer to Rasmussen & Williams (2005). In our case studies, we will consider the common choice[10] $\nu = 3/2$. The Matérn kernel offers advantages when higher curvature is required at some points to reduce the amplitude of oscillations in the approximation.

To estimate the unknown function from data, the following Bayesian approach will be applied: The unknown function μ is interpreted as a realization of a Gaussian process $M \sim \mathcal{GP}(m, c)$ with the given prior distribution. For any finite set $\mathbb{D}_* \subseteq \mathbb{D} \setminus \mathbb{D}^i$, M evaluated in $(\mathbb{D}^i, \mathbb{D}_*)$ follows a joint normal distribution. This still holds true, if we introduce independent noise $\varepsilon_k \sim \mathcal{N}(\mu(k), \tau_k^2)$ with $\tau_k \geq 0$ and set $\hat{M}_k = M(k) + \varepsilon_k$, $k \in \mathbb{D}^i$. The joint distribution with noise is

$$\begin{bmatrix} \hat{M}(\mathbb{D}^i) \\ M(\mathbb{D}_*) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\mathbb{D}^i) \\ m(\mathbb{D}_*) \end{bmatrix}, \begin{bmatrix} \Sigma(\mathbb{D}^i, \mathbb{D}^i) + \mathrm{diag}\left(\tau_1^2, \ldots, \tau_{|\mathbb{D}^i|}^2\right) & \Sigma(\mathbb{D}^i, \mathbb{D}_*) \\ \Sigma(\mathbb{D}_*, \mathbb{D}^i) & \Sigma(\mathbb{D}_*, \mathbb{D}_*) \end{bmatrix}\right) \tag{5.4.1}$$

with $\hat{M}(\mathbb{D}^i) = (\hat{M}_l)_{l \in \mathbb{D}^i} = (M(l) + \varepsilon_l)_{l \in \mathbb{D}^i}$, $M(\mathbb{D}_*) = (M(l))_{l \in \mathbb{D}_*}$ and $\Sigma(\mathbb{D}^i, \mathbb{D}^i) = (c(l, l'))_{l,l' \in \mathbb{D}^i}$, $\Sigma(\mathbb{D}^i, \mathbb{D}_*) = (c(l, l'))_{l \in \mathbb{D}^i, l' \in \mathbb{D}_*}$, $\Sigma(\mathbb{D}_*, \mathbb{D}^i) = \Sigma(\mathbb{D}^i, \mathbb{D}_*)^\top$, $\Sigma(\mathbb{D}_*, \mathbb{D}_*) = (c(l, l'))_{l,l' \in \mathbb{D}_*}$.

We assume that *noisy observations* are obtained on $\mathbb{D}^i$. Then a Bayesian update for the Gaussian process is computed. A Bayesian estimator of μ is given by the posterior mean function, and its uncertainty can be captured by the posterior variance. The *posterior distribution*, i.e., the distribution of $M \mid \hat{M}(\mathbb{D}^i) = \hat{\mu}(\mathbb{D}^i)$, is characterized by the following theorem.

Theorem 5.4.1 (Gaussian Process Regression). *Suppose that the Gaussian process*

[9] Rasmussen & Williams (2005) offer an extensive discussion on covariance functions.

[10] For $\nu = 3/2$, the Matérn kernel simplifies into $c_{\mathrm{M}}(k, k'; 3/2) = \left(1 + \frac{\sqrt{3}\|k-k'\|}{l}\right) \exp\left(-\frac{\sqrt{3}\|k-k'\|}{l}\right)$.

$M \sim \mathcal{GP}(m, c)$ has a mean function $m\colon \mathbb{D} \to \mathbb{R}$ and a covariance function $c\colon \mathbb{D} \times \mathbb{D} \to [0, \infty)$. We assume that $\hat{M}_k = M(k) + \varepsilon_k$, $k \in \mathbb{D}^i$, where the random variables $\varepsilon_k \sim \mathcal{N}(0, \tau_k^2)$, $k \in \mathbb{D}^i$, are jointly independent and also independent of M. For given noisy observations $\hat{\mu}(\mathbb{D}^i)$ of $\hat{M}$ on $\mathbb{D}^i$, the conditional law of the process is $(M \mid \hat{M}(\mathbb{D}^i) = \hat{\mu}(\mathbb{D}^i)) \sim \mathcal{GP}(m^i, c^i)$ where

$$m^i(k) = m(k) + \Sigma(\mathbb{D}^i, k)^\top \left(\Sigma(\mathbb{D}^i, \mathbb{D}^i) + \operatorname{diag}\left(\tau_1^2, \ldots, \tau_{|\mathbb{D}^i|}^2\right)\right)^{-1} (\hat{\mu}(\mathbb{D}^i) - m(\mathbb{D}^i)),\ k \in \mathbb{D},$$

$$c^i(k, k') = c(k, k') - \Sigma(\mathbb{D}^i, k)^\top \left(\Sigma(\mathbb{D}^i, \mathbb{D}^i) + \operatorname{diag}\left(\tau_1^2, \ldots, \tau_{|\mathbb{D}^i|}^2\right)\right)^{-1} \Sigma(\mathbb{D}^i, k'), \quad k, k' \in \mathbb{D}.$$

with $\hat{\mu}(\mathbb{D}^i) = (\hat{\mu}_l)_{l \in \mathbb{D}^i}$, $m(\mathbb{D}^i) = (m(l))_{l \in \mathbb{D}^i}$, $\Sigma(\mathbb{D}^i, k) = (c(l, k))_{l \in \mathbb{D}^i}$, and $\Sigma(\mathbb{D}^i, \mathbb{D}^i) = (c(l, l'))_{l, l' \in \mathbb{D}^i}$. The standard deviations are given by $\sigma^i(k) = \sqrt{c^i(k, k)}$.

Proof. See, for example, Goldberg, Williams & Bishop (1997). □

The previous theorem requires that the hyperparameters are determined in advance. The prior distribution of the Gaussian distribution and the sampling distribution of the noisy observation specify the distribution of the observed data given the hyperparameters. To select the hyperparameters θ, we maximize the *marginal likelihood*[11] of the observed data $\hat{\mu}(\mathbb{D}^i)$ using

$$\hat{M}(\mathbb{D}^i) \mid \theta \sim \mathcal{N}\left(m(\mathbb{D}^i), \Sigma(\mathbb{D}^i, \mathbb{D}^i) + \operatorname{diag}\left(\tau_1^2, \ldots, \tau_{|\mathbb{D}^i|}^2\right)\right),$$

cf. equation (5.4.1).

5.4.3 Active Learning Framework

We propose an algorithm that combines GPR and the successive acquisition of new sets of points $\mathbb{D}^0 \subseteq \mathbb{D}^1 \subseteq \mathbb{D}^2 \subseteq \ldots$. The goal is to approximate the set of acceptable designs $\mathcal{D}$. The general structure is described in Algorithm 2, which we will discuss in more detail in the following subsections. We would like to emphasize that each part of this algorithm is related to active learning approaches with stochastic kriging that have been used before (see, for example. Binois, Gramacy & Ludkovski (2018), Lyu, Binois & Ludkovski (2021), and Gotovos et al. (2013)); however, as far as we are aware, our particular combination of stochastic search and GPR is novel. Our estimation procedure consists of two major steps:

- **Phase 1: Initialize.** First, we create an initial set of points $\mathbb{D}^0$ and estimate values $\hat{\mu}_k$ with a target noise $(\tau^0)^2$ for each $k \in \mathbb{D}^0$. The hyperparameters of the

[11] Maximizing the marginal likelihood is also referred to as empirical Bayes, evidence approximation, or type-II maximum likelihood (cf. Rasmussen & Williams (2005) or Schulz, Speekenbrink & Krause (2018)). This is a non-convex problem and typical methods may converge to local maxima. However, numerical experiments indicate that prediction based on the squared exponential function is robust with respect to the estimation of its hyperparameters (see Chen & Wang (2018)). Alternatively, one could place a hyperprior on θ at the expense of tractability. We refer to Lalchand & Rasmussen (2020) for more details.

GPR are estimated by maximizing the marginal likelihood.[12] We then compute our first GPR[13] estimate $m^0\colon \mathbb{D} \to \mathbb{R}$ and $\sigma^0\colon \mathbb{D} \to [0,\infty)$ according to Theorem 5.4.1.

- **Phase 2: Loop.** Second, we repeat an active learning procedure to improve our GPR estimate. We use an acquisition function to randomly construct $\mathbb{D}^i \setminus \mathbb{D}^{i-1}$. Function values at the new arguments are estimated[14] and used to improve the estimates of the posterior mean and variance of GPR (see again Algorithm 6).

  The algorithm either terminates after a maximum number of iterations, or an upper bound for the approximation error can be specified to control the quality of the approximation.[15]

The set estimate $\hat{\mathcal{D}}^i$ is a function of the point estimate of the mean m^i. Additionally, the measure of uncertainty σ^i will be used to construct $\mathbb{D}^i \setminus \mathbb{D}^{i-1}$, i.e., selecting points to sample. It also provides information on the approximation errors associated with this procedure.

Selecting Points to Sample. The goal of selecting new points $\mathbb{D}^i \setminus \mathbb{D}^{i-1}$ after iteration $i-1$ is to improve the estimate of the superlevel set $\mathcal{D} = \{k \in \mathbb{D}\colon \mathbb{E}(u(Q_k)) \geq \gamma\}$. To examine $\mathbb{D}$, we first randomly select new points. Sampled points $k \in \mathbb{D}$ will only be used for further improvements in step i in the GPR, if for some constant $c_1 > 1$ the inequality $c_1 \cdot \tau^i < \sigma^{i-1}(k)$ holds, i.e., if Monte Carlo sampling can substantially decrease the uncertainty at that point. We use $c_1 = 5$ in our implementations of the algorithm.

Moreover, points that are close to the previously estimated boundary $m^{i-1}(k) = \gamma$ are preferred. This can be encoded by an *acquisition function* $\mathcal{I}^i\colon \mathbb{D} \to [0,\infty)$ that seeks to capture the informative potential of estimating $\mathbb{E}(u(Q_k))$ at a new design parameter $k \in \mathbb{D}$. We use[16] $\mathcal{I}^i(k)\colon = \Phi\left(-c_2^i \cdot |m^{i-1}(k) - \gamma|\right)$, $i = 1, 2, \ldots$, where Φ is the standard normal CDF and $c_2^i > 0$ is an increasing sequence. Up to its normalizing constant, we treat $\mathcal{I}^i$ as a density in order to simulate new points in $\mathbb{D}$.

In order to sample from this acquisition function, we employ rejection sampling (see, e.g., Glasserman (2003) for more details on the method) to sample from $\mathcal{I}^i$. We note that $\mathcal{I}^i(k) \leq 1/2$ for all $k \in \mathbb{D}$ by construction. Therefore by sampling points uniformly

[12] The detailed procedure is described in Algorithm 5.

[13] The details are provided in Algorithm 6.

[14] The GPR from each iteration yields a prior $\mathcal{N}(m^{i-1}(k), (\sigma^{i-1}(k))^2)$ which could be exploited in the subsequent iteration for sampling. This idea of Bayesian sampling is briefly described in Appendix 5.8.1.

[15] Such an upper bound is discussed in more detail in Section 5.4.4.

[16] Alternatively, one could use $\mathcal{I}^i(k)\colon = \Phi\left(-c_2 \cdot \frac{|m^{i-1}(k)-\gamma|}{\sigma^{i-1}(k)}\right)$, $i = 1, 2, \ldots$, with constant $c_2 > 0$. It is large when k is close to the estimated boundary and when the posterior uncertainty is large. This acquisition function is comparable to that used in Lyu, Binois & Ludkovski (2021); in that work, the acquisition function is optimized using a genetic algorithm to determine a single next point. Here, one would directly implement a stochastic sampling routine to generate $n_{\text{loop}} \geq 1$ new points to construct $\mathbb{D}^i \setminus \mathbb{D}^{i-1}$.

Algorithm 2 Active learning framework.

Phase 1: Initialize

(i) Sample an initial data set $\mathbb{D}^0$ of n_{initial} points uniformly in $\mathbb{D}$

(ii) Estimate values at points $k \in \mathbb{D}^0$ such that the variance is bounded by the target variance $(\tau^0)^2$

(iii) Estimate the hyperparameters for GPR (which will be fixed hereafter)

(iv) Compute the posterior mean and variances according to Theorem 5.4.1

Phase 2: Loop

(i) Build the acquisition function from the last posterior estimates. The acquisition encodes tradeoffs between the distance from the estimated boundary and the posterior uncertainty, when selecting additional points

(ii) Sample n_{loop} new points according to the acquisition function via rejection sampling. This set is $\mathbb{D}^i \setminus \mathbb{D}^{i-1}$

(iii) Simulate values at points $k \in \mathbb{D}^i \setminus \mathbb{D}^{i-1}$ such that the variance is bounded by the target variance $(\tau^i)^2$

(iv) Use all simulated values (i.e., the simulations for $\mathbb{D}^i$) to compute the new posterior mean and variance by Bayesian updating of the original prior according to Theorem 5.4.1

(v) Determine whether to stop

in $\mathbb{D}$ (i.e., with density $1/\text{vol}(\mathbb{D})$), we can accept the point k with probability $2\mathcal{I}^i(k)$ to recover samples from our acquisition function $\mathcal{I}^i$.[17] This algorithm is modified by first checking the inequality $c_1 \cdot \tau^i < \sigma^{i-1}(k)$. The procedure[18] is provided by Algorithm 3.

Algorithm 3 Rejection sampling.

```
for j = 1, 2, ..., n_loop do
  Set flag = true.
  while flag = true do
    Sample Û_j ~ Unif(𝔻).
    if c_1 · τ^i < σ^{i-1}(k) then
      Sample p ~ Unif(0, 1).
      if p < 2I^i(Û_j) then
        Set flag = false.
      end if
    end if
  end while
end for
```

5.4.4 Sandwich Principle and Bounds on the Approximation Error

In this section, we evaluate the approximation error of the estimate $\hat{\mathcal{D}}^i$ of the set $\mathcal{D}$. We construct inner and outer approximations $\hat{\mathcal{D}}^i_-$ and $\hat{\mathcal{D}}^i_+$ of $\mathcal{D}$ that *sandwich* the true set. We find these inner and outer approximations by constructing lower and upper approximations $m^i_-, m^i_+ \colon \mathbb{D} \to \mathbb{R}$ of $\mu \colon \mathbb{D} \to \mathbb{R}$, in the context of Gaussian process regression. We study approximation errors under the assumptions that are described in Theorem 5.4.1. The metric we use is given by the commonly studied Nikodym metric[19]

$$d_N\left(\hat{\mathcal{D}}^i, \mathcal{D}\right) = \text{vol}\left(\hat{\mathcal{D}}^i \Delta \mathcal{D}\right),$$

where $\hat{\mathcal{D}}^i \Delta \mathcal{D} = \left(\hat{\mathcal{D}}^i \setminus \mathcal{D}\right) \cup \left(\mathcal{D} \setminus \hat{\mathcal{D}}^i\right)$ is the symmetric difference between $\hat{\mathcal{D}}^i$ and $\mathcal{D}$, and the volume $\text{vol}(\cdot)$ refers to the l-dimensional Lebesgue measure. Given inner and outer approximations $\hat{\mathcal{D}}^i_-$ and $\hat{\mathcal{D}}^i_+$ that sandwich $\mathcal{D}$, we can upper bound the true error $d_N\left(\hat{\mathcal{D}}^i, \mathcal{D}\right)$ as follows.

[17] The upper bound on the likelihood ratio for the rejection sampling can be given by $\sup_{k \in \mathbb{D}} \mathcal{I}^i(k)/(1/\text{vol}(\mathbb{D})) \leq \text{vol}(\mathbb{D})/2$.

[18] The implementation of Algorithm 3 includes a maximal number of trials for the while loop. If no point is found that satisfies $c_1 \cdot \tau^i < \sigma^{i-1}(k)$, the algorithm continues with the next iteration $i+1$. Moreover, after the estimation of a function value, we check if its sample noise is small enough so that including the new data point is beneficial. For $c_3 > 0$ (we use $c_3 = 2$ in our simulations), we discard a new data point $(k, \hat{\mu}_k)$ if $\hat{\sigma}_k/\sqrt{n} \geq c_3 \cdot \tau^i$.

[19] There are other distance metrics between sets. We refer, for example, to Cuevas (2009) or Brunel (2018) for an overview. Another common metric is the Hausdorff metric, which has a more visual character. Here, the distance between two sets $A_1, A_2 \subseteq \mathbb{D}$ is defined by $d_H(A_1, A_2) = \inf\{\varepsilon > 0 \mid A_1 \subseteq \bigcup_{k \in A_2} B(k, \varepsilon)\ , A_2 \subseteq \bigcup_{k \in A_1} B(k, \varepsilon)\}$ where $B(k, \varepsilon)$ is the open ball of radius ε centered in k.

Lemma 5.4.2 (Sandwich Principle and Error Bound). *Let $\hat{\mathcal{D}}^i_-$, $\hat{\mathcal{D}}^i$, and $\hat{\mathcal{D}}^i_+$ be estimators of $\mathcal{D}$ such that $\hat{\mathcal{D}}^i_- \subseteq \hat{\mathcal{D}}^i \subseteq \hat{\mathcal{D}}^i_+$ and $P(\hat{\mathcal{D}}^i_- \subseteq \mathcal{D} \subseteq \hat{\mathcal{D}}^i_+) \geq 1-\delta$ for $\delta \in (0,1)$. Then it holds*

$$P\left(d_N\left(\hat{\mathcal{D}}, \mathcal{D}\right) \leq \operatorname{vol}\left(\hat{\mathcal{D}}^i_+ \setminus \hat{\mathcal{D}}^i_-\right)\right) \;\geq\; 1-\delta.$$

Proof. See Section 5.8.4.1. □

We consider two alternatives to define the lower and upper approximations $m^i_-, m^i_+ \colon \mathbb{D} \to \mathbb{R}$ and apply them in the sandwich principle Lemma 5.4.2. These choices are associated to uniform and pointwise bounds, respectively.[20]

Uniform Bounds. Functions $m^i_-, m^i_+ \colon \mathbb{D} \to \mathbb{R}$ are called *uniform bounds* for some $\delta \in (0,1)$, if

$$P\left(\forall\, k \in \mathbb{D} \colon m^i_-(k) \leq M(k) \leq m^i_+(k) \mid \hat{M}(\mathbb{D}^i) = \hat{\mu}(\mathbb{D}^i)\right) \;\geq\; 1-\delta. \tag{5.4.2}$$

The region in between is referred to as a *credible band* for the unknown function $\mu \colon \mathbb{D} \to \mathbb{R}$.

Lederer, Umlauft & Hirche (2019)[21] discuss the derivation of such uniform bounds for Gaussian process regression. They impose Lipschitz conditions on the true function $\mu \colon \mathbb{D} \to \mathbb{R}$ and the covariance function $c \colon \mathbb{D} \times \mathbb{D} \to \mathbb{R}$. The uniform bounds can then be constructed[22] by applying a bound on values at sampled points and bounding the values in between using the Lipschitz assumptions. While a Lipschitz constant on μ is often unknown, a Lipschitz condition is satisfied by commonly used covariance functions. As an immediate consequence, we can relate such uniform bounds to probabilistic bounds of the set $\mathcal{D}$, so that the sandwich principle can be applied. We set $\mathfrak{D} = \{k \in \mathbb{D} \colon M(k) \geq \gamma\}$.

Corollary 5.4.3 (Credible Band for the Acceptable Design and Error Bound). *Define the estimators $\hat{\mathcal{D}}^i = \{k \in \mathbb{D} \colon m^i(k) \geq \gamma\}$, $\hat{\mathcal{D}}^i_- = \{k \in \mathbb{D} \colon m^i_-(k) \geq \gamma\}$, and $\hat{\mathcal{D}}^i_+ = \{k \in \mathbb{D} \colon m^i_+(k) \geq \gamma\}$. If condition (5.4.2) is satisfied, then*

$$\begin{aligned} P(\hat{\mathcal{D}}^i_- \subseteq \mathfrak{D} \subseteq \hat{\mathcal{D}}^i_+ \mid \hat{M}(\mathbb{D}^i) = \hat{\mu}(\mathbb{D}^i)) &\geq 1-\delta, \\ P(d_N(\mathfrak{D}, \hat{\mathcal{D}}^i) \leq \operatorname{vol}(\hat{\mathcal{D}}^i_+ \Delta \hat{\mathcal{D}}^i_-) \mid \hat{M}(\mathbb{D}^i) = \hat{\mu}(\mathbb{D}^i)) &\geq 1-\delta \end{aligned}$$

where $\operatorname{vol}(\hat{\mathcal{D}}^i_+ \Delta \hat{\mathcal{D}}^i_-) = \operatorname{vol}(\hat{\mathcal{D}}^i_+ \setminus \hat{\mathcal{D}}^i_-) = \operatorname{vol}\left\{k \in \mathbb{D} \colon m^i_+(k) \geq \gamma > m^i_-(k)\right\}$.

Proof. See Section 5.8.4.2. □

[20] Appendix 5.8.2 discusses how to compute the proposed upper bounds. Appendix 5.8.3 presents additional ideas for robustifying the pointwise bound.

[21] See, in particular, Theorem 3.1 in Lederer, Umlauft & Hirche (2019). These results are extended in Lederer, Umlauft & Hirche (2021).

[22] For $s \in \mathbb{R}_+$, the bounds can be defined by

$$m^i_\pm(k) = m^i(k) \pm \sqrt{\alpha(s)}\sigma^i(k) \pm \beta(s), \quad k \in \mathbb{D},$$

where $\alpha(s) = 2\log\left(\frac{M(s,\mathbb{D})}{\delta}\right)$ and $\beta(s) = (L_{m^i} + L_\mu)s + \sqrt{\alpha(s)}\omega_{\sigma^i}(s)$ for Lipschitz constants $L_{m^i}, L_\mu \geq 0$, $\omega_{\sigma^i}(\cdot)$ being the modulus of continuity of σ^i, and $M(s,\mathbb{D})$ the s-convering number of $\mathbb{D}$. We refer to Lederer, Umlauft & Hirche (2019) for details.

Controlling the approximation error uniformly with high probability typically results in a large set $\hat{\mathcal{D}}^i_+ \setminus \hat{\mathcal{D}}^i_-$ used to locate the boundary of $\mathfrak{D}$, which is not desirable. Moreover, uniform bounds require more prior information on the true function $\mu\colon \mathbb{D} \to \mathbb{R}$, i.e., the Lipschitz constant. These problems do not occur with pointwise bounds.

Pointwise Bounds. In practice, it is often sufficient to assess the quality of a single candidate design k. This issue can be approached as follows.

Lemma 5.4.4 (Pointwise Approximation Error). *Let $\delta \in (0,1)$. It holds*

$$\forall\, k \in \mathbb{D}\colon P\left(|M(k) - m^i(k)| \leq \Phi^{-1}\left(1 - \frac{\delta}{2}\right)\sigma^i(k) \;\middle|\; \hat{M}(\mathbb{D}^i) = \hat{\mu}(\mathbb{D}^i)\right) \geq 1 - \delta$$

where Φ^{-1} denotes the inverse CDF of the standard normal distribution.

Proof. This is a standard argument, see Section 5.8.4.3 for a proof. □

This pointwise credible band can be attained with the functions $\bar{m}^i_-(k) := m^i(k) + \Phi^{-1}\left(\frac{\delta}{2}\right)\sigma^i(k)$ and $\bar{m}^i_+(k) := m^i(k) + \Phi^{-1}\left(1 - \frac{\delta}{2}\right)\sigma^i(k)$ for the GPR estimates m^i, σ^i, i.e.,

$$P\left(\bar{m}^i_-(k) \leq M(k) \leq \bar{m}^i_+(k) \mid \hat{M}(\mathbb{D}^i) = \hat{\mu}(\mathbb{D}^i)\right) \geq 1 - \delta \quad \forall\, k \in \mathbb{D}.$$

To set up a sandwich principle, we define the inner and outer approximations

$$\hat{\mathcal{D}}^i_\pm = \{k \in \mathbb{D}\colon \bar{m}^i_\pm(k) \geq \gamma\}$$

and evaluate the Nikodym metric $d_N\left(\hat{\mathcal{D}}^i_-, \hat{\mathcal{D}}^i_+\right)$. For $k \in \hat{\mathcal{D}}^i_+ \setminus \hat{\mathcal{D}}^i_-$, we have that $\gamma \in (\bar{m}^i_-(k), \bar{m}^i_+(k)]$, thus, due to Lemma 5.4.4 and the choice of $\bar{m}^i_-(k)$ and $\bar{m}^i_+(k)$,

$$P\left(|M(k) - \gamma| < 2 \cdot \Phi^{-1}\left(1 - \frac{\delta}{2}\right)\sigma^i(k) \;\middle|\; \hat{M}(\mathbb{D}^i) = \hat{\mu}(\mathbb{D}^i)\right) \geq 1 - \delta.$$

5.5 Case Studies

We study two traffic networks, one with two signalized intersections and another one with variable capacities of highways, speed limits and bottlenecks due to roundabouts. Appendix 5.9.1 provides a pseudocode for the implementation of our traffic model for general networks. We investigate acceptable designs based on demand proportional flows. For selected design parameters, we compare these with the cooperative driving benchmark model.

5.5.1 Urban Network

Traffic signals are essential control elements in modern urban networks. Their main function is to temporarily block certain traffic flows so that competing traffic flows can pass safely. Efficient placement and design are nontrivial problems; issues include the

choice of traffic flow to be interrupted and the duration of the interruption. Complex interdependencies arise in networks, for example, when there is more than one traffic light in a network.

5.5.1.1 Set-Up

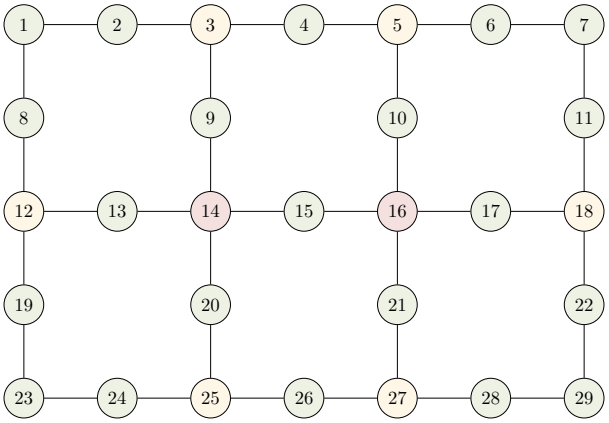


Figure 5.2: Signalized urban network.

Network. We consider a stylized urban network as depicted in Figure 5.2. It consists of 29 nodes $V = \{1, \dots, 29\}$, which are interconnected in a grid-like manner. The network features three types of nodes (cf. Section 5.2.3):

- *Signalized Intersection.* We consider two signalized intersections $\mathcal{R} = \{14, 16\}$, marked in red. Both are connected to four adjacent intersections and allow vehicles to travel horizontally or vertically through the network while blocking vehicles in orthogonal directions. We study the duration of their green time T^g and the displacement of the two green phases T^s of nodes 14 and 16 as important design parameters. Details on the implementation of the traffic lights are given in Section 5.9.2.1.

- *Unsignalized Intersection.* On the periphery are six unsignalized intersections $\mathcal{Y} = \{3, 5, 12, 15, 25, 27\}$, highlighted in yellow. Each of these intersections connects traffic flows from three adjacent nodes.

- *Bidirectional Road.* The remaining nodes of the network are simple bidirectional roads $\mathcal{G} = V \setminus (\mathcal{R} \cup \mathcal{Y})$ which are highlighted in green.

We assume that the signalized and unsignalized intersections have the same diameter of 40 m while the green connecting bidirectional roads are of length 120 m. Vehicles move at a free-flow speed of 50 km/h. We assume symmetric and constant turning fractions.[23]

[23] The turning fractions are $f_{(x,u,v)\to w}(t+1) = 1$ for $v \in \mathcal{G}$ $w \in \mathcal{O}(v)$, $w \neq u \in \mathcal{I}(v)$, $u \neq x \in \mathcal{I}(u)$ and, in analogy, $f_{(x,u,v)\to w}(t+1) = 1/2$ for $v \in \mathcal{Y}$ and $f_{(x,u,v)\to w}(t+1) = 1/3$ for $v \in \mathcal{R}$.

The connecting bidirectional roads $\mathcal{G}$ are of length 120 m. We normalize traffic densities, i.e., we let $l_v = 1$ for $v \in \mathcal{R} \cup \mathcal{Y}$ and set $l_v = 3$ for $v \in \mathcal{G}$. As an initial configuration, we set $\rho_{(\cdot,v,\cdot)}(0) = 5$ for $v \in \mathcal{G}$ and $\rho_{(\cdot,v,\cdot)}(0) = 1$ for $v \in \mathcal{Y} \cup \mathcal{R}$. At 50 km/h, intersections can be passed in 2.88 s, i.e., the interval $[t, t+1]$ corresponds to $t^{\text{real}} = 2.88$ s. We simulate $T = 1,250$ time steps corresponding to 1 h of traffic. The remaining parameters of the modules are listed in Table 5.1.

Table 5.1: Parameter choice.

	$s_v^{\max}$	$\rho_v^{\max}$	a_v	b_v	c_v	ζ_v
$v \in \mathcal{R}$	5	16	1	1	1	1/10
$v \in \mathcal{Y}$	5	10	1	1	1	1/10
$v \in \mathcal{G}$	5	30	1	1	1	-

Random Environment. We place the traffic network described above in a random environment by introducing random sources and sinks. Specifically, we implement two stochastic processes to model $q^{\text{net}}_{(6,7,11)}(t)$ and $q^{\text{net}}_{(24,23,19)}(t)$ for $t = 1, \dots, T$. As building blocks, we take two autoregressive models with a given dependence structure. The details are described in Section 5.9.2.2. The key idea is that both time series models include white noise $\varepsilon_{(6,7,11)}(t+1)$ and $\varepsilon_{(24,23,19)}(t+1)$. For each time step, these are normal random variables centered around 0 with standard deviations $\sigma_{(6,7,11)} \geq 0$ and $\sigma_{(24,23,19)} \geq 0$. We assume a particular dependence structure on $\varepsilon_{(6,7,11)}(t+1)$ and $\varepsilon_{(24,23,19)}(t+1)$ modeled by the Frank copula[24]

$$C^r(u_1, u_2) = -\frac{1}{r} \log\left(1 + \frac{(e^{-ru_1} - 1)(e^{-ru_2} - 1)}{e^{-r} - 1}\right), \quad u_1, u_2 \in (0, 1),$$

which is parametrized by $r \in \mathbb{R}$. The Frank copula interpolates from full countermonotonicity for $r \to -\infty$ (i.e., $\varepsilon_{(6,7,11)}(t+1)$ is large when $\varepsilon_{(24,23,19)}(t+1)$ is small and vice versa) to full comonotonicity for $r \to \infty$ (i.e., $\varepsilon_{(6,7,11)}(t+1)$ and $\varepsilon_{(24,23,19)}(t+1)$ move in the same direction). In the limit $r \to 0$, $\varepsilon_{(6,7,11)}(t+1)$ and $\varepsilon_{(24,23,19)}(t+1)$ are stochstically independent.

5.5.1.2 Acceptable Configurations and Design

Construction of Acceptance Sets. To assess the performance of traffic systems, we employ the normative approach of Section 5.3.2 and compare acceptable designs $\mathcal{D}_{u,\gamma} = \{k \in \mathbb{D} \colon \mathbb{E}(u(Q_k)) \geq \gamma\}$ for different utility functions u and levels γ. Specifically,

[24] A copula is a multivariate distribution function with uniform marginals. It captures dependence among the marginals of a random vector by virtue of Sklar's theorem. We refer to McNeil, Frey & Embrechts (2015) for more details.

we consider expectation, polynomial utility, expectile utility, and square root utility.[25]

The comparison across different utility functions is facilitated by calibrating the thresholds to benchmark flow distributions $\tilde{Q}^A$, $\tilde{Q}^B$, $\tilde{Q}^C$. For all utility functions, we choose the corresponding threshold levels[26] as $\gamma_u^A = \mathbb{E}(u(\tilde{Q}^A))$, $\gamma_u^B = \mathbb{E}(u(\tilde{Q}^B))$, and $\gamma_u^C = \mathbb{E}(u(\tilde{Q}^C))$.

Simulation Set-Up. We consider five design parameters that characterize traffic models, $k = \big(r, \sigma_{(6,7,11)}, \sigma_{(24,23,19)}, T^g, T^s\big)$:

- *Sources and Sinks.* We vary the dependence structure of the autoregressive models determining the source and sink flows, i.e., the dependence parameter r and the respective standard deviations of the noise terms $\sigma_{(6,7,11)}$ and $\sigma_{(24,23,19)}$.

- *Signal Control.* We vary the duration of the green phases of the two traffic lights T^g and the displacement of the green phases T^s.

Performance is characterized by average network traffic flow defined by

$$Q = Q_k = \frac{1}{T}\sum_{t=0}^{T-1}\sum_{v\in V}\sum_{u\in\mathscr{I}(v)}\sum_{x\in\mathscr{I}(u)} q^{\text{out}}_{(x,u,v)}(t+1)$$

for $k \in \mathbb{D} := [-50, 50] \times [0, 0.025] \times [0, 0.025] \times [0, 100] \times [0, 100]$. The network flow is simulated[27] and evaluated by our stochastic search algorithm.[28]

[25] While the square-root utility is a standard utility function (increasing and concave), polynomial utility and expectile utility, with appropriately chosen constants $c_p, c_e \in \mathbb{R}$, place special emphasis on downside risk. We set $c_p = 2\mathbb{E}(Q_{k^*})$ and $c_e = \mathbb{E}(Q_{k^*})$ for a "good" design parameter $k^* \in \mathbb{D}$. Polynomial utility evaluates only flows smaller than c_p, and expectile utility evaluates random fluctuations around c_e asymmetrically, with a stronger penalty for outcomes below this value.

[26] Specifically, let $X = X(\beta) \sim \text{Beta}(\beta, \beta)$ be a beta distribution with mean 1/2 and standard deviation $\sigma(X) = 1/\sqrt{8\beta + 4}$. We compute β^A, β^B and β^C such that $\sigma(X(\beta^A)) = 0.1$, $\sigma(X(\beta^B)) = 0.15$ and $\sigma(X(\beta^C)) = 0.2$. We obtain benchmark flow distributions $\tilde{Q}^A$, $\tilde{Q}^B$, $\tilde{Q}^C$ by setting $\tilde{Q}^A = e^A \cdot 2X(\beta^A)$, $\tilde{Q}^B = e^B \cdot 2X(\beta^B)$, and $\tilde{Q}^C = e^C \cdot 2X(\beta^C)$ for chosen benchmark expectations $e^A > e^B > e^C > 0$. Numerical evaluation yields the corresponding thresholds $\gamma^A = \mathbb{E}(u(\tilde{Q}^A))$, $\gamma^B = \mathbb{E}(u(\tilde{Q}^B))$, and $\gamma^C = \mathbb{E}(u(\tilde{Q}^C))$ for the different utility functions. Beta distributions were chosen as benchmarks because they are simple two-parametric distributions on compact intervals that generalize the uniform distribution; they are uniquely specified by their mean and variance.

[27] Since this is a discrete-time model, this property must be respected for each simulation run by T^g, T^s. This issue can be addressed by independently sampling discrete numbers for each simulation run from $\{\lfloor T^g \rfloor, \lceil T^g \rceil\}$ and $\{\lfloor T^s \rfloor, \lceil T^s \rceil\}$ such that their respective expected value is T^g or T^s. The implementation used for the case studies is based on an approximation where samples are taken uniformly from $\{\lfloor T^g \rfloor, \lceil T^g \rceil\}$ and $\{\lfloor T^s \rfloor, \lceil T^s \rceil\}$.

[28] For the level set estimation, we apply the following computational budget: We consider 8 iterations of our algorithm, where $n_{\text{initial}} = 150$ points are sampled uniformly in the initial phase and $n_{\text{loop}} = 50$ are sampled in the following 7 iterations according to the acquisition function. We define target noises as $\{5\%, 10\%, 8\%, 6\%, 5\%, 4\%, 3\%, 2\%\} \cdot (\gamma^C - \gamma^A)$ for the respective utility functions. We consider at least $n_{\min} = 20$ independent simulations and set $n_{\max} = 500, 150, 200, 300, 400, 650, 1200, 3000$. In our case studies, we find that a design parameter k^* for the first network is good if $\mathbb{E}(Q_{k^*}) = 60$; therefore, we set $e^A = 60$, $e^B = 55$, $e^C = 50$ and use these to calibrate γ^A, γ^B, and γ^C as described above.

5.5.1.3 Results

Impact of Sources and Sinks. Fixing $T^g = 10$, $T^s = 0$, we vary the parameters $r, \sigma_{(6,7,11)}, \sigma_{(24,23,19)}$ under the condition $\sigma_{(6,7,11)} = \sigma_{(24,23,19)}$. The impact of the dependence structure on performance is small. Increasing noise decreases the system performance considerably.[29]

Acceptable Traffic Lights Design. Second, we fix the parameters of the random environment as $r = 2.5$ and $\sigma_{(6,7,11)} = \sigma_{(24,23,19)} = 0.01$, and examine the acceptable designs of the two traffic lights. Figure 5.3 shows acceptable configurations of green time duration T^g and shift T^s. The resulting quantities are nontrivial:

$\mathbb{E}(u(Q))$, as a function of T^g and T^s, is concave in T^g – first increasing and then decreasing, since too short green times lead to low traffic flow and are unacceptable, and the same applies to values that are too high; moreover, it is periodic in T^s. The simulations show that for longer green times T^g (i.e., $T^g \geq 40$), acceptable designs can also be found on a diagonal in the (T^g, T^s) plane. The comparison across different utility functions indicates qualitatively similar behavior. However, different normative assessments of risk are reflected in the different sizes of the domains that are preferred over the three benchmark levels.

Comparison with the Cooperative Driving Benchmark Model. The preceding results are based on the demand proportional flows as an interaction rule. We investigate the benchmark model of cooperative driving as a theoretical alternative for selected design parameters.[30] Table 5.2 compares the expected traffic flows for four selected design parameters. We find that the myopically optimized cooperative driving model yields moderately but consistently higher traffic flows in these examples.

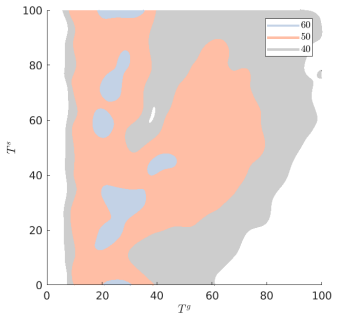
Table 5.2: Comparison with the benchmark model for cooperative driving.

	(T^g, T^s)			
	$(20, 75)$	$(20, 10)$	$(30, 30)$	$(30, 20)$
DPF	60.48	60.49	61.75	59.17
CDBM	65.78	64.71	64.29	64.28

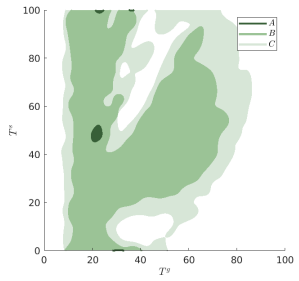
Approximation of $\mathbb{E}(Q)$ for demand proportional flows (DPF) and the cooperative driving benchmark model (CDBM) based on 500 independent simulations per traffic light configuration.

[29] This is shown in Figure 5.6 and Figure 5.7 in the appendix.

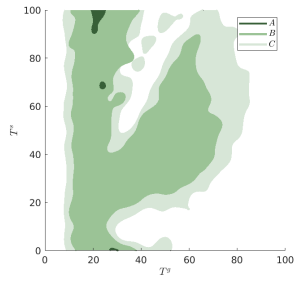
[30] The simulations are more complex because they have to solve a linear program for each node per time step. We use Matlab's built-in solver for these linear programs.



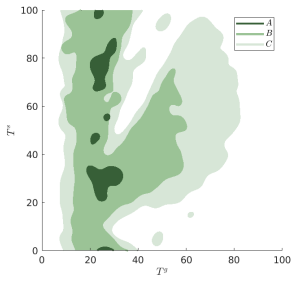
(a) $u(x) = x$



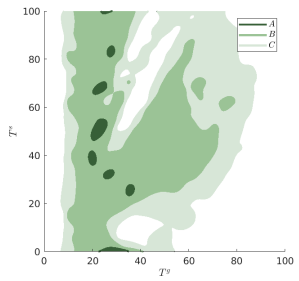
(b) $u(x) = 0.1(x-60)_+ - 0.9(x-60)_-$



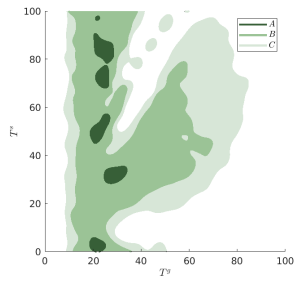
(c) $u(x) = 0.2(x-60)_+ - 0.8(x-60)_-$



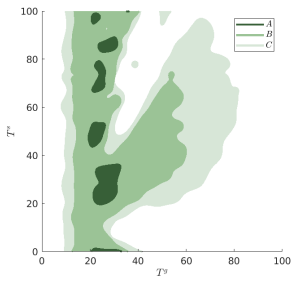
(d) $u(x) = \sqrt{x}$



(e) $u(x) = -|x-2\cdot 60|^3 \mathbb{1}\{x \leq 2\cdot 60\}$



(f) $u(x) = -|x-2\cdot 60|^2 \mathbb{1}\{x \leq 2\cdot 60\}$



(g) $u(x) = -|x - 2\cdot 60|^{3/2} \mathbb{1}\{x \leq 2\cdot 60\}$

Figure 5.3: Acceptable traffic lights configurations for $r = 2.5$ and $\sigma_{(6,7,11)} = \sigma_{(24,23,19)} = 0.01$. GPR is based on the Matérn kernel.

5.5.2 Highway Network

Highways are an important part of the road network as they allow efficient travel over longer distances. Vehicles can move at higher speeds, with the risk of accidents limited by the fact that the directions of travel are physically separated. Control mechanisms (such as different speed limits, optional driving on the shoulder, etc.) can be used to increase efficiency, depending on the traffic situation.

5.5.2.1 Set-Up

Network. We consider a stylized highway network as shown in Figure 5.1. The network consists of 33 nodes $V = \{1, \dots, 33\}$. Three roundabouts $\mathcal{Y} = \{1, 12, 23\}$ connect two types of highway modules. Those in the core $\mathcal{R} = \{7, \dots, 11, 18, \dots, 22, 29, \dots, 33\}$ allow optional driving on the shoulder, while those in the periphery $\mathcal{G} = V \setminus \{\mathcal{Y} \cup \mathcal{R}\}$ do not. When the shoulder is open to traffic, the maximum density and maximum traffic flow are increased while the maximum speed is reduced.

All highway modules $\mathcal{G} \cup \mathcal{R}$ have the same length of 3 km, while the roundabouts have a diameter of 200 m. We assume a free-flow speed on the highways of 100 km/h and an average speed of 40 km/h on the roundabout. Highway modules $v \in \mathcal{R}$ allow for an optional driving on the hard shoulder: In this case, maximal flow and density are increased by 50 %, while the free-flow speed is decreased to 80 km/h. We normalize traffic densities in terms of the roundabouts, i.e., we let $l_v = 1$ for $v \in \mathcal{Y}$ and set $l_v = 15$ for $v \in \mathcal{G} \cup \mathcal{R}$. Moreover, we specify symmetric and constant turning fractions, i.e., $f_{(x,u,v)\to w}(t+1) = 1/3$ for $v \in \mathcal{Y}$ and $w \in \mathcal{O}(v),\ w \neq u \in \mathcal{I}(v),\ u \neq x \in \mathcal{I}(u)$ and $f_{(x,u,v)\to w}(t+1) = 1$ for $v \in \mathcal{G} \cup \mathcal{R}$ in analogy.

At 100 km/h, a roundabout would be traversed in 7.2 s, i.e., the interval $[t, t+1]$ corresponds to $t^{\text{real}} = 7.2\,\text{s}$. Initially, the system is homogeneously filled with 5 % of the maximum density.[31] We simulate $T = 500$ time steps corresponding to 1 h of traffic. The remaining parameters of the modules are given in Table 5.3. In this section, (CLOSED) means that the shoulder is closed to traffic, while (OPEN) denotes the configuration in which it is open.

Random Sources and Sinks. We focus mainly on traffic flowing from the bottom to the top of the network. For this purpose, we consider sources and sinks at the periphery, so that the corresponding traffic has to cross two roundabouts. These can be considered as bottlenecks in the network. We mix this traffic with additional traffic in the core of

[31] To be precise, let $\rho^0 = \sum_{v\in V} \sum_{u\in\mathcal{I}(v),\ u\neq w\in\mathcal{O}(u)} \rho_{(u,v,w)}(0) / \sum_{v\in V} \rho_v^{\max} \in [0,1]$ denote the total initial density, as a fraction of the maximal density. For given $\rho^0 = 0.05$, we distribute the density homogeneously via

$$\rho_{(u,v,w)}(0) = \frac{\rho_v^{\max}}{\#\{(u,v,w)\colon u \in \mathcal{I}(v),\ u \neq w \in \mathcal{O}(u)\}} \cdot \rho^0.$$

Table 5.3: Parameter choice.

	$s_v^{\max}$	$\rho_v^{\max}$	a_v	b_v	c_v	d_v
$v \in \mathcal{G}$	20	100	1	1	1	1
$v \in \mathcal{Y}$	5	30	2/5	1	1	1
$v \in \mathcal{R}$ (CLOSED)	20	100	1	1	1	1
$v \in \mathcal{R}$ (OPEN)	30	150	4/5	1	1	1

the network that uses only the highways, which have a high capacity; however, heavy traffic can cause congestion. Opening the shoulder with a lower speed limit can alleviate this problem.

- *Periphery.* The sources at the bottom of the network are associated with auxiliary flows $q^{\text{aux}}_{(5,4,3)}(t+1) = q^{\text{aux}}_{(14,15,16)}(t+1)$, where $q^{\text{aux}}_{(5,4,3)}(t+1) \sim \mathcal{N}(\xi_1, \psi_1^2 \cdot \xi_1^2)$. At the top, deterministic sinks are defined via auxiliary flows $q^{\text{aux}}_{(27,26,25)}(t+1) = q^{\text{aux}}_{(25,26,27)}(t+1) = -2\xi_1$.

- *Core.* The additional traffic on the highways is specified by sources and sinks associated with the following auxiliary flows:

$$
\begin{aligned}
q^{\text{aux}}_{(18,19,20)}(t+1) &\sim \mathcal{N}(\xi_2, \psi_2^2 \cdot \xi_2^2), \quad q^{\text{aux}}_{(20,21,22)}(t+1) = -q^{\text{aux}}_{(18,19,20)}(t+1),\\
q^{\text{aux}}_{(33,32,31)}(t+1) &\sim \mathcal{N}(\xi_2, \psi_2^2 \cdot \xi_2^2), \quad q^{\text{aux}}_{(31,30,29)}(t+1) = -q^{\text{aux}}_{(33,32,31)}(t+1),\\
q^{\text{aux}}_{(7,8,9)}(t+1) &\sim \mathcal{N}(\xi_2, \psi_2^2 \cdot \xi_2^2), \quad q^{\text{aux}}_{(9,10,11)}(t+1) = -q^{\text{aux}}_{(7,8,9)}(t+1),\\
q^{\text{aux}}_{(11,10,9)}(t+1) &\sim \mathcal{N}(\xi_2, \psi_2^2 \cdot \xi_2^2), \quad q^{\text{aux}}_{(9,8,7)}(t+1) = -q^{\text{aux}}_{(11,10,9)}(t+1).
\end{aligned}
$$

All random variables are assumed to be independent. As in Case Study 1, $q^{\text{net}}_{(u,v,w)}(t+1)$ is equal to $q^{\text{aux}}_{(u,v,w)}(t+1)$, if this leads to $0 \leq \rho_{(u,v,w)}(t+1) \leq (\rho_v^{\max})/2$; otherwise, the absolute value of $q^{\text{aux}}_{(u,v,w)}(t+1)$ is reduced, such that one of these boundaries is attained. The quantity $q^{\text{aux}}_{(u,v,w)}(t+1)$ should be interpreted as the flow of vehicles that attempt to enter the network in the considered time period. In our simulations, we set the coefficient of variation to be $\psi_1 = \psi_2 = 0.1$ and vary ξ_1 and ξ_2.

5.5.2.2 Acceptable Configurations and Design

We study the impact of varying the design parameters ξ_1 and ξ_2, which control the volume and fluctuation of traffic originating from the periphery and the core, respectively. We compare two highway configurations: driving on shoulder prohibited (CLOSED) vs. driving on shoulder allowed (OPEN). We focus on the mean performance $\mathbb{E}(\cdot)$ and consider two different performance measures.

The set $\mathcal{N}$ contains all travel directions (u, v, w) that are sources or sinks. Letting

$$Q^a = \frac{\frac{1}{T}\sum_{t=0}^{T-1}\sum_{(u,v,w)\in\mathcal{N}}\left(q^{\text{net}}_{(u,v,w)}(t+1)\right)_-}{\frac{1}{T}\sum_{t=0}^{T-1}\sum_{v\in V}\sum_{(u,v,w)\in\mathcal{N}}\left(q^{\text{aux}}_{(u,v,w)}(t+1)\right)_+},$$

Q^a can be regarded as measure of the actual throughput: The sum of the flows that are actually removed from the network is divided by the sum of the available flows that attempt to enter the network.

Another performance measure is

$$Q^b = \frac{1}{T}\sum_{t=0}^{T-1}\left(\frac{q^{\text{out}}_{(12,18,19)}(t+1)}{\rho_{(12,18,19)}(t)} + \frac{q^{\text{out}}_{(1,33,32)}(t+1)}{\rho_{(1,33,32)}(t)}\right).$$

that measures average velocity on $(12, 18, 19)$ and $(1, 33, 32)$ by considering the fraction of flow that actually moves divided by the available density. We compute acceptable designs[32] based on the two performance measures Q^a and Q^b in the two regions[33] $\mathbb{D}^a = [1, 61] \times [1, 61]$ and $\mathbb{D}^b = [1, 31] \times [1, 31]$.

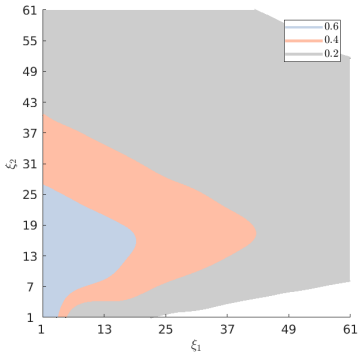
5.5.2.3 Results

We evaluate the two highway configurations (CLOSED) and (OPEN) based on the two performance measures Q^a and Q^b. To better compare the driving configurations, we also investigate the differences for (CLOSED) and (OPEN). The results are presented in Figure 5.4.

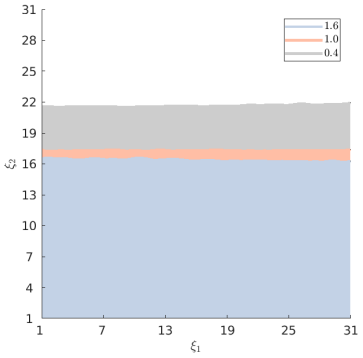
Figure 5.4a, Figure 5.4c, and Figure 5.4e show the set of acceptable designs based on Q^a. We can clearly distinguish the effects of ξ_1 (traffic generated in the periphery) and ξ_2 (traffic generated in the core). If we increase ξ_1 (traffic from the periphery) and keep ξ_2 constant, performance deteriorates because the former traffic needs to pass through roundabouts, which are bottlenecks. Increasing ξ_2 while ξ_1 is fixed initially increases the overall performance of the system. This is because the highway has enough capacity for core traffic; increasing ξ_2 increases the proportion of traffic that performs well. The statistic Q^a measures overall performance, and its expected value is therefore increasing. If ξ_2 becomes even larger, congestion will occur on the highway, again reducing performance. Opening the shoulder to traffic is advantageous when traffic density is higher, as can be seen in Figure 5.4e. However, the advantages are less pronounced when the system as a whole is too congested.

[32] For the level set estimation, we apply the following computational budget: We consider 8 iterations of our algorithm, where $n_{\text{initial}} = 100$ points are sampled uniformly in the initial phase and $n_{\text{loop}} = 50$ are sampled in the following 7 iterations according to the acquisition function. We define target noises as $\{5\%, 10\%, 8\%, 6\%, 5\%, 4\%, 3\%, 2\%\} \cdot 0.1$. We consider at least $n_{\min} = 20$ independent simulations and set $n_{\max} = 500, 150, 200, 300, 400, 650, 1200, 3000$.

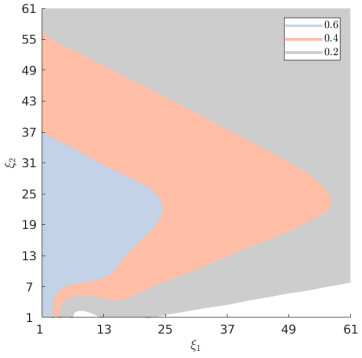
[33] We assume $\xi_1, \xi_2 \geq 1 > 0$ to exclude simulations with almost no traffic that might lead to small values in the denominators of the performance measures.



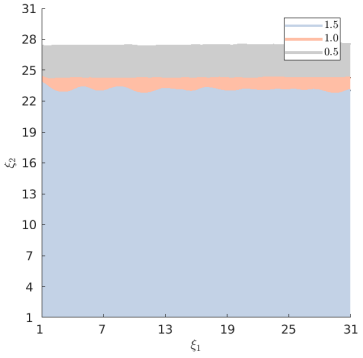
(a) Q^a: (CLOSED)



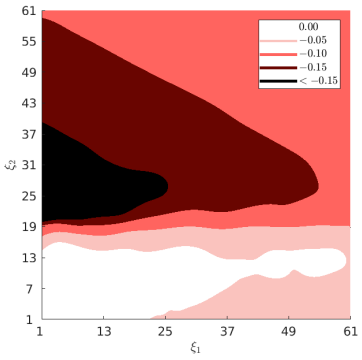
(b) Q^b: (CLOSED)



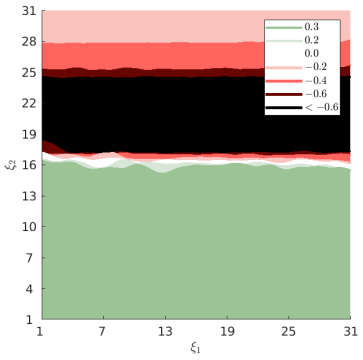
(c) Q^a: (OPEN)



(d) Q^b: (OPEN)



(e) Q^a: Difference



(f) Q^b: Difference

Figure 5.4: Acceptable designs and configurations.

Figure 5.4b, Figure 5.4d, and Figure 5.4f show the set of acceptable designs based on Q^b. Q^b essentially measures the speed of traffic originating from the periphery that has just entered the core area after passing through a roundabout. It can be clearly observed that the acceptable designs do not depend on ξ_1: The roundabouts serve as bottlenecks that control flow into the core area so that no additional congestion is caused by these traffic participants and therefore no reduction in speed. An increase in ξ_2, in contrast, leads to a decrease in speed; this is due to congestion at nodes 19, 20 and 32, 31. In low density regimes, traffic flows with constant free-flow speed. Once a critical density is reached, the speed decreases relatively quickly. Depending on ξ_2, there is a clear region where it is beneficial to open the shoulder to traffic (see Figure 5.4d). This effect is again less pronounced when the system is too congested.

5.6 Conclusion

In this work, we introduced a rigorous framework for stochastic cell transmission models for general traffic networks. The performance of traffic systems was evaluated based on preference functionals. The numerical implementation combined simulation, Gaussian process regression, and a stochastic exploration procedure. The approach was illustrated in two case studies that served as proofs of concept.

Future research should address the following tasks: a) Our flexible framework can be applied to many traffic systems. This requires careful calibration and validation at both the traffic cell level and the traffic system level. b) These models can then be used to answer specific questions in traffic planning. c) As shown in a simple example in the appendix, the setting can be extended to multiple interacting populations. This requires a closer look at model extensions. d) The algorithm combines stochastic search and Gaussian process regression. The latter could be replaced by other techniques, e.g., Bayesian neural networks (cf. Goan & Fookes (2020)), and the performance of different techniques should be compared. e) The normative criteria in this chapter were based on expected utility. Other preference functionals might be appropriate in the face of uncertainty, for example. Their implementation requires adapted estimation procedures.

5.7 Appendix: Further Examples of Cells

5.7.1 Roundabout

Unidirectional Roundabout. Consider an unidirectional roundabout # with four entries/exits enumerated counterclockwise and identified with $\mathbb{Z}_4 = \mathscr{I}(\#) = \mathcal{O}(\#)$ as shown in Figure 5.5. In right-hand traffic, vehicles travel counterclockwise through the roundabout. For simplicity, we assume that the roundabout is completely symmetric.

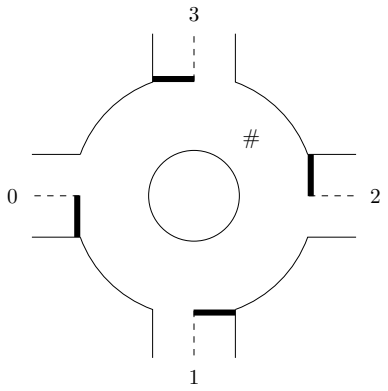


Figure 5.5: Symmetric roundabout.

One possibility is to assume that sending functions are as for highways, but receiving functions capture that different paths overlap in the roundabout. This leads to sending function of the following form:

$$S_{(u,\#,w)}\left(\left(\rho_{(u',\#,w')}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#)}\right) = \min\left(s_{\#}^{\max}, a\rho_{(u,\#,w)}\right), \quad w \neq u.$$

The receiving functions have a similar shape as for bidirectional linear interfaces, but the counterdensity of vehicles traveling in opposite direction is replaced by the densities on overlapping paths. These densities must be adjusted by a factor corresponding to the length of the overlap in the roundabout. This is due to the fact that the densities are normalized for each node and are proportional to the number of vehicles on each path within each node. We assume that for each path the vehicles are uniformly distributed over the segments of the path. Under these assumptions we obtain the following receiving functions:

$$\begin{aligned}
R_{(u,\#,u+1)}\left(\left(\rho_{(u',\#,w')}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#)}\right) &= \max\Bigg(b\Bigg(\frac{\rho_{\#}^{\max}}{4} - c\rho_{(u,\#,u+1)} - d\Bigg(\frac{1}{2}\rho_{(u,\#,u+2)} \\
&\quad + \frac{1}{3}\rho_{(u,\#,u+3)} + \frac{1}{3}\rho_{(u+2,\#,u+1)} + \frac{1}{2}\rho_{(u+3,\#,u+1)} + \frac{1}{3}\rho_{(u+3,\#,u+2)}\Bigg)\Bigg), 0\Bigg), \\
R_{(u,\#,u+2)}\left(\left(\rho_{(u',\#,w')}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#)}\right) &= \max\Bigg(b\Bigg(\frac{2\rho_{\#}^{\max}}{4} - c\rho_{(u,\#,u+2)} - d\Bigg(\rho_{(u,\#,u+1)} \\
&\quad + \frac{2}{3}\rho_{(u,\#,u+3)} + \rho_{(u+1,\#,u+2)} + \frac{1}{2}\rho_{(u+1,\#,u+3)} + \frac{1}{3}\rho_{(u+1,\#,u)} \\
&\quad + \frac{1}{3}\rho_{(u+2,\#,u+1)} + \frac{1}{2}\rho_{(u+3,\#,u+1)} + \frac{2}{3}\rho_{(u+3,\#,u+2)}\Bigg)\Bigg), 0\Bigg),
\end{aligned}$$

$$R_{(u,\#,u+3)}\left(\left(\rho_{(u',\#,w')}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#)}\right) = \max\Bigg(b\Bigg(\frac{3\rho^{\max}_{\#}}{4} - c\rho_{(u,\#,u+3)} - d\Bigg(\rho_{(u,\#,u+1)} + \rho_{(u,\#,u+2)} + \rho_{(u+1,\#,u+2)} + \rho_{(u+1,\#,u+3)} + \frac{2}{3}\rho_{(u+1,\#,u)} + \rho_{(u+2,\#,u+3)} + \frac{1}{2}\rho_{(u+2,\#,u)} + \frac{1}{3}\rho_{(u+2,\#,u+1)} + \frac{1}{2}\rho_{(u+3,\#,u+1)} + \frac{2}{3}\rho_{(u+3,\#,u+2)}\Bigg)\Bigg), 0\Bigg).$$

Bidirectional Roundabout. The roundabout model discussed above can be easily extended to bidirectional traffic flows of pedestrians in a bidirectional traffic area that has the form of a roundabout. The main difference is that the densities of traffic on overlapping paths for participants moving in the same direction and in the opposite direction must be considered in the receiving functions. In addition, pedestrians are assumed to choose the shortest path in the roundabout. If two paths have the same length, half of the pedestrians will use the first path and the other half will use the second path. These assumptions lead to the following formalization:

$$S_{(u,\#,w)}\left(\left(\rho_{(u',\#,w')}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#)}\right) = \min\left(s^{\max}_{\#}, a\rho_{(u,\#,w)}\right), \quad w \neq u,$$

$$R_{(u,\#,u+1)}\left(\left(\rho_{(u',\#,w')}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#)}\right) = \max\Bigg(b\Bigg(\frac{\rho^{\max}_{\#}}{4} - c\rho_{(u,\#,u+1)} - d\Bigg(\frac{1}{4}\rho_{(u,\#,u+2)} + \rho_{(u+1,\#,u)} + \frac{1}{4}\rho_{(u+1,\#,u+3)} + \frac{1}{4}\rho_{(u+2,\#,u)} + \frac{1}{4}\rho_{(u+3,\#,u+1)}\Bigg)\Bigg), 0\Bigg),$$

$$R_{(u,\#,u+2)}\left(\left(\rho_{(u',\#,w')}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#)}\right) = \max\Bigg(b\Bigg(\rho^{\max}_{\#} - c\rho_{(u,\#,u+2)} - d\sum_{\substack{u'\in\mathscr{I}(\#)\setminus\{u\},\\ w'\in\mathcal{O}(\#)\setminus\{u+2\}}}\rho_{(u',\#,w')}\Bigg), 0\Bigg),$$

$$R_{(u,\#,u+3)}\left(\left(\rho_{(u',\#,w')}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#)}\right) = \max\Bigg(b\Bigg(\frac{\rho^{\max}_{\#}}{4} - c\rho_{(u,\#,u+3)} - d\Bigg(\frac{1}{4}\rho_{(u,\#,u+2)} + \frac{1}{4}\rho_{(u+1,\#,u+3)} + \frac{1}{4}\rho_{(u+2,\#,u)} + \rho_{(u+3,\#,u)} + \frac{1}{4}\rho_{(u+3,\#,u+1)}\Bigg)\Bigg), 0\Bigg).$$

Roundabout with Vehicles and Pedestrians. The conceptual framework we develop can be generalized to multiple populations. In this chapter, for simplicity, we focus only on explaining generalized cell transmission models for traffic participants of one type. However, in the current example, we describe how an extension to more than one population is feasible.

Again, we focus on a node # with $\mathscr{I}(\#) = \mathcal{O}(\#) = \mathbb{Z}_4$ with vehicles ($k = 1$) moving as in the unidirectional roundabout. The cell transmission model can be implemented on different time scales. Here we assume that each time step corresponds to a relatively short real time span. Pedestrians ($k = 2$) move in both directions, but – according to their lower speed – only up to the next exit. Pedestrians have priority in the roundabout. The dynamics of the pedestrians is independent of the movement of the vehicles. Vehicles move as in the unidirectional roundabaout, but can be blocked by pedestrians who have priority. This canonically leads to the following formalization, where we introduce an

additional subscript for the type ($k = 1, 2$):

$$S_{(u,\#,w),1}\left(\left(\rho_{(u',\#,w'),k'}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#),k'=1,2}\right) =$$
$$\min\left(s^{\max}_{\#,1}, a_1\rho_{(u,\#,w),1}\right)\cdot\mathbb{1}\{\rho_{(w-1,\#,w),2}+\rho_{(w,\#,w-1),2}=0\},\ w\neq u$$
(pedestrians may block exit),

$$S_{(u,\#,w),2}\left(\left(\rho_{(u',\#,w'),k'}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#),k'=1,2}\right) =$$
$$\min\left\{s^{\max}_{\#,2}, a_2\rho_{(u,\#,w),2}\right\},\ w\in\{u-1,u+1\}\quad\text{(pedestrians move independently of vehicles)},$$

$$R_{(u,\#,u+1),1}\left(\left(\rho_{(u',\#,w'),k'}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#),k'=1,2}\right) =$$
$$\mathbb{1}\{\rho_{(u,\#,u+1),2}+\rho_{(u+1,\#,u),2}=0\}\cdot\max\left(b_1\left(\frac{\rho^{\max}_{\#,1}}{4}-c_1\rho_{(u,\#,u+1),1}-d_1\left(\frac{1}{2}\rho_{(u,\#,u+2),1}\right.\right.\right.$$
$$\left.\left.\left.+\frac{1}{3}\rho_{(u,\#,u+3),1}+\frac{1}{3}\rho_{(u+2,\#,u+1),1}+\frac{1}{2}\rho_{(u+3,\#,u+1),1}+\frac{1}{3}\rho_{(u+3,\#,u+2),1}\right)\right),0\right)$$
(pedestrians may block entrance),

$$R_{(u,\#,u+2),1}\left(\left(\rho_{(u',\#,w'),k'}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#),k'=1,2}\right) =$$
$$\mathbb{1}\{\rho_{(u,\#,u+1),2}+\rho_{(u+1,\#,u),2}=0\}\cdot\max\left(b_1\left(\frac{2\rho^{\max}_{\#,1}}{4}-c_1\rho_{(u,\#,u+2),1}-d_1\left(\rho_{(u,\#,u+1),1}\right.\right.\right.$$
$$+\frac{2}{3}\rho_{(u,\#,u+3),1}+\rho_{(u+1,\#,u+2),1}+\frac{1}{2}\rho_{(u+1,\#,u+3),1}+\frac{1}{3}\rho_{(u+1,\#,u),1}+\frac{1}{3}\rho_{(u+2,\#,u+1),1}$$
$$\left.\left.\left.+\frac{1}{2}\rho_{(u+3,\#,u+1),1}+\frac{2}{3}\rho_{(u+3,\#,u+2),1}\right)\right),0\right)\quad\text{(pedestrians may block entrance)},$$

$$R_{(u,\#,u+3),1}\left(\left(\rho_{(u',\#,w'),k'}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#),k'=1,2}\right) =$$
$$\mathbb{1}\{\rho_{(u,\#,u+1),2}+\rho_{(u+1,\#,u),2}=0\}\cdot\max\left(b_1\left(\frac{3\rho^{\max}_{\#,1}}{4}-c_1\rho_{(u,\#,u+3),1}-d_1\left(\rho_{(u,\#,u+1),1}\right.\right.\right.$$
$$+\rho_{(u,\#,u+2),1}+\rho_{(u+1,\#,u+2),1}+\rho_{(u+1,\#,u+3),1}+\frac{2}{3}\rho_{(u+1,\#,u),1}+\rho_{(u+2,\#,u+3),1}$$
$$\left.\left.\left.+\frac{1}{2}\rho_{(u+2,\#,u),1}+\frac{1}{3}\rho_{(u+2,\#,u+1),1}+\frac{1}{2}\rho_{(u+3,\#,u+1),1}+\frac{2}{3}\rho_{(u+3,\#,u+2),1}\right)\right),0\right)$$
(pedestrians may block entrance),

$$R_{(u,\#,w),2}\left(\left(\rho_{(u',\#,w'),k'}\right)_{u'\in\mathscr{I}(\#),w'\in\mathcal{O}(\#),k'=1,2}\right)=\max\left(b_2\left(\frac{\rho^{\max}_{\#,2}}{4}-c_2\rho_{(u,\#,w),2}-d_2\rho_{(w,\#,u),2}\right),0\right),$$
$$w\in\{u-1,u+1\},\quad\text{(pedestrians move independently of vehicles)}$$

Here, for types $k = 1, 2$, $\rho^{\max}_{\#,k} > 0$ is the maximum density, $s^{\max}_{\#,k} > 0$ is the maximum flow, $0 < a_k \leq 1$ is the free-flow speed, $0 < b_k \leq 1$ is the congestion wave speed, and $c_k, d_k > 0$ are interaction parameters.

5.8 Appendix: The Algorithm

5.8.1 A Bayesian Approach to Sampling

In most applications, generating the samples of the complex system Q_k is expensive. A Bayesian approach exploits the previous GPR (m^i, σ^i) in order to do variance reduction and, thus, reduce computational costs. Specifically, we propose utilizing the GPR (with its normal distribution) as a prior for the estimation of $\hat{\mu}_k$. With this structure, we

consider Bayesian inference for normal mean conditional on the variance (see, e.g., Hoff (2009, Section 5.2)) with a single sample taken at a time. Mathematically, we have the following structure:

- *Prior distribution*: $\mu(k) \sim \mathcal{N}(m^i(k), \sigma^i(k)^2)$,
- *Sampling distribution*: $\hat{\mu}_k^n \mid \mu(k) \sim \mathcal{N}\left(\mu(k), \frac{(\hat{\sigma}_k^n)^2}{n}\right)$ as a central limit theorem heuristic by prior assumptions, and
- *Posterior distribution*: $\mu(k) \mid \hat{\mu}_k^n \sim \mathcal{N}\left(t_{\text{post},n}, s^2_{\text{post},n}\right)$ where

$$t_{\text{post},n} = \frac{\frac{1}{(\sigma^i(k))^2} m^i(k) + \frac{n}{(\hat{\sigma}_k^n)^2}\hat{\mu}_k^n}{\frac{1}{(\sigma^i(k))^2} + \frac{n}{(\hat{\sigma}_k^n)^2}} \quad \text{and} \quad s^2_{\text{post},n} = \frac{1}{\frac{1}{(\sigma^i(k))^2} + \frac{n}{(\hat{\sigma}_k^n)^2}}.$$

Note that $s^2_{\text{post},n} \leq \frac{(\hat{\sigma}_k^n)^2}{n}$, i.e., the variance (and corresponding sample size) is reduced compared to the purely frequentist view described in Section 5.4.3. At the same time, precision may be reduced as t_{post} in general may not be an unbiased estimator of $\mathbb{E}(u(Q_k))$. The updated stopping criterion in the Bayesian approach, i.e., so that the sample variance drops below $(\tau^i)^2$, is given by

$$n = \min\left\{\min\left\{n_{\min} \leq \bar{n} \colon s^2_{\text{post},\bar{n}} \leq (\tau^i)^2\right\}, n_{\max}\right\}.$$

5.8.2 Computing the Error Bounds

We wish to return to our discussion of the error bounds with some remarks on its computation with Monte Carlo estimation. As shown in Section 5.4.4, the estimation error can be upper bounded by integrals of the form

$$\text{vol}\left\{k \in \mathbb{D} \colon m^i_+(k) \geq \gamma > m^i_-(k)\right\},$$

where $m^i_-, m^i_+ \colon \mathbb{D} \to \mathbb{R}$ are constructed as either uniform or pointwise error bounds.

The computation of these integrals is not trivial as the functions m^i_- and m^i_+ are typically not analytically accessible. Yet, values at specific positions $k \in \mathbb{D}$ can be evaluated. This provides a natural setting for approximation via Monte Carlo simulation:

- For a fixed budget $n_{\text{eval}} \in \mathbb{N}$, let $U_1, \ldots, U_{n_{\text{eval}}} \sim \text{Unif}(\mathbb{D})$.
- An approximation is given by

$$\begin{aligned}\text{vol}\left\{k \in \mathbb{D} \colon m^i_+(k) \geq \gamma > m^i_-(k)\right\} &= \int_{\mathbb{D}} \mathbb{1}\{m^i_+(k) \geq \gamma > m^i_-(k)\}\mathrm{d}k \\ &\approx \frac{\text{vol}(\mathbb{D})}{n_{\text{eval}}} \sum_{j=1}^{n_{\text{eval}}} \mathbb{1}\left\{m^i_+(U_j) \geq \gamma > m^i_-(U_j)\right\}\end{aligned}$$

- In order to eliminate random fluctuations in the comparison of the error bound for different iterations i, we fix a particular sequence[34] of samples $\hat{U}_1, \dots, \hat{U}_{n_{\text{eval}}}$.

The details of the evaluation procedure with the Monte Carlo approximation are given in Algorithm 4.

Algorithm 4 Evaluation procedure for the approximation error.

Input: $n_{\text{eval}} \in \mathbb{N}$, $m^i_+, m^i_- : \mathbb{D} \to \mathbb{R}$, samples $\hat{U}_1, \dots, \hat{U}_{n_{\text{eval}}} \in \mathbb{D}$.
Compute

$$\hat{e}^i = \frac{\text{vol}(\mathbb{D})}{n_{\text{eval}}} \sum_{j=1}^{n_{\text{eval}}} \mathbb{1}\left\{ m^i_+(\hat{U}_j) \geq \gamma > m^i_-(\hat{U}_j) \right\}.$$

Output: $\hat{e}^i$.

5.8.3 Robustification of the Pointwise Error Bound

The pointwise error bound can be extended *locally* if we impose a Lipschitz assumption on M. This can be understood as a robustification of the pointwise bounds. We present the following statement in analogy to the uniform error bounds by Lederer, Umlauft & Hirche (2019).

Proposition 5.8.1 (Local Credible Band). *Let $\delta \in (0,1)$ and $\varepsilon > 0$. For fixed $k^* \in \mathbb{D}$ with $B_\varepsilon(k^*) \subseteq \mathbb{D}$, let $L = L(k^*) > 0$ and assume that $|M(k^*) - M(k)| \leq L\|k^* - k\|$ P-a.s. for all $k \in B_\varepsilon(k^*)$. It holds*

$$P\left(\forall\, k \in B_\varepsilon(k^*) \colon |M(k) - m^i(k^*)| \leq \Phi^{-1}\left(1 - \frac{\delta}{2}\right) \sigma^i(k^*) + L\|k - k^*\| \mid \hat{M}(\mathbb{D}^i) = \hat{\mu}(\mathbb{D}^i) \right) \geq 1 - \delta.$$

Proof. Let $k^* \in \mathbb{D}$ be fixed. Applying the pointwise approximation error for k^*, we have

$$P\left(|M(k^*) - m^i(k^*)| \leq \Phi^{-1}\left(1 - \frac{\delta}{2}\right) \sigma^i(k^*) \mid \hat{M}(\mathbb{D}^i) = \hat{\mu}(\mathbb{D}^i) \right) \geq 1 - \delta.$$

For all $k \in \mathbb{D}$, triangular inequality and Lipschitz assumption imply

$$|M(k) - m^i(k^*)| \leq |M(k) - M(k^*)| + |M(k^*) - m^i(k^*)| \leq L\|k - k^*\| + |M(k^*) - m^i(k^*)|.$$

We conclude

$$P\left(\forall\, k \in B_\varepsilon(k^*) \colon |M(k) - m^i(k^*)| \leq \Phi^{-1}\left(1 - \frac{\delta}{2}\right) \sigma^i(k^*) + L\|k - k^*\| \mid \hat{M}(\mathbb{D}^i) = \hat{\mu}(\mathbb{D}^i) \right) \geq 1 - \delta.$$

[34] This is a natural application for Quasi-Monte Carlo methods in order to decrease the approximation error; here, we use the *Sobol sequence*. We refer to Glasserman (2003) for an overview on Quasi-Monte Carlo methods.

□

The local credible band gives rise to a local sandwich principle; we can upper bound the approximation error $\hat{\mathcal{D}}^i$ locally by intersecting it with $B_\varepsilon(k^*) := \{k \in \mathbb{D} \colon \|k - k^*\| < \varepsilon\}$.

Corollary 5.8.2 (Local Credible Band for the Acceptable Design and Error Bound). *In the setting of Proposition 5.8.1, let* $m^i_{\pm,k^*}(k) := m^i(k^*) \pm \Phi^{-1}\left(1 - \frac{\delta}{2}\right)\sigma^i(k^*) \pm L\|k - k^*\|$ *and define the estimators* $\hat{\mathcal{D}}^i = \{k \in \mathbb{D} \colon m^i(k) \geq \gamma\}$, $\hat{\mathcal{D}}^i_{-,k^*} = \{k \in \mathbb{D} \colon m^i_{-,k^*}(k) \geq \gamma\}$, *and* $\hat{\mathcal{D}}^i_{+,k^*} = \{k \in \mathbb{D} \colon m^i_{+,k^*}(k) \geq \gamma\}$. *Let* $\mathfrak{D} = \{k \in \mathbb{D} \colon M(k) \geq \gamma\}$ *be the corresponding prior for* $\mathcal{D}$. *Then, for all* $k^* \in \mathbb{D}$, *it holds*

$$P(\hat{\mathcal{D}}^i_{-,k^*} \cap B_\varepsilon(k^*) \subseteq \mathfrak{D} \cap B_\varepsilon(k^*) \subseteq \hat{\mathcal{D}}^i_{+,k^*} \cap B_\varepsilon(k^*) \mid \hat{M}(\mathbb{D}^i) = \hat{\mu}(\mathbb{D}^i)) \geq 1 - \delta$$

and

$$P\Bigg(d_N(\mathfrak{D} \cap B_\varepsilon(k^*), \hat{\mathcal{D}}^i \cap B_\varepsilon(k^*)) \leq d_N\left((\hat{\mathcal{D}}^i_{+,k^*} \cap B_\varepsilon(k^*),\ \hat{\mathcal{D}}^i_{-,k^*}) \cap B_\varepsilon(k^*)\right) \mid \hat{M}(\mathbb{D}^i) = \hat{\mu}(\mathbb{D}^i)\Bigg) \geq 1 - \delta.$$

Proof. Clear. □

The preceding statement tells us that widening the pointwise lower and upper approximations *locally* allows to probabilistically bound the approximation error of the acceptable design *locally*. In analogy to the uniform error bounds by Lederer, Umlauft & Hirche (2019), this requires an additional Lipschitz assumption with a (local) Lipschitz constant L which, in practice, is typically unknown.

5.8.4 Proofs

5.8.4.1 Proof of Lemma 5.4.2

We compute

$$\begin{aligned}\operatorname{vol}\left(\hat{\mathcal{D}}^i \Delta \mathcal{D}\right) &= \operatorname{vol}\left(\left(\hat{\mathcal{D}}^i \setminus \mathcal{D}\right) \cup \left(\mathcal{D} \setminus \hat{\mathcal{D}}^i\right)\right) = \operatorname{vol}\left(\hat{\mathcal{D}}^i \setminus \mathcal{D}\right) + \operatorname{vol}\left(\mathcal{D} \setminus \hat{\mathcal{D}}^i\right) \\ &\leq \operatorname{vol}\left(\hat{\mathcal{D}}^i \setminus \hat{\mathcal{D}}^i_-\right) + \operatorname{vol}\left(\hat{\mathcal{D}}^i_+ \setminus \hat{\mathcal{D}}^i\right) = \operatorname{vol}\left(\hat{\mathcal{D}}^i \setminus \hat{\mathcal{D}}^i_-\right) + \operatorname{vol}\left(\hat{\mathcal{D}}^i_+ \setminus \hat{\mathcal{D}}^i_-\right) - \operatorname{vol}\left(\hat{\mathcal{D}}^i \setminus \hat{\mathcal{D}}^i_-\right) \\ &= \operatorname{vol}\left(\hat{\mathcal{D}}^i_+ \setminus \hat{\mathcal{D}}^i_-\right).\end{aligned}$$

Depending on the type of inclusion, also the inequality is guaranteed P-a.s. or with probability greater than $1 - \delta$.

5.8.4.2 Proof of Corollary 5.4.3

It is clear that, for all $k \in \mathbb{D}$, $m^i_-(k) \leq m^i(k) \leq m^i_+(k)$. This implies the inclusion of the corresponding set estimators. More precisely, we have

$$\hat{\mathcal{D}}^i_- = \{k \in \mathbb{D} \colon m^i_-(k) \geq \gamma\} \subseteq \{k \in \mathbb{D} \colon m^i(k) \geq \gamma\} = \hat{\mathcal{D}}^i \subseteq \{k \in \mathbb{D} \colon m^i_+(k) \geq \gamma\} = \hat{\mathcal{D}}^i_+.$$

Correspondingly, $P\left(\forall\, k \in \mathbb{D}\colon m^i_-(k) \leq M(k) \leq m^i_+(k)\right) \geq 1-\delta$ implies

$$P(\hat{\mathcal{D}}^i_- \subseteq \mathfrak{D} \subseteq \hat{\mathcal{D}}^i_+) \geq 1-\delta.$$

Thus, Theorem 5.4.2 yields the claimed error bound $P(\mathrm{vol}(\mathfrak{D}\Delta\hat{\mathcal{D}}^i) \leq \mathrm{vol}(\hat{\mathcal{D}}^i_+\Delta\hat{\mathcal{D}}^i_-)) \geq 1-\delta$ and, due to the inclusion $\hat{\mathcal{D}}^i_- \subseteq \hat{\mathcal{D}}^i_+$, it follows

$$\begin{aligned}\mathrm{vol}(\hat{\mathcal{D}}^i_+\Delta\hat{\mathcal{D}}^i_-) &= \mathrm{vol}(\hat{\mathcal{D}}^i_+ \setminus \hat{\mathcal{D}}^i_-)\\ &= \mathrm{vol}\left(\{k \in \mathbb{D}\colon m^i_+(k) \geq \gamma\} \setminus \{k \in \mathbb{D}\colon m^i_-(k) < \gamma\}\right)\\ &= \mathrm{vol}\left\{k \in \mathbb{D}\colon m^i_+(k) \geq \gamma > m^i_-(k)\right\}.\end{aligned}$$

5.8.4.3 Proof of Lemma 5.4.4

The GPR based on observed data $\hat{\mu}(\mathbb{D}^i)$ yields

$$\forall\, k \in \mathbb{D}\colon M(k) \mid \hat{M}(\mathbb{D}^i) = \hat{\mu}(\mathbb{D}^i) \sim \mathcal{N}\left(m^i(k), (\sigma^i(k))^2\right).$$

The pointwise approximation error directly follows from standard confidence intervals for the mean of the normal distribution $\mathcal{N}\left(m^i(k), (\sigma^i(k))^2\right)$.

5.8.5 Algorithms

Algorithm 5 includes a pre-processing of the data: Subtracting the sample mean resembles the prior assumption $m \equiv 0$ (see also Schulz, Speekenbrink & Krause (2018)); the additional standardization of the data by its sample standard deviation serves to circumvent numerical issues. In our implementation, we use Matlab's built-in optimization routine. In case of numerical issues, we restart with a random initial point or reduce the number of points considered in the log likelihood.

Algorithm 5 Pre-processing and estimation of hyperparameters.

Input:

- Noisy data $(k, \hat{\mu}_k)_{k\in\mathbb{D}^0}$ such that $\hat{\mu}_k = \mu(k)+\varepsilon_k$ with $\varepsilon_k \sim \mathcal{N}(0, \tau_k^2)$ independent,
- prior mean $m \equiv 0$,
- prior covariance function $c\colon \mathbb{D} \times \mathbb{D} \to [0, \infty)$ depending on hyperparameters $\sigma_c, l > 0$.

Pre-Process Data: Let $\bar{\mu}^0 = 1/|\mathbb{D}^0| \sum_{k\in\mathbb{D}^0} \hat{\mu}_k$, $\bar{\varsigma}^0 = \sqrt{1/(|\mathbb{D}^0|-1) \sum_{k\in\mathbb{D}^0} (\hat{\mu}_k - \bar{\mu}^0)^2}$ and define $\hat{\nu}_k = (\hat{\mu}_k - \bar{\mu}^0)/\bar{\varsigma}^0$, $k \in \mathbb{D}^0$.
Model Selection: Determine $(\hat{\sigma}_c, \hat{l})$ by maximizing the (log) marginal likelihood

$$\ell(\hat{\nu}(\mathbb{D}^0); \sigma_c, l) = -\frac{1}{2}\hat{\nu}(\mathbb{D}^0)^\top \left(\Sigma(\mathbb{D}^0, \mathbb{D}^0) + \operatorname{diag}\left(\tau_1^2, \ldots, \tau_{|\mathbb{D}^0|}^2\right)\right)^{-1} \hat{\nu}(\mathbb{D}^0) - \frac{1}{2}\det\left(\Sigma(\mathbb{D}^0, \mathbb{D}^0) + \operatorname{diag}\left(\tau_1^2, \ldots, \tau_{|\mathbb{D}^0|}^2\right)\right) - \frac{|\mathbb{D}^0|}{2}\log(2\pi).$$

Output: $\bar{\mu}^0$, $\bar{\varsigma}^0$, $\hat{\sigma}_c$, $\hat{l}$.

Algorithm 6 Estimation procedure with Gaussian process regression.

Input:

- Noisy data $(k, \hat{\mu}_k)_{k\in\mathbb{D}^i}$ such that $\hat{\mu}_k = \mu(k)+\varepsilon_k$ with $\varepsilon_k \sim \mathcal{N}(0, \tau_k^2)$ independent,
- $\bar{\mu}^0$, $\bar{\varsigma}^0$, $\hat{\sigma}_c$, $\hat{l}$ from Algorithm 5

Transformation: Define $\hat{\nu}_k = (\hat{\mu}_k - \bar{\mu}^0)/\bar{\varsigma}^0$, $k \in \mathbb{D}^i$.
Bayesian Update: Based on $(\hat{\sigma}_c, \hat{l})$ and the data $(k, \hat{\nu}_k)_{k\in\mathbb{D}^i}$, compute $m_\nu^i\colon \mathbb{D} \to \mathbb{R}$ and $\sigma_\nu^i\colon \mathbb{D} \to [0, \infty)$ given by $\sigma_\nu^i(k) = \sqrt{c_\nu(k,k)}$ according to Theorem 5.4.1.
Retransformation: Define $m^i\colon \mathbb{D} \to \mathbb{R}$, $m^i(k) := m_\nu^i(k)\bar{\varsigma}^0 + \bar{\mu}^0$ and $\sigma^i\colon \mathbb{D} \to [0, \infty)$, $\sigma^i(k) := \sigma_\nu^i(k)\bar{\varsigma}^0$.
Output: $\hat{\mathcal{D}}^i = \{k \in \mathbb{D}\colon m^i(k) \geq \gamma\}$, m^i, σ^i.

5.9 Appendix: Supplement to the Case Studies

5.9.1 Traffic Simulation

The implementations of our traffic models adhere to the structure of the following pseudo-code.

Algorithm 7 Basic traffic simulation.

Input:

- Adjacency matrix: $A^E \in \{0,1\}^{|V|\times|V|}$ for a set of enumerated nodes $V = \{1,\dots,|V|\}$.
- Initial traffic configuration: $\rho_{(u,v,w)}(0) \geq 0$ for all $v \in V$, $u \in \mathscr{I}(v)$, $u \neq w \in \mathcal{O}(v)$
- Parameters (including terminal time: $T \in \mathbb{N}$)

for $t = 0,\dots,T-1$ **do**
  Phase 1: Compute sending and receiving constraints.
  for $v \in V$ **do**
    for $u \in \mathscr{I}(v)$ and $u \neq w \in \mathcal{O}(v)$ **do**
      Compute sending function $S_{(u,v,w)}\left((\rho_{(u',v,w'),k'}(t))_{u'\in\mathscr{I}(v),w'\in\mathcal{O}(v)}\right)$ and receiving function $R_{(u,v,w)}\left((\rho_{(u',v,w')}(t))_{u'\in\mathscr{I}(v),w'\in\mathcal{O}(v)}\right)$.
    end for
  end for
  Phase 2: Compute outflows.
  for $u \in V$ **do**
    for $x \in \mathscr{I}(u)$ and $x \neq v \in \mathcal{O}(u)$ **do**
      Compute $q^{\text{out}}_{(x,u,v)}(t+1)$.
    end for
  end for
  Phase 3: Compute inflows.
  for $v \in V$ **do**
    for $u \in \mathscr{I}(v)$ and $u \neq w \in \mathcal{O}(v)$ **do**
      Compute $q^{\text{in}}_{(u,v,w)}(t+1) = \sum_{x\in\mathscr{I}(u)} f_{(x,u,v)\to w}(t+1)\cdot q^{\text{out}}_{(x,u,v)}(t+1)$.
    end for
  end for
  Phase 4: Compute source/sink flows.
  for $v \in V$ **do**
    for $u \in \mathscr{I}(v)$ and $u \neq w \in \mathcal{O}(v)$ **do**
      Compute $q^{\text{net}}_{(u,v,w)}(t+1)$.
    end for
  end for
  Phase 5: Update densities.
  for $v \in V$ **do**
    for $u \in \mathscr{I}(v)$ and $u \neq w \in \mathcal{O}(v)$ **do**
      Compute $\rho_{(u,v,w)}(t+1) = \rho_{(u,v,w)}(t) + q^{\text{in}}_{(u,v,w)}(t+1) - q^{\text{out}}_{(u,v,w)}(t+1) + q^{\text{net}}_{(u,v,w)}(t+1)$.
    end for
  end for
end for

5.9.2 Urban Network

5.9.2.1 Traffic Light Implementation

Let $v \in \mathcal{R} = \{14, 16\}$. For any $u \in \mathcal{I}(v)$ and $u \neq w \in \mathcal{O}(v)$, let $LA_{(u,v,w)} \in [0,1]$ model the traffic light adjustment for traffic users with traveling direction (u, v, w) which is based on the respective traffic light signal $LS_{(u,v,w)} \in \{0, 1\}$.

In the following, we identify $\{13, 20, 15, 9\}$ and $\{15, 21, 17, 10\}$ with $\mathbb{Z}_4$ and set

$$
\begin{aligned}
S_{(u,v,u+1)}\left(\left(\rho_{(u',v,w')}\right)_{u'\in\mathcal{I}(v),w'\in\mathcal{O}(v)}, LA_{(u,v,u+1)}\right) &= \min\left\{s_v^{\max}, a_v LA_{(u,v,u+1)}\rho_{(u,v,u+1)}\right\},\\
S_{(u,v,u+2)}\left(\left(\rho_{(u',v,w')}\right)_{u'\in\mathcal{I}(v),w'\in\mathcal{O}(v)}, LA_{(u,v,u+2)}\right) &= \min\left\{s_v^{\max}, a_v LA_{(u,v,u+2)}\rho_{(u,v,u+2)}\right\},\\
S_{(u,v,u+3)}\left(\left(\rho_{(u',v,w')}\right)_{u'\in\mathcal{I}(v),w'\in\mathcal{O}(v)}, LA_{(u,v,u+3)}\right) &= \min\Bigg\{s_v^{\max}, a_v LA_{(u,v,u+3)}\rho_{(u,v,u+3)}\cdot\\
&\qquad \exp\left(-\zeta_v\left(\rho_{(u+2,v,u)} + \rho_{(u+2,v,u+3)}\right)\right)\Bigg\},\\
R_{(u,v,w)}\left(\left(\rho_{(u',v,w')}\right)_{u'\in\mathcal{I}(v),w'\in\mathcal{O}(v)}\right) &= \max\left(b_v\Big(\frac{\rho_v^{\max}}{4} - \sum_{w'\in\mathcal{O}(v)}\rho_{(u,v,w')}\Big), 0\right).
\end{aligned}
$$

We implement the signal policy as follows. Let $T^g \in \mathbb{N}$ be the duration of the green phase and $T^s \in \mathbb{N}$ the shift between the green times of the two traffic lights. Let $\mathcal{I}_{14} = \{13, 15\}$, $\mathcal{J}_{14} = \{9, 20\}$ and $\mathcal{I}_{16} = \{15, 17\}$, $\mathcal{J}_{16} = \{10, 21\}$. We set

$$
LS_{(u,14,w)}(t) = \begin{cases}
1, & t \mod 2T^g \in \{0, 1, \ldots, T^g - 1\},\ u \in \mathcal{I}_{14}, w \neq u,\\
0, & t \mod 2T^g \in \{0, 1, \ldots, T^g - 1\},\ u \in \mathcal{J}_{14}, w \neq u,\\
0, & t \mod 2T^g \in \{T^g, T^g + 1, \ldots, 2T^g - 1\},\ u \in \mathcal{I}_{14}, w \neq u,\\
1, & t \mod 2T^g \in \{T^g, T^g + 1, \ldots, 2T^g - 1\},\ u \in \mathcal{J}_{14}, w \neq u.
\end{cases}
$$

and

$$
LS_{(u,16,w)}(t) = \begin{cases}
1, & t + T^s \mod 2T^g \in \{0, 1, \ldots, T^g - 1\},\ u \in \mathcal{I}_{16}, w \neq u,\\
0, & t + T^s \mod 2T^g \in \{0, 1, \ldots, T^g - 1\},\ u \in \mathcal{J}_{16}, w \neq u,\\
0, & t + T^s \mod 2T^g \in \{T^g, T^g + 1, \ldots, 2T^g - 1\},\ u \in \mathcal{I}_{16}, w \neq u,\\
1, & t + T^s \mod 2T^g \in \{T^g, T^g + 1, \ldots, 2T^g - 1\},\ u \in \mathcal{J}_{16}, w \neq u.
\end{cases}
$$

We assume that vehicles accelerate comfortably with $a^{\text{real}} = 1.5\,\text{m/s}^2$ when a traffic lights switches from red to green. We introduce $t^{\text{safe}} = 2$ to model the acceleration delay caused by safety and reaction time and set

$$
LA_{(u,v,w)}(t) = \max\left\{0, \min\left\{1, \left(t^{\text{switch}}_{(u,v,w)}(t) - t^{\text{safe}}\right)\cdot t^{\text{real}}\cdot\frac{\text{a}^{\text{real}}}{v^{\text{real}}}\right\}\right\}\cdot LS_{(u,v,w)}(t)
$$

with the time intervals since the last switch being

$$t^{\text{switch}}_{(u,v,w)}(t) = \inf\{s \in \mathbb{N}\colon LS_{(u,v,w)}(t) \neq LS_{(u,v,w)}(t-s)\}.$$

5.9.2.2 Net Flows

Net flows are modelled as follows: For $(u,v,w) \in \{(6,7,11),(24,23,19)\}$, we define autoregressive models of order 1:

$$q^{\text{ar}}_{(u,v,w)}(t+1) = q^{\text{ar}}_{(u,v,w)}(t) + \varepsilon_{(u,v,w)}(t+1)$$

where $\varepsilon_{(u,v,w)}(t+1) \sim \mathcal{N}\left(0, \sigma_{(u,v,w)}\right)$, $\sigma^2_{(u,v,w)} \geq 0$, and innovations are stochastically independent across time. The initial value is $q^{\text{ar}}_{(u,v,w)}(0) = 0$. The dependence of $\varepsilon_{(6,7,11)}(t+1)$ and $\varepsilon_{(24,23,19)}(t+1)$ is governed by a Frank copula, parametrized by $r \in \mathbb{R} \setminus \{0\}$. In order to respect non-negativity constraints and maximal densities, we set

$$\begin{aligned} q^{\text{net}}_{(u,v,w)}(t+1) = \min\Bigg(& \max\left(q^{\text{ar}}_{(u,v,w)}(t+1),\ q^{\text{out}}_{(u,v,w)}(t+1) - q^{\text{in}}_{(u,v,w)}(t+1) - \rho_{(u,v,w)}(t)\right), \\ & \rho^{\max}_v + q^{\text{out}}_{(u,v,w)}(t+1) - q^{\text{in}}_{(u,v,w)}(t+1) - \rho_{(u,v,w)}(t)\Bigg). \end{aligned}$$

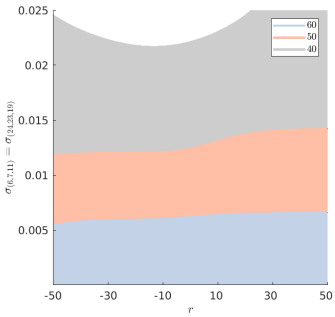
5.9.2.3 Dependence Parameter and Noise

To study the effects of the random environment, we set $T^g = 10$ and $T^s = 0$ and run simulations[35] in the three-dimensional subset $\left\{\left(r, \sigma_{(6,7,11)}, \sigma_{(24,23,19)}, 10, 0\right) \in \mathbb{D}\right\}$. Figure 5.6 shows acceptable designs in terms of dependence structure and noise (where we set $\sigma_{(6,7,11)} = \sigma_{(24,23,19)}$) for the considered utility functions. The impact of the dependence parameter r is small. In Figure 5.7, we set $r = 0$, corresponding to independent noise at the sources. The system performance decreases with increasing noise present in the system.

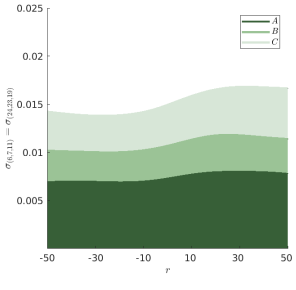
5.9.2.4 Comparison of Squared Exponential and Matérn Kernel

Figure 5.8 compares the squared exponential and Matérn kernel in the situation of Figure 5.3. The Matérn kernel can more flexibly adapt to functions that require higher curvature at some points. The surface plots have smaller amplitudes of the fluctuations than the squared exponential kernel.

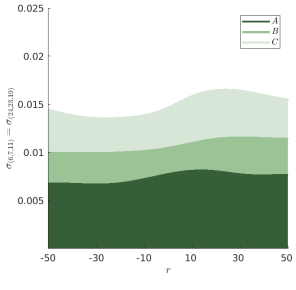
[35] GRP is based on the Matérn kernel. We also compared this to the squared exponential kernel which leads to almost the same results.



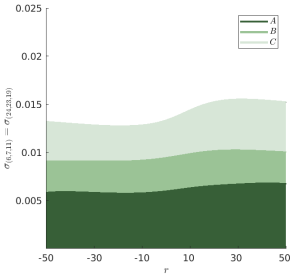
(a) $u(x) = x$



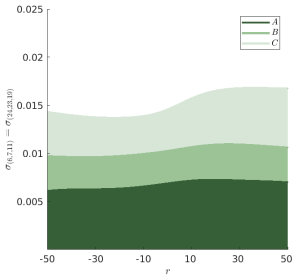
(b) $u(x) = 0.1(x - 60)_+ - 0.9(x - 60)_-$



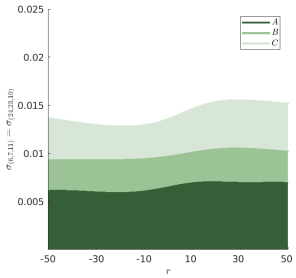
(c) $u(x) = 0.2(x - 60)_+ - 0.8(x - 60)_-$



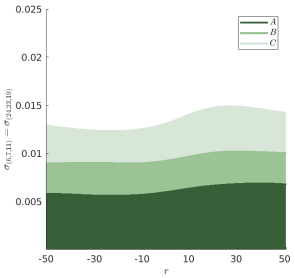
(d) $u(x) = \sqrt{x}$



(e) $u(x) = -|x-2\cdot 60|^3 \mathbb{1}\{x \leq 2 \cdot 60\}$

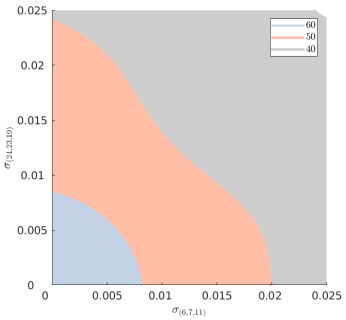


(f) $u(x) = -|x-2\cdot 60|^2 \mathbb{1}\{x \leq 2 \cdot 60\}$

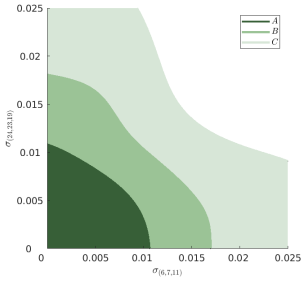


(g) $u(x) = -|x - 2 \cdot 60|^{3/2} \mathbb{1}\{x \leq 2 \cdot 60\}$

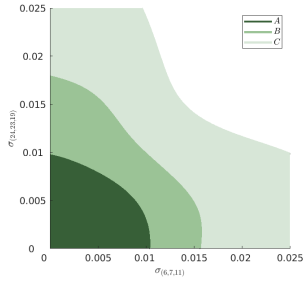
Figure 5.6: Acceptable dependence and noise. GRP is based on the Matérn kernel.



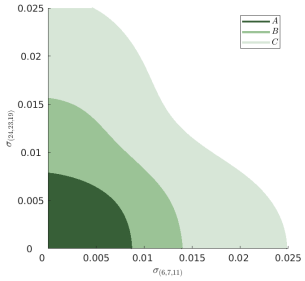
(a) $u(x) = x$



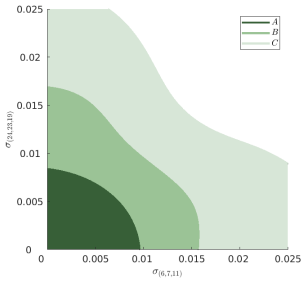
(b) $u(x) = 0.1(x-60)_+ - 0.9(x-60)_-$



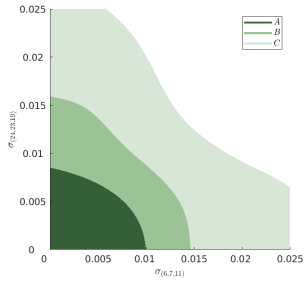
(c) $u(x) = 0.2(x-60)_+ - 0.8(x-60)_-$



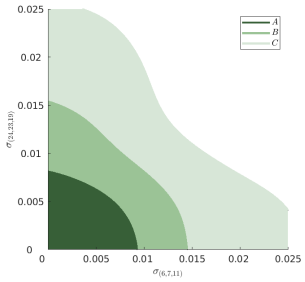
(d) $u(x) = \sqrt{x}$



(e) $u(x) = -|x-2\cdot 60|^3 \mathbb{1}\{x \leq 2\cdot 60\}$

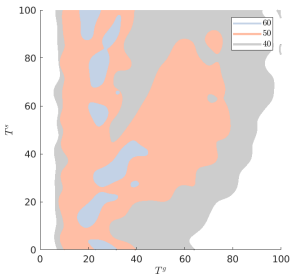


(f) $u(x) = -|x-2\cdot 60|^2 \mathbb{1}\{x \leq 2\cdot 60\}$

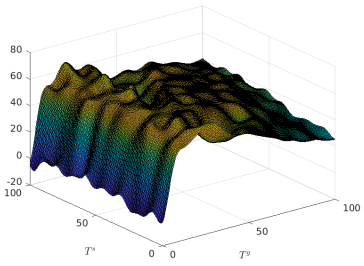


(g) $u(x) = -|x - 2\cdot 60|^{3/2} \mathbb{1}\{x \leq 2\cdot 60\}$

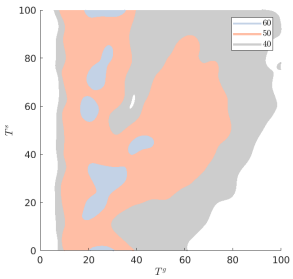
Figure 5.7: Acceptable noise for $r = 0$. GPR is based on the Matérn kernel.



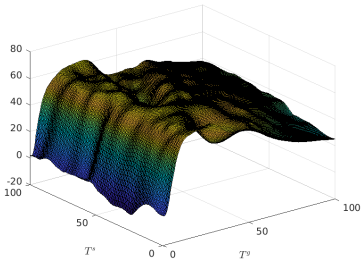
(a) Level sets: Squared exponential kernel



(b) Surface plot: Squared exponential kernel ($\gamma = 60$)



(c) Level sets: Matérn kernel



(d) Surface plot: Matérn kernel ($\gamma = 60$)

Figure 5.8: Comparison of kernels.

A Microscopic Traffic Simulations with the MODIS Framework

This chapter is based on the working paper Timmermann, Kleiber & Weber (2023).

A.1 Introduction

The design of safe and efficient traffic areas is one of the core tasks of a traffic planner. These conflicting goals can be resolved through attentive and appropriate traffic behavior. It can be directly enforced or indirectly brought about:

- *Direct Measures.* Traffic regulations such as maximal speeds or minimal safety distances explicitly control individual driving behavior in order to increase safety. Technological innovations can further improve the driving behavior of vehicles.
- *Indirect Measures.* Proactive placements of street elements such as road pavement design, incorporation of natural barriers, etc. allow traffic to flow in a controlled manner.

A comprehensive view is particularly relevant for traffic areas in which different traffic modes such as cars, bikes, and pedestrians coexist, i.e., where vulnerable road users meet motorized traffic. Here, traffic modeling enables the investigation of relevant scenarios in a controlled simulation environment.

MODIS – the acronym of **M**ulti m**OD**al **I**ntersection **S**imulation – is a microscopic traffic simulator developed to address these questions. MODIS refers to a traffic modeling framework and accompanying implementation in which road users move through a given traffic space based on minimal assumptions. The simulator is capable of representing small to medium road sections, with its focus on intersections. The motivating example for MODIS is the classical Shared Space: An intersection which is crossed by different traffic modes without a priori designated lanes. This chapter introduces the main features of MODIS and explains how to access it and perform simulations, focusing on the implementation aspects.

The Modeling Framework. MODIS is a 2D microscopic traffic model that can represent the motion of traffic participants in a bounded two-dimensional space given their entry point, entry time, destination, and their respective mode. Conceptually, the movement is governed by a system of coupled ordinary differential equations which are solved

in discrete time steps. The modeling approach is based on the Social Force Model by Helbing & Molnar (1995). For each traffic participant a resultant acceleration is computed as a function of present and past observations of positions and velocities of all visible participants. In the next time step, velocity and position are updated by integrating the acceleration.

A key difference from most competing microscopic traffic simulators is that the 2D traffic model MODIS does not require pre-defined lanes[1]. Traffic participants navigate from their current position towards their destination, performing evasive maneuvers to avoid collisions. This is achieved by computing appropriate acceleration vectors in a complex procedure. In the style of an agent-based model, the following actions are performed to calculate a resulting acceleration for a time step:

- *Free-Flow Trajectory.* Each traffic participant computes a desired free-flow trajectory. On this free-flow trajectory they would move from their current position to their destination if they do not encounter conflicting road users.
- *Perception.* Before moving, each traffic participant perceives their surroundings and estimates the positions and velocities of all visible road users.
- *Trajectory Extrapolation.* From current and past perceived positions and velocities, each traffic participant extrapolates the trajectories of the other visible traffic participants into the future.
- *Conflict Detection.* Each traffic participant compares their desired trajectory with the previously extrapolated trajectories. A conflict is detected when the predicted minimum future distance to another road user falls below a certain threshold.
- *Conflict Classification.* The conflict situation is classified based on the circumstances such as the number of conflicting road users, their traffic modes, the time to the first conflict, etc.
- *Decision for Strategy.* A behavioral model is implemented that distinguishes offensive and defensive strategies to resolve the detected conflicts. At this point, a strategy is decided which results in subsequent reactions.
- *Conflict Reaction.* Depending on the situation and chosen strategy, a conflict reaction is applied. This relies on either updating the planned trajectory by rerouting through a safe point or applying a long range reaction force so that a collision is avoided (e.g., slowing down or accelerating).

Up to this point, the motion of a vehicle is deterministic. This can be generalized by including stochastic components such as, e.g., random perception models, random effects, and random environments. The inclusion of randomness broadens the modeling

[1] Note that lane-based driving behavior can also be represented.

perspective and allows for more diverse conclusions about traffic systems in terms of safety, efficiency, and risk.

The Implementation. The multi-layered modeling approach requires an accompanying implementation for simulations. The current implementation offers three options for performing traffic simulations:

- *Graphical User Interface.* A graphical user interface is provided that allows to create deterministic traffic scenarios and visualize the movement of road users as they are simulated.
- *Command Line Interface.* Alternatively, the same deterministic simulations can also be executed with a command line interface. This saves the computational overhead for the visualization.
- *Monte Carlo Simulations.* While the first two approaches are useful for deterministic simulations, MODIS is also prepared for Monte Carlo simulations, i.e., repeated simulations when random components are included in the traffic system under consideration. Monte Carlo simulations are also controlled by a command line interface.

A.1.1 Outline

This chapter describes the main features of the traffic simulator MODIS with a focus on its implementation. The first section concludes with details on how the code can be accessed and a brief literature review of applications of MODIS. Section A.2 is intended for users who are generally interested in performing microscopic simulations with MODIS. Here, the various input files, parameters, and required file formats are described. These concepts are of general relevance whether the goal is to perform simulations with or without graphical output on the local computer or extensive Monte Carlo simulations on an HPC cluster. Section A.3 describes the implementation in more detail. The goal of this section is to enable the reader to actively work with the code so that they may perform own modifications or extensions of the framework. Section A.4 contains a guided exercise for making first changes to the framework. Appendix A.5 collects longer code examples that are referred to in the previous sections.

A.1.2 Starting Point

The microscopic traffic simulator MODIS is publicly available at:

`https://gitlab.com/LUH-HOI/modis`

The project landing page displays a `README.md` file that contains instructions on how to obtain precompiled executables and run the code. The software itself is implemented

in Java. It is accompanied by code documentation in HTML format generated with Javadoc. This can be found at:

`https://luh-hoi.gitlab.io/modis/`

A.1.3 History of MODIS & Applications

Conceptually, MODIS extends the Social Force Model; its code base originates from the simulation of pedestrian dynamics (see Höcker (2010)). In the initial phase, the multi-modal traffic modeling approach and accompanying implementation was developed jointly between the Institute of Transportation and Urban Engineering at TU Braunschweig and the Institute for Risk and Reliability at Leibniz Universität Hannover. From 2013 to 2022, it was financially supported by the German Research Foundation[2]. As of 2017, it was further expanded in collaboration with the House of Insurance at Leibniz Universität Hannover. MODIS is a collaborative project that many people have worked on, with Chris Timmermann being the lead developer. To date, it has resulted in several publications and dissertations.

The basic concept of the research project is described in Pascucci et al. (2015). The traffic modeling approach involves many reaction parameters; Schiermeyer et al. (2016) demonstrate a calibration of these using a genetic algorithm for exemplary conflict scenarios. Rinke et al. (2017) discuss the different reaction mechanisms for cyclists and pedestrians in more detail. While previously only conflict situations between exactly two road users were considered, Schiermeyer et al. (2019) extend the modeling approach for multiple simultaneous conflicts. Schiermeyer et al. (2017) provide an overall overview towards the end of the first phase of the DFG project. As a further modeling layer, Pascucci et al. (2018) develop a decision model for conflict situations that can be resolved by different strategies. The decision model is statistically calibrated using real data. Trifunović et al. (2021) demonstrate the research progress of the second phase of the DFG project: A key focus lies on more detailed and automated collection of real data, which enables a quantitative analysis of longer observation periods. The results of the DFG project are also collected in the dissertations by Pascucci (2020) and Timmermann (2022).

In a second stream of applications, the MODIS framework is extended for Monte Carlo simulations to include randomness as an additional modeling paradigm. For this, a sampling of random variables and parallel execution of the previously deterministic simulations are implemented. The focus of the applications lies on characterizing the trade-off between safety and efficiency of traffic systems. Berkhahn et al. (2018) suggest random misperception as a cause of traffic accidents. To model this, stochastic processes are included in car-following models. The approach is evaluated in case studies on simple traffic scenarios (one-lane roads and t-junctions). Berkhahn et al. (2022) extend this idea

[2]See DFG project 248905318.

for the more complex scenario of an unsignalized intersection and discuss state-of-the-art numerical schemes for the involved random ordinary differential equations.

A.2 Performing Simulations

This section explains how to run simulations using the provided pre-compiled executables. This includes single simulation runs with or without a GUI as well as Monte Carlo simulations (i.e., repeated independent simulations). At the same time, the section introduces important concepts and terms together with the explanation of the corresponding control elements of the GUI.

Three different executable Java-`jar`-Files are provided in the MODIS Gitlab repository:

- `ModisView.jar`: For running simulations using the GUI (see Section A.2.2),
- `ModisCore.jar`: For running a single simulation run without visualization (see Section A.2.3),
- `ModisMonteCarlo.jar`: For Monte Carlo simulations (see Section A.2.4).

The commands presented in this section are appropriate for these pre-compiled jar files. However, the syntax for specifying required or optional execution parameters is the same when the code is run from a particular IDE (e.g., when the project is imported using the provided Gradle build files). Since the exact procedure in this case depends on the IDE, it is not discussed further in this section.

A.2.1 Before Getting Started

Regardless of whether the simulation is to be started with or without a GUI, some prerequisites must be met. These prerequisites and terms are explained in the following sections.

A.2.1.1 Scenario

The term "scenario" refers to the stationary parts of the simulation. This includes the different parts of a traffic area with their surfaces and the intended use of the different traffic modes. It also includes elements such as road markings, traffic lights (if applicable), vegetation, and other obstacles. As a rule of thumb, the scenario includes everything that is still in place after all simulated road users have left the simulated area.

From a technical point of view, there are three ways to specify a scenario in MODIS:

- It can be hard coded by extending the class `ModisScenario` (see Section A.3.1.1; for technical details refer to the Javadoc).

- It can be loaded from a CityGML[3] file.
- It can be loaded as an OpenDRIVE[4] file.

It should be noted that in the latter two cases some features provided by the specific file formats are ignored if they are not relevant for the simulation. On the other hand, some features of the Modis framework can only be used if the scenario is defined as an extension of `ModisScenario`, e.g., preference rates for traffic areas (see Section A.3.2).

When working with the GUI, a scenario can be loaded using the corresponding control elements (see Section A.2.2). When working with the CLI, the scenario must be specified in the startup arguments (see Section A.2.3).

Example Scenario. A simple scenario, used as an example in several parts of this chapter, is shown in Figure A.1.

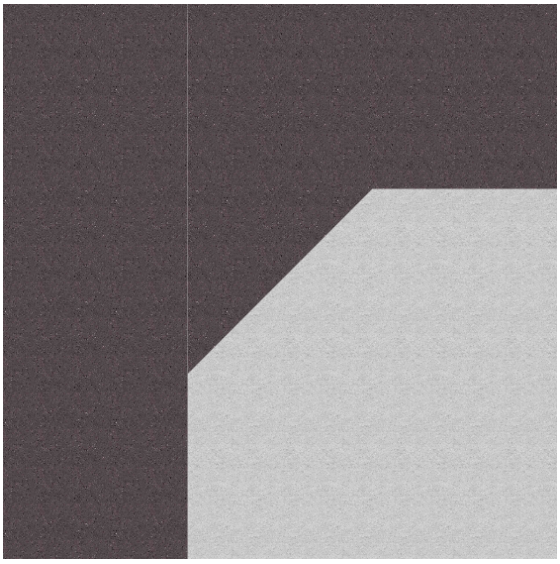


Figure A.1: Example scenario `DocumentationDemoScenario`.

This demo scenario is intended to resemble a simplified right-turn on an intersection. The dark gray areas represent a paved roadway and the light gray area marks a sidewalk. The exact specification of these properties in a `ModisScenario` is explained in Section A.3.1.1.

A.2.1.2 Simulation Properties

In order to simulate a specific traffic situation, more specifications are required than the static scenario provides. These include, for example, the initial positions of the road users, their individual destinations, optionally pre-defined free-flow trajectories or, if applicable, parameters for the dynamic generation of road users (see also Section

[3] See `www.ogc.org/standards/citygml` for more information on this file format.
[4] See `www.asam.net/standards/detail/opendrive/` for more information on this file format.

A.3.2.1). To store this kind of data, an xml file named Simulation Properties is used in the Modis framework. A complete simulation properties xml can be found in Appendix A.5.1.

For illustration purposes, the following Listing A.1 shows an excerpt from this file, specifying the origin and destination of a pedestrian.

Listing A.1: Specifying a pedestrian in a Simulation Properties File.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <properties schemaVersion="1.4">
3   <infrastructureID>DocumentationDemoScenario</infrastructureID>
4   ...
5   <odLocations>
6     <odLocation id="0">
7       <polyline>
8         <point>
9           <x>10.112</x>
10          <y>4.621</y>
11        </point>
12      </polyline>
13    </odLocation>
14  </odLocations>
15  <users>
16    <user id="0" mode="Pedestrian">
17      <position>
18        <x>7.165</x>
19        <y>12.442</y>
20      </position>
21      ...
22      <dest_id>0</dest_id>
23    </user>
24  </users>
25 </properties>
```

Line 3 specifies that this simulation properties file is for use with the Documentation-DemoScenario. Lines 6–13 define an "odLocation" with ID 0 that serves as the destination in this example. It consists of a single point with a given x- and y-coordinate. Lines 16–22 describe a pedestrian with ID, starting position, and destination, which is referenced by its ID[5].

Figure A.2 shows the example scenario presented in the previous section after loading

[5]It should be noted that IDs need to be unique only within an entity, i.e., both a road user and an odLocation are allowed to have the ID 0, while, e.g., a pedestrian and a car with the same ID would raise an error.

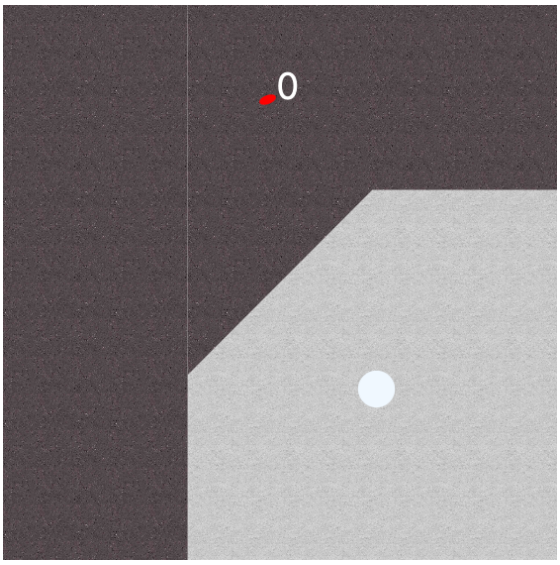


Figure A.2: `DocumentationDemoScenario` after loading Simulation Properties.

the simulation properties. The pedestrian is represented by a red ellipse labeled with its ID. The destination is marked by a light blue circle.

A.2.1.3 Real Data

It is possible to import trajectory data into the Modis framework. This can serve various purposes:

- Qualitative (visual) comparison of simulation results,
- Calibration of model parameters,
- Forced movement of some road users to study the reactions of simulated road users to certain behaviors.

The trajectory data can be provided either as a simple csv-file or as a SQLite db-file. The requirements for both file formats are explained in the following sections. In both cases it is possible to either load only the trajectory data or to automatically generate simulation properties from the trajectories. In the latter case, a road user with the respective traffic mode is generated for each loaded trajectory, using the first time stamp of a trajectory as the entry time. The first and the last point of the trajectory are used as origin and destination.

Trajectories From a CSV File. Any trajectory belongs to a specific road user. In a valid csv file for trajectory data, the header line must describe ID and traffic mode of the respective road user. In the following lines of the file, each trajectory takes exactly three columns. Of these three columns, the first column contains the time stamp, the second

TRAJECTORY_ID	FRAME_NO	X	Y	LABEL
0	0	16.42	38.4	person
0	1	17.34	38.58	person
	...			
1	0	19.74	46.85	car
1	1	19.79	46.65	car

Table A.1: Required structure for an SQLite file.

an x-coordinate, and the third a y-coordinate (both in meters). Multiple trajectories are stored by concatenating these three columns, as shown in the example.

Listing A.2: Example trajectory csv file.

```
1  0,   Pedestrian, ,      1,   Motorist,
2  0,   16.42,      38.4,  0,   19.74,    46.85
3  0.5, 17.34,      38.58, 0.5, 19.79,    46.65
4  1,   18.09,      38.87, 1,   19.8,     46.4
5  1.5, 18.91,      39.19, 1.5, 19.87,    46.12
6  2,   19.75,      39.74, 2,   19.85,    45.86
7  ...
```

Currently[6], valid traffic modes are `Pedestrian`, `Cyclist`, and `Motorist`. The IDs do not need to be sequential, the only requirement is uniqueness.

Trajectories From an SQLite Database File. Trajectory data can also be imported from an SQLite file (ending with .db). Since this was originally implemented to import trajectories generated by the automated approach presented by Trifunović et al. (2021), the naming and units are slightly different than in the rest of the framework. In Table A.1, the required table structure is shown.

Valid strings for the column "LABEL" are `person`, `bicycle`, `car`, and `truck`. Attention: This differs from the traffic modes used in csv files. Both `car` and `truck` are mapped to `Motorist`.

By default, which is also used when loading a .db file from the GUI, this file is expected to contain a table named `trajectories`. Also, the units for coordinates must be in cm and the values in the column "FRAME_NO" correspond to a frame rate of 30 fps. The class `PropertiesFromTrajectoryDB` provides methods for loading trajectories with a different frame rate, table name, or distance unit. This is currently not possible when using the pre-compiled jar files.

[6] An extension for further traffic modes can be implemented relatively easy in `TrajectoryReader.addUserFromTrajectory()`.

A.2.1.4 Structure of the MODIS Framework

The framework is divided into five modules, three of which can be executed to run simulations. A dependency graph between the modules is shown in Figure A.3.

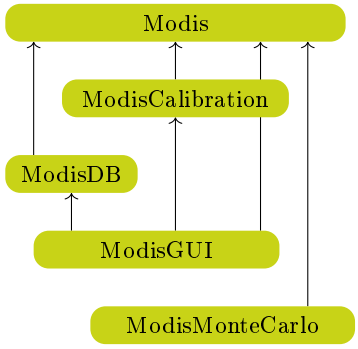


Figure A.3: Structure of the MODIS framework.

- *Modis.* The module named "Modis" is the core of the model. It contains everything that is necessary to perform one simulation run. An executable jar-file containing this module is provided. Section A.2.3 explains how to enter required parameters and start a simulation using the CLI.
- *ModisDB.* ModisDB provides a database of tested scenarios, mostly used and presented in previous publications. This is only for convenience for the GUI, there is nothing to execute directly here.
- *ModisView.* ModisView contains the Modis GUI written in JavaFX. For the reasons explained, it depends on the Modis core module as well as ModisCalibration and ModisDB. The use of the GUI is discussed in more detail in Section A.2.2.
- *ModisMonteCarlo.* ModisMonteCarlo provides helper classes to combine multiple single simulation runs into a Monte Carlo simulation. These tools mainly manage parallel execution of simulations, sampling of random variables, and variation of model parameters. The module depends only on the core Modis module. The execution of Monte Carlo simulations is described in Section A.2.4.
- *ModisCalibration.* ModisCalibration corresponds to research presented in Schiermeyer et al. (2016) and is not discussed further in this chapter. It contains classes and methods to perform calibration of single or multiple similar scenarios using a genetic algorithm. The module itself depends only on the core Modis module, but is controlled via the GUI.

A.2.2 Simulations Using the Graphical User Interface

If one's operating system is configured to execute jar files by double-clicking, one can navigate to the location where ModisView.jar is stored and start the GUI by double-

clicking. Otherwise, one can use the CLI to navigate to the location where ModisView.jar is stored and execute:

```
java -jar ModisView.jar
```

No parameters are required, as everything else is configured in the GUI. After successfully starting the GUI, a view similar to the one shown in Figure A.4 should appear.

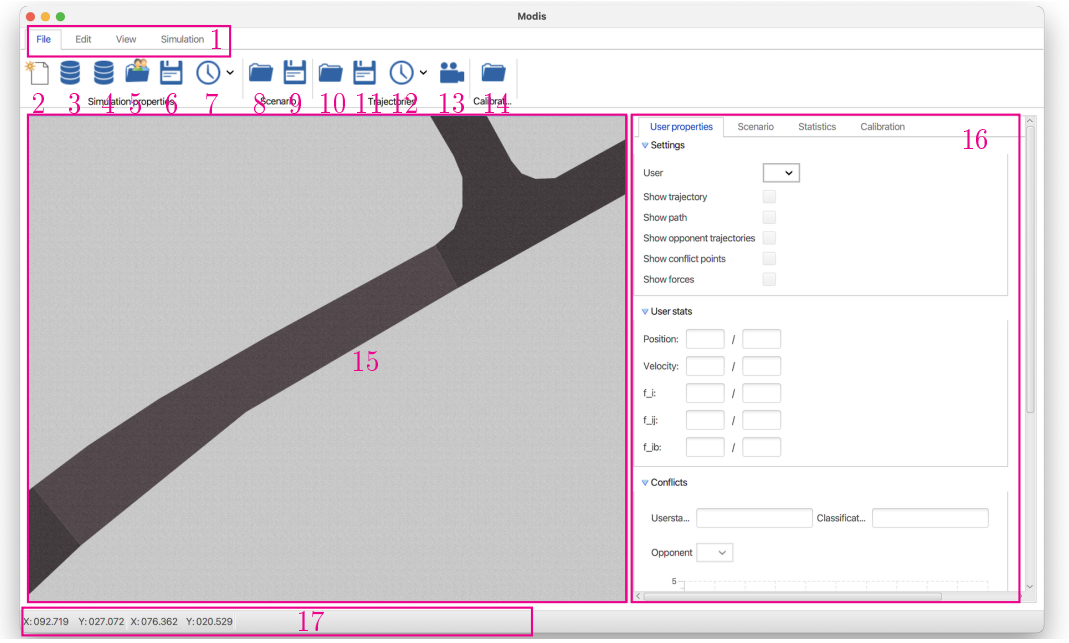


Figure A.4: ModisView after application start.

Overview of GUI Elements.

1. Tab bar for navigating menu categories
2. Clear simulation properties (currently deactivated)
3. Load simulation properties from integrated scenario database
4. Load trajectory data from external database file
5. Load simulation properties from xml file
6. Save simulation properties to xml file
7. Load recent simulation property xml file
8. Open scenario from CityGML or OpenDRIVE file
9. Save scenario to CityGML file (currently deactivated)

10. Load trajectory data from csv file
11. Save simulated trajectories to csv file
12. Open recent trajectory csv file
13. Save animation as gif file
14. Load calibration properties file
15. Main simulation view
16. Simulation details view
17. Status bar

Loading or Creating Simulation Properties. Before a simulation can be started, at least one road user needs to be created. This can be done either by loading simulation properties or by creating them interactively using the GUI. Simulation properties can be loaded from the integrated scenario database (3), automatically created from trajectories stored in a database file (4), or loaded from a simulation properties xml file (5). If one has previously worked with an xml file, one can also quickly load the five most recent xml files using the Recent Simulation Properties button (7).

For interactive creation of road users, one needs to switch to the tab labeled "Scenario" in the Simulation Details View (16). A tabular view similar to the one shown in Figure A.5 appears.

The upper of these tables contains road users, the lower destinations for road users. In the depicted tables, one destination with ID 0 and one pedestrian, also with ID 0, are already present. If simulation properties were loaded using one of the described methods, these two tables should also already contain at least one entry. If one starts with an empty scenario, both tables will be empty. In this case, the first step is to create the first destination.

Technically, a destination is modeled by so-called `ODLocation`s, which in turn are represented by a polygonal line with a theoretically unlimited number of vertices. A road user has arrived at their polygonal destination when any point on the polyline is reached. In the GUI, a destination is created using the input fields labeled with x_1, y_1 and so on below the "Destinations" table. An input of up to three vertices is supported[7]. The coordinates for the vertices of the polyline can be entered directly into the input fields. After clicking in an input field, it is also possible to select a point from the main simulation view (15). This is also indicated by the message "Waiting for mouse click..."

[7] More complex polylines need to be specified using xml files. In this case, only the first three vertices of the polygonal line will be displayed in the table, but for the simulation, the whole polyline will be used.

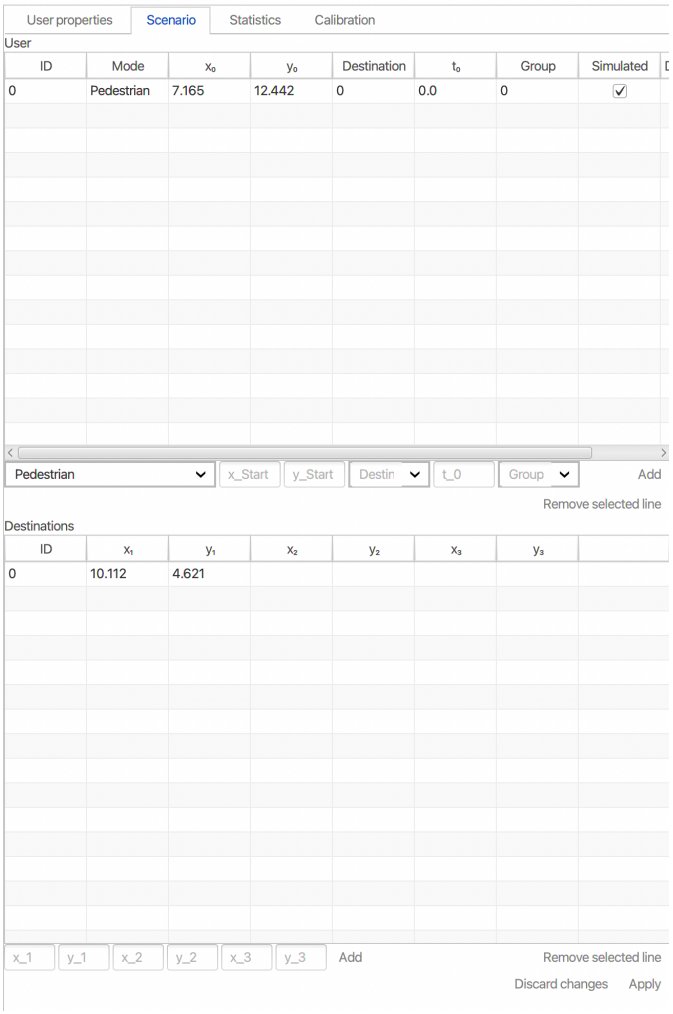


Figure A.5: Scenario tab in the Simulation Details View of ModisView.

in the status bar (17) at the bottom of the window. Clicking the "Add"-button next to the input fields adds the destination to the table and makes it available for user creation.

Road users are created in a similar manner using the controls below the upper table. First, a user mode must be selected in the drop down menu. A start coordinate can be entered or selected by mouse click. The drop-down menu labeled "Destination" provides a choice between all available destinations (note that destinations can be shared by road users). The input field labeled t_0 specifies the time in seconds after the simulation start at which the road user will appear at their start coordinate. Finally, a group ID can be selected. A 0 indicates that the road user does not belong to a specific group. A specific group behavior is currently only considered for pedestrians, all other traffic modes behave the same way regardless of which group they belong to. Again, clicking on the "Add"-button adds the selected values to the table.

Finally, when all destinations and users were added to the tables, clicking the "Apply"-button on the lower right will add everything to the model. This is indicated by the fact that only now newly created road users appear in the main simulation view.

The save button (6) can also be used to save interactively created simulation properties as an xml file for later reuse.

A.2.3 Simulations Using the Command Line Interface

Simulations without the GUI cannot be started by double-clicking the jar-file, since at least a Simulation Properties file has to be supplied. The simulation is started by executing the following command:

```
java -jar ModisCore.jar -<options>
```

To get a summary of the required and optional arguments, use the –help option (or -h). The output looks as follows:

Listing A.3: Runtime parameters for ModisCore.

```
1  usage: ModisCore
2   -h,--help                              print this message
3      --ignoreFollowsRealTrajectory       Users specified in a properties xml
4                                          file to follow a real trajectory are
5                                          simulated regardless of whether
6                                          followsRealTrajectory is set.
7      --listscenarios                     Print list of valid scenario names for
8                                          use with -s
9   -o,--output <arg>                      Output filename, ending either with
10                                         .csv, .db or .h5. If no output is
11                                         specified, the name of the
12                                         <propertiesfile>.db will be used to
13                                         store the simulated trajectories.
```

```
14 -p,--properties <arg>            Simulation Properties (required).
15                                  Either as .xml-file or as trajectory
16                                  data stored in a .csv or .db file. If
17                                  simulation properties are derived from
18                                  trajectory data, specifying a scenario
19                                  with --scenario is required.
20 -r,--realdata <arg>              Real data file, either as .csv or .db
21                                  (optional). This option is ignored if
22                                  --properties is already a .csv or .db
23                                  file.
24 -s,--scenario <arg>              Scenario (optional). Either contained
25                                  in a file (ending with .gml or .xodr)
26                                  or as a string representation from the
27                                  list of built-in scenarios (see
28                                  --listscenarios). Caution: Providing a
29                                  scenario will override a scenario that
30                                  may be specified in the simulation
31                                  properties file. This can lead to
32                                  errors or unexpected simulation
33                                  results.
```

A.2.4 Monte Carlo Simulations

Monte Carlo simulations must also be started from the CLI. The parameters for one Monte Carlo run (Scenario, properties file, number of simulations, model parameters to vary etc.) are specified in a different type of xml-file called Monte Carlo properties file. The structure of such a file is exemplarily shown in Appendix A.5.2.

Listing A.4: Runtime parameters for ModisMonteCarlo.

```
1 usage: ModisMonteCarlo
2  -h,--help              print this message
3  -i,--input <arg>       Input filename (required).
4                         If <arg> is a file name ending with .xml
5                         it is treated as a single Monte-Carlo
6                         properties file. In every other case,
7                         a text file containing one xml file name
8                         per line is expected.
```

A.2.5 Postprocessing / Simulation Results

Microscopic simulations enable the generation of large data sets that describe the detailed evolution of traffic systems over time. These data may need to be post-processed for their specific use. For each simulated road user, the GUI provides built-in access to:

- a visualisation of current and past positions (i.e., the traveled trajectory),
- the current velocity and active forces,
- for all perceived users: their predicted trajectories, distance function, conflict point, and situation classification,
- distance-time graph and velocity-time graph (these are displayed and, if available, compared with those of the corresponding trajectories from real data).

Apart from the live supervision of simulation parameters, simulation results can always be saved as trajectory data. In the context of importing real data to the simulation, trajectories stored in CSV and SQLite files were already discussed in Section A.2.1.3. The export of simulation results to one of these two file formats follows the same file structure.

In addition, simulation results can also be stored in a HDF5 database file. This is especially useful for storing the output of Monte Carlo simulations, since HDF5 is a file format designed to store large data sets with a homogeneous structure. A HDF5 database consists of a number of tables, which are organized in a file-system-like structure. Unlike the other file formats, when exporting the simulation results, the trajectory of each simulated user is saved in a separate table. Figure A.6 shows a screenshot of the software HDFView[8] with an open simulation result .h5 file.

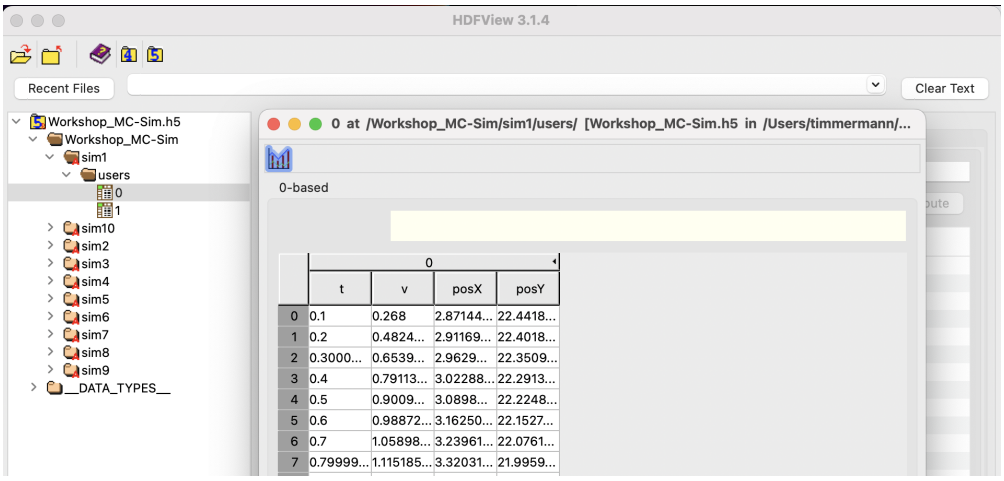


Figure A.6: HDFView.

[8] https://hdfgroup.org/downloads/hdfview/

The left part of the figure indicates the structure of the file: A folder is created for each numbered simulation run. In its subfolder named users a table is stored for each simulated user, named after the user ID. The table itself contains four columns for time stamp, velocity, x-, and y-coordinate.

A.2.6 Next Steps

The instructions explained above all rely on the pre-compiled jar files provided in the Gitlab repository. These are fully sufficient if the framework is to be used as published. However, own changes or additions to the existing code[9] require compilation of the code.

Compiling the Modis framework requires a number of external dependencies as well as the interdependency of the Modis modules as explained above. To simplify this task, build configuration files for the *build automation software* Gradle are supplied. A local installation of the Gradle software itself is optional, as the repository includes a so-called "wrapped" version of Gradle. Therefore, to make the following commands work, only a copy of the repository is required.

In Gradle, pre-configured actions such as compilation or execution are called "tasks". The task to execute a program is usually called "run". Task and module names are seperated by a colon, therefore the run task for a module has the structure

```
:<ModisModule>:run
```

A run task is available for the modules `ModisCore`, `ModisView`, and `ModisMonteCarlo`. The run task automatically checks whether a (re-)compilation is necessary and triggers it if required. The syntax to use the included wrapped Gradle commands slightly differs depending on the operating system. For Linux and MacOS, a run task, in this example for ModisView, is triggered by executing

```
./gradlew :ModisView:run
```

from the root folder of the project in the terminal. For Windows, a similar command is

```
.\gradlew.bat :ModisView:run
```

If, such as for ModisCore, arguments have to be specified, they have to be passed enclosed in double quotes as an argument to the Gradle command in the following way:

```
./gradlew :ModisCore:run --args="-p=properties.xml
                                -r=trajectories.csv"
```

Compile to jar Files. The supplied gradle configuration also provides a configuration to build a so-called "fat jar", a self-contained jar file which contains all external dependencies apart from the Java runtime itself. A fat jar can be used to distribute packages of

[9] A first example for custom implementations is presented in the exercise in Section A.4.

own code to collaborators who in turn only need to execute the commands as explained in Sections A.2.2 – A.2.4. Jar files of own implementations are also useful if one wants to perform simulations on a machine with limited access, e.g., a computation cluster.

To build a fat jar, one needs to execute the Gradle task `:shadowJar`, for Linux/Mac OS:

```
./gradlew -PfatJar :<ModisModule>:shadowJar
```

or for Windows:

```
.\gradlew.bat -PfatJar :<ModisModule>:shadowJar
```

where `<ModisModule>` has to be replaced by either `ModisCore`, `ModisView`, or `ModisMonteCarlo`. For external dependencies that are platform dependent, the configuration adds versions for Linux, Mac OS and Windows, independently of the operating system where the compilation is executed.

A.3 Simulation in Detail

The previous section described how one can interact with the MODIS framework without any further modifications and how to use it for traffic simulations. This section now dives deeper into the structure of the code. It serves as a starting point to understand where and how to implement own modifications or extensions to the existing framework.

A.3.1 Simulation Setup

A.3.1.1 Model Data Structure

The main class in which the data required for a simulation is stored is called `ModisScenario`. Within this class, several other classes are nested. An overview of the class structure is shown in Figure A.7.

As the class diagram shows, not many attributes are stored in `ModisScenario` itself. Instead, most of the attributes relevant for describing a traffic simulation are stored in a `TrafficNet`. The traffic net describes both the static (static as in immovable, not in the sense of Java) parts of a traffic situation (e.g., infrastructure) and the dynamic parts (e.g., road users). In the terms introduced in Section A.2.1, the `TrafficNet` represents both the Scenario as well as the Simulation Properties. Thus, all attributes of a `TrafficNet` listed in the class diagram belong to either of these.

Representation of a Scenario. The scenario itself is represented by a list of `TrafficRegions`, while the information from a simulation properties file is stored in lists of `Users` and `ODLocations`.

As discussed in Section A.2.1.1, a scenario can either be imported from files following the CityGML or OpenDRIVE file formats or defined directly in Java code. Since the

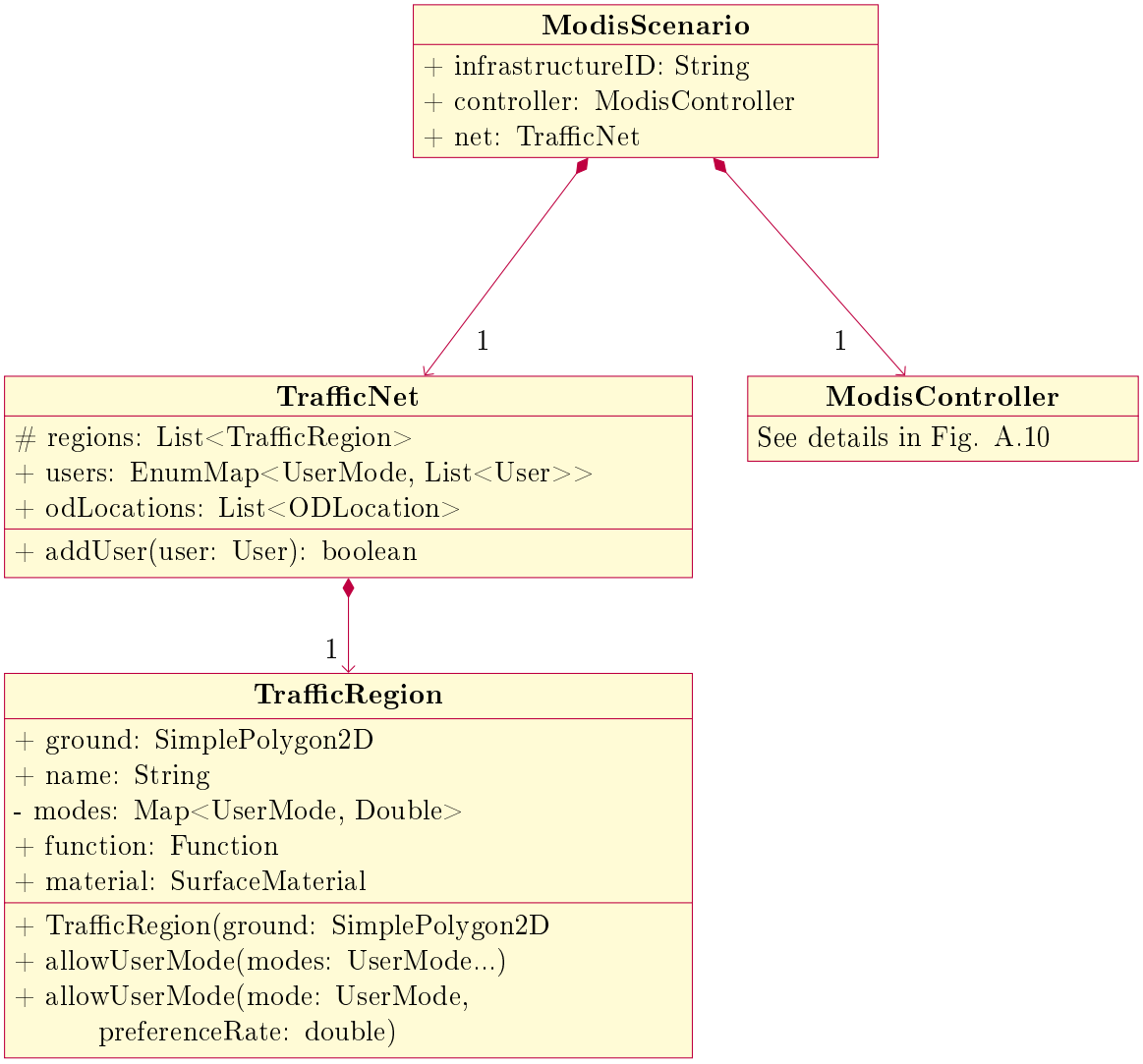


Figure A.7: Class diagram (excerpt) for ModisScenario and nested data classes.

external file formats lack some features of the MODIS framework, defining a scenario in code is often the best choice. On the other hand, a scenario defined in Java code comes with the disadvantage that the pre-compiled jar files cannot be used; in this case, the newly defined scenario needs to be compiled along with the rest of the framework, as described in Section A.2.6.

The specification of road networks in OpenDRIVE or city models in CityGML is outside the scope of this chapter and is better explained in other sources (see the references in Section A.2.1.1). Instead, this section discusses the internal data format and briefly demonstrates it using the simple scenario "DocumentationDemoScenario" that was already introduced in the previous section.

Listing A.5 shows an excerpt from the definition of `DocumentationDemoScenario.java`. The full code of the Java class is printed in Appendix A.5.3.

Listing A.5: Defining the traffic net.

```
1 @Override
2 public void initTrafficNet() {
3     // Define the polygonal areas of the Scenario
4     SimplePolygon2D roadStraightPoly = new SimplePolygon2D();
5     roadStraightPoly.addVertex(new Point2D(0, 0));
6     roadStraightPoly.addVertex(new Point2D(0, 15));
7     roadStraightPoly.addVertex(new Point2D(5, 15));
8     roadStraightPoly.addVertex(new Point2D(5, 0));
9     ...
10    // Define traffic regions
11    TrafficRegion roadStraight = new TrafficRegion(roadStraightPoly);
12    roadStraight.allowUserMode(UserMode.Motorist, UserMode.Cyclist);
13    roadStraight.material = SurfaceMaterial.light_asphalt;
14    roadStraight.function = Function.DRIVINGLANE;
15    roadStraight.name = "Straight Road Segment";
16    ...
17    // Add traffic regions to traffic net
18    this.net.addRegion(roadStraight);
19    ...
20 }
```

A scenario in MODIS consists of one ore more traffic regions represented by simple polygons in the two-dimensional plane. The excerpt shows the three mandatory steps to create a scenario from polygonal traffic regions:

- **Step 1: Define Polygon.** As can be seen from the class diagram, a `TrafficNet` consists of one or more `TrafficRegions`, whose outer shape is described by an object of the class `SimplePolygon2D`. In the example, an empty polygon is first

created (line 4) and its vertices are subsequently added (lines 5–8). The shape of the polygon is determined by connecting the vertices consecutively, where the last vertex is again connected to the first one.

- **Step 2: Define Traffic Region.** The newly defined polygon serves as the attribute for the constructor of a `TrafficRegion` (line 11). The user modes that are allowed to use a traffic region need to be explicitly specified. This can be done either by granting a "full permission", as in the example code snippet (line 12), or by specifying a preference rate between 0.0 and 1.0 (the method used here corresponds to a preference rate of 1.0). The effects of preference rates on the navigation are explained in Section A.3.2.2.

  The attributes `name`, `function`, and `material` are not strictly necessary for the simulation to work, but it is advisable to use sensible values here for displaying the scenario in the GUI. Both `Function` and `SurfaceMaterial` are Java-Enumerations. A full list of valid values can be found in the Javadoc.

- **Step 3: Add Traffic Region to Traffic Net.** Finally, the traffic region is added to the traffic net (line 18).

An image of the complete DocumentationDemoScenario has already been shown in Figure A.1. The part of the scenario specified by the excerpt in Listing A.5 is highlighted in magenta in Figure A.8.

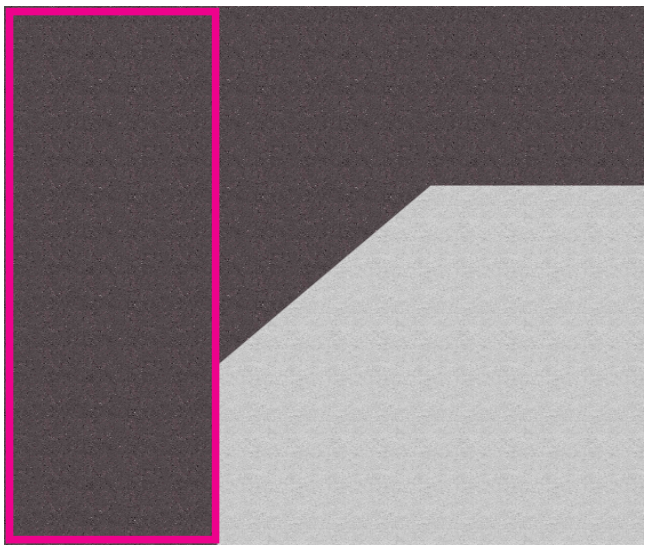


Figure A.8: Part of DocumentationDemoScenario as described by the example code.

This small example demonstrates the minimal steps necessary to specify a scenario. It is possible to represent more complex scenarios:

- One can specify lanes for cyclists and cars using directed edges. These will be added to the respective navigation graph; more details on this are given in Section A.3.2.2.

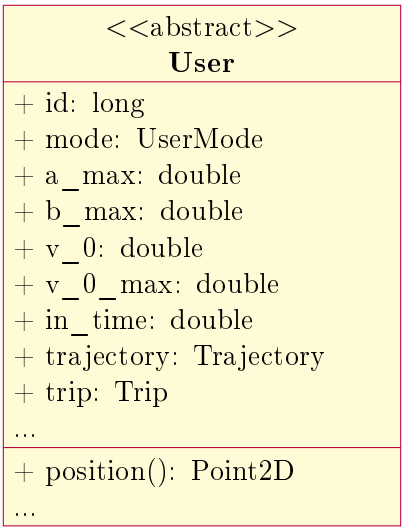


Figure A.9: Class diagram (excerpt) for the class `User`.

For now, just note that the logic for lanes is fully integrated in the continuous movement model, i.e., a road user is not aware whether they move along a lane or not.

- Traffic rules can be added using a "right-of-way graph", which specifies right of way relations between traffic regions. More on this can be found in Diekmann & Schiermeyer (2019).

Note that when using the importers for CityGML or OpenDRIVE files, the geometric descriptions of a scenario are converted into the same structure as described in the above example. As such, after importing such a scenario, a further distinction between externally or internally specified scenarios is not necessary.

The User Class. Almost all information previously introduced as Simulation Properties is stored in the class `User`. The abstract class `User` serves as the superclass for all implementations of the various traffic modes. Therefore, it contains attributes that are valid for all traffic modes, such as position, destination, or current velocity.

The class diagram in Figure A.9 shows an excerpt of the attributes and methods of the abstract class `User`. Note that the class has a lot more attributes than listed in the diagram; the ones listed here are the ones that are necessary when initializing a simulation. Other attributes of `User` are omitted for now and will be introduced as needed.

All implementations of road users extend the abstract class `User`. For a detailed overview of the inheritance hierarchy of the implemented subtypes of `User` please refer to the accompanying Javadoc (see Section A.1.2).

A.3.1.2 Populating the Data Structure

Before a simulation can be started, the data structure that was described in the last sections needs to be populated. As already explained, the scenario can either be specified in Java code or loaded from standardized file formats. Simulation properties, on the other hand, can either be supplied as xml files as described in Section A.2.1.2 or automatically generated from trajectory data. The latter serves the purpose that often situations observed in recorded data should be simulated for comparison. As described in Section A.2.1.3, the MODIS framework offers the possibilities to load trajectory data from csv files or SQLite database files.

The implementations of the different importers can be found in the package `modis.io` of the ModisCore module:

- The class `PropertiesReader` contains methods to import simulation properties from xml files.
- `TrajectoryReader` contains the code to load trajectory data from csv files.
- Methods to load trajectory data from a database file are implemented in `PropertiesFromTrajectoryDB`.

The two implementations to import trajectory data generate objects described by the class `TrajectoryData`, which contains only the space-time-coordinates of a trajectory. The actual generation of road users from `TrajectoryData` objects is implemented in `TrajectoryReader.addUserFromTrajectory()`. In all cases, the generated single road users are loaded into `TrafficNet.users` by calling the method `TrafficNet.addUser()`.

Note that from a conceptual point of view, the scenario and the simulation properties are independent of each other. However, in the MODIS framework, the scenario is always loaded before the simulation properties. This has the advantage that basic validity checks are possible when loading simulation properties, e.g., checking whether origins and destinations of road users are within the coordinates specified by the scenario and in the traffic areas permitted for the respective user mode.

A.3.2 Simulation State Flow: The ModisController Class

The entry point for starting and subsequently controlling a simulation is the class named `ModisController`. Thus, almost everything described in this section is implemented in this class, or at least triggered directly by it. Figure A.10 shows a (shortened) class diagram listing the most relevant methods and attributes of `ModisController`.

The MODIS framework follows the Model-View-Controller (MVC) design pattern. As such, ModisController serves as a logical link between the model and the user interface (user interface in this case refers not only to the GUI, but to any kind of interaction with the software). As a consequence, the methods implemented in ModisController can roughly be divided into two groups:

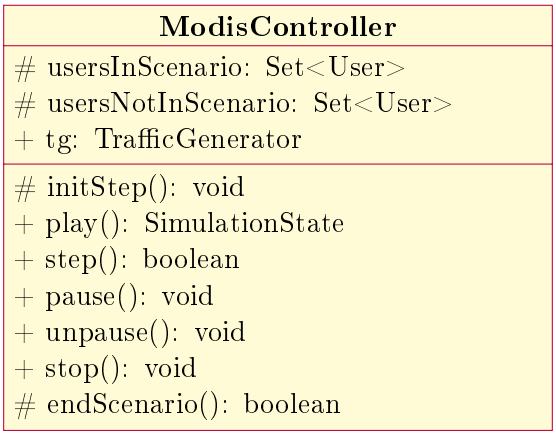


Figure A.10: Class diagram (excerpt) for `ModisController`.

- **External Controls.** The methods `play()`, `step()`, `pause()`, `unpause()`, `stop()` provide the ability to control the simulation[10]. Therefore, they need to have public access. These methods usually call the second kind of methods.

- **Model Controls.** Methods which directly trigger logical parts of the traffic model, either once per simulation run (e.g., `initiStep()` before the first time step) or once per simulation time step (e.g., `detectConflicts()`, `react()`, etc.). Most of these methods have protected or private access. Since the actual traffic model is mostly implemented in methods that are called once per time step, the following sections will mainly focus on the different methods of this type.

A.3.2.1 User Generation

In general, there are two ways to add road users in MODIS. We distinguish "single road users" and "generated road users". While single road users are specified separately, generated road users are created automatically using the so-called `TrafficGenerator` class. To make a clear distinction, only the latter road users are referred to as "generated road users" in this chapter, although technically the single road users are also "generated" at some point.

Single Road Users. Single road users are defined by their specific attributes such as user mode, start coordinate and velocity, destination, and in_time. The in_time describes the time span in seconds from the simulation start, after which the user appears at their start coordinate. This implies that each road user has to be specified separately, regardless of whether they use the same Origin-Destination (OD) relation.

[10] Even if the controls in the GUI suggest otherwise, the function for reversing time steps is currently not supported, i.e., simulation in the reverse direction is not possible.

The single road user is the category that can also be specified using the tabular view of the GUI, as explained in Section A.2.2. The syntax for specifying single road users in the input file has also already been shown in the excerpt in Listing A.1.

Note that, regardless of their in_time, all single road users are already initialized at initialization of the controller in the method `ModisController.initStep()`, i.e., before the first simulation time step. Road users with an in_time of 0 are stored in `ModisController.usersInScenario` and those with a larger in_time in `ModisController.usersNotInScenario`, for later entry into the system.

Generated Road Users. The framework provides the `TrafficGenerator` class, which is known to the `ModisController` as its attribute `tg`. The traffic generator provides automated, repeated generation of road users for specific OD-relations. The class diagram in Figure A.11 lists the methods and attributes relevant for understanding the class `TrafficGenerator`.

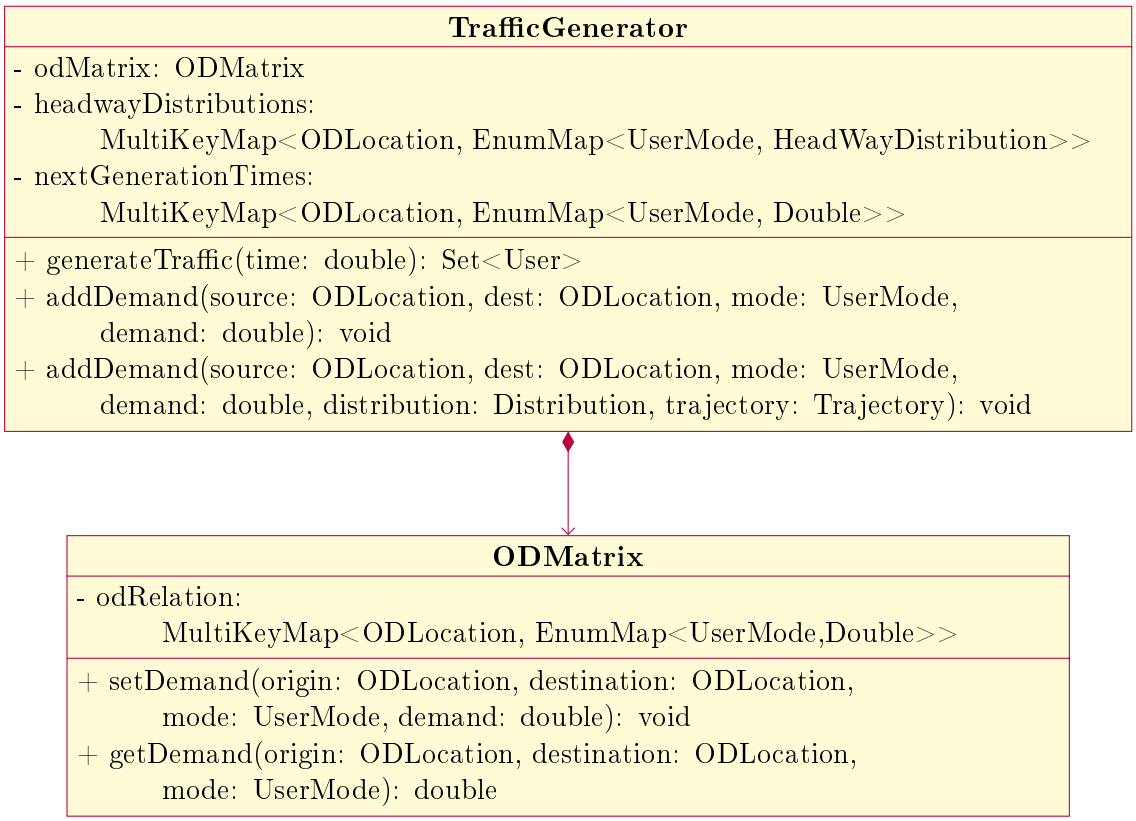


Figure A.11: Class diagram (excerpt) for `TrafficGenerator`.

The implementation of traffic demand in MODIS utilizes the `MultiKeyMap` class from the Apache Commons Collections framework[11] in several places. This class provides

[11] See `https://commons.apache.org/proper/commons-collections/` for details.

mappings from combinations of more than one key to single values. For example, a matrix can be modeled using `MultiKeyMap`s by using exactly two keys per value, where each key corresponds to either a row or a column of the matrix.

In the MODIS framework, the demand is stored in an object of the class `ODMatrix`, which is known to the traffic generator as its attribute `odMatrix`. Within this class, there is a nested `MultiKeyMap` that maps OD pairs to a `EnumMap<UserMode, Double>`. The nested `EnumMap` contains the actual demand (measured in road users per hour) for each user mode. The methods `ODMatrix.setDemand()` and `ODMatrix.getDemand()` allow to interact with the content of the OD matrix, without having to directly access this nested map structure.

Furthermore, headway distributions for each OD pair are stored in the traffic generator in `TrafficGenerator.headwayDistributions` using the same logic as for the scalar demand. Headway refers to the duration between the generation of two successive road users of the same OD-relation. This duration may be deterministic or random; currently, a constant value, a uniform distribution, or an exponential distribution are supported. These distributions are implemented such that their expectation equals the desired demand. Other probability distributions can easily be added by implementing the interface `HeadwayDistribution` specified in `TrafficGenerator`. When the traffic generator is initialized or a new demand is added, the next generation time point for each OD relation is calculated by sampling the headway distribution and stored in the `TrafficGenerator.nextGenerationTimes` attribute. Optionally, a pre-defined trajectory can be stored in `TrafficGenerator.manualTrajectories` for each OD pair. If no manual trajectory is present for an OD pair, the route search is used as described in Section A.3.2.2.

To avoid having to directly deal with these nested map structures, there are two implementations of `TrafficGenerator.addDemand()` methods with different signatures: The simpler one requires origin and destination of a relation as well as the respective user mode and the (scalar) demand as road users per hour. The second one takes the headway distribution as well as a manual trajectory as additional arguments (see operations block of class TrafficGenerator in Figure A.11). The first one initializes the manual trajectory as null and the headway as a constant value.

The method `TrafficGenerator.generateTraffic()` finally generates the actual road users. It has to be called by the controller, which supplies the current simulation time. Internally, the traffic generator creates a road user for each OD pair where the according entry in `TrafficGenerator.nextGenerationTimes` is less or equal to the current simulation time.

Listing A.6 shows an excerpt of a Simulation Properties xml file where parameters for the traffic generator are specified. This is a slight modification of the example already presented in Section A.2.1.2. Instead of a single pedestrian, pedestrians are generated endlessly in this example, with an exponentially distributed headway with an expectation

of 200 pedestrians per hour.

Listing A.6: Specifying periodically generated pedestrians in a Simulation Properties File.

```
<?xml version="1.0" encoding="UTF-8"?>
<properties schemaVersion="1.4">
  ...
  <odLocations>
    <odLocation id="0">
      <polyline>
        <point>
          <x>10.112</x>
          <y>4.621</y>
        </point>
      </polyline>
    </odLocation>
    <odLocation id="0">
      <polyline>
        <point>
          <x>7.165</x>
          <y>12.442</y>
        </point>
      </polyline>
      <demand>
        <dest_id>1</dest_id>
        <user_mode>Pedestrian</user_mode>
        <demand_per_mode>200.0</demand_per_mode>
        <distribution>Exponential<distribution>
      </demand>
    </odLocation>
  </odLocations>
</properties>
```

An important syntactical difference becomes apparent: For a single road user, it is not necessary to explicitly specify an `<odLocation>` element for the starting coordinate. In contrast, generated traffic requires this, since the `<demand>` element has to be specified as a child of the `<odLocation>` element of the origin.

Remark on the Choice of MultiKeyMaps to Implement OD Matrices. In the field of traffic demand modeling, data are often represented using matrices[12]. In its

[12] See, for example, Patriksson (2015) for a general treatment of the traffic assignment problem and Bera & Rao (2011) for an overview on estimating the OD-matrix from empirical data.

simplest form, the entries of an OD matrix are scalar values representing traffic demand, usually measured in road users per time span. Here, rows and columns of the matrix relate to origins or destinations.

Depending on the modeled road network, the resulting matrix may be sparse or densely populated. Considering, e.g., a simple highway network, there are few origins and destinations in contrast to the geographical size covered by the network. For such networks, demand exists between most of the possible combinations. Technically, the corresponding OD matrix can simply be implemented as a two dimensional array of floating point values, with an additional mapping between array indices and origins or destinations.

However, typical MODIS scenarios deviate from this example: Here, we may have many origins and destinations (which are modeled using the class `ODLocation`, as described earlier), some on the road, some on the sidewalk, etc. Yet, for most of the possible OD relations, there is no traffic demand (e.g., no demand between road and sidewalk). This results in a large but sparse matrix. Using methodology from numerical mathematics, it would be feasible to implement the OD matrix efficiently as a two dimensional array. But considering that we store not only the actual demand but rather a number of encapsulated objects (demand, headway distribution, optional fixed trajectory, next generation time), the MultiKeyMap is more appropriate.

A.3.2.2 Route Search

This section describes the route search for a road user. According to the previous section, there are three slightly different cases how road users are generated in a MODIS simulation:

- Single road users with in_time $= 0$, i.e., they are generated before the first time step,
- Single road users with in_time > 0, i.e., they are generated during the simulation,
- Generated road users which are created by the traffic generator during the simulation.

At a fixed time step, both currently active road users (i.e., their in_time is less than or equal to the current time and they have not reached their destination yet) and newly generated road users are contained in `ModisController.usersInScenario`.

All current, past, and future road users (if already known) are stored in `TrafficNet.users`. A traffic user is always added to the traffic net by calling the method `TrafficNet.addUser()`, while its actual creation may vary depending on the three cases above (i.e., the place where the constructor of the respective subclass of `User` is called).

Overview of the Route Search Implementation. Successfully adding a new road user to the traffic net with `TrafficNet.addUser()` also calls the method `TrafficNet.defineUserStartOrientation()`. As its name suggests, this method mainly calculates

the body orientation of a user who has not moved yet. For this operation, a free-flow trajectory is necessary, i.e., a desired trajectory that a road user will follow if there are no conflicts. The method checks if there is an optional pre-defined free-flow trajectory – otherwise (which is the usual case) it triggers the calculation which is implemented in the class `TrajectoryBuilder`. This class is known to the traffic net as its attribute `TrafficNet.trajectoryBuilder` and is initialized during initialization of the traffic net.

It is a key modeling innovation of MODIS that road users can navigate from their current position towards their destination without pre-defined lanes. The computation of a realistic and smooth free-flow trajectory is separated into two major steps:

- First, a shortest path from the current position of a road user to their destination is determined. The computation relies on building a so-called navigation graph and performing a shortest path search on the graph.
- Second, the shortest path is transformed into a trajectory.

Both steps are discussed in the following paragraphs.

Building the Navigation Graph and Finding the Shortest Path. The computation of the shortest path from current position to destination is treated as a graph-theoretic problem. For this a navigation graph[13] is determined, as it is commonly used in the field of pedestrian simulation (see, e.g., Höcker et al. (2010)). Given a graph-based representation, suitable algorithms can be used to find a shortest path. More precisely, the following steps are executed and exemplarily visualized in Figure A.12 for the "DocumentationDemoScenario".

- **Static Navigation Graph.** An initial graph is computed using only the geometry of the traffic scenario. It is built by adding edges from all vertices of the polygonally shaped regions to all respectively reachable[14] vertices. The result is called the static navigation graph; it is constant over time and only depends on the traffic mode. Thus, it is only calculated once in the beginning of a simulation. Examples are shown in Figure A.12a for pedestrians (which cannot enter the road on the left side) and in Figure A.12b for motorists (which cannot enter the sidewalk).
- **Dynamic Navigation Graph.** For a fixed road user, the current position and the destination are added as two additional vertices to the static navigation graph. Again, edges are added for all reachable vertices. The result is called the dynamic navigation graph for the road user under consideration, which changes over time

[13] The concept originates from the "visibility graph" developed in the field of robotics (see De Berg et al. (2000)) for navigating through an environment with polygonally shaped obstacles. Here, "visibility" is interpreted as reachibility, whereas visibility is more generally discussed in terms of a perception model in Section A.3.3.3.

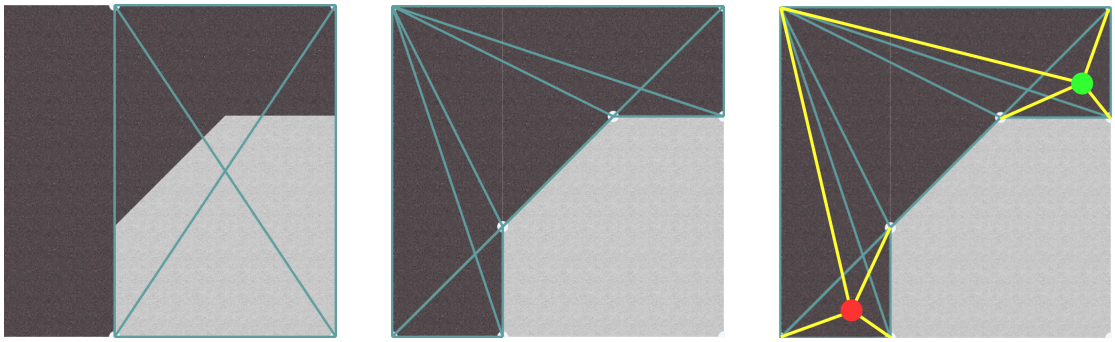
[14] A point is reachable from a given point if the direct connection does not intersect with any obstacle. It is clear that reachability is a symmetric relation, and this is also true for the corresponding graph.

as it depends on the current position. In Figure A.12c this is illustrated for a motorist: The current position is colored red, the destination is green, and newly added edges are highlighted in yellow.

- **Shortest Path.** Based on the dynamic navigation graph, a search for the shortest path from the current position to the destination is performed using the classical A* algorithm (see Hart, Nilsson & Raphael (1968)). The result is a directed path W defined as a tuple of vertices, say

$$W = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_i, \ldots, \boldsymbol{w}_n)\,, \tag{A.3.1}$$

which links the current position $\boldsymbol{w}_1$ to the destination $\boldsymbol{w}_n$ via $n-1$ edges.



(a) Static navigation graph for pedestrians.

(b) Static navigation graph for motorists.

(c) Dynamic navigation graph for a motorist.

Figure A.12: Example for routing through polygonally shaped obstacles using a navigation graph.

Computing a Clothoid Trajectory. The path W does yet not resemble realistic behavior for two reasons: First, it lies on the boundary of traffic regions. Since the path is intended to model the movement of the center of a road user, movement along W would imply contact with obstacles. Second, it is not smooth but makes sharp turns[15]. Moreover, it is not yet a trajectory, i.e., not time-dependent.

In the following, let $\boldsymbol{x}_i^{\mathrm{des}}(t)$ denote the desired trajectory of road user i. The modeling assumption is that a road user will follow their desired trajectory in the absence of conflicts with other road users. To obtain $\boldsymbol{x}_i^{\mathrm{des}}(t)$ from the previously calculated path W, the following steps are executed:

- **General Clothoid Computation.** Smooth curves commonly used in traffic engineering are clothoids (see, e.g., Meek & Walton (2004) and references therein).

[15] Empirical observations (see Pascucci (2020)) indicate that road users prefer smoother trajectories regardless of the traffic type.

These have different desirable properties from an engineering point of view and are parametrized by their arc length.

MODIS replaces each edge of the path W by a single clothoid segment and connects the individual segments while maintaining C^1 smoothness. For this purpose, the algorithm proposed by Bertolazzi & Frego (2015) is implemented. It numerically computes a discretized clothoid and requires start point, end point, and tangent slopes at these points. The algorithm is sequentially evaluated for each path segment. More details can be found in Timmermann (2022).

- **Position Adjustment.** The trajectory still needs to be moved away from the boundary. For this purpose, all nodes are moved away from the obstacle edge by half their body width plus an individual obstacle distance orthogonal to the tangent. Now the clothoid sections can be determined and assembled to form the complete trajectory, as previsoyly described.

The described procedure yields a parametrized curve in x-y-coordinates. Clothoids have the essential feature that they are parameterized by their arc length s. By asserting a time relation $s = s(t)$, a (one-dimensional) motion along the curve is obtained, which can contain arbitrary accelerations (see also Section A.3.3.2 for more details on the acceleration). This finally yields the time-dependent trajectory $\boldsymbol{x}_i^{\text{des}}(t)$.

Preference Rates. An empirical observation at multimodal intersections is that cyclists use the sidewalk in *some* situations, while they most often stay on the road. To model such partial use of a traffic region, MODIS provides the concept of preference rates: Edges in less preferred regions are formally penalized when calculating the shortest path. The resulting "shortest path" is deflected from these regions. The concept is implemented as follows:

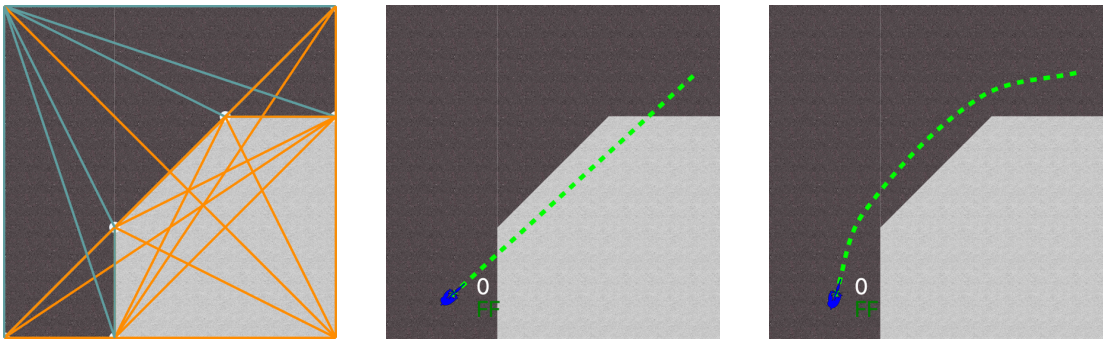
- Each traffic region of a scenario is assigned a preference rate between 0.0 and 1.0 for each allowed user mode. If no preference rate is specified, the default value 1.0 is used. Note that a value of 0.0 is equivalent to the traffic region not being allowed for that user mode.
- Each edge of the navigation graph is assigned the lowest preference rate of all traffic areas that the edge traverses.
- When computing the shortest path, edge lengths are artificially increased by dividing the actual edge length by the preference rate (e.g., a preference rate of 0.5 corresponds to a doubling of the length).

Listing A.7 shows a modified excerpt of the method `initTrafficNet()` introduced in Listing A.5 to define the traffic net of the demo scenario. Here, the traffic region `sidewalk` is assigned a preference rate of 1.0 for pedestrians and 0.5 for cyclists.

Listing A.7: A traffic region using preference rates.

```
1 ...
2 TrafficRegion sidewalk = new TrafficRegion(sidewalkPoly);
3 sidewalk.allowUserMode(UserMode.Pedestrian);
4 sidewalk.allowUserMode(UserMode.Cyclist, 0.5);
5 sidewalk.material = SurfaceMaterial.pavement;
6 sidewalk.function = Function.FOOTPATH;
7 sidewalk.name = "Sidewalk";
8 ...
```

The result of this adjustment can be illustrated in a small example. Figure A.13a shows the navigation graph for cyclists for the `DocumentationDemoScenario`. The edges highlighted in orange are affected by the preference rate specified in line 4 of Listing A.7. Figure A.13b shows: a right-turning cyclist would cross the gray sidewalk if all preference rates for the entire scenario were set to 1.0 (or any constant value). If the preference rates for the sidewalk are set to 0.5, the new "shortest path" is to stay on the road, as shown in Figure A.13c.



(a) Navigation graph for cyclists.

(b) All preference rates 1.0.

(c) Preference rates of orange edges set to 0.5.

Figure A.13: Comparison of navigation graph and shortest path with and without preference rates.

Lanes. As briefly mentioned earlier, the MODIS framework also provides the option to specify lanes. This is mainly motivated by two factors:

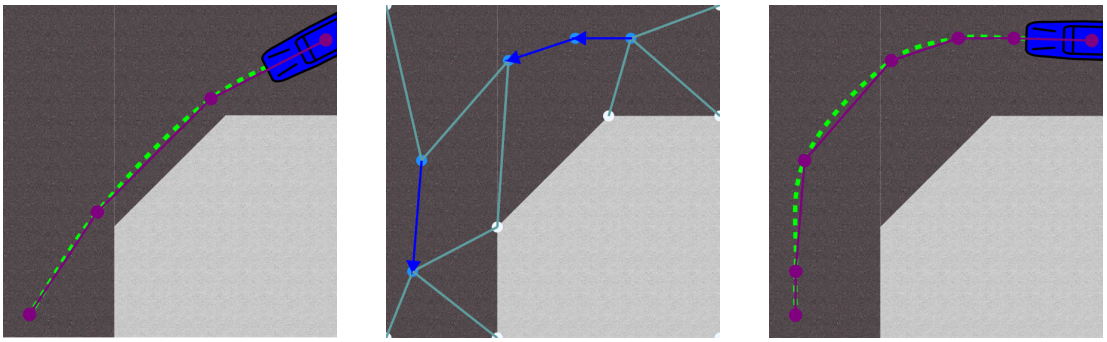
- **Compatibility.** File formats such as OpenDrive, which can be used to import a scenario into MODIS, rely on lanes to describe the geometry of a scenario. An important feature of lanes is that they usually limit the direction of movement for the underlying traffic area. This cannot be modeled if only undirected traffic areas are used without separate consideration of lanes.
- **Behavior.** Some of the observed behavior patterns are closely related to the fact that road users are used to driving lanes. One example is the tendency to move

to the right side of a traffic area. Although it would theoretically be possible to incorporate such behavior into other parts of the model, i.e., in the case of the right-turn tendency, into the free-flow model, a lane model is the more intuitive choice.

The central goal for the design of the MODIS lane model was a full integration into the existing navigation graph. This has the advantage that subsequent parts of the model, such as decision or reaction, are not forced to differentiate between situations with or without lanes (whereas such a differentiation is still possible if needed). To fulfil this requirement, lanes are simply added to the navigation graph as directed edges (so far, all considered edges in the graph were undirected). The nodes and edges describing a lane have to be specified in the scenario description.

In a similar fashion to the generation of the static navigation graph around obstacles (as described above or in more detail in Pascucci et al. (2015) or Timmermann (2022)), nodes of the lanes are automatically connected to the closest vertices of the surrounding polygonal boundaries or the closest other lane nodes. This way, it is ensured that undirected lanes do not cross a lane which would defy their purpose.

The following example demonstrates how to add lanes to the specification of `DocumentationDemoScenario` and briefly shows the impact using a simple situation. We consider a car starting at the upper right corner with the intention to turn left. The navigation graph for motorists without lanes was already shown in Figure A.12b. Using this navigation graph, the route search algorithm will choose the shortest possible path, thus cutting the inner curve (see Figure A.14a).



(a) Free-flow trajectory without lanes.

(b) Navigation graph with lanes.

(c) Free-flow trajectory with lanes.

Figure A.14: Comparison of navigation graph and resulting free-flow trajectories with and without lanes. Lane edges painted in dark blue, lane nodes in light blue, undirected edges in aqua, free-flow trajectory in green, and the underlying shortest path in purple.

For this example, lane edges are added in two places. The example demonstrates the two methods to add lane edges. Both methods require a `Lanes` object, which is later added to the `TrafficRegion`. The first option, as shown in lines 7–8 of Listing A.8, is

to add a single directed edge specified by two `LanePoints`. The resulting lane edge is visible in the left part of Figure A.14b. The second option (lines 16–20) is to add a list of coordinates describing continuous edges[16].

Listing A.8: Extending DocumentationDemoScenario.initTrafficNet().

```
1 ...
2 // Define traffic regions
3 TrafficRegion roadStraight = new TrafficRegion(roadStraightPoly);
4 ...
5
6 Lanes lanesRoadStraight = new Lanes(roadStraight);
7 lanesRoadStraight.addDirectedEdge(new LanePoint2D(1.6, 8.0),
8         new LanePoint2D(1.2, 3.0));
9 roadStraight.addLanesForMode(UserMode.Motorist, lanesRoadStraight);
10 ...
11
12 TrafficRegion roadRightTurn = new TrafficRegion(roadRightTurnPoly);
13 ...
14
15 Lanes lanes = new Lanes(roadRightTurn);
16 ArrayList<Point2D> directedLane = new ArrayList<>();
17 directedLane.add(new Point2D(11.0,13.5));
18 directedLane.add(new Point2D(8.5,13.5));
19 directedLane.add(new Point2D(5.5,12.5));
20 lanes.addDirectedLane(directedLane);
21 roadRightTurn.addLanesForMode(UserMode.Motorist, lanes);
22
23 // Add traffic regions to traffic net
24 this.net.addRegion(roadStraight);
25 ...
26 }
```

A.3.3 One Simulation Time Step

A sequence of many different calculations is executed for each time step. This is controlled by the method `runStep()` of `ModisController` and its content is shown in Listing A.9.

Listing A.9: `ModisController.runStep()`.

[16] Note that the demonstrated modified navigation graph is not suitable for practical purposes. A car intending to turn right would not find a path without specifying an additional lane for that relation.

```
1     protected void runStep() {
2
3         // remove all forces from the previous time step
4         for (User u : usersInScenario) {
5             u.clearForces();
6         }
7
8         // Update orientation of the traffic users according to the
9         // direction of movement within the last time step (moving the
10        // users has only an effect on the users position but not their
11        // orientation)
12        this.orientate();
13
14        // Compute all the driving forces for the users, which do not
15        // depend on conflicts, etc.
16        this.computeDrivingForces();
17
18        // update the perception of the users
19        this.updatePerceivedUsers();
20
21        // Risk assesment, if enabled
22        this.calculateRisk();
23
24        // MODIS extensions to the classical social force model:
25        // Classify the situation for each user active user
26        // The classification of the situation includes two steps:
27        //  1. detect conflicts
28        //  2. classify conflicts
29        this.detectConflicts();
30
31        // Evaluate detected conflicts in respect to traffic rules
32        this.checkTrafficRules();
33
34        // Classify the current situation
35        this.classifySituation();
36        this.adaptUserStatusToClassification();
37
38        // determine the behavior (like aggressive, defensive,
39        // do-nothing) for each user in a conflict situation
40        this.decide();
```

```
41
42          // calculates the influence of traffic lights
43          this.handleTrafficLights();
44
45          // React on conflicts based on the situation classification and
46          // the determined behavior. The reaction implies additional
47          // forces for the conflicting users or the recalculation of the
48          // trajectory
49          this.react();
50
51          // adapt the desired speed of the motor vehicles according to
52          // the properties of the traffic region
53          this.setDesiredSpeed();
54
55          // move the users according to the determined forces
56          // in other words: solve the equations by time integration
57          // scheme
58          this.move();
59
60          // check, whether collisions occurred after moving the users
61          this.checkForCollisions();
62
63          // generate new users according to:
64          //  - the traffic generator for periodically generated users
65          //  - single users whose in_time is below current time step
66          //    for the first time
67          // activate the generated users and add them to the simulation
68          this.enterUsersIntoScenario();
69
70          // actually, the evaluate method is used for some postprocessing
71          //  - record the walked trajectory (stored for each user)
72          //  - record the history in field stat (ModisStatistics)
73          //  - update stops for public transportation systems, and do some
74          //    settings for passengers
75          this.evaluate();
76
77          // automatic rerouting (not as an reaction, but as an correction)
78          this.automaticRerouting();
79
80          // update current simulation time
```

```
81          super.stat.t += super.stat.dt;
82          super.stat.nt++;
83      }
```

Most of the methods called in `ModisController.runStep()` iterate over all road users currently in the system. This means that parallelization can be easily achieved in each of these methods. In many of the methods, a generic interface is used. The following subsections explain the most important of the methods called here and, where applicable, the interfaces used. A more abstract representation of the simulation logic in a simulation time step is shown in Figure A.15.

A.3.3.1 Orientation

In the x-y-plane, the body orientation of a road user is the current tangent to the curve defined by the desired trajectory. At the beginning of a time step, `ModisController.orientate()` computes the road user's body orientation according to the movement from the previous time step. Edge cases without previous movement are also handled. The correct body orientation is important for several direction-dependent operations later in the time step, especially when determining the field of view for the perception model.

A.3.3.2 Driving Forces

Recall that the MODIS framework extends the dynamics of the Social Force Model: The overall approach is maintained that the movement of road users is governed by an impulse (or driving term) as well as repellent forces from other road users and obstacles. This general idea manifests in the interface `DynamicsModel`, which specifies three methods for calculating the three forces. `ModisController.computeDrivingForces()` calls the method for computing the driving term for each road user in the system. Note that the implementation of `DynamicsModel` originates from the work of Höcker (2010) and is called `SFMHoecker`. While the different reaction mechanisms are presented in Section A.3.3.6, the following paragraph describes the driving term.

Implementation of the Driving Term. In the absence of other conflicts, the driving term is responsible for any motion. It is a two-dimensional force that makes a road user move towards their destination while accelerating to a desired velocity along the path. Common formulations of the Social Force Model include a driving term $\boldsymbol{f}_i^{\mathrm{SFM}}(t)$ that points from the current position of the road user i towards their destination.

A core paradim of MODIS consists in splitting the driving term $\boldsymbol{f}_i^{\mathrm{SFM}}(t)$ into a tangential $\boldsymbol{f}_i^{\parallel}(t)$ and a radial part $\boldsymbol{f}_i{}^{\perp}(t)$, i.e.,

$$\boldsymbol{f}_i^{\mathrm{SFM}}(t) = \boldsymbol{f}_i^{\parallel}(t) + \boldsymbol{f}_i{}^{\perp}(t).$$

With this approach, the tangential part is exclusively responsible for regulating the

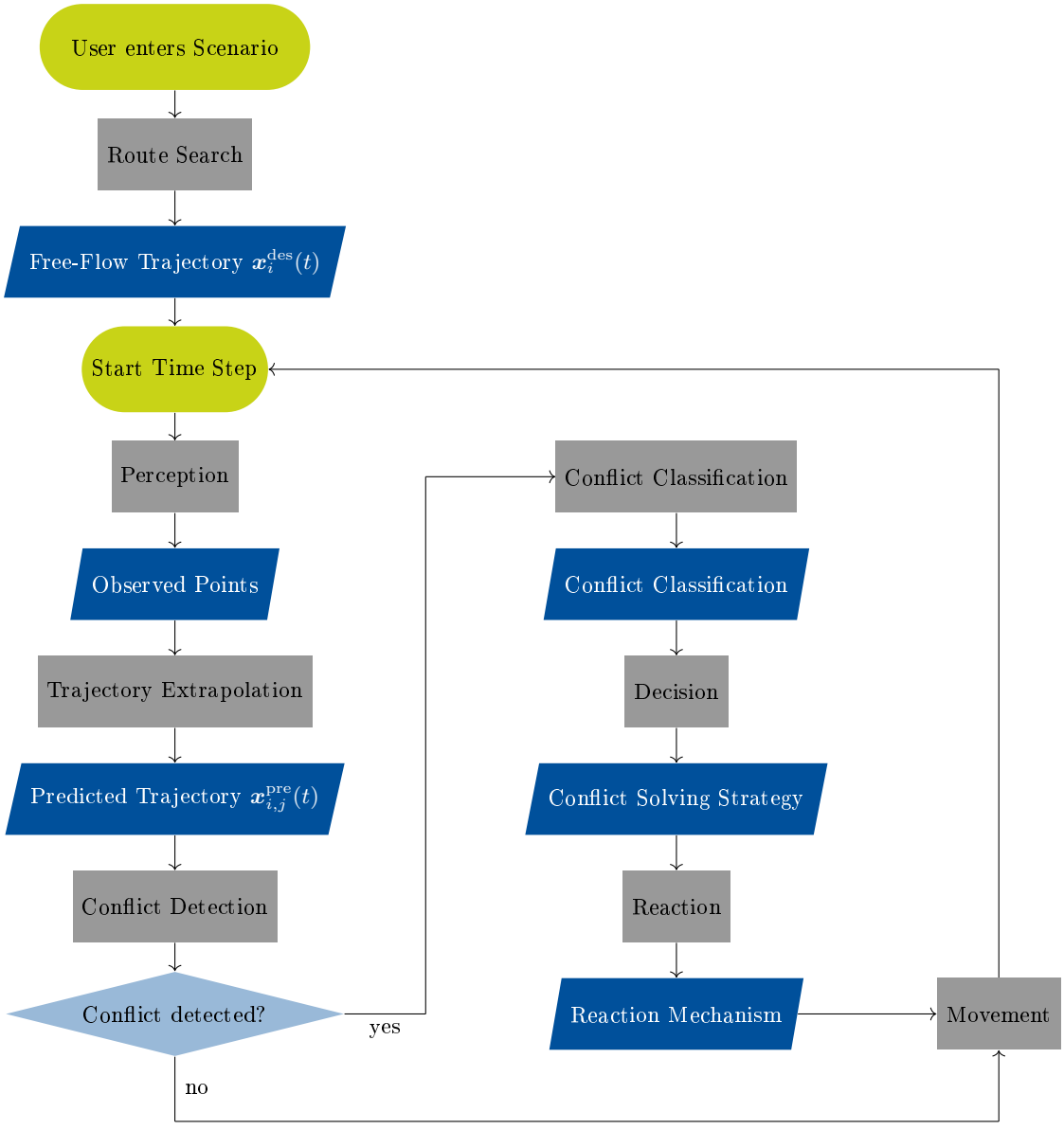


Figure A.15: Flow chart for one simulation time step.

absolute value of the velocity and makes the vehicle accelerate to a desired speed v_i^0, while the radial part controls that the vehicle moves along the desired path from the desired trajectory $\boldsymbol{x}_i^{\text{des}}(t)$.

- **Tangential Term.** The calculation depends on the situation and the respective traffic mode. For example, depending on the situation, the tangential component for motor vehicles can be calculated by a car-following model or by decelerating towards a braking a point (see also Section A.3.3.6).

- **Radial Term.** The radial component ensures compliance with the desired trajectory $\boldsymbol{x}_i^{\text{des}}(t)$. It acts as a centripetal force that is perpendicular to the current direction of motion, pointing towards the center of the osculating circle of the curve. Its absolute value depends on the absolute value of the velocity $|\boldsymbol{v}_i(t)|$ and the curvature $\kappa_i(t)$ of the trajectory (for a clothoid, the curvature at each point of the curve is known). More precisely, assuming unit mass, it holds

$$|\boldsymbol{f}_i^{\perp}(t)| = |\boldsymbol{v}_i(t)|^2 \kappa_i(t).$$

  The radial force $\boldsymbol{f}_i^{\perp}(t)$ has a second modeling role in MODIS, as it can be limited. The consequence is that road users cannot follow arbitrarily tight curve radii. The particular lower limit depends on the respective traffic mode (e.g., it exists for cars but not for pedestrians). Furthermore, the (predicted) radial force of a planned trajectory can be taken into account when determining an evasive trajectory (see also Section A.3.3.6).

A.3.3.3 Perception

Perception refers to the road user's ability to detect other road users within an area around the current position (the currently perceived area). It is implemented in `Modis-Controller.updatePerceivedUsers()` and uses the interface `PerceptionModel`. This partly differs from SFM dynamics where another user is perceived if they have some form of influence (see, e.g., Yang et al. (2014) for more details).

Perceived Area. In MODIS, the area that is perceived by a road user i is modeled as a circle segment which depends on the current position of the road user and their body orientation. More precisely, the circle segment is characterized by its center located in the point of view $\boldsymbol{p}_i^{\text{p}}$ of the road user[17], the perception radius r^{p}, and the field of view angle ϕ^{p} (centered around the body orientation). Note that, for simplicity, road users are fully perceived if they are only partially located within the perceived area. This general perception model is extended by introducing a visual height of objects and road users.

[17] The exact location of the point of view relative to the road user's body dimensions depends on the traffic mode and its implementation.

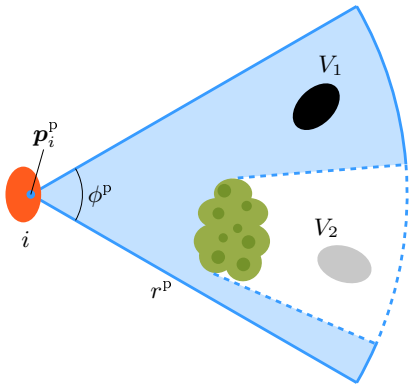


Figure A.16: Illustration of the perception model.

Illustration. The concept is illustrated in Figure A.16 where the perceived area is highlighted in blue. Note that parts of the perceived area may be obscured by obstacles. In the example shown, road user V_1 is perceived by i, while V_2 is obscured and thus not perceived by i.

However, if visual height is taken into account (e.g., V_2 is higher than the obstacle), V_2 may be visible. The implementation of such a modeling of occluded perception considering static and dynamic obstacles is discussed by Meier (2019).

A.3.3.4 Conflict Detection and Situation Classification

In the MODIS framework, conflict detection and situation classification is based on predicting road users' movements within the perceived area. This is implemented in the method `ModisController.detectConflicts()`, which uses the interface `ConflictAlgorithm`. The class `LoopConflictDetection` is used for prediction: It is an implementation of `ConflictAlgorithm` that uses Lagrange polynomials to extrapolate trajectories.

Movement Prediction. Another core ingredient of MODIS is the prediction of other vehicles' trajectories: If another road user j is within the perceived area of road user i, i tries to anticipate their behavior by observing j. Discrete-time observed positions $\boldsymbol{p}_j(t_k)$ are extrapolated spatially and temporally to predict the evolution of j's trajectory. This prediction, however, is not meant to be as accurate as possible. It is intended to represent the perception of humans who, in reality, also need time to detect changes in direction or speed. It can be regarded as a feature of MODIS' prediction model that some situations also lead to unrecognized or falsely detected conflicts.

Lagrange Polynomials. In geometric modeling, a variety of methods is available for the extrapolation of a trajectory from discrete support points. MODIS uses Lagrange polynomials for this purpose[18].

[18] Basic material on Lagrange polynomials can be found in the literature, e.g., Hoschek & Lasser (1992), Farin (1994). Their specific application in MODIS is also presented in Timmermann (2022, Section

Based on the Lagrange polynomials for spatio-temporal extrapolation of motions, the time-dependent predicted trajectory $\boldsymbol{x}_{i,j}^{\mathrm{pre}}(t)$ for the road user j (as predicted by i) is given by

$$\boldsymbol{x}_{i,j}^{\mathrm{pre}}(t) = \sum_{k=0}^{n} \boldsymbol{p}_j(t_k) L_k^n(t), \tag{A.3.2}$$

where $\boldsymbol{p}_j(t_k)$ is the observed position of $\boldsymbol{x}_{i,j}^{\mathrm{pre}}(t)$ at time t_k and $L_k^n(t)$ is the k-th Lagrange polynomial of degree n[19].

Data Set Reduction. The number of observation points is typically larger than $n+1$ which are required for the Lagrange extrapolation. To reduce the data set, the Ramer-Douglas-Peucker algorithm (RDP) (see Ramer (1972)) is implemented. This algorithm originates from the fields of computer graphics and cartography and is used to simplify polygonal curves while preserving their rough shape.

The RDP algorithm requires a tolerance criterion $\epsilon > 0$ as its sole input parameter. It connects all observed points to form a polygon curve and eliminates those which do not exceed a distance of ε from the new polygon curve if removed from the data set. More details can be found in Timmermann (2022).

If after the reduction by the RDP algorithm a larger number of points is available than the maximum curve degree n allows, only the most recent $n+1$ points are considered and the remaining points are discarded for the extrapolation. From these the anticipated trajectory $\boldsymbol{x}_{i,j}^{\mathrm{pre}}(t)$ is calculated according to (A.3.2).

Conflict Detection. Based on the predicted trajectories, the road user i detects potential conflicts with other perceived road users. For this purpose, the time interval $[t_n; t_n + t_{\mathrm{hor}}^{\mathrm{c}}]$ is considered, where t_n is the last observation time (which usually corresponds to the current simulation time t, unless the other road user is hidden at time t, but was previously perceived) and $t_{\mathrm{hor}}^{\mathrm{c}}$ is the temporal horizon for conflict detection. In the above time interval, if the minimum distance between the desired trajectory of i and the predicted trajectories of the other road users falls below a certain threshold, a conflict is detected.

Situation Classification. As a next step, the current situation is classified so that each active user has both a situation classification (stored in `User.situation_Classification`) and a user status (`User.userStatus`):

- User status is a value from the enum `UserStatus`. Possible values are listed and

4.2.2.).

[19] Note that higher degree Lagrangian polynomials tend to oscillate undesirably (see, e.g., Farin (1994)), so the degree n must be small. Experiments have shown that a degree of $n = 3$ yields plausible extrapolated trajectories for points observed each 0.5 s.

explained in Table A.2[20].

- Situation classification is a value from the enum `SituationClassification`. Possible values are listed and explained in Table A.3.

User Status	Description
`FREE_FLOW`	User has not detected a conflict with any other road user and follows their desired trajectory.
`SOLVING`	User has detected a conflict with at least one other road user and is actively solving it.
`WAITING`	User purposely has a velocity of 0. Can imply `SOLVING`, but not necessarily (e.g., when waiting at traffic light).
`NON_FREE_FLOW`	User has detected a conflict with at least one other road user, but is currently not actively solving it.
`COLLISION`	Distance to another road user is 0 or less.
`INACTIVE`	User with `t_0` less than the current time (see Section A.3.2.1).

Table A.2: Overview of possible values for user status.

Number of Conflicts	Time to Collision	Situation Classification	Description
1	$> t_{\text{SR}}$	General: `M`$_i$`_M`$_j$, e.g., `PED_PED`, `CYC_GROUP`, ...	Conflict with exactly one other road user
$1 < n_{\text{conf}} \leq n_{\max}$		General: `M`$_i$`_M`$_j$`s`, e.g., `PED_PEDs`, `CYC_CARs`, ...	Conflict with multiple road users of the same traffic mode
		General: `M`$_i$`_MULT`, e.g., `PED_MULT`, `CAR_MULT` ...,	Conflict with multiple road users of multiple traffic modes
$> n_{\max}$		`CROWDED`	Situation involving too many road users for sensible anticipatory behavior
irrelevant	$\leq t_{\text{SR}}$	`SHORT_RANGE`	Ad hoc situation
0	-	`NO_CONF`	No conflict
> 0	> 0	e.g., `OTHER_LANE`, `FOLLOWING`	Situations in which a deliberate undercutting of the safety distance can be permitted, e.g., in the presence of lanes. Determination based on further criteria required.

Table A.3: Overview of situation classifications. Classifications printed in `typewriter font`.

A.3.3.5 Decision

The modular structure of MODIS enables decision-making based on any criteria. The decision level has the task of determining the strategy for the subsequent reaction in a conflict situation. A distinction is made between offensive, defensive, and (initially) no

[20] The status `NON_FREE_FLOW` is of technical nature: This may be the case if the time to conflict is large or if conflict detection does not return a conflict status due to a past reaction, but it is not guaranteed that the conflict will not occur again. In the latter case, the status must be explicitly terminated.

reaction. At this stage, no directly visible change in behavior takes place. The decision is implemented in the method `ModisController.decide()`.

Pascucci et al. (2018) examine empirical data to determine parameters that drive the choice of a strategy. Based thereon, a stochastic decision model is implemented in MODIS for conflict situations between pedestrians and motor vehicles. Specifically, the significant parameters are the predicted minimum distance to the conflicting road user, the associated remaining time until the minimum distance, the predicted temporal distance of arrival at the intersection of the trajectories, and the acceleration of the pedestrian at the time of decision. Based on these four parameters, a logistic regression model is formulated to determine individual probabilities for all strategies common to each conflict type[21].

A.3.3.6 Reaction

MODIS specifies the interface `ReactionAlgorithm` to implement reactions for conflict situations. This interface consists of only one method, namely `ReactionAlgorithm.reactOnSituation(User user)`. As the name indicates, this method is intended to compute a reaction appropriate to the current situation for a given road user. From a general perspective, `ModisController.react()` iterates over all active users in the system and for every user in a state of conflict, `ReactionAlgorithm.reactOnSituation()` is called.

Currently there are multiple implementations of this interface. The active one is defined in `ModisConfiguration`, which in the default setting is the implementation `DecisionBasedReaction`. Note that for some situations there are calls in between different implementations. The following implementations of `ReactionAlgorithm` are available:

- `DecisionBasedReaction`: This implementation currently serves as the entry point for reactions. It maps the decision taken in the decision step (see Section A.3.3.5) to the corresponding reaction mechanism.
- `MultiLayerReaction`: Contains implementations of reaction mechanisms that solve conflicts by modifying the desired trajectory.
- `ForceBasedReaction`: Implements all reaction mechanisms which involve additional forces.
- `SFMBasedReaction`: Here, the logic from the original SFM is implemented. It is active when the long range reaction has not solved the conflict or a conflict is detected too late (these are also referred to as *ad hoc situations*).

[21] Note that Pascucci et al. (2018) denote the three possible strategies *Aggressive*, *Prudent*, and *No Reaction*. Moreover, they find that all three strategies are observed in `PED_CAR` conflicts, whereas only defensive or no reactions occur in `CAR_PED` conflicts.

- `TrafficRuleBasedReaction`: This can be understood as an intermediate layer: In scenarios where traffic rules are to be observed, it is checked whether right of way applies in a given situation and then maps to the reaction mechanism that implements appropriate behavior.
- `SFMOnlyReaction`: This ensures that only the classic SFM is used, resulting in roughly unmodified SFM (except for movement along trajectories). This implementation is not active in the default configuration; it is intended for test or comparison purposes only.

Note that all long range reaction mechanisms that model anticipatory driving are implemented either in `MultiLayerReaction` (reaction mechanisms based on trajectory re-routing) or `ForceBasedReaction` (reaction mechanisms based on forces). In the following, we provide more details on these.

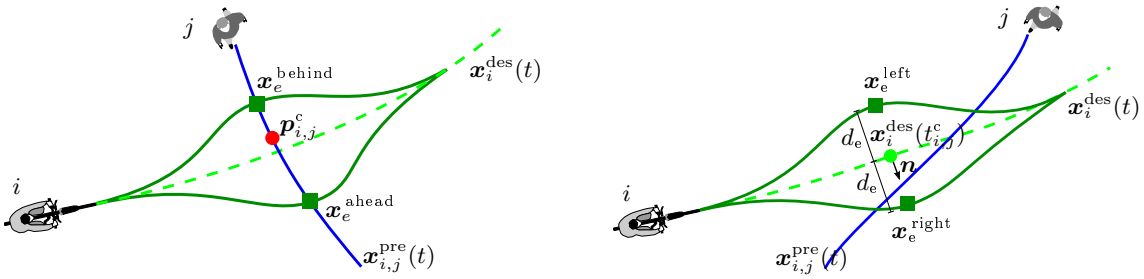
Reaction Mechanisms Based On Trajectory Re-Routing. `MultiLayerReaction` contains reaction mechanisms that deliberately reroute the current desired trajectory in order to solve a conflict situation directly. In can be regarded as a one shot mechanism that determines a new trajectory which has no conflict with the other road users' predicted trajectories.

This reaction mechanism requires predicted trajectories of conflicting road users. Based thereon, a fixed point in a safe distance is determined. This point is additionally inserted into the desired path W (see (A.3.1)); as before a desired trajectory is calculated from this path (see A.3.2.2 for details). Note that, from a modeling perspective, no additional reaction force is applied, but rather the driving term is modified (see Section A.3.3.1)[22].

This general idea of determining such a fixed point is implemented in two ways: The trajectory can be adjusted spatially and temporally by re-routing through a safe point, or only temporally by adding a waiting point. The following paragraphs highlight the ideas, more details (in particular on the involved parametrizations) can be found in Timmermann (2022).

- **Re-Routing the Desired Trajectory Through a Safe Point.** The calculation of possible evasion points, as implemented for various user modes in `MultiLayerReaction`, differs depending on the relative body angle of the conflicting users. For lateral conflicts, evasion points can be determined directly on the predicted trajectory $\boldsymbol{x}_{i,j}^{\mathrm{pre}}(t)$. This is not possible for frontal and rear conflicts, as this would not always guarantee sufficient distance. The difference is exemplified in Figure A.17.
  - *Lateral Conflicts.* Two evasion points are calculated based on the predicted trajectory of the conflicting user and the conflict point. Essentially, these are

[22] To avoid a collision caused by an incorrectly estimated fixed point, a successive re-evaluation of the reaction is possible, but unlike force-based reactions, it is not mandatory.



(a) Lateral conflict (b) Frontal conflict

Figure A.17: Calculation of possible evasion points. Rearwards conflict analogous to frontal conflict.

found by going forward or backward (for a specified time[23]) on the predicted trajectory.

– *Frontal Conflicts.* The above approach does not always solve frontal or rearwards conflicts. An alternative is implemented as shown schematically in Figure A.17b for a frontal conflict: Evasion points are simply calculated by adding (or subtracting) a safe distance to (or from) the predicted conflict point, perpendicular to the direction of motion.

– *Choosing an Evasion Point.* The methods above require a road user to choose between two possible evasion points (except in cases where the choice is further constrained by the environment). Inspired from empirical observations of cyclist behavior, MODIS takes the maximal centripetal acceleration required to follow a possible evasion path as the decision criterion.

  Specifically, MODIS assumes an individual maximum comfortable centripetal acceleration for each user[24]. The following strategy uses this value to find acceptable evasion trajectories and is already implemented in several `MultiLayerReaction.reactOnCyc...()`-methods:

  * Calculate trajectories trough all evasion points.
  * Calculate required maximum centripetal acceleration under the assumption of free-flow for each evasion trajectory candidate (this also means that acceleration to the desired speed must be considered if the current speed is lower).
  * Select the evasion point associated with the lowest required centripetal acceleration, if it does not exceed the maximum comfortable centripetal acceleration.

[23] Such a parameterization as temporal distances entails a velocity dependence of the evasion distances. This corresponds to the empirical observations, according to which in case of crossing trajectories, road users tend to prefer evasion distances which grow with the predicted velocity of the conflicting user.

[24] The current default values have been calibrated using sparse data and may be improved in the future.

  - If all required centripetal accelerations exceed the maximum comfortable centripetal acceleration, calculate the deceleration required for maintaining a comfortable centripetal acceleration and choose the evasion trajectory with the least required deceleration.

- **Adding a Waiting Point.** While moving along the same path, a trajectory can also be changed by a targeted braking maneuver. Here, the goal is to add a waiting point, so that the conflicting road user may pass the conflict point safely. This is implemented in `MultiLayerReaction.calculateSmoothDeceleration()`.

  The waiting point, denoted by $\boldsymbol{x}_{\text{stop}}$ in the following, can either be given by the environment (e.g., road markings or change of surfaces) or estimated based on the predicted trajectory $\boldsymbol{x}_{i,j}^{\text{pre}}(t)$ and the safety distance $d_{i,j}^{\text{s}}$. An example of the waiting point estimation is shown in Figure A.18.

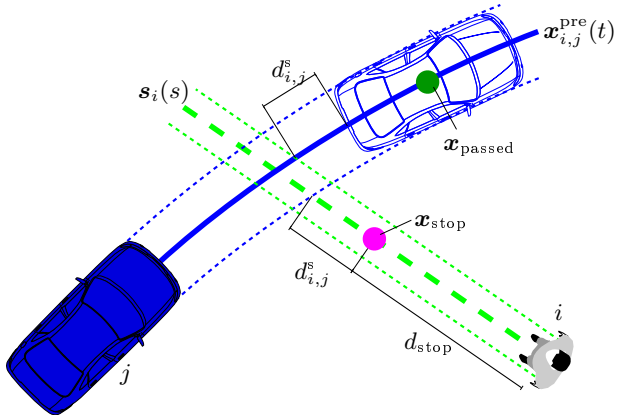


Figure A.18: Estimating a waiting point based on trajectory prediction.

In this example, the waiting point is determined by first estimating the crossing point between the trajectories. Then, the body dimensions as well as the desired safety distance $d_{i,j}^{\text{s}}$ are projected back along the trajectory of i.

In the implementation of the targeted braking maneuver, a constant deceleration is assumed. For the choice of this deceleration, two cases need to be distinguished:

(i) The predicted time until j passes is small enough that i only needs to decelerate, but not come to a complete stop. For this case, the required braking acceleration is determined so that the desired distance can be maintained.

(ii) The predicted time until j passes is so long that i must decelerate to a complete stop and, if necessary, wait for j to pass. For this case, the required braking acceleration for a complete stop at the point $\boldsymbol{x}_{\text{stop}}$ is determined.

The determined braking acceleration for the targeted braking maneuver is applied by projecting it onto the tangential part of the driving term (see Section A.3.3.2).

The above case distinction is implemented by first calculating the constant braking

acceleration a_{stop} required to come to a complete stop from the current velocity v_i exactly at x_{stop}. The corresponding required time span is computed and compared to the predicted time needed for the conflicting road user to pass the conflict point. More details on the computation of the involved accelerations can be found in Berkhahn et al. (2022).

Reaction Mechanisms Based On Forces. Originating from the SFM, vehicles move through a given space due to forces. Next to the driving term (see Section A.3.3.1) which makes a road user move along a desired trajectory, additional forces can be added to perform evasive maneuvers. These are mathematically added to the driving term and can be regarded as long range reaction forces that represent anticipatory driving (opposed to short range reactions in, e.g., car-following models).

In `ForceBasedReaction`, four different long range reaction forces are currently implemented. Note that these forces are mostly suited to specific traffic situations (mostly pedestrians), i.e., not well suited for a general approach. While a long range reaction force is active, the parallel component of the driving term is inactive. Otherwise, the two approaches could cancel each other out, leading to a possible acceleration back to the desired velocity. The four forces are:

- **Safe Zone Force.** This is a defensive reaction mechanism that is applied when the following conditions are met:
  - The traffic user under consideration is a pedestrian who is in conflict with a less vulnerable traffic mode (e.g., cyclist or car).
  - The user is currently in a part of the traffic area where the conflicting user is not allowed to be (e.g., a sidewalk). This area is considered the user's safe zone.
  - The desired trajectory will lead the user out of the safe zone through a mixed zone, where the impending conflict is expected.
  - The user decides to react defensively.

  Empirical observations show that in such cases users tend to stay in the safe zone until the conflict is resolved, but still try to move as far as possible towards their destination.

  Therefore, the Safe Zone Force gradually deflects the road user from the border between safe and mixed zone. This leads to a movement parallel to this border until the conflict is solved. Depending on the location of the destination, waiting at the border before entering the mixed zone can also occur.

- **Defensive Long Range Reaction Force.** This force can be understood as a generalization of the Safe Zone Force. Instead of the border of a safe traffic area, a certain safe distance is kept from the predicted trajectory of the conflicting user.

Thus, the restrictions regarding a specific zone as well as the combinations of traffic modes do not apply to the Defensive Long Range Reaction Force.

- **Offensive Long Range Reaction Force.** This force also uses the predicted trajectory of the conflicting traffic user. Instead of moving parallel to it and letting the other traffic user pass, it is directed towards the predicted trajectory so that the area around the conflict point is left before the conflicting user approaches[25].

- **Trajectory-Based Evasion Force.** This force models the behavior in conflict situations between pedestrians. It is often characterized by a simultaneous gradual evasion by the involved road users. The simultaneous reaction leads to constantly changing predictions; the force models a deflection away from the vicinity of the predicted conflict point. With some restrictions, the trajectory-based evasion force can also be applied to cyclists.

A more detailed description of the formulation of these forces and their interpretation can be found in Timmermann (2022).

A.3.3.7 Moving the Road Users

The actual movement of a traffic user is implemented in the method `ModisController.move()` which iterates over all traffic users and determines the position and velocity at the next time step in exactly one of the following ways:

- If an external trajectory is supplied (see also Section A.2.1.3), position and velocity are determined from it.

- If the user is involved in a collision, it does not move, i.e., velocity and acceleration are set to 0 so that the users remains at the same position.

- If none of the above applies, the movement is obtained via numerical integration of the resulting acceleration.

The numerical integration is specified in the interface `NumIntegrationAlgorithm` which contains one method, namely `NumIntegrationAlgorithm.move(user: User, dt: double)`. This method updates position and velocity of a road user for the next time step.

By default, MODIS uses the two-step Adams–Bashforth method, a standard multistep method for approximating the initial value ODE problem. Two-step in this context means that the next position is calculated from the current position and velocity and that of the previous time step. For this purpose, `AdamsBashforth2OIntegration` implements the interface `NumIntegrationAlgorithm`; other numerical schemes can be implemented in a similar way.

[25] It is only possible to solve a conflict situation using this mechanism if there is enough time to cross the predicted trajectory. Therefore, the decision model needs to take this into account.

A.3.3.8 Evaluation and End of Time Step

The end of a time step is split into several methods: `ModisController.checkForCollisions()`, `ModisController.enterUsersIntoScenario()`, `ModisController.evaluate()` and `ModisController.automaticRerouting()`. These methods perform the following tasks:

Checking for Collisions. Caused by misperception, wrong decisions due to misjudgement of the situation, or other reasons, the previous call of the `move()`-method can result in collisions. A collision is defined as a situation where the distance between two road users is equal to or less than 0. If that occurs, the following is executed:

- Collided road users are stopped immediately by setting their speed and acceleration to zero.
- They are added to the controller's `CollisionManager`.
- The `CollisionManager` specifies a time after which the collided users are removed from the traffic system. This time can be deterministic or random.

Add Scheduled Users to the Scenario. This method actually serves as a preparation for the next time step. `ModisController.enterUsersIntoScenario()` is called to check if new road users want to enter the scenario. This can result from one of the following two cases:

- There is at least one single road user in `ModisController.usersNotInScenario()` whose `in_time` is smaller than the current time.
- The traffic generator, i.e., the call of `ModisController.tg.generateTraffic(currentTime)` returns a non-empty set of road users.

Both types of road users are then added to `ModisController.usersInScenario` and considered in the next time step.

Evaluating Arrival at Destinations. `ModisController.evaluate()` iterates over all simulated road users, mainly checking if they reached their destination with the previous movement.

Although not discussed in this chapter, the framework provides the option that road users have more than one destination. This sequence of destinations is encapsulated in an object described by the `Trip` class, as introduced by Diekmann & Schiermeyer (2018). Originally motivated to model transportation of road users by other road users (such as buses or trams), the trip concept was designed in a more general way: It allows the representation of road users who take a break at a certain point and then continue the trip to another destination.

Following this concept, each user's trip has at least one element and a user has a current destination (i.e., the current trip element's destination) at any given time. Given this context, the main purpose of `ModisController.evaluate()` can now be more clearly defined as choosing the appropriate action upon arrival of a road user at their (intermediate) destination.

A road user arriving at their trip's last destination is immediately removed from the system. This includes removal from `ModisController.usersInScenario` as well as from all other road user's lists of perceived users, in order to avoid wrong detection of conflicts with users that have already left the system. Behaviors for other types of destination, such as stopping the road user's movement for resting or calculating a new trajectory to enter another mode of transportation, are described in more detail in Diekmann & Schiermeyer (2018).

Apart from road users that arrived at their last specified destination, the `CollisionManager` is queried for collided users whose removal time is reached. These are subsequently also removed from `ModisController.usersInScenario`.

Re-routing the Desired Trajectory. Caused by force-based reactions or numerical inaccuracy during the `move()`-method (i.e., the numerical solving of the ODE), a road user can leave their desired trajectory over the course of several simulation time steps.

As explained in Section A.3.3.2, the direction of the tangent at the closest point of the desired trajectory majorly determines the movement. A small deviation from the desired trajectory does not notably affect the simulation results and is therefore tolerated (up to a certain threshold). As such, `ModisController.automaticRerouting()` iterates over all simulated road users and checks the current distance to the desired trajectory. If the distance exceeds the given threshold value[26], a trajectory recalculation is triggered. The rerouting algorithm follows the same principle as described in Section A.3.2.2; in contrast to the initial route search, it uses only the remaining nodes of the path as well as eventually calculated evasion points.

A.4 Exercise: Implementing a Custom Reaction

In this exercise, we will learn how to implement a custom response. For demonstration purposes, we will use a simple behavior that is neither necessarily realistic nor guaranteed to resolve a conflict. The intended behavior is that any road user who detects a conflict will slow down with a certain probability. Since each user decides this independently, a collision may occur if both users decide not to wait (it is also possible that both stop). To implement this custom reaction, two main steps are required:

[26] The threshold value is set in `ModisParameter.AUTOMATIC_REROUTING_DISTANCE_THRESHOLD`. Depending on various factors such as the scenario or the traffic modes involved, a sensible threshold value was observed in the range between 0.5 and 1.5 m.

(i) **Implement the Decision.** When making a decision, each user randomly decides whether to resolve the conflict by slowing down (decision "defensive") or not (decision "no reaction").

(ii) **Implement the Reaction.** The sole purpose of the reaction is to check whether a user has decided to react defensively in the event of a conflict. If this is the case, a fixed negative acceleration is applied until a full stop is reached.

Finally, we need to activate our new decision and reaction implementations to be used by the simulation.

A.4.1 Decision

We start by creating a new class that extends the abstract class `DecisionAlgorithm`:

Listing A.10: New decision class.

```
1 package modis.model.situation.decision;
2
3 import modis.control.ModisController;
4 import modis.model.traffic.User;
5
6 public class RandomWaitDecision extends DecisionAlgorithm {
7
8     double waitProbability = 0.5;
9
10    public RandomWaitDecision(ModisController controller) {
11        super(controller);
12    }
13
14    @Override
15    public void decideOnSituation(User user) { }
16 }
```

In addition to the necessary implementations given by the abstract parent class, we introduce a double value to be used as the probability that a user will slow down. Technically, this code is already compileable and can be used for simulation. However, the simulated road users just do not show a reaction yet, because no decision is made. To achieve this, we need to fill the method `decideOnSituation()` with meaningful content:

Listing A.11: Extending `decideOnSituation()`.

```
1 public void decideOnSituation(User user) {
2     if (user.situation_Classification == Classification.NO_CONF) {
3         return;
```

```
4     }
5     ConflictBehaviourType behaviour = sampleRandomBehaviour();
6     user.decisionHistory.add(new Decision(behaviour, controller.stat.t,
7             user.conflictingUsersThisTimestep.size()));
8 }
9
10 private ConflictBehaviourType sampleRandomBehaviour() {
11     double r = Math.random();
12
13     if (r < waitProbability) {
14         return ConflictBehaviourType.Defensive;
15     }
16     return ConflictBehaviourType.None;
17 }
```

For clarity, we write the sampling of a random number in a separate method called `sampleRandomBehaviour()`. Here, a uniform random number is sampled first. We compare it with the reaction probability and return either `ConflictBehaviourType.Defensive` or `ConflictBehaviourType.None`.

In our implementation of `decideOnSituation()`, we first make a check if the user is actually in a state of conflict. If this is not the case, we end the method without any decision. Otherwise, we sample a decision by calling `sampleRandomBehaviour()`. To make the decision available for the subsequent reaction, it needs to be encapsulated in a `Decision` object, which is added to the road user's decision history. A decision object needs three parameters: The actual `ConflictBehaviourType`, which we already chose, the current time, which we can get from the `ModisController`, and the number of other road users currently in conflict with the current user.

This logic now works as a somewhat sensible input for implementing a reaction. Still, we can make one simple improvement. The way `decideOnSituation()` is implemented in Listing A.11, a user in a state of conflict makes a new decision in each time step, which means that the behaviour can change. To avoid this, we make the simple assumption that once a decision has been taken, it does not change anymore, thus there is no need to make a new decision once the user's decision history is non-empty:

Listing A.12: Improvement of `decideOnSituation()`.

```
1 public void decideOnSituation(User user) {
2     if (user.situation_Classification == Classification.NO_CONF) {
3         return;
4     }
5     if (!user.decisionHistory.isEmpty()) {
6         return;
```

```
7     }
8     ...
9 }
```

The full implementation of `RandomWaitDecision.java` is printed in Appendix A.5.4.

A.4.2 Reaction

Our custom reaction starts with a new class implementing the interface `ReactionAlgorithm`:

Listing A.13: Implementing the interface `ReactionAlgorithm`.

```
1 package modis.model.situation.reaction;
2
3 import modis.control.ModisController;
4 import modis.model.traffic.User;
5
6 public class RandomWaitReaction implements ReactionAlgorithm {
7
8     double b = 1;
9     ModisController controller;
10
11    public RandomWaitReaction(ModisController controller) {
12        this.controller = controller;
13    }
14
15    @Override
16    public void reactOnSituation(User user) { }
17 }
```

Similarly to the decision, this is almost only the bare minimum required to make the class compileable. Furthermore, we added a parameter `b` to model the deceleration a user will perform. The implementation as it stands will not do anything yet, so let us start by extending the `reactOnSituation()` method:

Listing A.14: Extending the `reactOnSituation()` method.

```
1 public void reactOnSituation(User user) {
2     if (user.situation_Classification == Classification.NO_CONF) {
3         return;
4     }
5     if (!user.decisionHistory.isEmpty() &&
6             user.decisionHistory.last().behaviour
7             == ConflictBehaviourType.Defensive) {
```

```
8         // Here we will call the actual deceleration:
9         decelerate(user);
10        user.userStatus = UserStatus.SOLVING;
11    }
12 }
```

What is missing is the actual deceleration. We implement it in a separate method `decelerate()`, which is called by `reactOnSituation`:

Listing A.15: New method `decelerate()`.

```
1 private void decelerate(User user) {
2     User.Force reactionForce =
3         new User.Force(-b * user.e_b.x(), -b * user.e_b.y(),
4                 ForceType.REACTION);
5     Reaction reaction = new Reaction(controller.stat.t,
6             Reaction.ReactionType.DECELERATION, reactionForce);
7     user.react(user.conflictingUsersThisTimestep.first().u_j, reaction);
8 }
```

First, we define a reaction force which is the projection of the parameter b against the body orientation of the road user. Then, we create a reaction object where we specify the current time, the type of reaction and submit our reaction force.

The full implementation of `RandomWaitReaction.java` is printed in Appendix A.5.4.

A.4.3 Activate the Newly Implemented Behavior

The settings for all modularly designed algorithms of the framework can be found in the `ModisConfiguration` class. To use our newly implemented classes as the standard decision and reaction algorithms, we have to change two lines in `ModisConfiguration.reset()`, where the initialization of the respective algorithms takes place:

Listing A.16: Activating our new algorithms in ModisConfiguration.reset().

```
1 public class ModisConfiguration {
2     ...
3     public void reset() {
4         ...
5         decisionAlgorithm = new RandomWaitDecision(controller);
6         reactionAlgorithm = new RandomWaitReaction(controller);
7         ...
8     }
9 }
```

Keep in mind that with this change all conflict situations for all user modes are handled in the same way. A more realistic approach would be to define one implementation

each for both decision and reaction as an entry point which then branches to different implementations to be able to handle different types of conflict. In the existing framework this is performed in `StatisticDecisionAlgorithm` and `MultiLayerReaction`.

A.4.4 Where to Go from Here?

The new decision and reaction can be tested by creating a simple situation, e.g., two crossing pedestrians. If one simulates the situation several times, it will become clear: Even for seemingly simple situations like two crossing pedestrians, comparatively complex reaction mechanisms are required to model a reasonably realistic behavior.

A.5 Appendix: Code Examples

A.5.1 Simulation Properties File

Listing A.17: Example simulation properties xml file.

```
<?xml version="1.0" encoding="UTF-8"?>
<properties schemaVersion="1.4">
  <infrastructureID>Pockelsstrasse-Demo-Szenario</infrastructureID>
  <seed>1671345436</seed>
  <timeStep>0.1</timeStep>
  <odLocations>
    <odLocation id="0">
      <polyline>
        <point>
          <x>28.285</x>
          <y>26.78</y>
        </point>
      </polyline>
    </odLocation>
  </odLocations>
  <users>
    <user id="0" mode="Pedestrian">
      <position>
        <x>13.115</x>
        <y>46.301</y>
      </position>
      <v_0>
        <x>0.0</x>
        <y>0.0</y>
      </v_0>
```

```
      <v_max>
        <value>1.34</value>
      </v_max>
      <perception_radius>30.0</perception_radius>
      <phi_s>200.0</phi_s>
      <in_time>0.0</in_time>
      <dest_id>0</dest_id>
    </user>
  </users>
</properties>
```

A.5.2 Monte Carlo Properties File

Listing A.18: Example Monte Carlo properties xml file.

```
<?xml version="1.0" encoding="UTF-8"?>
<simulationRun>
    <n_sim>10</n_sim>
    <t_sim>600</t_sim>
    <mean_removal_time>300</mean_removal_time>
    <trafficGenerator>
        <odRelation>
            <source>0</source>
            <dest>1</dest>
            <demand>200</demand>
        </odRelation>
    </trafficGenerator>
    <safetyDistance>5</safetyDistance>
    <IDM>
        <s0>2.0</s0>
        <T>1.5</T>
        <delta>4</delta>
        <a_max>2.0</a_max>
        <b_max>3.5</b_max>
        <b_comf>1.67</b_comf>
    </IDM>
    <perception>
        <distance>
            <sigma>0.1</sigma>
            <alpha>1.0</alpha>
        </distance>
        <own_velocity>
```

```
            <sigma>0.1</sigma>
            <alpha>1.0</alpha>
        </own_velocity>
    </perception>
    <simulationProperties>Unidirectional_2_km.xml</simulationProperties>
    <outputDir>SimulationResults</outputDir>
</simulationRun>
```

A.5.3 Example ModisScenario

Listing A.19: DocumentationDemoScenario.java.

```
package demo;

import math.geom2d.Point2D;
import math.geom2d.polygon.SimplePolygon2D;
import modis.control.ModisScenario;
import modis.model.traffic.TrafficRegion;
import modis.model.traffic.UserMode;
import modis.model.traffic.infrastructure.Function;
import modis.model.traffic.infrastructure.SurfaceMaterial;

public class DocumentationDemoScenario extends ModisScenario {

    @Override
    public void initTrafficNet() {

        // Define the polygonal areas of the Scenario
        SimplePolygon2D roadStraightPoly = new SimplePolygon2D();
        roadStraightPoly.addVertex(new Point2D(0, 0));
        roadStraightPoly.addVertex(new Point2D(0, 15));
        roadStraightPoly.addVertex(new Point2D(5, 15));
        roadStraightPoly.addVertex(new Point2D(5, 0));

        SimplePolygon2D sidewalkPoly = new SimplePolygon2D();
        sidewalkPoly.addVertex(new Point2D(5, 0));
        sidewalkPoly.addVertex(new Point2D(5, 5));
        sidewalkPoly.addVertex(new Point2D(10, 10));
        sidewalkPoly.addVertex(new Point2D(15, 10));
        sidewalkPoly.addVertex(new Point2D(15, 0));

        SimplePolygon2D roadRightTurnPoly = new SimplePolygon2D();
```

```
31          roadRightTurnPoly.addVertex(new Point2D(5, 15));
32          roadRightTurnPoly.addVertex(new Point2D(15, 15));
33          roadRightTurnPoly.addVertex(new Point2D(15, 10));
34          roadRightTurnPoly.addVertex(new Point2D(10, 10));
35          roadRightTurnPoly.addVertex(new Point2D(5, 5));
36
37          // Define traffic regions
38          TrafficRegion roadStraight = new TrafficRegion(roadStraightPoly);
39          roadStraight.allowUserMode(UserMode.Motorist, UserMode.Cyclist);
40          roadStraight.material = SurfaceMaterial.light_asphalt;
41          roadStraight.function = Function.DRIVINGLANE;
42          roadStraight.name = "Straight Road Segment";
43
44          TrafficRegion sidewalk = new TrafficRegion(sidewalkPoly);
45          sidewalk.allowUserMode(UserMode.Pedestrian);
46          sidewalk.allowUserMode(UserMode.Cyclist, 0.5);
47          sidewalk.material = SurfaceMaterial.pavement;
48          sidewalk.function = Function.FOOTPATH;
49          sidewalk.name = "Sidewalk";
50
51          TrafficRegion roadRightTurn = new TrafficRegion(roadRightTurnPoly);
52          roadRightTurn.allowUserMode(UserMode.Motorist, UserMode.Cyclist,
53              UserMode.Pedestrian);
54          roadRightTurn.material = SurfaceMaterial.light_asphalt;
55          roadRightTurn.function = Function.DRIVINGLANE;
56          roadRightTurn.name = "Right Turn Road Segment";
57
58          // Add traffic regions to traffic net
59          this.net.addRegion(roadStraight);
60          this.net.addRegion(sidewalk);
61          this.net.addRegion(roadRightTurn);
62      }
63  }
```

A.5.4 Classes for Section A.4

Listing A.20: `RandomWaitDecision.java`.

```
1  package modis.model.situation.decision;
2
3  import modis.control.ModisController;
4  import modis.model.situation.Classification;
```

```
5 import modis.model.situation.reaction.ConflictBehaviourType;
6 import modis.model.traffic.User;
7
8 public class RandomWaitDecision extends DecisionAlgorithm {
9
10     double waitProbability = 0.5;
11
12     public RandomWaitDecision(ModisController controller) {
13         super(controller);
14     }
15
16     @Override
17     public void decideOnSituation(User user) {
18         if (user.situation_Classification == Classification.NO_CONF) {
19             return;
20         }
21         if (!user.decisionHistory.isEmpty()) {
22             return;
23         }
24         ConflictBehaviourType behaviour = sampleRandomBehaviour();
25         user.decisionHistory.add(new Decision(behaviour,
26              controller.stat.t,
27                 user.conflictingUsersThisTimestep.size()));
28     }
29
30     private ConflictBehaviourType sampleRandomBehaviour() {
31         double r = Math.random();
32
33         if (r < waitProbability) {
34             return ConflictBehaviourType.Defensive;
35         }
36         return ConflictBehaviourType.None;
37     }
38 }
```

Listing A.21: `RandomWaitReaction.java`.

```
1 package modis.model.situation.reaction;
2
3 import modis.control.ModisController;
4 import modis.model.dynamics.ForceType;
```

```
5 import modis.model.situation.Classification;
6 import modis.model.traffic.User;
7 import modis.model.traffic.UserStatus;
8
9 public class RandomWaitReaction implements ReactionAlgorithm {
10
11     double b = 1;
12     ModisController controller;
13
14     public RandomWaitReaction(ModisController controller) {
15         this.controller = controller;
16     }
17
18     @Override
19     public void reactOnSituation(User user) {
20         if (user.situation_Classification == Classification.NO_CONF) {
21             return;
22         }
23         if (!user.decisionHistory.isEmpty()
24                 && user.decisionHistory.last().behaviour
25                 == ConflictBehaviourType.Defensive) {
26             decelerate(user);
27             user.userStatus = UserStatus.SOLVING;
28         }
29     }
30
31     private void decelerate(User user) {
32         User.Force reactionForce = new User.Force(-b * user.e_b.x(),
33                 -b * user.e_b.y(), ForceType.REACTION);
34         Reaction reaction = new Reaction(controller.stat.t,
35                 Reaction.ReactionType.DECELERATION, reactionForce);
36         user.react(user.conflictingUsersThisTimestep.first().u_j, reaction);
37     }
38 }
```

B Companion to Chapter 4

We provide detailed tables that contain selected statistical functionals of the total loss in our different case studies. We evaluate

(i) *Expectation.* $\mathbb{E}(L)$,

(ii) *Variance.* $\mathrm{Var}(L) = \mathbb{E}(L^2) - \mathbb{E}(L)^2$,

(iii) *Skewness.* $\varsigma_L = \frac{\mathbb{E}\left[(L-\mathbb{E}(L))^3\right]}{(\mathrm{Var}(L))^{3/2}}$,

(iv) *Value-at-Risk.* $\mathrm{VaR}_p(L) = \inf\{x \in \mathbb{R} \colon P(L \leq x) \geq p\}$, $p = 0.9,\ 0.95,\ 0.99$,

(v) *Expected Shortfall.* $\mathrm{ES}_p(L) = \frac{1}{1-p}\int_p^1 \mathrm{VaR}_q(L)\,\mathrm{d}q$, $p = 0.9,\ 0.95,\ 0.99$.

These statistical functionals are presented for both unnormalized and normalized values of L. In total, we provide 18 tables, as shown in the list of tables below: The first 9 tables contain the statistical functionals for unnormalized total losses while the second 9 tables contain the results for normalized total losses.

Remarks on the Normalization. Recall that we normalize total losses in order to compare losses among fleet sizes $\rho^\Phi = 0.1,\ 0.5,\ 0.9$. More precisely, we normalize L by *100 expected insured vehicles* as follows:

- In the underlying SUMO scneario, the route files specify a number of vehicles for each *flow* belonging to the fleet Φ.
- For each traffic scenario k, we denote the sum of these values over all flows by n^k. We interpret this as the total number of insured vehicles in traffic scenario k. In our case studies, n^k takes two different values corresponding to the good and bad scenarios:
$$n^k = \begin{cases} n_g, & k = 1, \dots, 50, \\ n_b, & k = 51, \dots, 100. \end{cases}$$
- We denote the total number of insured vehicles by n^Φ. It is given by $n^\Phi = \sum_{k=1}^K \mu^k n^k$. Note that this number is random as μ is random.
- We evaluate the normalized total loss per 100 expected insured vehicles. According to our specific choice of μ, it is given by
$$\frac{L}{\frac{\mathbb{E}(n^\Phi)}{100}} = 100 \cdot \frac{L}{\sum_{k=1}^K n^k \mathbb{E}(\mu^k)} = \frac{200}{n_g + n_b} \cdot L.$$

Table B.1: Expectation of the total loss.

	Binomial Model						Poisson Model					
	Gamma			Log-Normal			Gamma			Log-Normal		
	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$
Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	546.7	544.7	544.2	547.0	544.2	546.5	545.5	546.7	546.2	543.4	547.2	542.5
ξ^{1b}	527.0	525.9	523.0	529.3	524.7	522.1	527.0	527.4	523.0	523.2	529.5	525.7
ξ^{2a}	1381.6	1376.5	1374.0	1384.3	1369.3	1367.1	1376.9	1377.7	1376.7	1376.1	1379.5	1364.3
ξ^{2b}	1399.7	1389.0	1380.1	1401.2	1387.2	1389.0	1395.5	1399.6	1392.0	1387.6	1394.9	1381.1
ξ^{3a}	2023.2	2009.8	2028.8	2027.8	2012.6	2019.9	2022.0	2026.4	2022.8	2003.2	2019.8	2002.1
ξ^{3b}	2168.9	2156.3	2148.7	2161.5	2144.3	2150.8	2163.1	2163.5	2158.5	2150.0	2164.1	2149.8
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	2173.0	2169.6	2174.0	2172.5	2170.7	2172.0	2174.9	2175.4	2176.2	2170.8	2165.0	2169.4
ξ^{1b}	2309.2	2312.7	2317.5	2310.9	2309.4	2311.8	2310.0	2314.6	2315.4	2309.9	2312.8	2308.5
ξ^{2a}	5712.1	5731.1	5753.4	5706.9	5719.8	5708.2	5715.9	5719.9	5707.2	5727.0	5705.6	5728.6
ξ^{2b}	7235.7	7258.7	7267.2	7260.1	7250.1	7281.1	7259.9	7262.6	7278.0	7246.7	7249.5	7243.4
ξ^{3a}	10020.9	10043.5	10034.8	10034.3	10012.5	10034.7	10037.9	10045.6	10028.9	10040.4	10040.6	10047.3
ξ^{3b}	13264.6	13275.8	13234.9	13271.4	13260.9	13277.3	13265.9	13272.4	13236.1	13251.3	13245.3	13241.8
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	3125.0	3133.0	3123.5	3130.7	3131.0	3119.0	3124.1	3128.5	3121.5	3128.1	3134.5	3133.0
ξ^{1b}	3772.7	3773.9	3768.3	3770.2	3778.0	3765.6	3778.6	3772.9	3770.4	3763.3	3773.1	3771.2
ξ^{2a}	10054.8	10066.1	10034.4	10069.6	10080.3	10060.3	10068.0	10065.4	10059.5	10054.0	10072.5	10104.8
ξ^{2b}	12538.0	12557.2	12527.5	12557.4	12545.2	12516.0	12554.5	12542.0	12552.6	12535.5	12555.9	12552.5
ξ^{3a}	20601.5	20620.3	20511.0	20585.1	20600.1	20587.4	20609.0	20571.2	20549.9	20562.7	20602.3	20603.8
ξ^{3b}	28596.4	28625.0	28470.8	28605.5	28633.4	28629.9	28595.0	28587.0	28595.0	28577.9	28647.2	28642.7
Non-Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	52.3	52.2	53.3	52.5	52.1	54.4	51.8	52.7	52.2	52.5	53.2	53.0
ξ^{1b}	52.9	52.7	52.3	52.9	52.0	52.8	52.6	52.3	52.6	52.4	52.0	54.1
ξ^{2a}	304.5	306.1	295.6	305.5	307.4	304.8	302.6	308.5	306.4	304.4	305.3	303.6
ξ^{2b}	325.5	328.6	331.2	322.7	329.7	324.6	327.2	325.4	327.8	327.3	325.1	322.0
ξ^{3a}	1003.9	1004.6	1006.5	1000.6	1010.8	1000.9	1003.4	993.5	1022.1	1006.5	1009.0	1008.5
ξ^{3b}	1208.6	1207.7	1202.9	1215.7	1207.0	1212.8	1206.1	1209.5	1196.5	1208.0	1211.4	1222.5
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	199.1	199.5	199.9	199.7	201.4	201.2	199.7	198.2	197.9	199.4	198.8	200.0
ξ^{1b}	228.6	228.0	229.6	229.5	229.7	226.7	228.3	227.7	229.3	227.2	228.5	230.2
ξ^{2a}	1577.8	1571.5	1578.4	1581.7	1576.2	1582.7	1579.1	1582.9	1583.0	1579.4	1579.2	1579.8
ξ^{2b}	1928.8	1912.8	1914.6	1923.6	1917.7	1911.4	1914.0	1917.7	1937.8	1913.3	1922.2	1911.7
ξ^{3a}	6946.2	6974.9	6913.7	6928.7	6937.4	6948.2	6945.0	6913.8	6942.0	6927.8	6943.6	6952.9
ξ^{3b}	9311.7	9349.5	9349.8	9329.4	9359.3	9343.5	9340.0	9330.0	9359.0	9334.5	9326.2	9308.5
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	335.7	335.2	335.6	335.0	335.1	333.7	334.7	334.7	334.3	337.0	336.5	334.9
ξ^{1b}	381.3	383.4	383.5	381.6	380.3	383.4	381.2	382.8	380.9	383.4	382.4	382.2
ξ^{2a}	2032.5	2020.2	2034.2	2030.7	2029.4	2040.2	2026.1	2022.3	2027.8	2025.6	2031.2	2035.6
ξ^{2b}	2562.1	2553.8	2550.5	2559.7	2556.1	2553.7	2562.5	2556.9	2568.5	2556.3	2567.8	2558.7
ξ^{3a}	12801.2	12825.5	12775.7	12795.2	12843.3	12815.3	12811.7	12808.1	12797.0	12818.1	12792.4	12772.1
ξ^{3b}	15767.8	15782.4	15757.3	15747.8	15789.4	15787.4	15766.4	15759.1	15736.7	15769.6	15744.5	15806.1

The expectation of the total loss is approximated using $10,000$ independent samples of L.

Table B.2: Variance of the total loss.

	Binomial Model						Poisson Model					
	Gamma			Log-Normal			Gamma			Log-Normal		
	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$
Uniform Accident Occurrence												
$\rho^{\clubsuit} = \mathbf{0.1}$:												
ξ^{1a}	18893.7	27665.8	63330.2	18797.9	28026.1	65488.9	18277.8	28087.5	63245.1	19071.7	28092.0	62943.3
ξ^{1b}	18087.2	26745.3	60657.5	18440.3	26108.7	61767.7	17887.3	26811.1	60802.6	17589.5	27311.8	61022.4
ξ^{2a}	173375.0	258237.7	600842.9	173092.1	258920.4	598414.0	172120.9	262482.2	613084.4	173021.5	265451.8	582242.0
ξ^{2b}	177214.8	265275.6	619897.7	178629.2	271430.9	631212.7	176634.1	272308.4	613676.7	174434.4	261582.3	590262.8
ξ^{3a}	485013.5	694041.4	1621207.6	499592.9	727156.0	1579947.2	495801.3	724757.7	1672718.2	495659.6	710318.7	1536261.7
ξ^{3b}	490118.8	722394.1	1688625.0	491584.1	734333.9	1711028.6	488696.3	721297.3	1687128.3	488968.6	752454.6	1606874.7
$\rho^{\clubsuit} = \mathbf{0.5}$:												
ξ^{1a}	121040.6	157007.0	323289.6	120936.5	166813.1	330798.2	122103.1	161366.7	326131.8	121952.2	160241.5	307114.2
ξ^{1b}	181320.5	230611.6	464392.6	179740.3	236318.9	433770.2	181861.1	233756.7	466338.5	181687.0	239151.9	475013.2
ξ^{2a}	960580.4	1331588.1	2810898.4	976795.7	1347996.6	2838794.5	953983.0	1324593.2	2730096.3	975379.6	1367343.2	2723201.3
ξ^{2b}	1476836.9	1960914.4	3775938.1	1483347.2	1965754.2	3854527.9	1486437.1	1935028.3	3828259.5	1476619.6	1924502.1	3735207.0
ξ^{3a}	2993260.8	4217438.8	8673871.0	3068922.2	4221409.4	8880079.8	3014765.5	4205157.8	8801550.9	3067141.8	4212840.4	8536435.4
ξ^{3b}	4676662.8	6224116.1	12156646.2	4809676.1	6329363.3	12781771.6	4762692.2	6219709.2	12509905.8	4676895.4	6141764.0	12152976.1
$\rho^{\clubsuit} = \mathbf{0.9}$:												
ξ^{1a}	230663.3	292653.8	557213.5	230110.7	299753.8	550714.7	223713.1	296432.7	556156.1	225082.5	298376.0	550281.9
ξ^{1b}	420227.2	514161.8	886441.8	419661.1	529744.0	890496.6	422378.1	506903.9	866223.8	410689.9	518008.1	838534.9
ξ^{2a}	3617972.2	4254283.7	6785282.6	3627251.6	4245970.8	6861536.3	3575807.1	4150777.2	6806228.4	3548344.3	4209067.1	6990333.1
ξ^{2b}	5148307.6	6076761.5	9320685.1	5301991.9	6039432.8	9419485.9	5157667.3	5835207.0	9384488.3	5055167.2	6010148.3	9398184.7
ξ^{3a}	13071136.0	15680586.9	24022504.1	13409575.9	15713162.4	25030090.4	13116476.7	15264724.3	25281776.3	12989610.2	15290800.5	25353912.1
ξ^{3b}	24501429.2	28291913.8	41813984.7	24611290.5	28767366.0	42224310.3	24362225.8	27467893.6	41963940.9	24379873.1	28203672.3	42459073.8
Non-Uniform Accident Occurrence												
$\rho^{\clubsuit} = \mathbf{0.1}$:												
ξ^{1a}	1438.0	2257.3	6100.2	1467.8	2237.1	6720.3	1440.0	2313.5	5776.6	1471.4	2425.6	5994.4
ξ^{1b}	1463.3	2378.6	5994.7	1455.2	2321.9	5858.0	1505.6	2300.6	5844.3	1413.0	2318.5	6845.5
ξ^{2a}	30953.2	49464.1	121528.1	31352.3	48965.0	109378.1	30342.7	49461.0	127032.7	30907.4	49753.9	111897.9
ξ^{2b}	33419.3	53075.3	143017.0	32834.7	55200.1	116193.0	34025.5	54202.3	135611.7	32757.6	54546.8	121344.0
ξ^{3a}	182937.0	295334.3	722181.9	184042.2	295803.4	738829.0	184663.6	288892.8	754295.9	184744.1	304452.4	796616.7
ξ^{3b}	222882.0	353001.2	886386.5	220849.6	360422.9	869560.6	218409.1	361988.8	862637.6	224628.2	344226.6	977182.7
$\rho^{\clubsuit} = \mathbf{0.5}$:												
ξ^{1a}	5313.9	8478.7	21793.0	5388.0	8619.9	24533.1	5398.9	8642.4	21592.3	5472.4	8465.0	22170.3
ξ^{1b}	6306.7	9851.5	24822.5	6414.5	10070.0	22808.0	6176.6	10229.3	24183.2	6242.0	9831.1	25914.1
ξ^{2a}	160179.5	247943.9	626190.6	162067.5	247069.8	614713.4	159388.6	250341.3	621806.1	159415.6	248641.8	620170.1
ξ^{2b}	193822.7	309370.7	772896.4	196901.0	307477.8	776261.6	189283.6	304666.2	777809.5	196095.0	319014.4	742031.7
ξ^{3a}	1212501.9	1988777.5	4900976.8	1218271.7	1967587.9	4929170.8	1214816.6	1973115.9	4820241.3	1212656.8	1951271.1	5212245.7
ξ^{3b}	2193293.6	3252845.0	7162873.4	2244429.9	3255329.8	7542890.4	2213441.4	3230865.7	7360691.8	2272519.9	3139292.2	7205238.8
$\rho^{\clubsuit} = \mathbf{0.9}$:												
ξ^{1a}	9418.1	14251.7	34904.2	8856.3	14557.9	32370.7	8959.2	14462.4	35612.6	9189.2	14733.7	35395.1
ξ^{1b}	10433.6	17198.0	41040.9	10625.6	16511.0	41640.4	10743.4	16890.3	41181.5	10870.6	17197.3	39666.6
ξ^{2a}	213730.9	326337.6	793409.8	210433.2	326920.7	848075.4	209536.2	325582.3	798698.7	213523.4	324714.3	804142.7
ξ^{2b}	246957.3	400989.1	989549.0	249269.2	399286.6	948864.4	249621.3	409990.9	984075.1	250842.2	410979.1	1030749.7
ξ^{3a}	2892886.1	4312952.8	9670130.8	2866821.7	4265176.9	9459010.8	2895553.5	4295589.2	9710683.1	2956058.6	4316705.3	9543766.6
ξ^{3b}	3320897.8	5093373.4	11733718.4	3380575.1	5108875.3	11667850.1	3205683.4	5100258.4	12004775.6	3294870.1	5014524.6	12241070.1

The variance of the total loss is approximated using 10,000 independent samples of L.

Table B.3: Skewness of the total loss.

	Binomial Model						Poisson Model					
	Gamma			Log-Normal			Gamma			Log-Normal		
	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$
Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	0.306	0.533	0.869	0.320	0.679	1.904	0.327	0.454	0.858	0.386	0.536	1.728
ξ^{1b}	0.292	0.468	0.868	0.357	0.666	1.764	0.330	0.468	0.818	0.366	0.563	1.756
ξ^{2a}	0.409	0.609	1.095	0.451	0.733	2.822	0.403	0.573	1.109	0.451	0.785	2.221
ξ^{2b}	0.431	0.639	1.114	0.435	0.886	2.495	0.359	0.598	1.073	0.448	0.650	1.987
ξ^{3a}	0.451	0.640	1.234	0.503	0.803	2.269	0.430	0.687	1.297	0.497	0.773	2.171
ξ^{3b}	0.463	0.584	1.142	0.453	0.865	2.763	0.399	0.607	1.142	0.461	0.785	2.084
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	0.192	0.254	0.492	0.130	0.323	1.454	0.200	0.222	0.502	0.164	0.384	0.891
ξ^{1b}	0.330	0.458	1.120	0.326	0.576	1.658	0.350	0.532	1.032	0.341	0.679	3.431
ξ^{2a}	0.266	0.338	0.563	0.216	0.385	1.318	0.219	0.306	0.550	0.221	0.406	1.246
ξ^{2b}	0.208	0.323	0.515	0.196	0.290	0.942	0.209	0.257	0.484	0.184	0.307	0.919
ξ^{3a}	0.211	0.285	0.566	0.237	0.358	1.206	0.182	0.314	0.609	0.229	0.358	1.146
ξ^{3b}	0.194	0.264	0.496	0.187	0.323	1.166	0.235	0.242	0.532	0.193	0.263	0.972
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	0.265	0.380	0.795	0.261	0.402	1.359	0.257	0.397	0.746	0.292	0.436	1.418
ξ^{1b}	0.249	0.386	0.836	0.218	0.392	2.090	0.234	0.351	0.687	0.237	0.393	1.154
ξ^{2a}	0.143	0.209	0.391	0.145	0.212	0.755	0.150	0.230	0.382	0.125	0.195	0.979
ξ^{2b}	0.155	0.195	0.386	0.132	0.200	0.650	0.123	0.177	0.394	0.125	0.209	0.724
ξ^{3a}	0.142	0.194	0.370	0.119	0.231	0.652	0.125	0.191	0.384	0.147	0.209	0.791
ξ^{3b}	0.104	0.163	0.335	0.104	0.170	0.541	0.102	0.140	0.292	0.113	0.171	0.483
Non-Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	0.893	1.482	2.832	0.934	1.875	7.599	0.948	1.516	2.779	0.938	1.916	7.127
ξ^{1b}	0.916	1.463	2.841	0.931	2.016	5.193	0.970	1.385	2.789	0.908	1.875	6.440
ξ^{2a}	0.770	1.255	2.486	0.768	1.372	3.274	0.747	1.182	2.367	0.805	1.647	3.572
ξ^{2b}	0.778	1.070	2.370	0.739	1.503	3.182	0.717	1.181	2.276	0.753	1.600	3.484
ξ^{3a}	0.560	0.941	1.727	0.646	1.216	6.279	0.565	0.927	1.663	0.612	1.291	7.033
ξ^{3b}	0.602	0.848	1.639	0.550	1.169	2.945	0.501	0.857	1.578	0.591	1.013	5.332
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	0.502	0.733	1.364	0.510	0.931	5.395	0.488	0.761	1.502	0.508	0.975	2.945
ξ^{1b}	0.412	0.676	1.274	0.476	0.913	2.348	0.410	0.674	1.249	0.469	0.936	3.525
ξ^{2a}	0.333	0.561	0.993	0.377	0.660	2.152	0.299	0.522	1.045	0.417	0.711	2.155
ξ^{2b}	0.333	0.489	0.891	0.343	0.620	2.241	0.292	0.466	0.934	0.305	0.631	1.921
ξ^{3a}	0.209	0.362	0.663	0.222	0.507	1.711	0.199	0.429	0.664	0.188	0.446	2.234
ξ^{3b}	0.170	0.291	0.548	0.182	0.324	1.787	0.222	0.343	0.626	0.194	0.354	1.358
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	0.375	0.595	0.986	0.380	0.702	1.992	0.340	0.588	1.078	0.388	0.716	2.678
ξ^{1b}	0.386	0.562	0.975	0.393	0.689	2.316	0.374	0.556	1.006	0.359	0.751	2.101
ξ^{2a}	0.298	0.440	0.813	0.319	0.626	2.733	0.274	0.509	0.918	0.306	0.627	2.219
ξ^{2b}	0.251	0.420	0.759	0.270	0.517	1.567	0.260	0.363	0.743	0.315	0.585	2.460
ξ^{3a}	0.203	0.310	0.502	0.194	0.325	1.069	0.161	0.266	0.469	0.239	0.359	1.044
ξ^{3b}	0.145	0.235	0.437	0.138	0.316	1.084	0.096	0.218	0.440	0.129	0.280	1.006

The skewness of the total loss is approximated using $10,000$ independent samples of L.

Table B.4: Value-at-Risk at level $p = 0.9$ of the total loss.

	Binomial Model						Poisson Model					
	Gamma			Log-Normal			Gamma			Log-Normal		
	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$
Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	726.8	766.2	882.9	727.7	763.1	861.4	725.3	772.8	887.0	723.0	771.9	851.7
ξ^{1b}	707.7	743.7	848.1	709.3	737.7	829.6	706.0	743.0	857.3	699.0	748.5	827.4
ξ^{2a}	1933.3	2052.4	2413.8	1937.5	2041.5	2288.0	1926.0	2058.1	2429.1	1928.5	2065.1	2267.1
ξ^{2b}	1958.5	2072.9	2437.3	1964.7	2070.4	2317.7	1958.2	2099.0	2451.4	1934.8	2083.1	2314.8
ξ^{3a}	2952.1	3134.7	3703.4	2987.1	3151.4	3531.9	2970.2	3154.5	3774.6	2949.0	3137.8	3483.6
ξ^{3b}	3112.3	3283.0	3887.7	3097.2	3288.6	3659.2	3096.9	3317.0	3921.2	3070.4	3328.9	3674.9
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	2639.3	2686.3	2929.1	2632.9	2708.7	2895.6	2637.8	2714.0	2927.2	2635.1	2687.8	2886.9
ξ^{1b}	2889.2	2953.6	3169.3	2880.1	2944.2	3121.2	2888.1	2957.7	3188.6	2891.1	2952.3	3114.5
ξ^{2a}	7028.2	7253.9	7995.6	7033.8	7242.1	7807.5	7029.6	7254.4	7909.1	7051.0	7260.4	7853.7
ξ^{2b}	8878.5	9109.5	9838.1	8901.2	9130.7	9830.7	8910.9	9109.8	9876.5	8890.4	9091.0	9780.9
ξ^{3a}	12338.0	12765.5	13936.0	12345.0	12762.5	13804.0	12337.5	12802.0	13882.5	12382.5	12750.5	13753.0
ξ^{3b}	16184.0	16600.0	17849.0	16194.5	16581.0	17812.0	16181.5	16578.0	17881.0	16130.5	16525.0	17747.0
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	3766.3	3847.8	4095.0	3777.0	3843.9	4045.0	3759.6	3856.4	4095.4	3762.1	3866.1	4065.3
ξ^{1b}	4643.1	4744.1	4968.7	4633.4	4743.4	4900.4	4641.6	4735.9	5007.1	4622.5	4740.3	4905.8
ξ^{2a}	12584.5	12848.5	13568.0	12590.5	12830.0	13436.5	12552.5	12775.0	13579.5	12541.5	12822.0	13505.5
ξ^{2b}	15531.0	15888.0	16692.0	15583.5	15805.0	16638.5	15568.0	15769.5	16635.5	15505.0	15797.5	16529.5
ξ^{3a}	25402.0	25952.0	27072.5	25422.0	25873.5	27115.0	25397.5	25802.5	27306.5	25371.0	25876.5	27061.0
ξ^{3b}	35096.5	35639.0	37232.0	35087.5	35789.5	37168.5	35042.0	35566.0	37240.0	35099.5	35698.5	37191.5
Non-Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	103.7	115.1	147.7	105.1	113.8	125.9	102.1	116.3	145.2	105.6	116.1	124.6
ξ^{1b}	105.0	119.7	147.2	104.8	113.7	124.5	105.2	118.3	144.5	103.0	114.6	125.3
ξ^{2a}	541.5	608.5	730.7	542.4	606.4	657.7	537.3	606.1	765.5	541.3	592.2	656.2
ξ^{2b}	569.6	644.4	809.9	568.4	631.2	686.1	577.3	633.0	808.1	573.4	634.3	691.4
ξ^{3a}	1576.8	1734.2	2138.4	1571.1	1715.8	1917.8	1582.2	1726.8	2173.7	1582.0	1725.8	1899.4
ξ^{3b}	1826.7	2013.5	2439.2	1835.4	2000.0	2240.9	1832.4	2007.2	2436.1	1841.4	1984.4	2258.1
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	295.1	322.5	396.1	295.2	324.0	360.6	297.5	321.2	392.9	296.8	319.7	365.7
ξ^{1b}	334.8	361.3	438.3	336.4	363.2	405.9	332.5	363.2	437.0	331.8	358.9	409.1
ξ^{2a}	2111.9	2219.1	2634.2	2112.9	2242.4	2526.6	2101.2	2244.4	2623.3	2098.1	2226.8	2509.3
ξ^{2b}	2512.1	2640.6	3081.1	2515.9	2653.5	2972.0	2484.1	2638.4	3128.9	2497.4	2669.7	2968.7
ξ^{3a}	8385.3	8836.3	9837.0	8360.5	8773.8	9648.6	8382.1	8764.2	9810.4	8361.8	8757.2	9736.3
ξ^{3b}	11227.5	11706.0	12932.0	11281.5	11695.0	12703.0	11290.5	11670.0	12979.0	11312.5	11630.0	12613.0
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	465.9	493.1	593.0	459.5	494.5	549.9	457.4	493.9	588.5	463.1	499.8	555.4
ξ^{1b}	516.2	557.2	661.0	517.7	549.2	621.4	517.8	555.6	654.1	523.6	554.4	624.3
ξ^{2a}	2645.5	2778.7	3241.0	2629.0	2781.6	3113.7	2623.5	2779.6	3242.5	2632.3	2784.2	3112.9
ξ^{2b}	3206.6	3388.2	3904.8	3213.4	3387.9	3760.7	3216.9	3408.2	3889.2	3209.2	3400.6	3760.1
ξ^{3a}	15023.5	15581.0	16960.0	15046.0	15562.5	16712.0	15090.0	15568.0	16927.0	15121.0	15542.5	16791.0
ξ^{3b}	18160.0	18762.0	20237.0	18102.0	18736.0	20122.0	18088.5	18705.5	20215.0	18106.5	18662.5	20252.5

The Value-at-Risk at level $p = 0.9$ of the total loss is approximated using 10,000 independent samples of L.

Table B.5: Expected Shortfall at level $p = 0.9$ of the total loss.

	Binomial Model						Poisson Model					
	Gamma			Log-Normal			Gamma			Log-Normal		
	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$
Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	801.9	866.5	1055.4	803.7	871.6	1096.5	797.8	868.0	1058.7	805.2	875.7	1079.0
ξ^{1b}	778.2	840.5	1025.3	786.6	844.8	1058.5	777.6	842.5	1022.9	773.4	853.6	1054.4
ξ^{2a}	2175.9	2378.3	2998.7	2184.2	2384.9	3029.9	2163.1	2372.4	3029.2	2170.4	2418.7	3010.3
ξ^{2b}	2205.7	2400.3	3026.6	2213.2	2441.4	3095.9	2186.3	2429.5	3030.8	2188.0	2411.8	3062.6
ξ^{3a}	3356.2	3661.7	4733.3	3395.1	3726.8	4760.5	3367.4	3718.1	4821.5	3361.2	3716.6	4715.2
ξ^{3b}	3513.6	3819.4	4899.6	3513.2	3879.1	4976.4	3494.5	3840.0	4904.8	3485.0	3924.3	4907.7
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	2799.2	2888.9	3270.3	2785.5	2935.4	3319.3	2799.9	2906.4	3278.1	2791.8	2899.6	3280.8
ξ^{1b}	3087.8	3225.4	3702.7	3090.7	3249.3	3646.8	3100.8	3248.5	3705.4	3098.0	3259.8	3668.1
ξ^{2a}	7510.7	7881.5	9017.4	7499.5	7906.8	9128.0	7490.1	7866.2	8908.5	7530.2	7925.5	9048.0
ξ^{2b}	9413.0	9842.3	11013.9	9434.8	9832.0	11238.5	9449.8	9817.9	11033.4	9422.1	9819.5	11135.3
ξ^{3a}	13182.2	13850.3	15809.1	13228.0	13870.7	16164.8	13168.1	13888.1	15843.7	13243.7	13869.9	16010.4
ξ^{3b}	17154.5	17851.0	19962.8	17194.5	17938.8	20513.8	17219.5	17833.8	20094.8	17124.6	17832.9	20268.2
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	3993.4	4143.9	4599.1	3996.1	4155.3	4613.5	3975.6	4152.8	4595.2	3989.6	4171.8	4612.8
ξ^{1b}	4912.4	5106.7	5633.6	4898.1	5119.9	5614.8	4915.7	5085.0	5590.8	4884.6	5102.2	5595.5
ξ^{2a}	13210.3	13668.8	14950.9	13273.1	13698.8	15088.9	13245.1	13701.9	14949.2	13177.9	13682.6	15224.2
ξ^{2b}	16347.2	16845.4	18250.6	16353.2	16839.6	18467.3	16314.3	16711.2	18310.7	16260.2	16831.1	18364.0
ξ^{3a}	26696.6	27567.5	29709.6	26704.5	27646.9	30247.4	26664.9	27421.2	30053.0	26627.2	27437.3	30369.4
ξ^{3b}	36557.6	37674.3	40459.2	36582.8	37775.4	40933.9	36529.7	37550.3	40509.2	36598.9	37750.6	40952.3
Non-Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	130.6	156.5	241.4	131.8	156.6	228.5	130.3	158.3	234.4	132.1	163.7	220.1
ξ^{1b}	131.5	160.8	239.2	131.0	158.5	220.7	133.3	158.4	235.4	130.1	160.0	230.1
ξ^{2a}	658.2	782.7	1109.9	662.3	788.1	1063.8	652.1	785.0	1135.4	661.0	791.9	1066.1
ξ^{2b}	692.2	815.4	1214.7	687.5	834.8	1100.9	695.5	820.5	1180.9	690.6	830.1	1110.8
ξ^{3a}	1838.5	2128.6	2906.0	1846.6	2164.8	2827.8	1847.1	2098.3	2995.7	1859.1	2182.8	2885.5
ξ^{3b}	2139.4	2420.7	3305.6	2126.6	2455.3	3272.6	2107.2	2437.5	3259.1	2137.0	2419.6	3328.8
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	339.7	383.8	523.7	342.2	390.8	527.5	340.6	386.0	520.0	342.7	388.3	528.1
ξ^{1b}	379.3	424.8	567.2	384.5	435.2	560.5	378.3	428.6	565.6	379.4	432.1	577.9
ξ^{2a}	2331.7	2538.8	3233.3	2342.4	2557.5	3280.3	2324.1	2555.6	3233.9	2336.6	2565.0	3267.6
ξ^{2b}	2753.2	2979.5	3705.3	2756.6	3011.0	3791.3	2730.9	2976.9	3753.3	2738.7	3030.6	3741.5
ξ^{3a}	8967.6	9655.6	11313.2	8955.7	9653.9	11593.3	8958.3	9595.8	11302.3	8922.0	9616.2	11665.8
ξ^{3b}	11979.9	12682.9	14571.5	12049.5	12751.3	14897.9	12063.2	12699.1	14705.5	12071.2	12689.7	14794.7
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	519.2	568.8	726.0	513.6	575.8	724.3	511.8	569.1	731.6	518.6	578.1	737.0
ξ^{1b}	574.4	640.2	806.4	576.4	634.2	821.1	576.8	636.2	805.5	579.1	645.1	813.9
ξ^{2a}	2897.3	3111.9	3856.7	2888.5	3159.2	3974.0	2875.6	3129.3	3868.2	2889.7	3156.2	3931.6
ξ^{2b}	3477.8	3761.0	4556.0	3481.6	3775.2	4613.1	3494.4	3763.1	4572.9	3489.5	3812.7	4679.3
ξ^{3a}	15904.3	16738.7	18784.8	15905.5	16719.6	19046.2	15883.1	16660.3	18772.6	15990.8	16718.3	19092.6
ξ^{3b}	19082.9	19943.9	22286.1	19081.7	19990.4	22609.8	18973.6	19903.4	22353.6	19040.1	19888.6	22843.3

The Expected Shortfall at level $p = 0.9$ of the total loss is approximated using 10,000 independent samples of L.

Table B.6: Value-at-Risk at level $p = 0.95$ of the total loss.

	Binomial Model						Poisson Model					
	Gamma			Log-Normal			Gamma			Log-Normal		
	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$
Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	781.9	840.4	1007.2	787.2	838.1	1012.7	779.0	844.5	1019.1	787.2	850.7	997.9
ξ^{1b}	760.3	818.7	982.2	769.6	814.9	978.3	760.0	820.8	978.6	757.8	823.5	977.2
ξ^{2a}	2119.8	2292.5	2840.8	2124.7	2275.7	2773.4	2097.7	2273.9	2865.1	2107.4	2305.6	2735.3
ξ^{2b}	2143.4	2317.1	2834.7	2152.1	2340.0	2803.0	2127.7	2344.1	2849.9	2118.2	2337.5	2805.7
ξ^{3a}	3251.4	3529.7	4454.2	3309.8	3561.7	4312.2	3262.0	3560.0	4545.6	3260.0	3560.8	4310.9
ξ^{3b}	3416.7	3691.4	4645.1	3414.2	3695.1	4506.5	3418.6	3704.2	4610.5	3374.7	3775.3	4425.9
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	2763.8	2834.9	3194.6	2752.6	2887.8	3166.0	2759.9	2857.7	3180.4	2757.9	2836.4	3154.5
ξ^{1b}	3035.9	3156.2	3517.7	3038.5	3165.7	3442.4	3063.8	3164.7	3533.1	3051.4	3165.4	3449.1
ξ^{2a}	7392.6	7719.9	8776.3	7392.8	7746.9	8611.8	7388.3	7723.1	8614.5	7427.2	7746.2	8626.5
ξ^{2b}	9288.1	9640.5	10722.0	9310.3	9654.7	10796.5	9298.9	9675.0	10743.0	9307.5	9662.4	10692.0
ξ^{3a}	13018.0	13556.5	15357.5	13035.5	13626.0	15409.5	12969.0	13618.0	15313.5	13044.5	13588.0	15245.5
ξ^{3b}	16932.0	17548.0	19415.5	16980.5	17664.0	19639.0	16992.0	17557.5	19533.5	16878.5	17556.5	19507.5
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	3938.9	4059.4	4430.8	3939.9	4069.6	4396.3	3921.6	4085.6	4449.4	3931.8	4094.0	4414.9
ξ^{1b}	4853.6	5023.9	5449.5	4833.3	5012.8	5355.8	4847.5	5015.1	5410.6	4819.0	5009.6	5345.6
ξ^{2a}	13059.5	13482.0	14591.5	13115.0	13535.0	14544.5	13117.5	13480.5	14580.5	13015.0	13508.5	14641.5
ξ^{2b}	16158.5	16606.0	17873.5	16176.0	16587.0	17866.0	16176.0	16490.0	17951.0	16087.5	16617.5	17774.5
ξ^{3a}	26393.5	27251.0	29085.5	26415.0	27246.5	29354.5	26378.5	27064.0	29458.0	26324.5	27095.0	29377.5
ξ^{3b}	36241.5	37249.0	39731.0	36210.5	37290.5	39735.0	36181.0	37099.5	39811.0	36260.0	37316.0	39890.0
Non-Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	123.5	144.5	210.9	124.6	141.3	183.1	123.4	146.4	206.3	125.3	148.4	177.3
ξ^{1b}	125.3	148.6	207.6	122.9	141.9	176.0	125.4	148.0	206.1	123.4	145.3	181.7
ξ^{2a}	626.3	730.9	989.5	634.1	733.8	917.8	618.9	730.9	1026.8	634.2	726.4	915.9
ξ^{2b}	659.5	772.7	1094.8	658.2	763.1	942.9	663.5	765.0	1054.2	654.6	772.0	932.8
ξ^{3a}	1771.6	2023.4	2654.7	1774.0	2043.9	2470.4	1780.2	2005.3	2778.8	1797.8	2043.5	2482.9
ξ^{3b}	2055.6	2300.0	3053.1	2040.2	2313.3	2804.2	2028.8	2307.4	2999.2	2048.2	2287.4	2877.1
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	330.2	366.6	488.5	331.5	371.9	454.9	329.6	371.2	480.8	331.4	367.1	458.4
ξ^{1b}	367.0	407.5	530.5	372.1	414.7	499.3	367.6	409.2	529.9	368.2	410.5	512.1
ξ^{2a}	2275.9	2471.3	3070.5	2288.4	2468.1	2996.2	2272.2	2477.4	3079.6	2270.2	2462.6	2942.6
ξ^{2b}	2697.7	2888.7	3504.4	2694.5	2911.8	3479.6	2668.7	2896.3	3584.6	2674.3	2923.4	3477.9
ξ^{3a}	8830.5	9463.1	10967.5	8813.8	9408.5	10854.0	8816.8	9377.5	10928.5	8760.9	9407.1	10932.5
ξ^{3b}	11796.5	12442.5	14155.5	11889.0	12499.0	13985.0	11885.0	12411.5	14236.0	11885.0	12410.0	13992.5
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	505.9	548.1	688.2	502.5	555.6	660.3	499.1	548.4	691.4	506.7	555.9	669.6
ξ^{1b}	558.4	620.2	765.3	560.2	610.1	741.1	562.4	613.7	765.2	562.5	620.3	749.9
ξ^{2a}	2844.1	3025.4	3710.1	2827.3	3054.6	3667.1	2818.9	3042.3	3679.9	2830.7	3055.9	3601.2
ξ^{2b}	3405.7	3671.3	4362.8	3413.2	3647.9	4343.6	3428.1	3674.9	4407.2	3418.0	3696.7	4324.0
ξ^{3a}	15668.5	16400.5	18286.0	15692.5	16435.0	18314.5	15650.0	16380.0	18291.5	15752.5	16465.0	18365.0
ξ^{3b}	18857.0	19640.0	21779.5	18820.5	19659.0	21783.5	18761.0	19613.0	21761.5	18853.0	19563.0	22036.5

The Value-at-Risk at level $p = 0.95$ of the total loss is approximated using 10,000 independent samples of L.

Table B.7: Expected Shortfall at level $p = 0.95$ of the total loss.

	Binomial Model						Poisson Model					
	Gamma			Log-Normal			Gamma			Log-Normal		
	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$
Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	851.6	934.7	1173.4	853.3	946.7	1262.1	845.7	931.5	1173.7	857.9	942.1	1242.0
ξ^{1b}	823.8	904.7	1144.5	835.0	916.0	1225.7	825.6	908.2	1133.5	821.0	922.4	1217.5
ξ^{2a}	2334.9	2594.2	3397.3	2344.5	2621.6	3559.6	2324.0	2589.8	3423.6	2332.3	2659.3	3547.2
ξ^{2b}	2366.7	2621.6	3437.6	2373.5	2696.0	3666.3	2339.4	2635.8	3425.6	2351.4	2626.8	3594.9
ξ^{3a}	3624.5	4006.8	5433.9	3665.2	4113.6	5655.5	3629.3	4097.5	5512.1	3636.9	4106.9	5601.3
ξ^{3b}	3777.9	4174.4	5577.8	3771.7	4291.1	5927.3	3752.1	4192.3	5583.6	3760.2	4323.6	5803.9
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	2898.7	3028.8	3498.3	2882.5	3077.2	3623.9	2904.3	3033.9	3516.1	2893.6	3043.4	3551.6
ξ^{1b}	3221.0	3416.3	4089.0	3233.5	3453.1	4030.9	3228.6	3441.0	4062.3	3234.1	3467.5	4065.1
ξ^{2a}	7826.8	8298.1	9708.8	7807.1	8337.4	10085.9	7789.9	8265.3	9599.9	7836.9	8362.8	9903.0
ξ^{2b}	9753.9	10322.3	11837.3	9778.9	10302.2	12208.7	9802.7	10280.7	11823.4	9755.0	10290.7	12088.7
ξ^{3a}	13710.3	14549.6	17042.7	13795.0	14599.6	17823.1	13735.7	14569.3	17153.8	13799.1	14636.8	17600.7
ξ^{3b}	17749.3	18688.9	21363.1	17828.7	18822.3	22410.6	17893.6	18635.4	21554.1	17764.5	18606.5	21964.2
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	4139.5	4334.8	4961.1	4140.0	4361.4	5018.8	4113.6	4347.7	4944.2	4139.6	4377.7	5009.2
ξ^{1b}	5085.8	5346.1	6103.3	5072.1	5372.5	6131.3	5104.2	5306.3	5999.9	5059.2	5341.5	6085.4
ξ^{2a}	13625.6	14201.9	15854.1	13753.8	14239.0	16235.7	13679.3	14260.6	15856.6	13592.9	14208.9	16426.1
ξ^{2b}	16787.7	17434.1	19231.3	16881.1	17471.0	19755.7	16864.2	17349.0	19389.1	16715.3	17492.8	19648.7
ξ^{3a}	27476.8	28605.6	31399.4	27584.1	28785.6	32373.0	27480.0	28547.4	31907.6	27436.2	28465.3	32590.9
ξ^{3b}	37482.8	39014.9	42578.5	37593.0	39062.6	43572.6	37505.9	38740.3	42632.3	37560.9	39069.8	43480.5
Non-Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	148.3	184.8	306.5	149.8	186.9	306.3	149.5	186.4	296.0	149.4	196.9	293.8
ξ^{1b}	149.3	189.2	302.9	149.0	190.4	294.6	152.0	184.6	299.0	148.2	191.9	311.7
ξ^{2a}	736.2	901.3	1374.9	741.1	911.8	1363.7	731.3	904.5	1394.5	739.1	929.9	1363.7
ξ^{2b}	774.1	930.9	1495.2	765.0	976.7	1410.5	774.1	948.6	1440.7	771.7	966.6	1422.6
ξ^{3a}	2013.8	2399.6	3446.4	2027.5	2469.6	3510.9	2023.9	2350.4	3541.5	2040.0	2496.7	3620.6
ξ^{3b}	2345.2	2693.1	3902.0	2330.7	2768.9	4055.8	2293.3	2735.5	3828.9	2334.5	2723.7	4141.1
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	368.9	424.7	609.5	373.6	437.1	652.1	369.4	429.0	607.5	372.4	436.4	651.4
ξ^{1b}	408.7	467.9	655.6	416.6	484.0	672.8	408.5	472.1	653.4	410.1	481.0	700.5
ξ^{2a}	2468.9	2748.0	3628.9	2491.4	2772.5	3838.4	2465.7	2763.6	3642.2	2497.0	2795.4	3828.2
ξ^{2b}	2909.2	3204.1	4134.0	2916.2	3244.9	4384.6	2880.8	3204.5	4180.7	2899.7	3277.2	4284.5
ξ^{3a}	9343.9	10179.6	12284.4	9346.8	10237.9	13004.8	9326.5	10148.6	12281.8	9304.3	10202.9	13054.7
ξ^{3b}	12470.7	13334.5	15670.3	12563.5	13421.6	16473.4	12562.4	13369.5	15857.4	12563.6	13378.6	16355.4
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	554.5	620.2	816.2	548.8	630.5	849.1	547.2	619.8	830.3	556.6	631.9	871.7
ξ^{1b}	613.5	694.9	903.0	615.3	691.5	969.5	615.3	689.6	906.8	616.9	707.2	945.7
ξ^{2a}	3059.7	3339.0	4261.7	3060.4	3408.5	4597.3	3040.1	3353.0	4287.3	3067.7	3402.9	4533.1
ξ^{2b}	3659.9	4002.8	5011.7	3657.2	4036.6	5229.1	3668.1	3994.5	5024.5	3675.0	4092.1	5359.3
ξ^{3a}	16481.6	17442.7	20001.9	16439.2	17472.6	20696.7	16424.2	17384.3	20015.2	16532.0	17528.5	20681.3
ξ^{3b}	19647.2	20706.5	23616.4	19686.9	20829.7	24327.4	19540.9	20700.7	23730.6	19638.1	20736.9	24616.0

The Expected Shortfall at level $p = 0.95$ of the total loss is approximated using $10,000$ independent samples of L.

Table B.8: Value-at-Risk at level $p = 0.99$ of the total loss.

	Binomial Model						Poisson Model					
	Gamma			Log-Normal			Gamma			Log-Normal		
	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$
Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	896.9	993.6	1276.6	898.9	1011.2	1411.3	886.1	977.0	1268.0	898.6	1000.2	1361.8
ξ^{1b}	865.4	964.3	1243.6	877.0	978.3	1369.8	872.9	960.5	1221.0	861.4	989.1	1338.5
ξ^{2a}	2483.1	2785.0	3686.9	2480.7	2833.6	4035.2	2479.2	2796.8	3755.3	2454.1	2886.9	3928.3
ξ^{2b}	2502.4	2795.9	3782.1	2527.6	2892.2	4124.7	2472.4	2811.9	3794.3	2481.4	2822.8	4037.1
ξ^{3a}	3891.8	4298.1	5973.5	3862.8	4465.4	6474.9	3854.3	4423.5	6184.8	3858.0	4523.4	6162.8
ξ^{3b}	4002.2	4481.0	6078.6	4012.8	4705.7	6641.1	3961.6	4464.7	6227.3	3976.1	4671.3	6630.1
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	2983.6	3131.1	3669.1	2970.4	3185.2	3863.3	2995.2	3139.9	3693.8	2978.8	3165.8	3791.1
ξ^{1b}	3349.2	3566.6	4412.6	3353.6	3613.6	4262.1	3321.5	3600.9	4357.7	3348.4	3623.4	4227.9
ξ^{2a}	8123.4	8697.1	10350.5	8062.2	8689.8	10804.0	8020.9	8653.8	10244.0	8080.7	8718.1	10599.0
ξ^{2b}	10039.5	10750.5	12515.5	10080.5	10716.0	12941.5	10128.5	10630.5	12436.0	10036.5	10621.0	12809.0
ξ^{3a}	14135.5	15131.5	18185.0	14277.5	15237.0	18926.0	14166.5	15121.0	18225.0	14340.0	15289.5	18866.5
ξ^{3b}	18232.5	19448.5	22665.5	18393.0	19544.5	24125.0	18438.0	19333.0	22882.5	18341.5	19306.5	23424.0
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	4260.4	4513.9	5301.9	4264.1	4538.0	5334.1	4225.9	4521.6	5218.5	4273.4	4561.3	5270.7
ξ^{1b}	5214.1	5546.6	6478.4	5239.3	5570.9	6477.9	5244.6	5483.3	6321.1	5202.1	5531.9	6546.0
ξ^{2a}	13995.5	14677.0	16738.5	14017.0	14587.0	17082.5	14041.0	14698.0	16579.5	13956.0	14607.5	17239.5
ξ^{2b}	17127.0	17903.0	20093.0	17271.0	17958.5	20898.0	17082.0	17817.5	20230.0	17110.5	18015.5	20562.5
ξ^{3a}	28147.0	29476.0	32885.5	28145.5	29813.5	34001.0	28184.0	29512.5	33622.0	28210.0	29224.0	34144.5
ξ^{3b}	38280.5	40071.0	44148.5	38580.0	40146.5	45749.0	38445.5	39832.0	44320.5	38388.5	40163.5	45604.0
Non-Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	164.7	210.3	362.5	166.9	213.3	374.9	165.3	213.6	353.1	165.4	222.0	350.3
ξ^{1b}	162.8	215.6	368.3	162.8	219.5	354.2	170.3	210.9	352.9	162.5	219.4	360.9
ξ^{2a}	809.0	999.2	1578.1	798.9	1012.0	1633.8	806.4	1012.0	1614.9	798.6	1046.0	1609.6
ξ^{2b}	853.9	1032.2	1732.3	824.7	1094.7	1695.8	846.4	1068.5	1645.8	839.1	1079.1	1698.4
ξ^{3a}	2160.9	2632.2	3980.2	2152.6	2681.7	3944.5	2181.3	2549.9	4028.3	2194.5	2754.2	4270.2
ξ^{3b}	2528.6	2935.9	4423.7	2522.4	3079.8	4732.6	2455.9	3016.1	4325.9	2499.2	2972.0	4680.1
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	392.4	465.3	691.5	399.3	475.6	731.0	394.1	468.0	698.5	400.7	486.4	763.7
ξ^{1b}	432.8	505.9	731.0	444.1	534.3	765.1	431.6	508.3	728.0	438.3	519.2	766.9
ξ^{2a}	2587.3	2891.3	3999.6	2608.8	2943.8	4228.6	2591.8	2945.6	4027.3	2645.9	2976.1	4331.8
ξ^{2b}	3050.5	3416.1	4543.4	3048.4	3436.7	4781.1	3021.5	3379.9	4549.7	3042.7	3491.1	4740.8
ξ^{3a}	9704.1	10611.5	13158.0	9698.8	10755.5	13988.0	9638.0	10633.0	13285.5	9626.8	10727.0	14134.0
ξ^{3b}	12911.5	13951.0	16611.0	12950.5	13999.0	17784.0	12959.5	14000.0	16881.0	13007.0	14023.5	17559.5
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	587.6	668.5	895.8	579.0	675.1	956.9	581.6	658.2	915.9	585.4	671.7	952.1
ξ^{1b}	642.5	741.8	983.3	644.6	735.1	1094.2	650.5	737.8	995.0	651.0	762.9	1065.0
ξ^{2a}	3192.7	3530.3	4616.1	3207.8	3630.6	4960.2	3183.9	3525.8	4671.0	3196.6	3646.0	5106.1
ξ^{2b}	3806.4	4207.4	5450.8	3804.6	4281.5	5688.8	3812.2	4193.3	5422.1	3816.1	4332.2	5851.1
ξ^{3a}	16938.0	18022.0	21093.5	16977.5	18117.5	21646.0	16903.0	17944.5	20990.0	16987.5	18148.0	21973.0
ξ^{3b}	20123.5	21342.5	24736.5	20158.0	21620.5	25502.5	20014.5	21395.5	24993.5	20158.0	21397.0	25959.0

The Value-at-Risk at level $p = 0.99$ of the total loss is approximated using $10,000$ independent samples of L.

Table B.9: Expected Shortfall at level $p = 0.99$ of the total loss.

	Binomial Model						Poisson Model					
	Gamma			Log-Normal			Gamma			Log-Normal		
	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$
Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	951.2	1084.1	1447.0	948.7	1127.3	1695.8	950.2	1059.9	1431.3	961.0	1078.2	1681.4
ξ^{1b}	915.9	1029.1	1413.4	928.7	1073.8	1659.5	920.4	1030.0	1374.7	918.9	1061.8	1648.4
ξ^{2a}	2654.9	3043.7	4297.2	2659.1	3128.0	5105.2	2650.5	3043.8	4364.6	2671.8	3207.2	5066.1
ξ^{2b}	2690.2	3103.9	4374.9	2698.3	3279.9	5353.4	2634.5	3090.6	4317.8	2693.4	3076.5	5001.7
ξ^{3a}	4187.5	4718.1	7110.1	4189.4	5017.0	8175.1	4131.5	4897.9	7083.8	4185.3	4955.8	7996.0
ξ^{3b}	4320.5	4866.9	7106.5	4324.5	5241.7	8614.3	4240.5	4894.3	7133.4	4357.5	5160.8	8317.0
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	3099.0	3309.4	3962.0	3070.2	3355.4	4425.6	3118.7	3296.9	4004.0	3096.6	3369.9	4227.4
ξ^{1b}	3505.7	3784.2	5074.4	3499.1	3904.6	5198.2	3487.4	3862.7	4992.3	3500.1	3993.5	5341.9
ξ^{2a}	8467.3	9158.7	11195.0	8377.4	9250.9	12681.4	8371.6	9053.2	11079.4	8452.7	9212.8	12179.8
ξ^{2b}	10421.0	11373.9	13474.1	10463.1	11298.2	14700.9	10463.5	11156.4	13432.2	10380.7	11198.0	14485.0
ξ^{3a}	14715.7	15947.6	19567.0	14843.1	16065.1	22068.2	14785.1	16096.1	20006.3	14838.7	16230.3	21487.1
ξ^{3b}	18891.3	20313.7	24232.7	19009.5	20538.2	27354.1	19170.6	20231.2	24565.4	19007.9	20150.4	26141.2
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	4421.5	4801.0	5886.6	4429.2	4820.3	6140.0	4389.9	4744.2	5773.8	4436.8	4802.4	6131.0
ξ^{1b}	5407.3	5833.0	7237.2	5420.8	5952.7	7674.0	5429.9	5757.1	7002.7	5401.7	5878.8	7461.1
ξ^{2a}	14427.3	15187.1	17553.2	14482.0	15437.3	19414.9	14476.1	15392.7	17721.5	14548.2	15239.8	19825.8
ξ^{2b}	17695.8	18550.5	21340.6	17778.1	18745.3	22878.3	17786.8	18480.9	21560.8	17579.8	18823.2	23142.6
ξ^{3a}	29304.2	30507.8	35084.0	28981.6	31022.9	37720.2	29094.8	30447.0	35160.0	29003.0	30561.7	37920.1
ξ^{3b}	39359.9	41395.3	46366.1	39618.1	41544.5	49882.2	39376.1	41024.1	46500.0	39576.5	41421.5	49062.2
Non-Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	185.2	249.7	461.4	188.1	260.3	568.9	187.8	255.0	453.6	187.6	279.5	529.1
ξ^{1b}	186.0	251.5	463.2	189.4	273.8	535.9	192.8	243.8	462.5	184.2	271.5	585.1
ξ^{2a}	906.6	1182.9	2021.8	908.6	1205.0	2183.3	893.1	1159.3	2036.2	906.1	1287.1	2234.7
ξ^{2b}	963.7	1182.3	2168.0	934.0	1332.0	2249.6	941.1	1230.0	2088.1	937.5	1282.9	2358.6
ξ^{3a}	2359.6	2954.8	4766.4	2422.3	3172.1	5564.7	2371.6	2903.7	4751.4	2397.0	3263.5	5842.4
ξ^{3b}	2770.6	3326.5	5297.6	2753.4	3547.2	6336.0	2673.2	3344.7	5124.3	2767.3	3409.4	6663.7
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	428.9	513.6	794.9	432.7	542.2	1069.6	431.4	514.0	815.0	437.0	544.7	1012.9
ξ^{1b}	469.3	558.6	843.1	480.6	595.6	1004.6	467.4	564.5	830.1	474.7	592.4	1071.8
ξ^{2a}	2755.9	3188.6	4490.2	2809.0	3234.4	5424.2	2757.9	3186.1	4545.0	2845.4	3314.1	5490.5
ξ^{2b}	3254.6	3696.7	5110.7	3239.9	3756.0	6132.3	3195.5	3677.9	5153.4	3229.7	3852.8	5744.2
ξ^{3a}	10075.0	11204.4	14320.5	10130.4	11480.0	16782.5	10063.9	11319.4	14315.2	10129.2	11383.8	17251.5
ξ^{3b}	13461.6	14719.7	17974.1	13510.3	14659.9	21196.4	13576.5	14764.1	18235.5	13562.2	14738.9	20600.3
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	630.7	730.5	1010.7	620.8	742.1	1178.7	623.7	728.3	1045.5	628.2	755.0	1266.3
ξ^{1b}	692.8	808.0	1122.2	697.0	819.4	1397.8	695.5	801.4	1120.5	697.2	842.6	1310.3
ξ^{2a}	3361.5	3787.5	5101.3	3412.6	3952.0	6459.9	3367.0	3806.7	5227.1	3399.7	3954.7	6271.1
ξ^{2b}	4017.8	4502.6	5993.1	4026.6	4604.9	6854.6	3989.4	4468.5	5904.9	4059.2	4717.9	7290.5
ξ^{3a}	17650.5	18979.3	22686.9	17523.6	18915.1	24851.9	17471.2	18754.6	22552.3	17703.0	18991.9	24740.3
ξ^{3b}	20759.0	22314.3	26371.2	20985.2	22672.7	28992.9	20625.9	22257.1	26754.0	20848.5	22388.8	29549.3

The Expected Shortfall at level $p = 0.99$ of the total loss is approximated using $10,000$ independent samples of L.

Table B.10: Expectation of the normalized total loss.

	Binomial Model						Poisson Model					
	Gamma			Log-Normal			Gamma			Log-Normal		
	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$
Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	140.7	140.2	140.1	140.8	140.1	140.7	140.4	140.7	140.6	139.9	140.8	139.7
ξ^{1b}	135.7	135.4	134.6	136.2	135.1	134.4	135.6	135.8	134.6	134.7	136.3	135.3
ξ^{2a}	355.6	354.3	353.7	356.3	352.5	351.9	354.4	354.6	354.4	354.2	355.1	351.2
ξ^{2b}	360.3	357.5	355.2	360.7	357.1	357.5	359.2	360.2	358.3	357.2	359.0	355.5
ξ^{3a}	520.8	517.3	522.2	522.0	518.0	519.9	520.5	521.6	520.7	515.6	519.9	515.3
ξ^{3b}	558.3	555.0	553.1	556.4	551.9	553.6	556.8	556.9	555.6	553.4	557.0	553.4
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	128.8	128.6	128.8	128.7	128.6	128.7	128.9	128.9	129.0	128.6	128.3	128.6
ξ^{1b}	136.8	137.0	137.3	136.9	136.9	137.0	136.9	137.2	137.2	136.9	137.1	136.8
ξ^{2a}	338.5	339.6	340.9	338.2	339.0	338.3	338.7	339.0	338.2	339.4	338.1	339.5
ξ^{2b}	428.8	430.1	430.7	430.2	429.6	431.5	430.2	430.4	431.3	429.4	429.6	429.2
ξ^{3a}	593.8	595.2	594.7	594.6	593.3	594.7	594.8	595.3	594.3	595.0	595.0	595.4
ξ^{3b}	786.0	786.7	784.3	786.5	785.8	786.8	786.1	786.5	784.4	785.3	784.9	784.7
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	103.4	103.7	103.3	103.6	103.6	103.2	103.4	103.5	103.3	103.5	103.7	103.7
ξ^{1b}	124.8	124.9	124.7	124.7	125.0	124.6	125.0	124.8	124.7	124.5	124.8	124.8
ξ^{2a}	332.7	333.0	332.0	333.2	333.5	332.8	333.1	333.0	332.8	332.6	333.3	334.3
ξ^{2b}	414.8	415.5	414.5	415.5	415.1	414.1	415.4	415.0	415.3	414.7	415.4	415.3
ξ^{3a}	681.6	682.2	678.6	681.1	681.6	681.1	681.9	680.6	679.9	680.3	681.6	681.7
ξ^{3b}	946.1	947.1	942.0	946.4	947.3	947.2	946.1	945.8	946.1	945.5	947.8	947.7
Non-Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	13.5	13.4	13.7	13.5	13.4	14.0	13.3	13.6	13.4	13.5	13.7	13.6
ξ^{1b}	13.6	13.6	13.5	13.6	13.4	13.6	13.5	13.5	13.5	13.5	13.4	13.9
ξ^{2a}	78.4	78.8	76.1	78.6	79.1	78.4	77.9	79.4	78.9	78.4	78.6	78.1
ξ^{2b}	83.8	84.6	85.3	83.1	84.9	83.6	84.2	83.8	84.4	84.2	83.7	82.9
ξ^{3a}	258.4	258.6	259.1	257.6	260.2	257.6	258.3	255.7	263.1	259.1	259.7	259.6
ξ^{3b}	311.1	310.9	309.6	312.9	310.7	312.2	310.5	311.3	308.0	310.9	311.8	314.7
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	11.8	11.8	11.8	11.8	11.9	11.9	11.8	11.7	11.7	11.8	11.8	11.9
ξ^{1b}	13.5	13.5	13.6	13.6	13.6	13.4	13.5	13.5	13.6	13.5	13.5	13.6
ξ^{2a}	93.5	93.1	93.5	93.7	93.4	93.8	93.6	93.8	93.8	93.6	93.6	93.6
ξ^{2b}	114.3	113.4	113.5	114.0	113.6	113.3	113.4	113.6	114.8	113.4	113.9	113.3
ξ^{3a}	411.6	413.3	409.7	410.6	411.1	411.7	411.6	409.7	411.4	410.5	411.5	412.0
ξ^{3b}	551.8	554.0	554.1	552.9	554.6	553.7	553.5	552.9	554.6	553.2	552.7	551.6
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	11.1	11.1	11.1	11.1	11.1	11.0	11.1	11.1	11.1	11.2	11.1	11.1
ξ^{1b}	12.6	12.7	12.7	12.6	12.6	12.7	12.6	12.7	12.6	12.7	12.7	12.6
ξ^{2a}	67.2	66.8	67.3	67.2	67.1	67.5	67.0	66.9	67.1	67.0	67.2	67.3
ξ^{2b}	84.8	84.5	84.4	84.7	84.6	84.5	84.8	84.6	85.0	84.6	85.0	84.7
ξ^{3a}	423.5	424.3	422.7	423.3	424.9	424.0	423.9	423.8	423.4	424.1	423.2	422.6
ξ^{3b}	521.7	522.2	521.3	521.0	522.4	522.3	521.6	521.4	520.7	521.7	520.9	522.9

The expectation of the normalized total loss is approximated using $10,000$ independent samples of L. Total losses are normalized by 100 expected insured vehicles.

Table B.11: Variance of the normalized total loss.

	Binomial Model						Poisson Model					
	Gamma			Log-Normal			Gamma			Log-Normal		
	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$
Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	1251.8	1833.0	4195.9	1245.5	1856.9	4339.0	1211.0	1860.9	4190.3	1263.6	1861.2	4170.3
ξ^{1b}	1198.4	1772.0	4018.9	1221.8	1729.8	4092.4	1185.1	1776.4	4028.5	1165.4	1809.5	4043.0
ξ^{2a}	11486.9	17109.5	39808.8	11468.2	17154.7	39647.9	11403.9	17390.7	40619.8	11463.5	17587.5	38576.4
ξ^{2b}	11741.3	17575.8	41071.3	11835.1	17983.6	41820.9	11702.9	18041.8	40659.1	11557.1	17331.1	39107.8
ξ^{3a}	32134.5	45983.6	107412.9	33100.5	48177.6	104679.2	32849.3	48018.7	110825.8	32839.9	47062.1	101784.9
ξ^{3b}	32472.8	47862.1	111879.7	32569.9	48653.2	113364.0	32378.5	47789.5	111780.5	32396.6	49853.8	106463.3
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	425.1	551.4	1135.3	424.7	585.8	1161.7	428.8	566.7	1145.3	428.3	562.7	1078.5
ξ^{1b}	636.7	809.8	1630.8	631.2	829.9	1523.3	638.6	820.9	1637.6	638.0	839.8	1668.1
ξ^{2a}	3373.2	4676.1	9870.9	3430.2	4733.7	9968.9	3350.1	4651.5	9587.2	3425.2	4801.6	9563.0
ξ^{2b}	5186.1	6886.1	13259.8	5209.0	6903.1	13535.8	5219.9	6795.2	13443.5	5185.4	6758.2	13116.8
ξ^{3a}	10511.3	14810.2	30459.7	10777.0	14824.2	31183.8	10586.8	14767.1	30908.1	10770.8	14794.1	29977.1
ξ^{3b}	16422.8	21857.0	42690.0	16889.9	22226.6	44885.2	16725.0	21841.5	43930.5	16423.7	21567.8	42677.1
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	252.5	320.3	609.9	251.9	328.1	602.8	244.9	324.5	608.8	246.4	326.6	602.4
ξ^{1b}	460.0	562.8	970.3	459.4	579.9	974.8	462.3	554.9	948.2	449.6	567.0	917.9
ξ^{2a}	3960.3	4656.9	7427.4	3970.5	4647.8	7510.8	3914.2	4543.6	7450.3	3884.1	4607.4	7651.8
ξ^{2b}	5635.5	6651.8	10202.7	5803.7	6610.9	10310.9	5645.7	6387.4	10272.5	5533.5	6578.9	10287.5
ξ^{3a}	14308.1	17164.4	26295.8	14678.5	17200.1	27398.7	14357.7	16709.2	27674.2	14218.8	16737.8	27753.2
ξ^{3b}	26820.0	30969.2	45770.8	26940.3	31489.6	46220.0	26667.6	30067.2	45935.0	26686.9	30872.6	46477.0
Non-Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	95.3	149.6	404.2	97.2	148.2	445.3	95.4	153.3	382.7	97.5	160.7	397.2
ξ^{1b}	96.9	157.6	397.2	96.4	153.8	388.1	99.8	152.4	387.2	93.6	153.6	453.5
ξ^{2a}	2050.8	3277.2	8051.8	2077.2	3244.2	7246.8	2010.4	3277.0	8416.5	2047.8	3296.4	7413.8
ξ^{2b}	2214.2	3516.5	9475.6	2175.5	3657.3	7698.4	2254.4	3591.2	8984.9	2170.4	3614.0	8039.6
ξ^{3a}	12120.5	19567.3	47848.1	12193.7	19598.4	48951.0	12234.9	19140.6	49975.8	12240.2	20171.5	52779.8
ξ^{3b}	14767.0	23388.1	58727.4	14632.4	23879.8	57612.6	14470.7	23983.5	57154.0	14882.7	22806.7	64743.1
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	18.7	29.8	76.5	18.9	30.3	86.2	19.0	30.3	75.8	19.2	29.7	77.9
ξ^{1b}	22.1	34.6	87.2	22.5	35.4	80.1	21.7	35.9	84.9	21.9	34.5	91.0
ξ^{2a}	562.5	870.7	2199.0	569.1	867.6	2158.7	559.7	879.1	2183.6	559.8	873.1	2177.8
ξ^{2b}	680.6	1086.4	2714.1	691.4	1079.8	2726.0	664.7	1069.9	2731.4	688.6	1120.3	2605.8
ξ^{3a}	4257.9	6983.9	17210.6	4278.2	6909.5	17309.6	4266.0	6928.9	16927.0	4258.4	6852.2	18303.6
ξ^{3b}	7702.1	11422.9	25153.6	7881.7	11431.6	26488.1	7772.9	11345.7	25848.2	7980.3	11024.1	25302.3
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	10.3	15.6	38.2	9.7	15.9	35.4	9.8	15.8	39.0	10.1	16.1	38.7
ξ^{1b}	11.4	18.8	44.9	11.6	18.1	45.6	11.8	18.5	45.1	11.9	18.8	43.4
ξ^{2a}	234.0	357.2	868.5	230.3	357.9	928.3	229.4	356.4	874.3	233.7	355.4	880.2
ξ^{2b}	270.3	438.9	1083.2	272.9	437.1	1038.7	273.2	448.8	1077.2	274.6	449.9	1128.3
ξ^{3a}	3166.6	4721.1	10585.2	3138.1	4668.8	10354.1	3169.6	4702.1	10629.6	3235.8	4725.2	10446.9
ξ^{3b}	3635.2	5575.4	12844.1	3700.5	5592.3	12772.0	3509.0	5582.9	13140.8	3606.7	5489.0	13399.4

The variance of the normalized total loss is approximated using $10,000$ independent samples of L. Total losses are normalized by 100 expected insured vehicles.

Table B.12: Skewness of the normalized total loss.

	Binomial Model						Poisson Model					
	Gamma			Log-Normal			Gamma			Log-Normal		
	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$
Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	0.306	0.533	0.869	0.320	0.679	1.904	0.327	0.454	0.858	0.386	0.536	1.728
ξ^{1b}	0.292	0.468	0.868	0.357	0.666	1.764	0.330	0.468	0.818	0.366	0.563	1.756
ξ^{2a}	0.409	0.609	1.095	0.451	0.733	2.822	0.403	0.573	1.109	0.451	0.785	2.221
ξ^{2b}	0.431	0.639	1.114	0.435	0.886	2.495	0.359	0.598	1.073	0.448	0.650	1.987
ξ^{3a}	0.451	0.640	1.234	0.503	0.803	2.269	0.430	0.687	1.297	0.497	0.773	2.171
ξ^{3b}	0.463	0.584	1.142	0.453	0.865	2.763	0.399	0.607	1.142	0.461	0.785	2.084
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	0.192	0.254	0.492	0.130	0.323	1.454	0.200	0.222	0.502	0.164	0.384	0.891
ξ^{1b}	0.330	0.458	1.120	0.326	0.576	1.658	0.350	0.532	1.032	0.341	0.679	3.431
ξ^{2a}	0.266	0.338	0.563	0.216	0.385	1.318	0.219	0.306	0.550	0.221	0.406	1.246
ξ^{2b}	0.208	0.323	0.515	0.196	0.290	0.942	0.209	0.257	0.484	0.184	0.307	0.919
ξ^{3a}	0.211	0.285	0.566	0.237	0.358	1.206	0.182	0.314	0.609	0.229	0.358	1.146
ξ^{3b}	0.194	0.264	0.496	0.187	0.323	1.166	0.235	0.242	0.532	0.193	0.263	0.972
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	0.265	0.380	0.795	0.261	0.402	1.359	0.257	0.397	0.746	0.292	0.436	1.418
ξ^{1b}	0.249	0.386	0.836	0.218	0.392	2.090	0.234	0.351	0.687	0.237	0.393	1.154
ξ^{2a}	0.143	0.209	0.391	0.145	0.212	0.755	0.150	0.230	0.382	0.125	0.195	0.979
ξ^{2b}	0.155	0.195	0.386	0.132	0.200	0.650	0.123	0.177	0.394	0.125	0.209	0.724
ξ^{3a}	0.142	0.194	0.370	0.119	0.231	0.652	0.125	0.191	0.384	0.147	0.209	0.791
ξ^{3b}	0.104	0.163	0.335	0.104	0.170	0.541	0.102	0.140	0.292	0.113	0.171	0.483
Non-Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	0.893	1.482	2.832	0.934	1.875	7.599	0.948	1.516	2.779	0.938	1.916	7.127
ξ^{1b}	0.916	1.463	2.841	0.931	2.016	5.193	0.970	1.385	2.789	0.908	1.875	6.440
ξ^{2a}	0.770	1.255	2.486	0.768	1.372	3.274	0.747	1.182	2.367	0.805	1.647	3.572
ξ^{2b}	0.778	1.070	2.370	0.739	1.503	3.182	0.717	1.181	2.276	0.753	1.600	3.484
ξ^{3a}	0.560	0.941	1.727	0.646	1.216	6.279	0.565	0.927	1.663	0.612	1.291	7.033
ξ^{3b}	0.602	0.848	1.639	0.550	1.169	2.945	0.501	0.857	1.578	0.591	1.013	5.332
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	0.502	0.733	1.364	0.510	0.931	5.395	0.488	0.761	1.502	0.508	0.975	2.945
ξ^{1b}	0.412	0.676	1.274	0.476	0.913	2.348	0.410	0.674	1.249	0.469	0.936	3.525
ξ^{2a}	0.333	0.561	0.993	0.377	0.660	2.152	0.299	0.522	1.045	0.417	0.711	2.155
ξ^{2b}	0.333	0.489	0.891	0.343	0.620	2.241	0.292	0.466	0.934	0.305	0.631	1.921
ξ^{3a}	0.209	0.362	0.663	0.222	0.507	1.711	0.199	0.429	0.664	0.188	0.446	2.234
ξ^{3b}	0.170	0.291	0.548	0.182	0.324	1.787	0.222	0.343	0.626	0.194	0.354	1.358
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	0.375	0.595	0.986	0.380	0.702	1.992	0.340	0.588	1.078	0.388	0.716	2.678
ξ^{1b}	0.386	0.562	0.975	0.393	0.689	2.316	0.374	0.556	1.006	0.359	0.751	2.101
ξ^{2a}	0.298	0.440	0.813	0.319	0.626	2.733	0.274	0.509	0.918	0.306	0.627	2.219
ξ^{2b}	0.251	0.420	0.759	0.270	0.517	1.567	0.260	0.363	0.743	0.315	0.585	2.460
ξ^{3a}	0.203	0.310	0.502	0.194	0.325	1.069	0.161	0.266	0.469	0.239	0.359	1.044
ξ^{3b}	0.145	0.235	0.437	0.138	0.316	1.084	0.096	0.218	0.440	0.129	0.280	1.006

The skewness of the normalized total loss is approximated using $10,000$ independent samples of L. Total losses are normalized by 100 expected insured vehicles.

Table B.13: Value-at-Risk at level $p = 0.9$ of the normalized total loss.

	Binomial Model						Poisson Model					
	Gamma			Log-Normal			Gamma			Log-Normal		
	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$
Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	187.1	197.2	227.2	187.3	196.4	221.7	186.7	198.9	228.3	186.1	198.7	219.2
ξ^{1b}	182.2	191.4	218.3	182.6	189.9	213.5	181.7	191.3	220.7	179.9	192.7	213.0
ξ^{2a}	497.6	528.3	621.3	498.7	525.5	588.9	495.7	529.8	625.2	496.4	531.6	583.6
ξ^{2b}	504.1	533.6	627.4	505.7	532.9	596.6	504.1	540.3	631.0	498.0	536.2	595.8
ξ^{3a}	759.9	806.9	953.3	768.9	811.2	909.1	764.5	812.0	971.6	759.1	807.7	896.7
ξ^{3b}	801.1	845.0	1000.7	797.2	846.5	941.9	797.2	853.8	1009.3	790.3	856.9	945.9
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	156.4	159.2	173.6	156.0	160.5	171.6	156.3	160.8	173.5	156.2	159.3	171.1
ξ^{1b}	171.2	175.0	187.8	170.7	174.5	185.0	171.1	175.3	189.0	171.3	175.0	184.6
ξ^{2a}	416.5	429.9	473.8	416.8	429.2	462.7	416.6	429.9	468.7	417.8	430.2	465.4
ξ^{2b}	526.1	539.8	583.0	527.5	541.1	582.6	528.1	539.8	585.3	526.8	538.7	579.6
ξ^{3a}	731.1	756.5	825.8	731.6	756.3	818.0	731.1	758.6	822.7	733.8	755.6	815.0
ξ^{3b}	959.1	983.7	1057.7	959.7	982.6	1055.5	958.9	982.4	1059.6	955.9	979.3	1051.7
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	124.6	127.3	135.5	125.0	127.2	133.8	124.4	127.6	135.5	124.5	127.9	134.5
ξ^{1b}	153.6	157.0	164.4	153.3	156.9	162.1	153.6	156.7	165.7	152.9	156.8	162.3
ξ^{2a}	416.4	425.1	448.9	416.6	424.5	444.5	415.3	422.7	449.3	414.9	424.2	446.8
ξ^{2b}	513.8	525.7	552.3	515.6	522.9	550.5	515.1	521.7	550.4	513.0	522.7	546.9
ξ^{3a}	840.4	858.6	895.7	841.1	856.0	897.1	840.3	853.7	903.4	839.4	856.1	895.3
ξ^{3b}	1161.2	1179.1	1231.8	1160.9	1184.1	1229.7	1159.4	1176.7	1232.1	1161.3	1181.1	1230.5
Non-Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	26.7	29.6	38.0	27.1	29.3	32.4	26.3	29.9	37.4	27.2	29.9	32.1
ξ^{1b}	27.0	30.8	37.9	27.0	29.3	32.1	27.1	30.5	37.2	26.5	29.5	32.3
ξ^{2a}	139.4	156.6	188.1	139.6	156.1	169.3	138.3	156.0	197.0	139.3	152.4	168.9
ξ^{2b}	146.6	165.9	208.5	146.3	162.5	176.6	148.6	162.9	208.0	147.6	163.3	178.0
ξ^{3a}	405.9	446.4	550.4	404.4	441.7	493.6	407.2	444.5	559.5	407.2	444.2	488.9
ξ^{3b}	470.2	518.3	627.8	472.4	514.8	576.8	471.7	516.7	627.0	474.0	510.8	581.2
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	17.5	19.1	23.5	17.5	19.2	21.4	17.6	19.0	23.3	17.6	18.9	21.7
ξ^{1b}	19.8	21.4	26.0	19.9	21.5	24.1	19.7	21.5	25.9	19.7	21.3	24.2
ξ^{2a}	125.1	131.5	156.1	125.2	132.9	149.7	124.5	133.0	155.5	124.3	132.0	148.7
ξ^{2b}	148.9	156.5	182.6	149.1	157.2	176.1	147.2	156.4	185.4	148.0	158.2	175.9
ξ^{3a}	496.9	523.6	582.9	495.4	519.9	571.8	496.7	519.4	581.4	495.5	518.9	577.0
ξ^{3b}	665.3	693.7	766.3	668.5	693.0	752.8	669.1	691.6	769.1	670.4	689.2	747.4
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	15.4	16.3	19.6	15.2	16.4	18.2	15.1	16.3	19.5	15.3	16.5	18.4
ξ^{1b}	17.1	18.4	21.9	17.1	18.2	20.6	17.1	18.4	21.6	17.3	18.3	20.7
ξ^{2a}	87.5	91.9	107.2	87.0	92.0	103.0	86.8	92.0	107.3	87.1	92.1	103.0
ξ^{2b}	106.1	112.1	129.2	106.3	112.1	124.4	106.4	112.8	128.7	106.2	112.5	124.4
ξ^{3a}	497.1	515.5	561.1	497.8	514.9	552.9	499.3	515.1	560.0	500.3	514.2	555.5
ξ^{3b}	600.8	620.7	669.5	598.9	619.9	665.7	598.5	618.9	668.8	599.1	617.5	670.1

The Value-at-Risk at level $p = 0.9$ of the normalized total loss is approximated using $10,000$ independent samples of L. Total losses are normalized by 100 expected insured vehicles.

Table B.14: Expected Shortfall at level $p = 0.9$ of the normalized total loss.

	Binomial Model						Poisson Model					
	Gamma			Log-Normal			Gamma			Log-Normal		
	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$
Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	206.4	223.0	271.7	206.9	224.4	282.2	205.3	223.4	272.5	207.3	225.4	277.7
ξ^{1b}	200.3	216.3	263.9	202.5	217.4	272.4	200.1	216.9	263.3	199.1	219.7	271.4
ξ^{2a}	560.1	612.2	771.9	562.2	613.9	779.9	556.8	610.7	779.7	558.7	622.6	774.8
ξ^{2b}	567.7	617.8	779.1	569.7	628.4	796.9	562.8	625.4	780.1	563.2	620.8	788.3
ξ^{3a}	863.9	942.5	1218.4	873.9	959.3	1225.4	866.8	957.0	1241.1	865.2	956.7	1213.7
ξ^{3b}	904.4	983.1	1261.2	904.3	998.5	1280.9	899.5	988.4	1262.5	897.0	1010.1	1263.2
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	165.9	171.2	193.8	165.1	173.9	196.7	165.9	172.2	194.3	165.4	171.8	194.4
ξ^{1b}	183.0	191.1	219.4	183.2	192.5	216.1	183.8	192.5	219.6	183.6	193.2	217.4
ξ^{2a}	445.1	467.1	534.4	444.4	468.6	540.9	443.9	466.1	527.9	446.2	469.7	536.2
ξ^{2b}	557.8	583.2	652.7	559.1	582.6	666.0	560.0	581.8	653.8	558.3	581.9	659.9
ξ^{3a}	781.2	820.8	936.8	783.9	822.0	957.9	780.3	823.0	938.9	784.8	821.9	948.8
ξ^{3b}	1016.6	1057.8	1183.0	1018.9	1063.0	1215.6	1020.4	1056.8	1190.8	1014.8	1056.8	1201.1
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	132.1	137.1	152.2	132.2	137.5	152.6	131.5	137.4	152.0	132.0	138.0	152.6
ξ^{1b}	162.5	169.0	186.4	162.1	169.4	185.8	162.6	168.2	185.0	161.6	168.8	185.1
ξ^{2a}	437.1	452.2	494.7	439.1	453.2	499.2	438.2	453.3	494.6	436.0	452.7	503.7
ξ^{2b}	540.9	557.3	603.8	541.0	557.1	611.0	539.8	552.9	605.8	538.0	556.9	607.6
ξ^{3a}	883.3	912.1	982.9	883.5	914.7	1000.7	882.2	907.2	994.3	881.0	907.8	1004.8
ξ^{3b}	1209.5	1246.5	1338.6	1210.4	1249.8	1354.3	1208.6	1242.4	1340.3	1210.9	1249.0	1354.9
Non-Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	33.6	40.3	62.1	33.9	40.3	58.8	33.5	40.7	60.3	34.0	42.1	56.6
ξ^{1b}	33.9	41.4	61.6	33.7	40.8	56.8	34.3	40.8	60.6	33.5	41.2	59.2
ξ^{2a}	169.4	201.5	285.7	170.5	202.9	273.8	167.8	202.1	292.2	170.1	203.8	274.4
ξ^{2b}	178.2	209.9	312.7	177.0	214.9	283.4	179.0	211.2	304.0	177.8	213.7	285.9
ξ^{3a}	473.2	547.9	748.0	475.3	557.2	727.9	475.5	540.1	771.1	478.5	561.9	742.7
ξ^{3b}	550.7	623.1	850.9	547.4	632.0	842.4	542.4	627.4	838.9	550.1	622.8	856.8
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	20.1	22.7	31.0	20.3	23.2	31.3	20.2	22.9	30.8	20.3	23.0	31.3
ξ^{1b}	22.5	25.2	33.6	22.8	25.8	33.2	22.4	25.4	33.5	22.5	25.6	34.2
ξ^{2a}	138.2	150.4	191.6	138.8	151.6	194.4	137.7	151.4	191.6	138.5	152.0	193.6
ξ^{2b}	163.2	176.6	219.6	163.4	178.4	224.7	161.8	176.4	222.4	162.3	179.6	221.7
ξ^{3a}	531.4	572.2	670.4	530.7	572.1	687.0	530.9	568.6	669.8	528.7	569.9	691.3
ξ^{3b}	709.9	751.6	863.5	714.0	755.6	882.8	714.9	752.5	871.4	715.3	752.0	876.7
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	17.2	18.8	24.0	17.0	19.1	24.0	16.9	18.8	24.2	17.2	19.1	24.4
ξ^{1b}	19.0	21.2	26.7	19.1	21.0	27.2	19.1	21.0	26.6	19.2	21.3	26.9
ξ^{2a}	95.9	103.0	127.6	95.6	104.5	131.5	95.1	103.5	128.0	95.6	104.4	130.1
ξ^{2b}	115.1	124.4	150.7	115.2	124.9	152.6	115.6	124.5	151.3	115.5	126.1	154.8
ξ^{3a}	526.2	553.8	621.5	526.2	553.2	630.1	525.5	551.2	621.1	529.1	553.1	631.7
ξ^{3b}	631.4	659.8	737.3	631.3	661.4	748.1	627.7	658.5	739.6	629.9	658.0	755.8

The Expected Shortfall at level $p = 0.9$ of the normalized total loss is approximated using $10,000$ independent samples of L. Total losses are normalized by 100 expected insured vehicles.

Table B.15: Value-at-Risk at level $p = 0.95$ of the normalized total loss.

	Binomial Model						Poisson Model					
	Gamma			Log-Normal			Gamma			Log-Normal		
	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$
Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	201.2	216.3	259.3	202.6	215.7	260.7	200.5	217.4	262.3	202.6	219.0	256.9
ξ^{1b}	195.7	210.7	252.8	198.1	209.7	251.8	195.6	211.3	251.9	195.1	212.0	251.5
ξ^{2a}	545.6	590.1	731.2	546.9	585.8	713.9	539.9	585.3	737.5	542.5	593.4	704.1
ξ^{2b}	551.7	596.4	729.6	554.0	602.3	721.5	547.7	603.4	733.6	545.2	601.7	722.2
ξ^{3a}	836.9	908.5	1146.5	852.0	916.8	1110.0	839.6	916.3	1170.1	839.1	916.5	1109.6
ξ^{3b}	879.5	950.2	1195.6	878.8	951.1	1160.0	880.0	953.5	1186.7	868.6	971.8	1139.2
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	163.8	168.0	189.3	163.1	171.1	187.6	163.5	169.3	188.5	163.4	168.1	186.9
ξ^{1b}	179.9	187.0	208.5	180.1	187.6	204.0	181.6	187.5	209.4	180.8	187.6	204.4
ξ^{2a}	438.1	457.5	520.1	438.1	459.1	510.3	437.8	457.7	510.5	440.1	459.0	511.2
ξ^{2b}	550.4	571.3	635.4	551.7	572.1	639.8	551.0	573.3	636.6	551.6	572.6	633.6
ξ^{3a}	771.4	803.3	910.1	772.5	807.5	913.2	768.5	807.0	907.5	773.0	805.2	903.4
ξ^{3b}	1003.4	1039.9	1150.5	1006.3	1046.8	1163.8	1006.9	1040.4	1157.5	1000.2	1040.4	1156.0
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	130.3	134.3	146.6	130.4	134.6	145.5	129.7	135.2	147.2	130.1	135.5	146.1
ξ^{1b}	160.6	166.2	180.3	159.9	165.8	177.2	160.4	165.9	179.0	159.4	165.7	176.9
ξ^{2a}	432.1	446.1	482.8	433.9	447.8	481.2	434.0	446.0	482.4	430.6	446.9	484.4
ξ^{2b}	534.6	549.4	591.3	535.2	548.8	591.1	535.2	545.6	593.9	532.3	549.8	588.1
ξ^{3a}	873.2	901.6	962.3	873.9	901.5	971.2	872.7	895.4	974.6	871.0	896.4	972.0
ξ^{3b}	1199.1	1232.4	1314.5	1198.0	1233.8	1314.6	1197.1	1227.4	1317.2	1199.7	1234.6	1319.8
Non-Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	31.8	37.2	54.3	32.1	36.4	47.1	31.8	37.7	53.1	32.2	38.2	45.6
ξ^{1b}	32.2	38.3	53.4	31.6	36.5	45.3	32.3	38.1	53.0	31.8	37.4	46.8
ξ^{2a}	161.2	188.1	254.7	163.2	188.9	236.2	159.3	188.1	264.3	163.2	187.0	235.8
ξ^{2b}	169.7	198.9	281.8	169.4	196.4	242.7	170.8	196.9	271.4	168.5	198.7	240.1
ξ^{3a}	456.0	520.8	683.3	456.6	526.1	635.9	458.2	516.2	715.3	462.8	526.0	639.1
ξ^{3b}	529.1	592.0	785.9	525.1	595.4	721.8	522.2	593.9	772.0	527.2	588.8	740.6
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	19.6	21.7	28.9	19.6	22.0	27.0	19.5	22.0	28.5	19.6	21.8	27.2
ξ^{1b}	21.7	24.2	31.4	22.1	24.6	29.6	21.8	24.3	31.4	21.8	24.3	30.3
ξ^{2a}	134.9	146.5	182.0	135.6	146.3	177.6	134.7	146.8	182.5	134.5	145.9	174.4
ξ^{2b}	159.9	171.2	207.7	159.7	172.6	206.2	158.1	171.6	212.4	158.5	173.2	206.1
ξ^{3a}	523.3	560.8	649.9	522.3	557.5	643.2	522.5	555.7	647.6	519.2	557.5	647.9
ξ^{3b}	699.1	737.3	838.8	704.5	740.7	828.7	704.3	735.5	843.6	704.3	735.4	829.2
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	16.7	18.1	22.8	16.6	18.4	21.8	16.5	18.1	22.9	16.8	18.4	22.2
ξ^{1b}	18.5	20.5	25.3	18.5	20.2	24.5	18.6	20.3	25.3	18.6	20.5	24.8
ξ^{2a}	94.1	100.1	122.8	93.5	101.1	121.3	93.3	100.7	121.8	93.7	101.1	119.1
ξ^{2b}	112.7	121.5	144.3	112.9	120.7	143.7	113.4	121.6	145.8	113.1	122.3	143.1
ξ^{3a}	518.4	542.6	605.0	519.2	543.8	605.9	517.8	541.9	605.2	521.2	544.7	607.6
ξ^{3b}	623.9	649.8	720.6	622.7	650.4	720.7	620.7	648.9	720.0	623.8	647.2	729.1

The Value-at-Risk at level $p = 0.95$ of the normalized total loss is approximated using $10,000$ independent samples of L. Total losses are normalized by 100 expected insured vehicles.

Table B.16: Expected Shortfall at level $p = 0.95$ of the normalized total loss.

	Binomial Model						Poisson Model					
	Gamma			Log-Normal			Gamma			Log-Normal		
	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$
Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	219.2	240.6	302.0	219.6	243.7	324.9	217.7	239.8	302.1	220.8	242.5	319.7
ξ^{1b}	212.0	232.9	294.6	214.9	235.8	315.5	212.5	233.8	291.8	211.3	237.4	313.4
ξ^{2a}	601.0	667.7	874.5	603.5	674.8	916.2	598.2	666.6	881.2	600.3	684.5	913.0
ξ^{2b}	609.2	674.8	884.8	610.9	693.9	943.7	602.2	678.5	881.8	605.2	676.1	925.3
ξ^{3a}	932.9	1031.4	1398.7	943.4	1058.8	1455.7	934.2	1054.7	1418.8	936.1	1057.1	1441.8
ξ^{3b}	972.4	1074.5	1435.7	970.8	1104.5	1525.7	965.8	1079.1	1437.2	967.9	1112.9	1493.9
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	171.8	179.5	207.3	170.8	182.4	214.7	172.1	179.8	208.4	171.5	180.3	210.5
ξ^{1b}	190.9	202.4	242.3	191.6	204.6	238.9	191.3	203.9	240.7	191.7	205.5	240.9
ξ^{2a}	463.8	491.7	575.3	462.6	494.1	597.7	461.6	489.8	568.9	464.4	495.6	586.8
ξ^{2b}	578.0	611.7	701.5	579.5	610.5	723.5	580.9	609.2	700.6	578.1	609.8	716.4
ξ^{3a}	812.5	862.2	1009.9	817.5	865.2	1056.2	814.0	863.4	1016.5	817.7	867.4	1043.0
ξ^{3b}	1051.8	1107.5	1266.0	1056.5	1115.4	1328.0	1060.4	1104.3	1277.3	1052.7	1102.6	1301.6
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	137.0	143.4	164.1	137.0	144.3	166.0	136.1	143.8	163.6	137.0	144.8	165.7
ξ^{1b}	168.3	176.9	201.9	167.8	177.7	202.9	168.9	175.6	198.5	167.4	176.7	201.3
ξ^{2a}	450.8	469.9	524.5	455.0	471.1	537.2	452.6	471.8	524.6	449.7	470.1	543.5
ξ^{2b}	555.4	576.8	636.3	558.5	578.0	653.6	558.0	574.0	641.5	553.0	578.8	650.1
ξ^{3a}	909.1	946.4	1038.9	912.6	952.4	1071.1	909.2	944.5	1055.7	907.7	941.8	1078.3
ξ^{3b}	1240.1	1290.8	1408.7	1243.8	1292.4	1441.6	1240.9	1281.7	1410.5	1242.7	1292.6	1438.6
Non-Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	38.2	47.6	78.9	38.6	48.1	78.8	38.5	48.0	76.2	38.4	50.7	75.6
ξ^{1b}	38.4	48.7	78.0	38.4	49.0	75.8	39.1	47.5	77.0	38.1	49.4	80.2
ξ^{2a}	189.5	232.0	353.9	190.8	234.7	351.0	188.2	232.8	359.0	190.2	239.4	351.0
ξ^{2b}	199.3	239.6	384.9	196.9	251.4	363.1	199.2	244.2	370.8	198.6	248.8	366.2
ξ^{3a}	518.4	617.7	887.1	521.9	635.7	903.7	521.0	605.0	911.6	525.1	642.7	932.0
ξ^{3b}	603.7	693.2	1004.4	599.9	712.7	1044.0	590.3	704.1	985.6	600.9	701.1	1065.9
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	21.9	25.2	36.1	22.1	25.9	38.6	21.9	25.4	36.0	22.1	25.9	38.6
ξ^{1b}	24.2	27.7	38.8	24.7	28.7	39.9	24.2	28.0	38.7	24.3	28.5	41.5
ξ^{2a}	146.3	162.8	215.0	147.6	164.3	227.5	146.1	163.8	215.8	148.0	165.7	226.9
ξ^{2b}	172.4	189.9	245.0	172.8	192.3	259.8	170.7	189.9	247.7	171.8	194.2	253.9
ξ^{3a}	553.7	603.2	728.0	553.9	606.7	770.7	552.7	601.4	727.8	551.4	604.6	773.6
ξ^{3b}	739.0	790.2	928.6	744.5	795.4	976.2	744.4	792.3	939.7	744.5	792.8	969.2
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	18.3	20.5	27.0	18.2	20.9	28.1	18.1	20.5	27.5	18.4	20.9	28.8
ξ^{1b}	20.3	23.0	29.9	20.4	22.9	32.1	20.4	22.8	30.0	20.4	23.4	31.3
ξ^{2a}	101.2	110.5	141.0	101.3	112.8	152.1	100.6	110.9	141.8	101.5	112.6	150.0
ξ^{2b}	121.1	132.4	165.8	121.0	133.6	173.0	121.4	132.2	166.2	121.6	135.4	177.3
ξ^{3a}	545.3	577.1	661.8	543.9	578.1	684.8	543.4	575.2	662.2	547.0	579.9	684.2
ξ^{3b}	650.0	685.1	781.4	651.3	689.2	804.9	646.5	684.9	785.1	649.7	686.1	814.4

The Expected Shortfall at level $p = 0.95$ of the normalized total loss is approximated using $10,000$ independent samples of L. Total losses are normalized by 100 expected insured vehicles.

Table B.17: Value-at-Risk at level $p = 0.99$ of the normalized total loss.

	Binomial Model						Poisson Model					
	Gamma			Log-Normal			Gamma			Log-Normal		
	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$
Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	230.9	255.8	328.6	231.4	260.3	363.3	228.1	251.5	326.4	231.3	257.4	350.5
ξ^{1b}	222.8	248.2	320.1	225.7	251.8	352.6	224.7	247.2	314.3	221.7	254.6	344.5
ξ^{2a}	639.2	716.9	949.0	638.5	729.4	1038.6	638.1	719.9	966.6	631.7	743.1	1011.2
ξ^{2b}	644.1	719.7	973.5	650.6	744.5	1061.7	636.4	723.8	976.7	638.7	726.6	1039.2
ξ^{3a}	1001.8	1106.3	1537.6	994.3	1149.4	1666.7	992.1	1138.6	1592.0	993.1	1164.3	1586.3
ξ^{3b}	1030.2	1153.4	1564.6	1032.9	1211.2	1709.4	1019.7	1149.2	1602.9	1023.4	1202.4	1706.6
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	176.8	185.5	217.4	176.0	188.7	228.9	177.5	186.1	218.9	176.5	187.6	224.7
ξ^{1b}	198.5	211.4	261.5	198.7	214.1	252.6	196.8	213.4	258.2	198.4	214.7	250.5
ξ^{2a}	481.4	515.4	613.4	477.8	515.0	640.2	475.3	512.8	607.1	478.9	516.6	628.1
ξ^{2b}	594.9	637.1	741.7	597.4	635.0	766.9	600.2	630.0	736.9	594.8	629.4	759.1
ξ^{3a}	837.7	896.7	1077.6	846.1	902.9	1121.5	839.5	896.1	1080.0	849.8	906.0	1118.0
ξ^{3b}	1080.4	1152.5	1343.1	1090.0	1158.2	1429.6	1092.6	1145.7	1356.0	1086.9	1144.1	1388.1
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	141.0	149.3	175.4	141.1	150.1	176.5	139.8	149.6	172.7	141.4	150.9	174.4
ξ^{1b}	172.5	183.5	214.3	173.3	184.3	214.3	173.5	181.4	209.1	172.1	183.0	216.6
ξ^{2a}	463.0	485.6	553.8	463.8	482.6	565.2	464.5	486.3	548.5	461.7	483.3	570.4
ξ^{2b}	566.7	592.3	664.8	571.4	594.2	691.4	565.2	589.5	669.3	566.1	596.0	680.3
ξ^{3a}	931.2	975.2	1088.0	931.2	986.4	1124.9	932.5	976.4	1112.4	933.3	966.9	1129.7
ξ^{3b}	1266.5	1325.8	1460.7	1276.4	1328.3	1513.6	1272.0	1317.8	1466.4	1270.1	1328.8	1508.8
Non-Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	42.4	54.1	93.3	43.0	54.9	96.5	42.6	55.0	90.9	42.6	57.1	90.2
ξ^{1b}	41.9	55.5	94.8	41.9	56.5	91.2	43.8	54.3	90.8	41.8	56.5	92.9
ξ^{2a}	208.2	257.2	406.2	205.6	260.5	420.6	207.6	260.5	415.7	205.6	269.2	414.3
ξ^{2b}	219.8	265.7	445.9	212.3	281.8	436.5	217.9	275.0	423.6	216.0	277.8	437.2
ξ^{3a}	556.2	677.5	1024.5	554.1	690.3	1015.3	561.5	656.4	1036.9	564.9	708.9	1099.2
ξ^{3b}	650.9	755.7	1138.7	649.3	792.7	1218.2	632.2	776.4	1113.5	643.3	765.0	1204.7
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	23.3	27.6	41.0	23.7	28.2	43.3	23.4	27.7	41.4	23.7	28.8	45.3
ξ^{1b}	25.6	30.0	43.3	26.3	31.7	45.3	25.6	30.1	43.1	26.0	30.8	45.4
ξ^{2a}	153.3	171.3	237.0	154.6	174.4	250.6	153.6	174.6	238.7	156.8	176.4	256.7
ξ^{2b}	180.8	202.4	269.2	180.6	203.7	283.3	179.1	200.3	269.6	180.3	206.9	280.9
ξ^{3a}	575.1	628.8	779.7	574.7	637.4	828.9	571.1	630.1	787.3	570.5	635.7	837.6
ξ^{3b}	765.1	826.7	984.4	767.4	829.6	1053.9	768.0	829.6	1000.4	770.8	831.0	1040.6
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	19.4	22.1	29.6	19.2	22.3	31.7	19.2	21.8	30.3	19.4	22.2	31.5
ξ^{1b}	21.3	24.5	32.5	21.3	24.3	36.2	21.5	24.4	32.9	21.5	25.2	35.2
ξ^{2a}	105.6	116.8	152.7	106.1	120.1	164.1	105.3	116.7	154.5	105.8	120.6	168.9
ξ^{2b}	125.9	139.2	180.3	125.9	141.7	188.2	126.1	138.7	179.4	126.3	143.3	193.6
ξ^{3a}	560.4	596.3	697.9	561.7	599.4	716.2	559.2	593.7	694.5	562.0	600.4	727.0
ξ^{3b}	665.8	706.1	818.4	666.9	715.3	843.8	662.2	707.9	826.9	666.9	707.9	858.9

The Value-at-Risk at level $p = 0.99$ of the normalized total loss is approximated using $10,000$ independent samples of L. Total losses are normalized by 100 expected insured vehicles.

Table B.18: Expected Shortfall at level $p = 0.99$ of the normalized total loss.

	Binomial Model						Poisson Model					
	Gamma			Log-Normal			Gamma			Log-Normal		
	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$	$c_v = 0.5$	$c_v = 1.0$	$c_v = 2.0$
Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	244.8	279.0	372.4	244.2	290.2	436.5	244.6	272.8	368.4	247.4	277.5	432.8
ξ^{1b}	235.8	264.9	363.8	239.0	276.4	427.1	236.9	265.1	353.9	236.5	273.3	424.3
ξ^{2a}	683.4	783.4	1106.1	684.5	805.2	1314.1	682.2	783.5	1123.5	687.7	825.5	1304.0
ξ^{2b}	692.5	798.9	1126.1	694.5	844.3	1378.0	678.1	795.5	1111.4	693.3	791.9	1287.4
ξ^{3a}	1077.9	1214.4	1830.1	1078.4	1291.4	2104.3	1063.4	1260.7	1823.4	1077.3	1275.6	2058.2
ξ^{3b}	1112.1	1252.7	1829.2	1113.1	1349.2	2217.3	1091.5	1259.8	1836.1	1121.6	1328.4	2140.8
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	183.6	196.1	234.8	181.9	198.8	262.3	184.8	195.4	237.3	183.5	199.7	250.5
ξ^{1b}	207.7	224.3	300.7	207.4	231.4	308.0	206.7	228.9	295.8	207.4	236.7	316.6
ξ^{2a}	501.8	542.7	663.4	496.4	548.2	751.5	496.1	536.5	656.6	500.9	545.9	721.8
ξ^{2b}	617.5	674.0	798.5	620.0	669.5	871.2	620.1	661.1	796.0	615.2	663.6	858.4
ξ^{3a}	872.0	945.0	1159.5	879.6	952.0	1307.7	876.2	953.8	1185.6	879.3	961.8	1273.3
ξ^{3b}	1119.5	1203.8	1436.0	1126.5	1217.1	1621.0	1136.0	1198.9	1455.7	1126.4	1194.1	1549.1
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	146.3	158.8	194.8	146.5	159.5	203.1	145.2	157.0	191.0	146.8	158.9	202.8
ξ^{1b}	178.9	193.0	239.4	179.3	196.9	253.9	179.6	190.5	231.7	178.7	194.5	246.9
ξ^{2a}	477.3	502.5	580.8	479.1	510.7	642.3	478.9	509.3	586.3	481.3	504.2	655.9
ξ^{2b}	585.5	613.7	706.1	588.2	620.2	756.9	588.5	611.4	713.3	581.6	622.8	765.7
ξ^{3a}	969.5	1009.4	1160.8	958.9	1026.4	1248.0	962.6	1007.3	1163.3	959.6	1011.1	1254.6
ξ^{3b}	1302.2	1369.6	1534.0	1310.8	1374.5	1650.4	1302.8	1357.3	1538.5	1309.4	1370.4	1623.2
Non-Uniform Accident Occurrence												
$\rho^{\Phi} = \mathbf{0.1}$:												
ξ^{1a}	47.7	64.3	118.8	48.4	67.0	146.4	48.3	65.6	116.8	48.3	72.0	136.2
ξ^{1b}	47.9	64.7	119.2	48.7	70.5	137.9	49.6	62.8	119.1	47.4	69.9	150.6
ξ^{2a}	233.4	304.5	520.4	233.9	310.2	562.0	229.9	298.4	524.1	233.2	331.3	575.2
ξ^{2b}	248.1	304.3	558.1	240.4	342.9	579.1	242.2	316.6	537.5	241.3	330.2	607.1
ξ^{3a}	607.4	760.6	1226.9	623.5	816.5	1432.4	610.4	747.4	1223.0	617.0	840.0	1503.8
ξ^{3b}	713.1	856.2	1363.6	708.7	913.1	1630.9	688.1	860.9	1319.0	712.3	877.6	1715.2
$\rho^{\Phi} = \mathbf{0.5}$:												
ξ^{1a}	25.4	30.4	47.1	25.6	32.1	63.4	25.6	30.5	48.3	25.9	32.3	60.0
ξ^{1b}	27.8	33.1	50.0	28.5	35.3	59.5	27.7	33.5	49.2	28.1	35.1	63.5
ξ^{2a}	163.3	189.0	266.1	166.5	191.7	321.4	163.4	188.8	269.3	168.6	196.4	325.4
ξ^{2b}	192.9	219.1	302.9	192.0	222.6	363.4	189.4	218.0	305.4	191.4	228.3	340.4
ξ^{3a}	597.0	664.0	848.6	600.3	680.3	994.5	596.4	670.8	848.3	600.3	674.6	1022.3
ξ^{3b}	797.7	872.3	1065.1	800.6	868.7	1256.1	804.5	874.9	1080.6	803.7	873.4	1220.8
$\rho^{\Phi} = \mathbf{0.9}$:												
ξ^{1a}	20.9	24.2	33.4	20.5	24.6	39.0	20.6	24.1	34.6	20.8	25.0	41.9
ξ^{1b}	22.9	26.7	37.1	23.1	27.1	46.2	23.0	26.5	37.1	23.1	27.9	43.4
ξ^{2a}	111.2	125.3	168.8	112.9	130.8	213.7	111.4	125.9	172.9	112.5	130.8	207.5
ξ^{2b}	132.9	149.0	198.3	133.2	152.4	226.8	132.0	147.8	195.4	134.3	156.1	241.2
ξ^{3a}	584.0	627.9	750.6	579.8	625.8	822.2	578.0	620.5	746.1	585.7	628.3	818.5
ξ^{3b}	686.8	738.3	872.5	694.3	750.1	959.2	682.4	736.4	885.2	689.8	740.7	977.6

The Expected Shortfall at level $p = 0.99$ of the normalized total loss is approximated using $10,000$ independent samples of L. Total losses are normalized by 100 expected insured vehicles.

Bibliography

Acemoglu, D., A. Makhdoumi, A. Malekian & A. Ozdaglar (2018). "Informational Braess' Paradox: The Effect of Information on Traffic Congestion". *Operations Research* 66 (4), pp. 893–917.

Adacher, L. & M. Tiriolo (2018). "A Macroscopic Model with the Advantages of Microscopic Model: A Review of Cell Transmission Model's Extensions for Urban Traffic Networks". *Simulation Modelling Practice and Theory* 86, pp. 102–119.

Ankenman, B., B. L. Nelson & J. Staum (2010). "Stochastic Kriging for Simulation Metamodeling". *Operations Research* 58 (2), pp. 371–382.

Arief, M., P. Glynn & D. Zhao (2018). "An Accelerated Approach to Safely and Efficiently Test Pre-Production Autonomous Vehicles on Public Streets". *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 2006–2011.

Artzner, P., F. Delbaen, J.-M. Eber & D. Heath (1999). "Coherent Measures of Risk". *Mathematical Finance* 9 (3), pp. 203–228.

Bando, M., K. Hasebe, A. Nakayama, A. Shibata & Y. Sugiyama (1994). "Structure Stability of Congestion in Traffic Dynamics". *Japan Journal of Industrial and Applied Mathematics* 11 (2), pp. 203–223.

Bando, M., K. Hasebe, A. Nakayama, A. Shibata & Y. Sugiyama (1995). "Dynamical Model of Traffic Congestion and Numerical Simulation". *Physical Review E* 51 (2), pp. 1035–1042.

Bera, S. & K. V. K. Rao (2011). "Estimation of Origin-Destination Matrix from Traffic Counts: The State of the Art". *European Transport / Trasporti Europei* 49, pp. 2–23.

Berkhahn, V., M. Kleiber, J. Langner, C. Timmermann & S. Weber (2022). "Traffic Dynamics at Intersections Subject to Random Misperception". *IEEE Transactions on Intelligent Transportation Systems* 23 (5), pp. 4501–4511.

Berkhahn, V., M. Kleiber, C. Schiermeyer & S. Weber (2018). "Modeling Traffic Accidents Caused by Random Misperception". *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 2568–2574.

Bertolazzi, E. & M. Frego (2015). "G1 Fitting with Clothoids". *Mathematical Methods in the Applied Sciences* 38 (5), pp. 881–897.

Bertoncello, M. & D. Wee (2015). *Ten Ways Autonomous Driving Could Redefine the Automotive World*. URL: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/ten-ways-autonomous-driving-could-redefine-the-automotive-world.

Biagini, F., J.-P. Fouque, M. Frittelli & T. Meyer-Brandis (2018). "A Unified Approach to Systemic Risk Measures via Acceptance Sets". *Mathematical Finance* 29 (1), pp. 329–367.

Binois, M., R. B. Gramacy & M. Ludkovski (2018). "Practical Heteroscedastic Gaussian Process Modeling for Large Simulation Experiments". *Journal of Computational and Graphical Statistics* 27 (4), pp. 808–821.

Blanco, M., J. Atwood, S. Russell, T. Trimble, J. McClafferty & M. Perez (2016). *Automated Vehicle Crash Rate Comparison Using Naturalistic Data*. Tech. rep. Virginia Tech Transportation Institute.

Brunel, V.-E. (2018). "Methods for Estimation of Convex Sets". *Statistical Science* 33 (4).

Bundesministerium für Verkehr und digitale Infrastruktur (2017). *German Road Traffic Regulations*. URL: https://www.bmvi.de/SharedDocs/EN/publications/german-road-traffic-regulations.html.

Carathéodory, C. (1918). *Vorlesungen über reelle Funktionen*. Teubner.

Carlino, D., S. D. Boyles & P. Stone (2013). "Auction-Based Autonomous Intersection Management". *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, pp. 529–534.

Casas, J., J. L. Ferrer, D. Garcia, J. Perarnau & A. Torday (2010). "Traffic Simulation with Aimsun". *International Series in Operations Research & Management Science*. Springer New York, pp. 173–232.

Cassidy, A., Z. Feinstein & A. Nehorai (2016). "Risk Measures for Power Failures in Transmission Systems". *Chaos: An Interdisciplinary Journal of Nonlinear Science* 26 (11), p. 113110.

Chen, L. H., L. Goldstein & Q.-M. Shao (2011). *Normal Approximation by Stein's Method*. Springer Berlin Heidelberg.

Chen, S., M. Kuhn, K. Prettner & D. E. Bloom (2019). "The Global Macroeconomic Burden of Road Injuries: Estimates and Projections for 166 Countries". *The Lancet Planetary Health* 3 (9), e390–e398.

Chen, Z. & B. Wang (2018). "How Priors of Initial Hyperparameters Affect Gaussian Process Regression Models". *Neurocomputing* 275, pp. 1702–1710.

Colini-Baldeschi, R., R. Cominetti, P. Mertikopoulos & M. Scarsini (2020). "When is Selfish Routing Bad? The Price of Anarchy in Light and Heavy Traffic". *Operations Research* 68 (2), pp. 411–434.

Cuevas, A. (2009). "Set Estimation: Another Bridge Between Statistics and Geometry". *Boletín de Estadística e Investigación Operativa* 25 (2), pp. 71–85.

Daganzo, C. F. (1981). "Estimation of Gap Acceptance Parameters Within and Across the Population from Direct Roadside Observation". *Transportation Research Part B: Methodological* 15 (1), pp. 1–15.

Daganzo, C. F. (1994). "The Cell Transmission Model: A Dynamic Representation of Highway Traffic Consistent with the Hydrodynamic Theory". *Transportation Research Part B: Methodological* 28 (4), pp. 269–287.

Daganzo, C. F. (1995). "The Cell Transmission Model, Part II: Network Traffic". *Transportation Research Part B: Methodological* 29 (2), pp. 79–93.

De Berg, M., M. Van Kreveld, M. Overmars & O. C. Schwarzkopf (2000). "Visibility Graphs". *Computational Geometry: Algorithms and Applications*. Springer. Chap. 15, pp. 307–317.

Denuit, M., X. Maréchal, S. Pitrebois & J. F. Walhin (2007). *Actuarial Modelling of Claim Counts: Risk Classification, Credibility and Bonus-Malus Systems*. John Wiley & Sons.

Diekmann, N. & C Schiermeyer (2018). "Modellierung und Simulation von ÖPNV auf gemeinsam genutzten Verkehrsflächen". *Forum Bauinformatik 2018. Von jungen Forschenden für junge Forschende*. 30. Forum Bauinformatik (Bauhaus Universität Weimar). Ed. by M. Steiner, M. Theiler & M. Mirboland.

Diekmann, N. & C Schiermeyer (2019). "Modellierung eines gerichteten Graphen zur Abbildung von Vorfahrtsregeln". *Forum Bauinformatik 2019. Von jungen Forschenden für junge Forschende*. 31. Forum Bauinformatik (Technische Universität Weimar). Ed. by M. Sternal, L.-C. Ungureanu, L. Böger & C. Bindal-Gutsche.

Dresner, K. & P. Stone (2008). "A Multiagent Approach to Autonomous Intersection Management". *Journal of Artificial Intelligence Research* 31, pp. 591–656.

El Karoui, N. & Y. Jiao (2009). "Stein's Method and Zero Bias Transformation for CDO Tranche Pricing". *Finance & Stochastics* 13, pp. 151–180.

Erdmann, J. & D. Krajzewicz (2014). "SUMO's Road Intersection Model". *Simulation of Urban Mobility*. Springer Berlin Heidelberg, pp. 3–17.

Esser, J. & M. Schreckenberg (1997). "Microscopic Simulation of Urban Traffic Based on Cellular Automata". *International Journal of Modern Physics C* 08 (05), pp. 1025–1036.

Evans, L. & R. Rothery (1974). "Detection of the Sign of Relative Motion When following a Vehicle". *Human Factors* 16 (2), pp. 161–173.

Farin, G. (1994). *Kurven und Flächen im Computer Aided Geometric Design*. Vieweg + Teubner Verlag.

Feinstein, Z., M. Kleiber & S. Weber (2023). "Stochastic Cell Transmission Models of Traffic Networks". *Working Paper*.

Feinstein, Z., B. Rudloff & S. Weber (2017). "Measures of Systemic Risk". *SIAM Journal on Financial Mathematics* 8 (1), pp. 672–708.

Fellendorf, M. & P. Vortisch (2010). "Microscopic Traffic Flow Simulator VISSIM". *International Series in Operations Research & Management Science*. Springer New York, pp. 63–93.

Föllmer, H. & S. Weber (2015). "The Axiomatic Approach to Risk Measures for Capital Determination". *Annual Review of Financial Economics* 7 (1), pp. 301–337.

Flötteröd, G., M. Bierlaire & K. Nagel (2011). "Bayesian Demand Calibration for Dynamic Traffic Simulations". *Transportation Science* 45 (4), pp. 541–561.

Flötteröd, G. & G. Lämmel (2015). "Bidirectional Pedestrian Fundamental Diagram". *Transportation Research Part B: Methodological* 71, pp. 194–212.

Freund, D., S. G. Henderson & D. B. Shmoys (2022). "Minimizing Multimodular Functions and Allocating Capacity in Bike-Sharing Systems". *Operations Research* 70 (5), pp. 2715–2731.

Frey, R., M. Popp & S. Weber (2008). "An Approximation for Credit Portfolio Losses". *The Journal of Credit Risk* 4 (1), pp. 3–20.

Gao, G., S. Meng & M. V. Wüthrich (2022). "What Can We Learn from Telematics Car Driving Data: A Survey". *Insurance: Mathematics and Economics* 104, pp. 185–199.

Gazis, D. C. (2002). "The Origins of Traffic Theory". *Operations Research* 50 (1), pp. 69–77.

Geiger, A., P. Lenz & R. Urtasun (2012). "Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite". *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE.

Geroliminis, N. & C. F. Daganzo (2008). "Existence of Urban-Scale Macroscopic Fundamental Diagrams: Some Experimental Findings". *Transportation Research Part B: Methodological* 42 (9), pp. 759–770.

Glasserman, P. (2003). *Monte Carlo Methods in Financial Engineering*. Springer New York.

Goan, E. & C. Fookes (2020). "Bayesian Neural Networks: An Introduction and Survey". *Case Studies in Applied Bayesian Data Science*. Springer International Publishing, pp. 45–87.

Goldberg, P., C. Williams & C. Bishop (1997). "Regression with Input-dependent Noise: A Gaussian Process Treatment". *Advances in Neural Information Processing Systems*. Ed. by M. Jordan, M. Kearns & S. Solla. Vol. 10. MIT Press.

Gotovos, A., N. Casati, G. Hitz & A. Krause (2013). "Active Learning for Level Set Estimation". *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. IJCAI '13. Beijing, China: AAAI Press, 1344–1350.

Hadiuzzaman, M. & T. Z. Qiu (2013). "Cell Transmission Model Based Variable Speed Limit Control for Freeways". *Canadian Journal of Civil Engineering* 40 (1), pp. 46–56.

Hamdar, S. H., M. Treiber, H. S. Mahmassani & A. Kesting (2008). "Modeling Driver Behavior as Sequential Risk-Taking Task". *Transportation Research Record* 2088 (1), pp. 208–217.

Han, X. & P. E. Kloeden (2017). *Random Ordinary Differential Equations and Their Numerical Solution*. Springer Singapore.

Hart, P. E., N. J. Nilsson & B. Raphael (1968). "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". *IEEE Transactions on Systems Science and Cybernetics* 4 (2), pp. 100–107.

Hawkes, A. G. (1968). "Gap-Acceptance in Road Traffic". *Journal of Applied Probability* 5 (1), pp. 84–92.

Helbing, D. (2001). "Traffic and Related Self-Driven Many-Particle Systems". *Reviews of Modern Physics* 73 (4), pp. 1067–1141.

Helbing, D. & P. Molnar (1995). "Social Force Model for Pedestrian Dynamics". *Physical Review E* 51 (5), p. 4282.

Henckaerts, R. & K. Antonio (2022). "The Added Value of Dynamically Updating Motor Insurance Prices with Telematics Collected Driving Behavior Data". *Insurance: Mathematics and Economics* 105, pp. 79–95.

Höcker, M. (2010). "Modellierung und Simulation von Fußgängerverkehr". PhD thesis. Leibniz Universität Hannover.

Höcker, M., V. Berkhahn, A. Kneidl, A. Borrmann & W. Klein (2010). "Graph-Based Approaches for Simulating Pedestrian Dynamics in Building Models". *eWork and eBusiness in Architecture, Engineering and Construction*, pp. 389–394.

Hoff, P. D. (2009). *A First Course in Bayesian Statistical Methods*. Springer New York.

Hoschek, J. & D. Lasser (1992). *Grundlagen der geometrischen Datenverarbeitung*. Vieweg + Teubner Verlag.

Husnjak, S., D. Peraković, I. Forenbacher & M. Mumdziev (2015). "Telematics System in Usage Based Motor Insurance". *Procedia Engineering* 100, pp. 816–825.

Jentzen, A. & P. E. Kloeden (2009). "Pathwise Taylor Schemes for Random Ordinary Differential Equations". *BIT Numerical Mathematics* 49 (1), pp. 113–140.

Jin, L. & S. Amin (2019). "Analysis of a Stochastic Switching Model of Freeway Traffic Incidents". *IEEE Transactions on Automatic Control* 64 (3), pp. 1093–1108.

Jones, D., C. Snider, A. Nassehi, J. Yon & B. Hicks (2020). "Characterising the Digital Twin: A Systematic Literature Review". *CIRP Journal of Manufacturing Science and Technology* 29, pp. 36–52.

Kanagawa, M., P. Hennig, D. Sejdinovic & B. K. Sriperumbudur (2018). "Gaussian Processes and Kernel Methods: A Review on Connections and Equivalences". arXiv: `1807.02582 [stat.ML]`.

Kesting, A. & M. Treiber (2008). "Calibrating Car-Following Models by Using Trajectory Data". *Transportation Research Record* 2088 (1), pp. 148–156.

Kesting, A., M. Treiber & D. Helbing (2007). "General Lane-Changing Model MOBIL for Car-Following Models". *Transportation Research Record* 1999 (1), pp. 86–94.

Kim, S., M. Kleiber & S. Weber (2022). "Microscopic Traffic Models, Accidents, and Insurance Losses". *Working Paper*.

Krauß, S. (1998). "Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics". PhD thesis. DLR Deutsches Zentrum für Luft- und Raumfahrt e.V.

Lalchand, V. & C. E. Rasmussen (2020). "Approximate Inference for Fully Bayesian Gaussian Process Regression". *Proceedings of The 2nd Symposium on Advances in Approximate Bayesian Inference*. Ed. by C. Zhang, F. Ruiz, T. Bui, A. B. Dieng & D. Liang. Vol. 118. Proceedings of Machine Learning Research. PMLR, pp. 1–12.

Laval, J. A., C. S. Toth & Y. Zhou (2014). "A Parsimonious Model for the Formation of Oscillations in Car-Following Models". *Transportation Research Part B: Methodological* 70, pp. 228–238.

Lederer, A., J. Umlauft & S. Hirche (2019). "Uniform Error Bounds for Gaussian Process Regression with Application to Safe Control". *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox & R. Garnett. Vol. 32. Curran Associates, Inc.

Lederer, A., J. Umlauft & S. Hirche (2021). "Uniform Error and Posterior Variance Bounds for Gaussian Process Regression with Application to Safe Control". arXiv: 2101.05328 [cs.LG].

Lehmann, H. (2000). "Microscopic Randomness in Follow-the-Leader Dynamics". *Traffic and Granular Flow '99*. Springer Berlin Heidelberg, pp. 395–400.

Levin, M. W. & S. D. Boyles (2016). "A Multiclass Cell Transmission Model for Shared Human and Autonomous Vehicle Roads". *Transportation Research Part C: Emerging Technologies* 62, pp. 103–116.

Lighthill, M. J. & G. B. Whitham (1955). "On Kinematic Waves II. A Theory of Traffic Flow on Long Crowded Roads". *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 229 (1178), pp. 317–345.

Liu, H., J. Wang, K. Wijayaratna, V. V. Dixit & S. T. Waller (2015). "Integrating the Bus Vehicle Class Into the Cell Transmission Model". *IEEE Transactions on Intelligent Transportation Systems* 16 (5), pp. 2620–2630.

Long, J., Z. Gao, X. Zhao, A. Lian & P. Orenstein (2008). "Urban Traffic Jam Simulation Based on the Cell Transmission Model". *Networks and Spatial Economics* 11 (1), pp. 43–64.

Lopez, P. A., E. Wiessner, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lucken, J. Rummel & P. Wagner (2018). "Microscopic Traffic Simulation using SUMO". *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 2575–2582.

Lord, D. & F. Mannering (2010). "The Statistical Analysis of Crash-Frequency Data: A Review and Assessment of Methodological Alternatives". *Transportation Research Part A: Policy and Practice* 44 (5), pp. 291–305.

Lücken, L., E. Mintsis, K. N. Porfyri, R. Alms, Y.-P. Flötteröd & D. Koutras (2019). "From Automated to Manual - Modeling Control Transitions with SUMO". *SUMO User Conference 2019*.

Lyu, X., M. Binois & M. Ludkovski (2021). "Evaluating Gaussian Process Metamodels and Sequential Designs for Noisy Level Set Estimation". *Statistics and Computing* 31 (4).

Mahmassani, H. & Y. Sheffi (1981). "Using Gap Sequences to Estimate Gap Acceptance Functions". *Transportation Research Part B: Methodological* 15 (3), pp. 143–148.

Maxwell, M. S., M. Restrepo, S. G. Henderson & H. Topaloglu (2010). "Approximate Dynamic Programming for Ambulance Redeployment". *INFORMS Journal on Computing* 22 (2), pp. 266–281.

Maze, T. H., M. Agarwal & G. Burchett (2006). "Whether Weather Matters to Traffic Demand, Traffic Safety, and Traffic Operations and Flow". *Transportation Research Record: Journal of the Transportation Research Board* 1948 (1), pp. 170–176.

McNeil, A. J., R. Frey & P. Embrechts (2015). *Quantitative Risk Management: Concepts, Techniques and Tools*. Revised Edition. Princeton University Press.

Meek, D. & D. Walton (2004). "A Note on Finding Clothoids". *Journal of Computational and Applied Mathematics* 170 (2), pp. 433–453.

Meier, R. (2019). "Modellierung des Verhaltens in Verkehrssituationen mit verdecktem Wahrnehmungsbereich". Bachelorarbeit. Institut für Risiko und Zuverlässigkeit, Leibniz Universität Hannover.

Mitra, P., A. Choudhury, V. R. Aparow, G. Kulandaivelu & J. Dauwels (2018). "Towards Modeling of Perception Errors in Autonomous Vehicles". *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3024–3029.

Moustaid, E. & G. Flötteröd (2021). "Macroscopic Model of Multidirectional Pedestrian Network Flows". *Transportation Research Part B: Methodological* 145, pp. 1–23.

Munoz, L., X. Sun, R. Horowitz & L. Alvarez (2003). "Traffic Density Estimation with the Cell Transmission Model". *Proceedings of the 2003 American Control Conference*. IEEE.

Nekovee, M. & J. Bie (2013). "Rear-End Collision: Causes and Avoidance Techniques". *Wireless Vehicular Networks for Car Collision Avoidance*. Springer New York, pp. 99–119.

NHTSA (2022). "Early Estimates of Motor Vehicle Traffic Fatalities And Fatality Rate by Sub-Categories in 2021". *US Department of Transportation*, pp. 1–10.

Nigro, M., E. Cipriani & A. del Giudice (2018). "Exploiting Floating Car Data for Time-Dependent Origin–Destination Matrices Estimation". *Journal of Intelligent Transportation Systems* 22 (2), pp. 159–174.

Nikolova, E. & N. E. Stier-Moses (2014). "A Mean-Risk Model for the Traffic Assignment Problem with Stochastic Travel Times". *Operations Research* 62 (2), pp. 366–382.

Norden, J., M. O'Kelly & A. Sinha (2019). "Efficient Black-box Assessment of Autonomous Vehicle Safety". arXiv: `1912.03618 [cs.LG]`.

Ortelli, N., M. de Lapparent & M. Bierlaire (2021). "Can We Infer on Behavioral Impacts of Public Policy on Accident Severity Outcomes?" *21st Swiss Transport Research Conference*.

Osorio, C. & M. Bierlaire (2013). "A Simulation-Based Optimization Framework for Urban Transportation Problems". *Operations Research* 61 (6), pp. 1333–1345.

Osorio, C. & K. Nanduri (2015). "Energy-Efficient Urban Traffic Management: A Microscopic Simulation-Based Approach". *Transportation Science* 49 (3), pp. 637–651.

Osorio, C. & V. Punzo (2019). "Efficient Calibration of Microscopic Car-Following Models for Large-Scale Stochastic Network Simulators". *Transportation Research Part B: Methodological* 119, pp. 156–173.

Ossen, S. & S. P. Hoogendoorn (2008). "Validity of Trajectory-Based Calibration Approach of Car-Following Models in Presence of Measurement Errors". *Transportation Research Record* 2088 (1), pp. 117–125.

Pagany, R. (2020). "Wildlife-Vehicle Collisions - Influencing Factors, Data Collection and Research Methods". *Biological Conservation* 251, p. 108758.

Pascucci, F., N. Rinke, C. Schiermeyer, B. Friedrich & V. Berkhahn (2015). "Modeling of Shared Space with Multi-Modal Traffic Using a Multi-Layer Social Force Approach". *Transportation Research Procedia* 10, pp. 316–326.

Pascucci, F. (2020). "A Microsimulation Based Method to Evaluate Shared Space Performances". PhD thesis. Technische Universität Carolo-Wilhelmina zu Braunschweig.

Pascucci, F., N. Rinke, C. Schiermeyer, V. Berkhahn & B. Friedrich (2018). "Should I Stay or Should I Go? A Discrete Choice Model for Pedestrian–Vehicle Conflicts in Shared Space". Transportation Research Board 97th Annual Meeting. Washington DC, United States.

Patriksson, M. (2015). *The Traffic Assignment Problem: Models and Methods*. Dover Pubn Inc.

Phanse, S., M. Chaturvedi & S. Srivastava (2022). "Modelling and Simulation of Road Traffic Under Rainy Conditions". *2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS)*. IEEE.

Pohlmann, T. & B. Friedrich (2010). "Online Control of Signalized Networks Using the Cell Transmission Model". *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE.

Pollatschek, M. A., A. Polus & M. Livneh (2002). "A Decision Model for Gap Acceptance and Capacity at Intersections". *Transportation Research Part B: Methodological* 36 (7), pp. 649–663.

Punzo, V., B. Ciuffo & M. Montanino (2012). "Can Results of Car-Following Model Calibration Based on Trajectory Data be Trusted?" *Transportation Research Record* 2315 (1), pp. 11–24.

Ramer, U. (1972). "An Iterative Procedure for the Polygonal Approximation of Plane Curves". *Computer Graphics and Image Processing* 1 (3), pp. 244 –256.

Rasmussen, C. E. & C. K. I. Williams (2005). *Gaussian Processes for Machine Learning*. MIT Press Ltd. 272 pp.

Reinhold Baier et al. (2006). *Richtlinien für die Anlage von Stadtstraßen - RASt 06*. Forschungsgesellschaft für Straßen- und Verkehrswesen e.V.

Richards, P. I. (1956). "Shock Waves on the Highway". *Operations Research* 4 (1), pp. 42–51.

Rinke, N., C. Schiermeyer, F. Pascucci, V. Berkhahn & B. Friedrich (2017). "A Multi-Layer Social Force Approach to Model Interactions in Shared Spaces Using Collision Prediction". *Transportation Research Procedia*. Vol. 25. Elsevier, pp. 1249–1267.

Ross, N. (2011). "Fundamentals of Stein's method". *Probability Surveys* 8, pp. 210–293.

Salomon, J., M. Broggi, S. Kruse, S. Weber & M. Beer (2020). "Resilience Decision-Making for Complex Systems". *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg* 6 (2).

Schiermeyer, C., F. Pascucci, N. Rinke, V. Berkhahn & B. Friedrich (2017). "MODIS - Ein Simulationsmodell zur Bewertung von Verkehrsflächen nach dem Shared-Space-Gestaltungsprinzip". *Bauingenieur. Die Richtungsweisende Zeitschrift im Bauingenieurwesen* (92) (Juli/August 2017): *Verkehr und Infrastruktur*, pp. 341–346.

Schiermeyer, C., F. Pascucci, N. Rinke, V. Berkhahn & B. Friedrich (2016). "A Genetic Algorithm Approach for the Calibration of a Social Force Based Model for Shared Spaces". *Proceedings of the 8th International Conference on Pedestrian and Evacuation Dynamics (PED)*, pp. 485–491.

Schiermeyer, C., F. Pascucci, N. Rinke, V. Berkhahn & B. Friedrich (2019). "Modeling and Solving of Multiple Conflict Situations in Shared Spaces". *Traffic and Granular Flow '17*. Springer International Publishing, pp. 451–458.

Schulz, E., M. Speekenbrink & A. Krause (2018). "A Tutorial on Gaussian Process Regression: Modelling, Exploring, and Exploiting Functions". *Journal of Mathematical Psychology* 85, pp. 1–16.

Segata, M. & R. L. Cigno (2013). "Automatic Emergency Braking: Realistic Analysis of Car Dynamics and Network Performance". *IEEE Transactions on Vehicular Technology* 62 (9), pp. 4150–4161.

Sharma, A., Z. Zheng & A. Bhaskar (2019). "Is More Always Better? The Impact of Vehicular Trajectory Completeness on Car-Following Model Calibration and Validation". *Transportation Research Part B: Methodological* 120, pp. 49–75.

Soriguera, F. (2012). "Deriving Traffic Flow Patterns from Historical Data". *Journal of Transportation Engineering* 138 (12), pp. 1430–1441.

Srinivas, N., A. Krause, S. M. Kakade & M. W. Seeger (2012). "Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting". *IEEE Transactions on Information Theory* 58 (5), pp. 3250–3265.

Srivastava, A., W.-L. Jin & J.-P. Lebacque (2015). "A Modified Cell Transmission Model with Realistic Queue Discharge Features at Signalized Intersections". *Transportation Research Part B: Methodological* 81, pp. 302–315.

Sumalee, A., R. Zhong, T. Pan & W. Szeto (2011). "Stochastic Cell transmission Model (SCTM): A Stochastic Dynamic Traffic Model for Traffic State Surveillance and Assignment". *Transportation Research Part B: Methodological* 45 (3), pp. 507–533.

Swiler, L., M. Gulian, A. Frankel, C. Safta & J. Jakeman (2020). "A Survey of Constrained Gaussian Process Regression: Approaches and Implementation Challenges". *Journal of Machine Learning for Modeling and Computing, 1(2):119-156 (2020)*. arXiv: 2006.09319 [cs.LG].

Taieb-Maimon, M. & D. Shinar (2001). "Minimum and Comfortable Driving Headways: Reality versus Perception". *Human Factors* 43 (1), pp. 159–172.

Tampère, C. M., R. Corthout, D. Cattrysse & L. H. Immers (2011). "A Generic Class of First Order Node Models for Dynamic Macroscopic Simulation of Traffic Flows". *Transportation Research Part B: Methodological* 45 (1), pp. 289–309.

Tang, L., Q. He, D. Wang & C. Qiao (2022). "Multi-Modal Traffic Signal Control in Shared Space Street". *IEEE Transactions on Intelligent Transportation Systems* 23 (1), pp. 392–403.

Tanner, J. C. (1962). "A Theoretical Analysis of Delays at an Uncontrolled Intersection". *Biometrika* 49 (1/2), p. 163.

Tiaprasert, K., Y. Zhang, C. Aswakul, J. Jiao & X. Ye (2017). "Closed-Form Multiclass Cell Transmission Model Enhanced with Overtaking, Lane-Changing, and First-In First-Out Properties". *Transportation Research Part C: Emerging Technologies* 85, pp. 86–110.

Timmermann, C. (2022). "Ein Modell zur mikroskopischen Simulation interaktionsbasierter Verkehrsdynamik". PhD thesis. Leibniz Universität Hannover.

Timmermann, C., M. Kleiber & S. Weber (2023). "Microscopic Traffic Simulations with the MODIS Framework". *Working Paper*.

Treiber, M., A. Hennecke & D. Helbing (2000). "Congested Traffic States in Empirical Observations and Microscopic Simulations". *Physical Review E* 62 (2), pp. 1805–1824.

Treiber, M. & A. Kesting (2017). "The Intelligent Driver Model with Stochasticity - New Insights into Traffic Flow Oscillations". *Transportation Research Procedia* 23, pp. 174–187.

Treiber, M. & A. Kesting (2018). "The Intelligent Driver Model with Stochasticity – New Insights into Traffic Flow Oscillations". *Transportation Research Part B: Methodological* 117, pp. 613–623.

Treiber, M., A. Kesting & D. Helbing (2006). "Delays, Inaccuracies and Anticipation in Microscopic Traffic Models". *Physica A: Statistical Mechanics and its Applications* 360 (1), pp. 71–88.

Trifunović, A., C. Timmermann, B. Friedrich & V. Berkhahn (2021). "Implications of Converting Low Capacity Intersection Adjacent to Park Into Shared Space". Transportation Research Board 100th Annual Meeting. Washington DC, United States.

Tsoi, A. H. & H. C. Gabler (2015). "Evaluation of Vehicle-Based Crash Severity Metrics". *Traffic Injury Prevention* 16 (sup2), S132–S139.

Tuerprasert, K. & C. Aswakul (2010). "Multiclass Cell Transmission Model for Heterogeneous Mobility in General Topology of Road Network". *Journal of Intelligent Transportation Systems* 14 (2), pp. 68–82.

U. S. Department of Transportation, Federal Highway Administration (2013). *Manual on Uniform Traffic Control Devices for Streets and Highways - 2009 Edition with 2012 Revisions*. Datamotion Publishing LLC. 860 pp.

Verbelen, R., K. Antonio & G. Claeskens (2018). "Unravelling the Predictive Power of Telematics Data in Car Insurance Pricing". *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 67 (5), pp. 1275–1304.

Wagner, P. (2016). "Traffic Control and Traffic Management in a Transportation System with Autonomous Vehicles". *Autonomous Driving*. Springer Berlin Heidelberg, pp. 301–316.

Weber, S. & K. Weske (2017). "The Joint Impact of Bankruptcy Costs, Fire Sales and Cross-Holdings on Systemic Risk in Financial Networks". *Probability, Uncertainty and Quantitative Risk* 2 (1).

Weber, T., P. Driesch & D. Schramm (2019). "Introducing Road Surface Conditions into a Microscopic Traffic Simulation". *SUMO User Conference 2019*.

Wegener, A., M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer & J.-P. Hubaux (2008). "TraCI". *Proceedings of the 11th Communications and Networking Simulation Symposium*. ACM Press.

World Health Organization (2019). *Global Status Report on Road Safety 2018*. 419 pp. URL: `https://www.ebook.de/de/product/36226085/world_health_organization_global_status_report_on_road_safety_2018.html`.

Wüthrich, M. V. (2013). "Non-Life Insurance: Mathematics & Statistics". *SSRN Electronic Journal*.

Xia, F., J. Wang, X. Kong, Z. Wang, J. Li & C. Liu (2018). "Exploring Human Mobility Patterns in Urban Scenarios: A Trajectory Data Perspective". *IEEE Communications Magazine* 56 (3), pp. 142–149.

Xie, B., M. Xu, J. Harri & Y. Chen (2013). "A Traffic Light Extension to Cell Transmission Model for Estimating Urban Traffic Jam". *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE.

Yang, X., H. Dong, Q. Wang, Y. Chen & X. Hu (2014). "Guided Crowd Dynamics via Modified Social Force Model". *Physica A: Statistical Mechanics and its Applications* 411, pp. 63–73.

Yu, R., Y. Zheng, M. Abdel-Aty & Z. Gao (2019). "Exploring Crash Mechanisms with Microscopic Traffic Flow Variables: A Hybrid Approach with Latent Class Logit and Path Analysis Models". *Accident Analysis & Prevention* 125, pp. 70–78.

Zhang, C., C. Osorio & G. Flötteröd (2017). "Efficient Calibration Techniques for Large-Scale Traffic Simulators". *Transportation Research Part B: Methodological* 97, pp. 214–239.

Zhao, D., X. Huang, H. Peng, H. Lam & D. J. LeBlanc (2018). "Accelerated Evaluation of Automated Vehicles in Car-Following Maneuvers". *IEEE Transactions on Intelligent Transportation Systems* 19 (3), pp. 733–744.

List of Figures

List of Tables

Acknowledgment

First of all, I would like to thank Stefan Weber for agreeing to supervise my thesis. His enthusiasm for research is infectious, and his willingness to engage in interdisciplinary topics is unsurpassed. Credit undoubtedly goes to him for the depth of this work.

Furthermore, I am grateful to Zachary Feinstein, Bernhard Friedrich, and Thomas Knispel for agreeing to review this work, and to Gunnar Friege for chairing the thesis committee.

As well as Stefan Weber, I would like to thank my co-authors Volker Berkhahn, Zachary Feinstein, Sojung Kim, Johannes Langner, and Chris Timmermann for our many stimulating discussions. Those fruitful exchanges form the basis of this work.

My thanks go to all my colleagues who have helped me along the way. In particular, I would like to thank Kerstin Awiszus, Sören Bettels, and Chris Timmermann. Without their encouragement, this work would not have been possible.

Finally, I would like to thank my friends and family who have always supported me. Above all, I am indescribably grateful to Friederike. She has the remarkable talent of always being able to help me find the way forward.

The results presented in this thesis were (partially) achieved by computations carried out on the cluster system at Leibniz Universität Hannover.

Publications

- The work presented in Chapter 2 of this thesis was previously published as *Modeling Traffic Accidents Caused by Random Misperception* in the proceedings of the *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. It is joint work with Volker Berkhahn, Chris Schiermeyer, and Stefan Weber.
- The work presented in Chapter 3 of this thesis was previously published as *Traffic Dynamics at Intersections Subject to Random Misperception* in *IEEE Transactions on Intelligent Transportation Systems*. It is joint work with Volker Berkhahn, Johannes Langner, Chris Timmermann, and Stefan Weber.
- The work presented in Chapter 4 is based on the working paper *Microscopic Traffic Models, Accidents, and Insurance Losses* which has been submitted for publication. It is joint work with Sojung Kim and Stefan Weber.
- The work presented in Chapter 5 is based on the working paper *Stochastic Cell Transmission Models of Traffic Networks* which has been submitted for publication. It is joint work with Zachary Feinstein and Stefan Weber.
- The work presented in Appendix A is based on the working paper *Microscopic Traffic Simulations with the MODIS Framework*. It is joint work with Chris Timmermann and Stefan Weber.