Traffic State Estimation with Multi-Agent Simulations

vorgelegt von
Diplom-Ingenieur
Gunnar Flötteröd
aus Bielefeld

Von der Fakultät V – Verkehrs- und Maschinensysteme
der Technischen Universität Berlin
zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
Dr.–Ing.

genehmigte Dissertation

Promotionsausschuß:

Vorsitzender: Prof. Dr. rer. nat. Volker Schindler
Gutachter: Prof. Dr. rer. nat. Kai Nagel
Gutachter: Prof. Michel Bierlaire, PhD in Mathematics
Gutachter: Dr. rer. nat. Peter Wagner

Tag der wissenschaftlichen Aussprache: 23. April 2008

Berlin 2008

D 83

Acknowledgments

I am indebted to Claudia, my family, and my friends, who were patient with me throughout these years and continuously offered encouragement and distraction.

I would like to express my gratitude to the members of my thesis committee, Prof. Kai Nagel, Prof. Michel Bierlaire, and Dr. Peter Wagner, for their interest and guidance.

Thanks to Johannes for programming much of the prototype simulator.

Prof. Tim Nattkemper and the Center for Biotechnology at the University of Bielefeld kindly made available their computing facilities at short notice. TU Berlin's mathematical faculty also provided computing time.

This research was partially funded by the German research society DFG under the grant "State estimation for traffic simulations as coarse grained systems".

Zusammenfassung

Die vorliegende Dissertation beschreibt ein neuartiges Verfahren zur gänzlich disaggregierten Nachführung des Mobilitätsverhaltens von Autofahrern auf Grundlage aggregierter Messungen von Verkehrsflüssen, -dichten oder -geschwindigkeiten, welche durch eine begrenzte Anzahl von Sensoren im Netzwerk aufgenommen werden. Das Problem wird mittels eines bayesschen Ansatzes gelöst, wobei das gegebene a priori Wissen über die Auswahlverteilung der Verhaltensalternativen eines jeden Individuums mit der Likelihood-Funktion der verfügbaren Messungen in eine geschätzte a posteriori Verhaltensverteilung kombiniert wird. Der Ansatz ist insofern simulationsbasiert, als daß (i) allein ein Simulationssystem zur Repräsentation der a priori Verhaltensannahmen benötigt wird und (ii) das Verfahren ausschließlich Ziehungen aus der a posteriori Verhaltensverteilung generiert.

Das Verfahren behandelt den Simulator des a priori Verhaltens soweit wie möglich als eine Black Box. Die nachführbaren Verhaltensdimensionen reichen von einfacher Routenwahl bis hin zur Auswahl von Plänen für einen ganzen Tag. Eine gleichgewichtsbasierte Modellierungsannahme ist ebenso zulässig wie ein Telematikmodell unvollständig informierter Fahrer.

Die Verwendung aggregierter Sensordaten zur disaggregierten Verhaltensschätzung wird durch eine kombinierte mikroskopische/makroskopische Mobilitätssimulation ermöglicht, welche individuelle Fahrzeuge auf Grundlage eines makroskopischen Modells der Verkehrsflussdynamik bewegt. Das Modell erlaubt eine lineare Vorhersage des Effektes von individuellem Verhalten auf den aggregierten Verkehrszustand und ermöglicht auf diese Weise eine lineare Approximation der logarithmierten Likelihood-Funktion der Sensordaten in Abhängigkeit von dem Verhalten der Fahrerpopulation. Diese Linearisierung wird von zwei operativen bayesschen Schätzern genutzt.

Der *accept/reject estimator* macht keine weitergehenden Annahmen über die a priori Verhaltensverteilung. Er zieht eine Anzahl von Realisierungen aus dieser Verteilung und behält nur eine Teilmenge dieser Ziehungen bei. Diese Teilmenge wird unter Berücksichtigung der Likelihood-Funktion der Messungen derartig ausgewählt, daß sie näherungsweise äquivalent zu einer Stichprobe aus der a posteriori Verhaltensverteilung ist. Der *utility-modification estimator* addiert einen Korrekturterm zu der Nutzenfunktion einer jeden Verhaltensalternative, die ein simulierter Reisender vor einer Entscheidung auswertet. Diese Korrektur ist ebenfalls durch die Likelihood-Funktion der Messungen bestimmt. Für eine bestimmte Form der a priori Verhaltensverteilung ist das resultierende Verhalten

näherungsweise äquivalent zu einer Ziehung aus der a posteriori Verhaltensverteilung.

Für die experimentellen Untersuchungen dient ein erweitertes *cell-transmission model* als Mobilitätssimulation und ein randomisierter Kurzwegalgorithmus als Platzhalter für eine vollständige Verhaltenssimulation. Die Experimente werden unter synthetischen Bedingungen durchgeführt, wobei die Sensordaten durch eine externe Modellinstanz erzeugt werden. Der Testfall umfasst ein Netzwerk von 2 459 Kanten und eine mikroskopische Population von 206 353 Fahrern. Die experimentellen Ergebnisse zeigen, daß das implementierte Verfahren die folgenden Eigenschaften aufweist: (i) Es nutzt in effizienter Weise eine begrenzte Menge verfügbarer Verkehrszählungen, um das individuelle Routenwahlverhalten in der Population derartig nachzuführen, daß eine deutlich realistischere globale Verkehrslage resultiert. (ii) Es ist sowohl auf ein gleichgewichtsbasiertes Simulationssystem als auch auf einen Simulator ohne Gleichgewichtsannahme anwendbar. (iii) Wenngleich der verfügbare Testfall etwas zu groß ist, um in Echtzeit nachgeführt zu werden, sind in dieser Hinsicht realisierbare Szenarien nicht um Größenordnungen kleiner.

Abstract

This dissertation describes a novel method for the fully disaggregate estimation of motorist behavior from aggregate measurements of flows, densities or velocities that are obtained at a limited set of network locations. The problem is solved in a Bayesian setting, where the prior assumption about an individual's choice distribution is combined with the available measurements' likelihood into an estimated posterior choice distribution. The approach is simulation-based in that it (i) only requires a simulation system to represent the behavioral prior distribution and (ii) only generates realizations from the behavioral posterior distribution.

The estimator treats the behavioral simulation system as a black box to the greatest possible extent. The possibly estimated behavioral aspects range from single route choice to the selection of full-day plans, and an equilibrium-based modeling assumption is just as feasible as a telematics model of imperfectly informed drivers.

The incorporation of aggregate sensor data into this behaviorally disaggregate estimation procedure is enabled by a mixed micro/macro mobility simulation that moves individual drivers through a macroscopic model of traffic flow dynamics. This model allows to linearly predict the effect of individual behavior on aggregate traffic conditions, and through this it provides a linear approximation of the sensor data's log-likelihood given a particular behavioral pattern in the driver population. This linearization is utilized by two operational Bayesian estimators.

The accept/reject estimator functions without further assumptions about the behavioral prior distribution. Its takes a number of draws from this prior and retains only a subset of these draws. This subset is chosen in consideration of the measurements' likelihood such that it is equivalent to a sample from the behavioral posterior. The utility-modification estimator adds a correction term to the utility of every behavioral alternative a simulated traveler evaluates before making a choice. This correction also is a function of the measurements' likelihood. Given a particular form of the behavioral prior, the resulting behavior is equivalent to a draw from the behavioral posterior.

For experimental investigations, an extended cell-transmission model is implemented as the mobility simulation, and a randomized best-path routing logic serves as a placeholder for a full behavioral simulator. The experiments are conducted in a synthetic setting, where the sensor data is generated by an external model instance. The test case comprises a network of 2 459 links and a

microscopic population of 206 353 drivers. The experimental results show that the implemented estimator has the following properties: (i) It efficiently utilizes limited traffic counts to adjust the population's individual-level route choice such that a significantly more realistic global traffic situation results. (ii) It is equally applicable to an equilibrium-based and to a non-equilibrium-based simulation system. (iii) While the available test case is somewhat too large to be monitored in real-time, a feasible scenario for an online application of the estimator is not smaller by orders of magnitude.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The 2007 world climate report emphasizes the significant influence of fossil fuel burning on the current and future climate change [81, 82], whereas a large share of the global greenhouse gas production stems from present transportation systems [105]. Mobility is an essential good that justifies a certain environmental price. However, its necessity as well as the very price it entails make it highly desirable to operate transportation systems at working points of greatest efficiency and to optimally exploit the available infrastructure. This goal needs to be pursued both in long-term planning considerations and in short-term traffic management efforts.

From an engineering perspective, a powerful tool to achieve such objectives are algorithms for model-based prediction and control. They allow to evaluate the performance of a traffic system in various settings before choosing the most promising measure. Pivotal to the success of these approaches is the availability of a realistic model. Usually, this is achieved by building a structurally correct model which is calibrated based on comparisons of its outputs and available measurements. Numerous methods have been developed to more or less automatically solve the latter task.

This thesis contributes to that field. It describes a method to estimate the travel behavior of individual motorists from measurements of aggregate traffic features such as flows, densities or velocities that are obtained at a limited set of network locations. Knowing what trips people will make allows to predict and possibly reduce congestion. But no matter if this information is used to choose control measures, for driver information services or to collect long-term data: It always provides a valuable basis for prosperous decision making.

1.1 Definition of Problem Domain

Traffic state estimation is a broad field, which necessitates the preliminaries given in this section. Their purpose is to outline this dissertation's work scope and to introduce some terminology.

A model-based estimation approach is pursued. "Blind" modeling techniques that provide general-purpose mappings of a system's inputs to its outputs without an underlying problem-specific model structure are excluded from consideration. For example, a neural network that maps local traffic volumes on network-wide travel times does not contain a structural model and thus is not in the scope of this thesis.

The notion of "state estimation" is introduced informally as the measurement-based adjustment of a structural model's time-dependent properties. This terminology is made increasingly precise as the considered class of models is specified throughout Chapters 2 and 3. This order of presentation accompanies the overall composition of this work, which is geared by the transportation specific aspects of the estimation problem.

1.1.1 Macro- and Microsimulation

Macroscopic traffic models treat a population of travelers as a continuous quantity and express mobility in terms of equally macroscopic traffic streams. Real travelers are discrete entities. This requires their aggregation into sufficiently large homogeneous groups for this approach to work. While being particularly amenable to a mathematical treatment, macroscopic models are unable to represent highly heterogeneous traveler populations. The possibilities to macroscopically represent behavioral constraints, which often are of a rule-based nature and might greatly vary across a population, are limited as well.

Microsimulations capture travelers and their behavior individually. This gives them a greater expressive power. Still, since their population model can only be a sample of the real population, it is inherently stochastic. The increased realism of a structurally detailed mapping of the real world on a microscopic simulation system also introduces the real world's mathematical intractabilities into the model. This opens a gap between the ease of implementing a microscopic model and the difficulties in understanding the resulting model dynamics.

This work adopts a microsimulation approach to the estimation of individual behavior. Microsimulation greatly simplifies the modeling and likewise complicates the estimation task. Consequently, every property of the model that is to be estimated has to be carefully matched by a formal representation that allows for a mathematical treatment. The formal requirements set up in this thesis aim to capture a wide variety of microscopic aspects while ensuring tractability of the mathematical estimation problem.

1.1.2 Behavioral and Physical Simulation

Microsimulations of vehicular traffic usually consist of at least two sub-models, one of traffic flow dynamics and one of travel behavior:

- **Traffic flow dynamics** describe the physical laws of the traffic system under consideration. They determine how well a road network serves a traveler's need of driving most conveniently along a route to a destination in a potentially congested traffic situation. To serve the purpose of

this thesis, driver behavior in terms of breaking, acceleration, and lane changing is subsumed in the physical representation of traffic flow.

- **Travel behavior** results from the demand for mobility across a network. Various aspects such as route, destination, and departure time choice can be modeled once a representation for the traveler population itself is found [71, 128]. If only motorists are considered, mode choice winds down to the decision if a car trip is made or not. Long-term decisions such as car ownership and residential choice are beyond the time scales considered in this thesis.

This work is restricted to the estimation of behavioral aspects. That is, the present approach assumes the traffic flow dynamics to be modeled without error. A possible augmentation towards the concurrent estimation of behavior and physics is outlined as a subject of future research.

Given the focus on behavioral estimation, no differentiation between freeway and intra-urban traffic is necessary in principle since their major difference consists in their traffic dynamics. Only the granularity of the physical modeling has a limiting effect on the proposed method's applicability.

1.1.3 Transportation Planning and Telematics

Microsimulation can be applied both in transportation planning and transportation telematics, and the proposed estimation method is applicable in both fields as well.

At first glance, this is not surprising since planning and telematics constitute two different views of the same system. Planning methods have evolved over many decades, while telematics appeared quite recently as an offspring of transportation planning and adopted many methods from this field. Still, there are systematic differences that must be accounted for:

- **Planning** models usually assume that travelers obtain global knowledge of average system states through many days of exploration and that the resulting behavioral patterns resemble some kind of equilibrium. Typically, such models work at the granularity of average within-day trajectories but do not reproduce within-day fluctuations of the system states as they occur in reality due to the stochastic nature of traffic [35].
- **Telematics** models explicitly deal with fluctuations within a day. They neither assume global driver knowledge nor do they assume an equilibrium. The behavioral model component in such a system may represent driver reactions to new and possibly unforeseeable traffic situations, to provided information, and to guidance [24, 25]. Without these fluctuations, there would be little use in guiding the system in one way or another since under normal conditions travelers have already found good travel options via day-to-day experimentation [74].

This distinction carries over to the temporal constraints for a traffic state estimation algorithm. In a planning application, there is at least one night to adjust a model to recently collected measurements. This is considered as an **offline** estimation problem. In a telematics application, usually just a few minutes are available to incorporate the most recent measurements into the current estimate. The adjustment takes place while the model progresses through (real) time, constituting an **online** estimation problem. However, a telematics estimator may also be used in offline mode for the ex post analysis of a particular day's traffic situation.

While the above distinction is clear, that of applicable estimation methods is not. Conceptually, it does not make a difference to a recursive algorithm if it is used for incremental over-night adjustment of a planning model or on a 5-minutes time scale in a real-time context. However, the portability of traditional planning tools to telematics applications is limited. The need for substantial research in this field has been recognized about two decades ago, e.g., [160], and has spawned ongoing investigation efforts both nationally, e.g., [123, 169], and internationally, e.g., [42, 58, 143]. Still, many methodological potentials are yet to be explored [139].

But there are not only limitations. Mutual benefits of different estimation approaches naturally result from their common object of investigation. Online traffic monitoring systems usually rely on some kind of a priori knowledge about the average system behavior as provided by a planning simulation. Vice versa, the daily generation of high-resolution state estimates provides valuable data for the continuous calibration of a planning model.

The proposed estimator is compatible with both a planning and a telematics modeling assumption. However, its immediate benefits are greatest in online traffic monitoring, and further processing of its outputs is likely to be necessary for typical planning purposes. The following literature review therefore focuses on the online traffic state estimation problem and gives references to more traditional planning methods only where their interplay with the online problem is of relevance.

1.2 State of the Art

Many approaches to the online traffic state estimation problem draw from transportation planning's established methods and enhance them by a dynamical component. Arguably, the most frequently adopted methods are those of static origin-destination (OD) matrix estimation. An OD matrix models the demand of a given time interval in terms of the number of trips from every origin to every destination of a traffic system. The originally considered problem was to estimate such a matrix from observed link volumes given a linear assignment mapping of demand on link flows ("assignment matrix"). Various methods such as entropy maximization and information minimization [168], Bayesian estimation [113], generalized least squares [12, 34], and maximum likelihood estimation [162] have been proposed to solve this task. Early overviews on this subject can be found in [37, 180]. Nonlinear assignment mappings can be incorporated by a bilevel-approach that iterates between a nonlinear assignment and a linearized

estimation problem [114, 181, 182] until a fixed point of this mutual mapping is reached [39]. The combined estimation of OD matrices at subsequent time slices was demonstrated in [36], and many originally static methods have been applied to dynamical problems in this vein, e.g., [111, 158] and the references in Section 1.2.2.2. Beyond the different modeling requirements, temporal constraints are most critical to the online deployment of these approaches.

Many advanced online applications employ systems engineering methodologies to a suitably formalized traffic model. The most prominent of these methods is without doubt the Kalman filter in one of its many guises. Assuming a stochastic disturbance upon an originally linear dynamical system [90], it has evolved to an estimator for systems with smooth, nonlinear dynamics [161] as well as for systems with a merely simulation-based representation [88, 89]. More generally applicable particle filters even track multimodal state distributions [6]. These developments have made Kalman filtering increasingly applicable to the high complexity of traffic systems. However, with these capabilities comes a growing computational burden that renders the real-time observation of truly large-scale systems still impossible. Because of its equivalence with a recursive least squares estimator, the Kalman filter can also be reformulated as a problem of mathematical programming, which broadens the field of potentially applicable algorithms [23].

The following presentation is organized with respect to the underlying model. It differentiates between estimation methods that use a behavioral model and those that do not. At the limit of this classification are approaches that rely on spatially non-correlated probabilities of turning move occurrences at intersections. These methods represent route choice merely as a sequence of independent turning decisions and thus are not considered to be based on a behavioral model.

1.2.1 Estimation Without Behavioral Modeling

No structural modeling at all is required if general-purpose system representations are used. Auto-regressive moving average models and artificial neural networks learn a regression-type relation between current measurements and current or future traffic states. Pattern matching techniques such as nonparametric regression or clustering methods compare previously collected traffic state trajectories to currently available information and provide most similar historical data for estimation and forecast. Lacking a structural model, these approaches are mentioned only for completeness. A comprehensive overview of data-driven methods in traffic estimation and prediction is given in [46].

A linear road does not allow for the type of behavioral decisions considered in thesis but is amenable to the modeling of traffic flow dynamics. Since traffic flow is a dynamically rather restricted system, this yields useful additional information. Models for flow on a link have gone from the fundamental diagram (where density and velocity are uniquely related, and flow is a function of either density or velocity [72]) via the Lighthill-Whitham-Richards theory of kinematic waves (where the fundamental diagram is inserted into an equation of continuity [108, 151]) to second-order models (where a second equation introduces inertia [144]).

Various approaches based on Kalman filters (and, more recently, particle filters) have been proposed to estimate parameters and/or states of traffic flow models from local measurements in a variety of settings, e.g., [75, 112, 122, 165]. As a typical example of these, the RENAISSANCE approach is described further below. ASDA and FOTO ("Automatische Staudynamikanalyse: Automatic Tracing of Moving Traffic Jams" and "Forecasting of Traffic Objects") constitute a pattern-based monitoring and prediction system that tracks traffic jams along a freeway [95, 96]. The "adaptive smoothing method" uses a nonlinear filter that accounts for the different directions of disturbance propagation in free and congested traffic conditions to interpolate and extrapolate stationary detector data on freeways [167].

If network traffic is considered, turning decisions at intersections need to be modeled. If no such model is at hand, a simple approach is to define turning probabilities. The simulation of individual vehicles by this method results in parameterized random walks through the network. In a macroscopic model, flows across an intersection diverge at ingoing links according to turning fractions that equal these probabilities and additively merge at outgoing links. For the resulting linear model, (recursive) least squares and Kalman filtering can be applied to track the turning fractions from link volume measurements [13, 50, 107, 135]. The incorporation of signal timing information was proposed in [93, 117], and the provision of estimated turning flows as supplementary measurements to a network-wide OD matrix estimator was found to significantly increase the overall estimation quality in [68, 118].

The "Urban Traffic Analyzer" UTA uses a macroscopic queuing model of inner-urban traffic flow to predict network-wide flows and travel times. However, it requires that likewise network-wide measurements of current flows and turning fractions are available, and no data fusion beyond a temporal averaging of measurements is described [94, 95].

A system that is in continuous operation in Germany is OLSIM ("Online Traffic Simulation") [45, 137, 174]. It uses a microscopic traffic model. Additional vehicles are inserted where sensors record more vehicles than the model, and vehicles are removed where sensors record fewer vehicles than the model [92]. Measurements are extrapolated by having the vehicles move forward along links according to realistic driving rules and having them turn at intersections according to historical or directly measured turning probabilities [116]. The system predicts network conditions based on a pre-clustering of typical measurement trajectories: At a given point in time, the measurements themselves are predicted as a weighted average of the most recent observations and representative historical trajectories. Based on this prediction, the aforementioned sensor adaptation procedure is continued into the future [46].

Recently, the RENAISSANCE ("Real-Time Freeway Network Traffic Surveillance Tool") traffic monitoring and prediction system has been operationalized [178]. Its estimation module consists of an extended Kalman filter [175, 176, 177], which is applied to the macroscopic traffic flow model METANET [101]. A random walk assumption is imposed on model parameters such as road capacities, free flow velocities, and turning fractions, which allows to estimate these parameters together with the traffic flow model's density and velocity states. Suchlike observed parameters improve the state estimation quality, e.g., in case

of varying weather conditions, and serve as incidents indicators.

Methods that rely on a priori collected turning proportions can be expected to work well in normal situations but to be rather problematic during exceptional events when turning fractions deviate from pre-specified values. In principle, every turning-probability driven approach can be supplied with a behavioral model for the generation of these parameters. However, this alone does not clarify how to adjust the behavioral model itself to given measurements. This problem is considered next.

1.2.2 Estimation With Behavioral Modeling

1.2.2.1 Static Traffic Assignment

The classical planning method for the modeling of network traffic is static assignment. The problem is stated as to assign a given demand of flows between origin-destination pairs (OD pairs) on the network. Typical assignment criteria are a Nash equilibrium (all actually used routes for each OD pair have equal cost and no unused route has smaller cost; also called user equilibrium (UE)) or stochastic user equilibrium (SUE; the assignment of OD flows on routes follows a given distribution which is based on link cost). In so-called congested assignment, cost on a link is an increasing function of link flow which is generated by flows on routes that use the link. Links that are heavily used become "expensive", thus diverting some of the flow to other routes, e.g., [35].

The only way to approximate within-day dynamics by means of static assignment is to run independent simulations on consecutive time slices. Within limits and in combination with dynamical model components, this approach can be integrated into a practically acceptable system representation for telematics purposes, as the following two examples show.

The naming "path flow estimator" (PFE) is usually associated with the approach proposed in [17]. It describes a macroscopic one-step network observer that estimates static path flows from link volume measurements based on a SUE modeling assumption in a congested network [14]. The estimation problem is transformed into one of smooth optimization, which is iteratively solved. The model has been enhanced by multiple user classes and a simple analytical queuing model to represent traffic flow dynamics [16] and has been successfully implemented in various research and development projects [15]. The limitations associated with its original assumption of a logit path choice model ("overlapping path problem", e.g., [18]) have been mitigated by the implementation of a C-logit path choice model [38, 173]. The PFE's static UE counterpart was proposed in [157, 159] and has been further advanced in [133, 134].

The traffic management center of Berlin ("Verkehrsmanagementzentrale" VMZ) also operates an online traffic monitoring system [170]. The fully macroscopic method comprises a substantial number of different adjustment steps. It predicts measurement trajectories by a clustering approach similar to that of OLSIM and uses either a static or a simplified queue-dynamical model to interpolate traffic flows between sensors. Route choice is assumed to be in a static UE that is simulated in time slices of one hour. The assigned OD matrix is selected based

on a similarity measure between currently prevailing measurements and those the matrix had previously been calibrated with [171].

A computationally costly but methodologically straightforward approach to track route choice at an aggregate level is to estimate the assignment matrix itself concurrently with the OD matrix. The resulting estimation problem is in general highly under-determined, so a prior assignment matrix is incorporated in much the same way a prior OD matrix ensures a unique solution to the common OD matrix estimation problem [109, 110].

1.2.2.2 Dynamic Traffic Assignment

The following presentation concentrates on simulation-based approaches to dynamic traffic assignment (DTA). This is justified by their microscopic vehicle representation which is a fundamental modeling assumption of this thesis. An overview of DTA that includes analytical approaches can be found in [146].

Most current network loading models use similar techniques [8, 19, 57, 115, 136]: They have individual, decision-making particles ("driver vehicle units (DVUs)") which usually are sampled from an OD matrix and are moved forward along links using functions that in some way or other couple speed to density. Most models include storage capacities on their links, that is, the density of vehicles is limited and once a link is full, no more vehicles can enter. This implies that upstream links form queues of vehicles that cannot leave the link because the downstream link is full.

Time-dependent Nash equilibria are computed on such models via iterations [130]: Start with some version of time-dependent demand which gives, for each time slot and OD pair, the number of vehicles leaving the origin during that time slot. Have each vehicle follow a pre-computed route. After the network loading has run, re-compute the time-dependent path choice information. For example, give some fraction of travelers a new route that would have been fastest in the last iteration ("best response"), or distribute travelers between path options according to a distribution function, e.g., a path size logit or a C-logit model [18, 38]. This procedure is iterated until an approximate fixed point is reached [132].

As noted before, a dynamic equilibrium is a reasonable assumption for planning purposes, while the modeling of within-day fluctuations requires additional efforts. Even more in such a setting, simulation-based approaches are the method of choice because of their inherent ability to deal with individual and spontaneous driver behavior.

There are two projects in the United States, namely DynaMIT ("Dynamic Network Assignment for the Management of Information to Travelers", [19, 60]) and DYNASMART ("Dynamic Network Assignment Simulation Model for Advanced Road Telematics", [61, 115]), which pursue conceptually similar approaches. For illustration, a minimal online state estimation scenario is outlined in the following. More elaborate descriptions can be found in [3, 7] for DynaMIT and in [183] for DYNASMART.

- Beyond structural information, both systems require at least a static OD matrix and an initial set of traffic counts to prepare their online (within-day) estimation schemes. They proceed by estimating a time-dependent OD matrix, using methods which are in principle similar to the seminal techniques proposed in [36].

- In online operations, either system uses a linear Kalman filter to estimate the deviation of OD flows from average historical trajectories. This allows to incorporate the latters' structural information. Both systems capture the dynamics of a time-dependent OD matrix in the Kalman filter's state transition equation: DynaMIT assumes that the OD flow deviations follow a within-day autoregressive process with a priori estimated parameters. DYNASMART uses a polynomial trend representation of the OD trajectories, which yields a linear state equation for the temporal evolution of these polynomials' derivatives. In either case, the dynamical model allows for a demand prediction and (by simulation) for a network-wide prediction of traffic conditions.

- Loading a current demand estimate on the network yields a dynamic assignment matrix that linearly maps OD flows on link flows and thus relates state variables and traffic counts. This mapping constitutes the Kalman filter's measurement equation.

- Both systems run in a rolling horizon mode where two procedures take turns: (i) The Kalman filter generates a current demand estimate based on the most recent assignment matrix and the current measurements. (ii) The network loading procedure assigns the estimated demand on the network in order to predict traffic conditions and to provide an updated assignment matrix.

- Both systems use the estimated demand trajectories of a given day to update a historical OD matrix as a basis for the next day's online estimation problem. While for DynaMIT various smoothing methods are proposed, DYNASMART assumes a day-to-day random walk of the true OD matrix, considers the demand estimate of a single day as measurement of this matrix, and updates the historical OD matrix by another Kalman filter.

Much like in the static case, a time-dependent assignment matrix can be estimated together with the demand. This results in a significant state space increase and requires nonlinear filtering techniques [7]. The state vector can also be extended by time-dependent network parameters. This improves the adaptive properties of the overall monitoring system but again requires nonlinear estimators, various of which are compared in [3]. The incorporation of additional data sources such as probe vehicle samples [4, 183] is subject of ongoing research as well as advanced numerical solution algorithms [5, 23]. Recently, the DynaMIT system shifted from the Kalman filtering approach to a sparse least squares solution procedure [179], which, however, does not impair the conceptual correctness of the outline given above.

1.2.2.3 Multi-Agent Traffic Simulation

This approach is characterized by the fully disaggregate representation of travelers throughout the entire modeling process, while in DTA time-dependent OD matrices are typically disaggregated and re-aggregated whenever convenient. The multi-agent approach is attractive in the traffic domain since it appears natural to represent every traveler by a software object, to put these individual models into a representation of the physical world of mobility, and to observe the resulting mobility patterns. Due to its structural resemblance of real-world processes, the method is easily communicated and increasingly applied in transportation modeling (see, e.g., the collection of articles in [100]).

Multi-Agent Simulation (MASim) can go beyond other simulation methods by including travelers' goals and commitments into the modeling. For example, it is possible with MASim to differentiate between a delayed person with a free evening and a delayed person with a time-restricted day-care pick-up. MASim for transportation planning applications typically consists of the following modules [10, 11, 65, 130, 149]:

- A synthetic population generation module generates, from demographic data, a synthetic population that, in all its statistical aspects, corresponds to the real population under investigation, while at the same time preserving privacy.
- An activity-based demand generation module generates, for each member of the synthetic population, complete daily plans including a sequence of activities (such as home, work, shop, leisure), activity locations, and a temporal schedule. Consecutive activities at different locations generate the demand for travel.
- A router module computes how that demand is actually executed on the network, possibly including mode choice. At this point, all synthetic travelers have plans that describe what they intend to do.
- There is now always some kind of module that puts the synthetic travelers in a simulated version of the physical network and has them execute their plans simultaneously. The physical interaction in that system generates congestion. Depending on the specific focus, this simulation has different names: supply simulation, network loading, traffic flow simulation.

It is not possible to compute the system in the linear way indicated above since plans depend on congestion but congestion is a consequence of the plans. This is solved by iterations that can be seen as modeling human day-to-day learning. This learning takes place on various time scales. On the long term, there are aspects such as choice of residence and employment. These and further characteristics of an agent constitute constraints on decisions that take place within dimensions of days, such as activity scheduling, location choice, and route choice. Although there are no strict temporal domains for different elements of a plan, a rough distinction with respect to transportation planning and telematics can be made by a separation of elements that are modified only on a day-to-day basis and those that can be reconsidered within a day.

The estimation of fully disaggregate travel behavior from aggregate sensor data with a multi-agent traffic simulation is a novel venture. In order to clarify this statement, the following related yet different problems need to be mentioned:

- The calibration of a mobility simulation from aggregate sensor data has been widely addressed in the literature, e.g., [47, 48, 59, 97, 103, 141, 142]. However, these approaches do not carry over to a calibration of the behavioral simulation component (unless one adopts a different terminology than defined in Section 1.1.2 and attributes, e.g., car-following parameters to the behavioral model).
- A DTA-based OD matrix estimator captures various behavioral aspects, yet only on an aggregate level. Since a time-dependent OD matrix maps (origin, destination, departure time) tuples on demand levels, it directly represents destination and departure time choice. A motorist OD matrix reflects mode choice at least in terms of decisions for or against the vehicular mode. Route choice, however, constitutes no additional degree of freedom but is a function of demand defined by the DTA procedure. The only exception to this are the (behaviorally static) path flow estimators mentioned above.

1.3 Thesis Contribution and Outline

1.3.1 Conceptual Outline

The complexity of modern traffic simulation systems renders the technological design of a flexibly applicable estimator a nontrivial task. Extensive prototypical programming was conducted in order to validate the proposed method's applicability. Since the resulting architecture structurally reflects the estimator's working, it is outlined before methodological contributions are described.

In order to be compatible with the proposed estimator, a traffic simulation system must be separable into the components shown in Figure 1.1. Most of the employed terminology is adopted from [27].

- The **mobility simulation** moves individual vehicles along their chosen routes through the road network. All physical interactions occur within this component. A linearizable state space representation of the mobility simulation must be available. This dissertation demonstrates that such a requirement is compatible with a microscopic driver representation.
- The trip sequence of every vehicle in the mobility simulation is chosen by an individual **agent** that represents the driver of that vehicle. The **travel behavior** of an agent is realized by one or two further components. Whenever a decision is required, the agent provides these components with its **individual parameters**.
  - The **utility function** provides an individually parameterized map from the **network conditions** on the systematic utility of any behavioral alternative available to the agent. This may include utilities

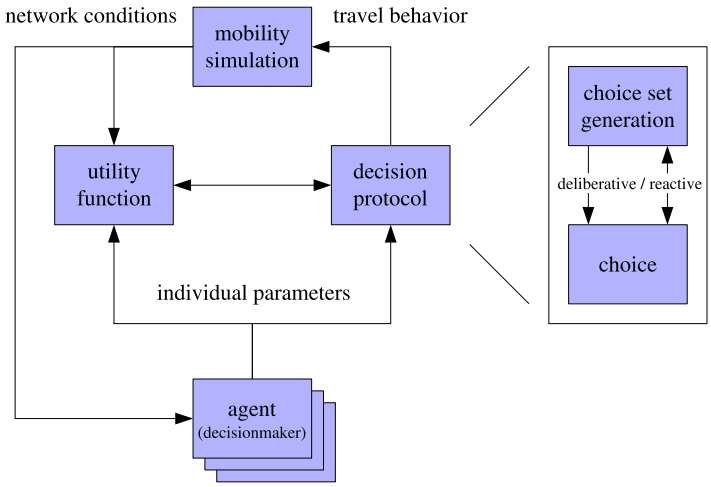


Figure 1.1: Simulation

Logical structure of a microscopic traffic simulator that is amenable to the proposed estimation methodology. The utility function is an optional component that may be omitted.

for partial choices if such a decomposition is required by the decision protocol. For example, a route choice decision protocol may only request utilities for single links in the network. The utility function is an optional component that may be omitted.

– The (likewise individually parameterized) **decision protocol** probabilistically generates a single decision based on this utility information. If there is no utility function, the choice is directly based on the network conditions. A decision protocol can be decomposed in the two aspects of **choice set generation** and **choice**. It may be **deliberative** in that the choice set of available alternatives is once enumerated before a choice is made. Alternatively, a **reactive** search may be implemented that iterates between the generation of some alternatives and their evaluation. In either case, one choice is finally realized by the agent.

This structure is independent of a particular planning or telematics context. For experimental purposes, all simulator components were exemplarily implemented similar to the according components of the MATSim ("Multi-Agent Transport Simulation Toolkit") simulation system [119], in the context of which this work was conducted.

Estimation is based on reasonable mathematical inference but follows a simple technical logic. As illustrated in Figure 1.2, the simulation structure is not changed at all. An **estimator** component is inserted between the decision protocol and the remaining simulation system. It is implemented transparently in that it provides unmodified interfaces to both the decision protocol and the remaining system. The estimator compares the output of the mobility simulation to **sensor data** from a surveillance system. Based on this comparison, it alters

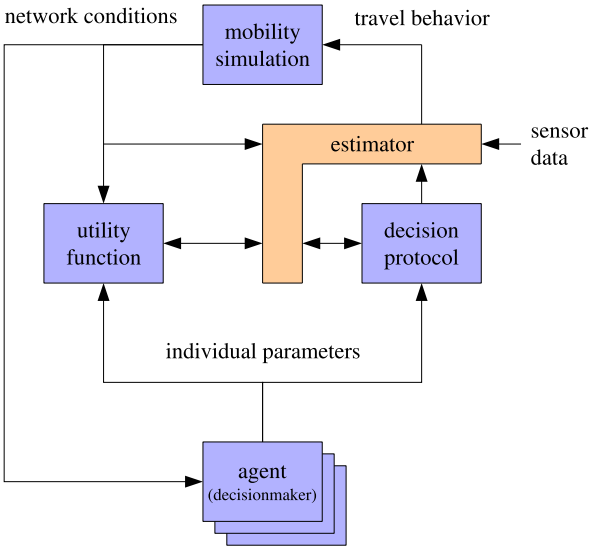


Figure 1.2: Estimation
Estimation is facilitated by the addition of a logical wrapper around the decision protocol. All interfaces within the original simulation system remain unchanged.

the data and control flow around the decision protocol such that the resulting agent behavior is most plausible given the measurements.

Two small route choice examples illustrate how this minor system extension allows to adjust simulated behavior:

- If the surveillance system observes a traffic jam where there is none in the simulation, the estimator increases the systematic utility of the according links until the agents start to favor these links and create the congestion as observed in reality. Vice versa, if there is congestion in the simulation but not in reality, the estimator decreases the involved links' utility until the agents start to avoid the critical area.
- Likewise, the estimator can encourage a certain behavioral pattern by asking the decision protocol to draw several alternatives in identical conditions for each agent. From this set of options, the estimator then passes only those decisions on to the mobility simulation that are most plausible given the measurements.

Either approach accesses only a subset of the interfaces touched by the estimator in Figure 1.2. This further relaxes the structural requirements on the simulation system. The apparent simplicity of this approach is confronted with (i) the difficulties to relate aggregate measurements and individual behavior through nonlinear traffic flow dynamics on large networks of general topology and (ii) the intention to be compatible with a broad variety of behavioral implementations.

The software prototype is single-threaded and written in the Java programming language [84]. Its interface-based design relies on standard software design pat-

terns [70] in order to simplify the (re-)composition of available software components. Likewise experimental implementations for the simulation of spontaneous route switching behavior [79, 80] and route guidance by feedback control [154] are integrated in the system.

1.3.2 Methodological Contribution

This thesis presents a novel approach to the fully disaggregate estimation of motorist behavior with a multi-agent simulation. The problem is solved by a combination of *prior* knowledge about the driver behavior with available measurements into most likely *posterior* estimates of this behavior:

- The prior knowledge about the driver behavior consists of two parts. First, an individually modeled agent exhibits likewise individual features that influence its behavior, e.g., socioeconomic features, preferences, and information availability. Second, every such agent has one or more individually generated plans it adheres to. These plans specify what the agent intends to do during a day.
- The measurements of aggregate traffic features such as flows, densities or velocities are available at a limited set of network locations. Beyond link related quantities, turning move counts can be directly utilized by the estimator. The amount of measurements may be arbitrarily small since the availability of individual plans guarantees an existing solution to the estimation problem.

Based on this information, arbitrary behavioral aspects ranging from single route choice to plan selection for a whole day are estimated in a fully disaggregate manner, agent by agent. Estimation methods of different complexity are proposed that allow for a problem-specific balance between computational speed and estimation precision. Experimental results are given and indicate the estimator's practical applicability.

The estimator can be used in a planning context (with an underlying equilibrium assumption) and for real time traffic monitoring (with a behavioral model that accounts for incomplete driver information and spontaneous behavior). If within-day estimates are fed back to a planning system for incremental adjustments on a day-to-day basis, improved prior information for the following day's online estimation problem can be generated.

The following results are also considered to be relevant contributions. They are obtained as intermediate steps on the way to a working estimator.

- A macroscopic traffic flow simulator is developed that is consistent with the phenomenology of the cell-transmission model and the requirements of first order traffic flow theory. It efficiently calculates linearized traffic flow dynamics, while its advanced simulation logic upholds a high computational performance that allows to simulate large networks of arbitrary topology. While linearization is required for estimation, the class of applicable mobility simulations is not restricted to this particular model.

- A simulation logic is proposed that runs a macroscopic traffic flow model based on the travel behavior of a fully microscopic agent population. This contribution to the field of "mesoscopic" modeling provides a broadly applicable link between behavioral microsimulation and physical macrosimulation.
- A method is developed that steers the behavior of simulated travelers such that a general objective function of aggregate network conditions is improved. Specifically, this result is employed to express and solve one instance of the behavioral state estimation problem. More generally, the method holds promise for further applications such as the generation of road pricing strategies.

1.3.3 Structure of Thesis

The remainder of this document is organized as follows. Chapter 2 describes the macroscopic mobility simulation. Chapter 3 treats the disaggregate modeling of behavior. Its first part describes how individual motorists are simulated in a macroscopic mobility simulation. Its second part specifies a formalism of driver behavior that is amenable to a mathematical estimator. Chapter 4 formulates the behavioral estimation problem and discusses different solution approaches. Chapter 5 verifies the estimator's computational feasibility for an application of practically relevant size. Finally, the work is concluded in Chapter 6, and a discussion of future research topics is given.

Chapter 2

Macroscopic Mobility Simulation

A model of physical reality maps demand for travel on network conditions. Basically, an inverse mapping is needed if travel behavior is to be deduced from these conditions. Such an inversion does generally not exist. Alternatively, a linearization of the mapping is used, and nonlinearities are accounted for in an iterative manner.

This chapter describes a mobility simulation that can be linearized. A reader with only a casual interest in traffic flow modeling may skip this material and continue reading at Section 2.7 without much loss of continuity.

2.1 Design Choices

The necessity of linearization calls for a macroscopic model. An aggregation of travelers into homogeneous groups can be avoided by the behavioral simulation scheme introduced later in Chapter 3 so that only single-commodity traffic is considered here.

Since the experimental validation of new phenomenological proposals would exceed the scope of this thesis, the model must build on established findings. This and the need to realize a large-scale test case calls for the simplest available model that still captures the most relevant traffic features with reasonable precision. Arguably, this is the *kinematic wave model* (KWM) [108, 151]. Within its phenomenological limitations, it is able to represent both freeway and intra-urban traffic flow. The choice of this model is well justified in light of the ongoing debate if more complex models yield a reasonable gain in expressive power [78, 131].

For numerical simulation of the KWM, the *cell-transmission model* (CTM) is adopted [53, 54, 55]. Various other macroscopic models had been considered before this choice was made [73, 76, 86, 101]. However, once higher order models are excluded from consideration, the CTM remains as the by far

most established model, with various applications, e.g., in freeway ramp metering and signal optimization [1, 66, 164], and thorough experimental validations [28, 126, 127]. The CTM is closely related to another implementation of the KWM, the STRADA model [29, 30]. Both approaches base on the numerical Godunov solution method [102, 106].

The model must allow to simulate a large and complex road network, provide linearized traffic flow dynamics, and maintain a high computational performance. These requirements motivate three in large parts novel adaptations of the CTM:

- To allow for linearization, all flow calculation rules of the CTM are unified in a formal calculation scheme, for which sensitivity analysis is conducted.
- Since the original CTM only specifies network topologies where at most three roads meet at an intersection, its established phenomenology is transferred to the modeling of general intersections.
- Spatially discretized macroscopic models imply a relatively high computational cost because of their large number of simulated entities. To ensure feasibility of large-scale applications, a simulation logic is adopted that assigns an individual simulation time step duration to every link in the network. The additional numerical imprecision introduced by this modification is investigated and is found to be countervailed by its computational benefits.

A simplified linearization of the CTM has been described before [125, 126]. This approach switches between linear sub-models according to the congestion status of a considered freeway stretch. It is a simplification even of the CTM and is not applicable to network traffic. A likewise constrained linearization is described in [165]. The originality of an earlier contribution is also acknowledged where CTM merges and diverges are recombined to generate more complex intersections and a simulation logic with variable time step lengths is enabled by the nesting of differently fast ticking cells [104].

Some elements of the KWM theory are given in Section 2.2. Before the CTM is considered, a general and linearizable flow calculation scheme is introduced in Section 2.3. The CTM and its extensions are then expressed in terms of this formalism in Section 2.4. The simulation logic on variable time scales is described in Section 2.5, a suitable spatiotemporal network discretization logic is proposed in Section 2.6, and, finally, a general state space representation of the mobility simulation is given in Section 2.7.

2.2 The Kinematic Wave Model

The KWM requires a minimal set of assumptions to model traffic flow on a linear road. Denote by $x \in \mathbb{R}$ a location on that road and by $t \in \mathbb{R}$ the continuous time. $\varrho(x,t)$ is the local density of traffic (in vehicles[1] (veh) per length unit), $q(x,t)$

[1] In the context of a macroscopic model, the notion of a "vehicle" is to be understood as a "macroscopic vehicle unit".

its flow (in vehicles per time unit), and $v(x,t)$ its velocity. These quantities are related by the first constituent equation of the KWM:

$$q(x,t) = v(x,t)\varrho(x,t). \tag{2.1}$$

The second modeling assumption is that of vehicle conservation. On smooth conditions, it is expressed by the continuity equation

$$\frac{\partial \varrho}{\partial t} + \frac{\partial q}{\partial x} = 0. \tag{2.2}$$

Finally, local flow is specified as a function only of local density. This relation is usually denoted as the fundamental diagram:

$$q(x,t) = Q(\varrho(x,t), x). \tag{2.3}$$

Since these specifications can still result in ambiguities, an additional condition must be instrumented to select the physically relevant solution. Given a concave fundamental diagram, the principle of local demand and supply provides a convenient technique to ensure uniqueness [102]. Denote by $x-$ ($x+$) the location immediately upstream (downstream) of x. For every x, the local flow $q(x,t)$ is then defined as the minimum of local flow *demand* $\Delta(\varrho(x-,t),x-)$ and local flow *supply* $\Sigma(\varrho(x+,t),x+)$:

$$q(x,t) = \min\{\Delta(\varrho(x-,t),x-), \Sigma(\varrho(x+,t),x+)\}. \tag{2.4}$$

Figure 2.1 illustrates this function.

To begin with, (2.4) reflects the self-evident constraint that local traffic flow is bounded by the flow that can be dismissed from the immediate upstream location and by the flow that can be absorbed by the immediately downstream location. But furthermore, the local flow is maximized subject to these constraints. This property enforces the physically relevant solution of the KWM-model [102]. Phenomenologically, it is a statement of drivers' ride impulse [2], which is equivalently expressed by the microsimulation rule for cellular automata "Drive as fast as you can and stop if you have to!" [45].

Beyond its ability to uniquely capture traffic flow along a link, this principle also holds for the modeling of general intersections, as illustrated in Figure 2.2. In such a setting, every upstream link i provides a demand $\Delta_i(t)$ equal to its greatest possible outflow towards the intersection, and every downstream link j provides a supply $\Sigma_j(t)$ equal to its greatest possible inflow. Additional phenomenological modeling is facilitated since these boundaries alone are generally not sufficient to uniquely define the flows across an intersection. However, every reasonable specification must adhere to the principle of local flow maximization.

2.3 Intersection Flow Calculation Scheme

This section describes a formalism for intersection traffic flow modeling denoted as the **general process of resource consumption** (GPRC). Since sensitivity

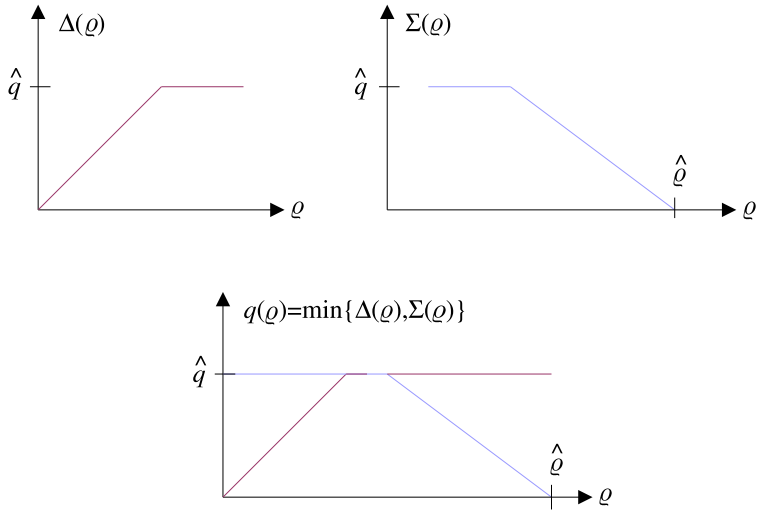


Figure 2.1: Local supply and demand comprise a fundamental diagram
The piecewise linear demand function $\Delta(\varrho)$ conforms to the original specification of the CTM, where it is denoted as the *sending* function. It consists of an increasing part with its slope equal to the free flow speed, and it is limited by the flow capacity $\hat{q}$. The supply function $\Sigma(\varrho)$ (also consistent with the original CTM, where it is called *receiving* function) is also limited by the flow capacity. The slope of its declining part equals the backward wave speed and intersects the abscissa at the greatest possible density $\hat{\varrho}$. The minimum of both functions yields a fundamental diagram.

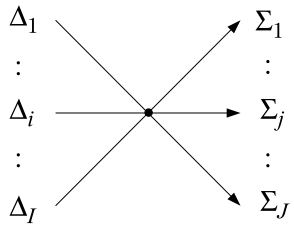


Figure 2.2: A point-like intersection with I ingoing and J outgoing links
Every upstream link i provides a demand Δ_i equal to its greatest possible outflow towards the intersection, and every downstream link j provides a supply Σ_j equal to its greatest possible inflow.

Algorithm 1 General process of resource consumption

$\boldsymbol{\xi}^{(0)}$ is given
$D^{(0)} = \{i;\, \xi_i^{(0)} > 0\}$
$m = 0$
while $(\exists i \in D^{(m)} : \varphi_i(D^{(m)}) > 0)$, do {
    for all $i \in D^{(m)}$, do: $\theta_i^{(m)} = \xi_i^{(m)} / \varphi_i(D^{(m)})$
    $\theta^{(m)} = \min\limits_{i \in D^{(m)}} \{\theta_i^{(m)}\}$
    $B^{(m)} = \arg\min\limits_{i \in D^{(m)}} \{\theta_i^{(m)}\}$
    $\boldsymbol{\xi}^{(m+1)} = \boldsymbol{\xi}^{(m)} - \theta^{(m)} \boldsymbol{\varphi}(D^{(m)})$
    $D^{(m+1)} = D^{(m)} \backslash B^{(m)}$
    $m{+}{+}$
}
$M = m$

analysis for the GPRC is available, every intersection model that conforms to its specification can be linearized.

Consider a dynamical process with time step index $m = 0 \ldots M$. Every element $\xi_i^{(m)} \in [0, \infty)$ of its state vector $\boldsymbol{\xi}^{(m)} = (\xi_i^{(m)})$ is considered as a resource that is used up during the process. Its rate of consumption equals a non-negative and finite value $\varphi_i^{(m)}$, which is constant throughout every time step m. Denote the duration of step m by $\theta^{(m)}$. The process dynamics are then defined by $\boldsymbol{\xi}^{(m+1)} = \boldsymbol{\xi}^{(m)} - \theta^{(m)} \boldsymbol{\varphi}^{(m)}$ where $\boldsymbol{\varphi}^{(m)} = (\varphi_i^{(m)})$. The resources must not become negative such that all zero states must have a zero consumption rate and $\theta^{(m)} \leq \xi_i^{(m)} / \varphi_i^{(m)}$ must hold for all nonzero states i.

The set $D^{(m)} = \{i;\, \xi_i^{(m)} > 0\}$ contains all resources that are strictly positive at the beginning of step m. The process terminates if all elements in $D^{(m)}$ have a zero consumption rate. Consumption rates only depend on the set $D^{(m)}$ of currently **available** resources such that $\boldsymbol{\varphi}^{(m)} = \boldsymbol{\varphi}(D^{(m)})$. Consequently, it is phrased that "step m is under **regime** $D^{(m)}$". The maximum duration of step m in exclusive consideration of resource i is $\theta_i^{(m)} = \xi_i^{(m)} / \varphi_i^{(m)} \in (0, \infty)$. Since every step m is specified to last until at least one resource in $D^{(m)}$ reaches a zero value, its duration is $\theta^{(m)} = \min_{i \in D^{(k)}} \{\theta_i^{(m)}\} > 0$. The set $B^{(m)} = \arg\min_{i \in D^{(m)}} \{\theta_i^{(m)}\}$ contains all resources that run dry at the end of step m.[2] This allows to give $D^{(m+1)} = D^{(m)} \backslash B^{(m)}$ as an update equation.

The temporal aspect of this process is not to be interpreted physically. Only its final state is of relevance to the physical simulation. Algorithm 1 gives an overview. An efficient implementation of the involved integer sets is described in Appendix A.

Sensitivity analysis for the GPRC is provided in Appendix B, where the following result is derived. It ensures linearizability of the subsequently developed traffic flow model.

If all consumption rates are monotonously increasing with respect to the number of available resources, i.e., if $\varphi_i(D \cup \{j\}) \geq \varphi_i(D)\ \forall i, j$, *and if the availability*

[2] The argmin function returns the set of all minimizing indices.

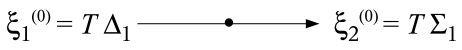


Figure 2.3: A straight connection
The mapping of upstream demands Δ and downstream supplies Σ on GPRC resources ξ is specified in (2.5).

of a resource with a zero consumption rate does not influence the process dynamics, i.e., if $\varphi_i(D \cup \{i\}) = 0 \Rightarrow \boldsymbol{\varphi}(D \backslash \{i\}) = \boldsymbol{\varphi}(D \cup \{i\})$*, then an approximate Jacobian* $\partial \boldsymbol{\xi}^{(M)} / \partial \boldsymbol{\xi}^{(0)}$ *can efficiently be computed concurrently with the GPRC. If, furthermore, the consumption rates are parameterized with a constant parameter vector* $\boldsymbol{\beta}$ *and the sensitivities* $\partial \boldsymbol{\varphi}(D) / \partial \boldsymbol{\beta}$ *are provided, an approximate Jacobian* $\partial \boldsymbol{\xi}^{(M)} / \partial \boldsymbol{\beta}$ *can be computed in a likewise efficient way.*

2.4 Intersection Specification

The CTM runs in discrete time and space. Denote the physical simulation time step length by T, the physical simulation time step counter by k, and the spatial segments of a link as **cells**. A **connector** is placed between every group of adjacent cells. Each such connector runs a GPRC implementation that calculates the flow transmissions between these cells.[3]

The demand $\Delta_i(k)$ of upstream cells $i = 1 \dots I$ and the supply $\Sigma_j(k)$ of downstream cells $j = 1 \dots J$ (both in vehicles per time step duration) are mapped on individual GPRC resources by

$$\begin{array}{rcll} \xi_i^{(0)}(k) & = & T\Delta_i(k) & \text{for } i \text{ upstream} \\ \xi_{I+j}^{(0)}(k) & = & T\Sigma_j(k) & \text{for } j \text{ downstream.} \end{array} \tag{2.5}$$

Transmitted vehicle counts and equivalent average out- and inflow rates $q_i^{\text{out}}(k)$, $q_j^{\text{in}}(k)$ result after the GPRC's termination from

$$\begin{array}{rcll} Tq_i^{\text{out}}(k) & = & \xi_i^{(0)}(k) - \xi_i^{(M)}(k) & \text{for } i \text{ upstream} \\ Tq_j^{\text{in}}(k) & = & \xi_{I+j}^{(0)}(k) - \xi_{I+j}^{(M)}(k) & \text{for } j \text{ downstream.} \end{array} \tag{2.6}$$

The original CTM flow calculation rules and their continuation into a general intersection model can now be expressed by appropriate specifications of the resource consumption rates $\boldsymbol{\varphi}(D)$.

2.4.1 Straight Connections

The CTM's basic flow calculation rule states that the number of transmitted vehicles between two succeeding cells equals the minimum of the available vehicles upstream, the available space downstream, and an upper flow constraint. This is the discrete-time equivalent of (2.4). The according straight connector

[3] Since a *flyweight design pattern* is used for implementation [70], the number of actually created GPRC objects winds down to the number of different intersection topologies.

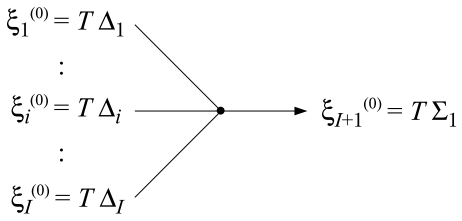


Figure 2.4: A merge with I ingoing links
The mapping of upstream demands Δ and downstream supplies Σ on GPRC resources ξ is specified in (2.5).

has one predecessor and one successor cell. Speaking in terms of the GPRC, its resource vector $\boldsymbol{\xi} = (\xi_1\ \xi_2)^T$ is two-dimensional: ξ_1 represents the number of available upstream vehicles and ξ_2 equals the available downstream space, cf. Figure 2.3. The superscript T denotes the transpose. The resource consumption vector

$$\boldsymbol{\varphi}(\{1,2\}) = (1\ 1)^T \tag{2.7}$$

corresponds to the only regime $\{1,2\}$ with a nonzero consumption rate. The resulting one-step GPRC run yields an identical vehicle transmission as the original CTM.

2.4.2 Merges

The original CTM allows for merge connections between exactly two upstream cells and one downstream cell. The according flow calculation rules state that both predecessors are allowed to send all their available vehicles as long as these can be accepted by the successor cell. If this is not the case, the successor's available space is shared between the predecessors in a ratio according to their priorities $\alpha_1 \in [0,1]$ and $\alpha_2 = 1 - \alpha_1$. If this causes all available vehicles of one predecessor to be transmitted but still leaves available space in the successor, this space is filled up as much as possible with vehicles from the complementary predecessor.

In terms of the GPRC, the merge resource vector is $\boldsymbol{\xi} = (\xi_1\ \xi_2\ \xi_3)^T$ where ξ_1 and ξ_2 denote the available vehicles in the predecessor cells and ξ_3 equals the available space in the successor cell. The evolution of the process is fully defined by three non-zero consumption rate vectors $\boldsymbol{\varphi}(\{1,2,3\}) = (\alpha_1\ \alpha_2\ \alpha_1+\alpha_2)^T$, $\boldsymbol{\varphi}(\{1,3\}) = (\alpha_1\ 0\ \alpha_1)^T$, and $\boldsymbol{\varphi}(\{2,3\}) = (0\ \alpha_2\ \alpha_2)^T$. Here, the priorities do not have to sum up to 1 but are required to be strictly positive. An inspection of the regime sequences $\{1,2,3\} \to \{1,3\}$ and $\{1,2,3\} \to \{2,3\}$ shows that this setup yields an identical behavior as the original CTM.

General merge connectors have an arbitrary number of $I \geq 2$ predecessor cells, as shown in Figure 2.4. The first I elements of the according resource vector are the available vehicles ξ_i in the predecessor cells $i = 1 \ldots I$. The available space ξ_{I+1} in the successor cell makes up one additional resource: $\boldsymbol{\xi} = (\xi_1\ \ldots\ \xi_I\ \xi_{I+1})^T$.

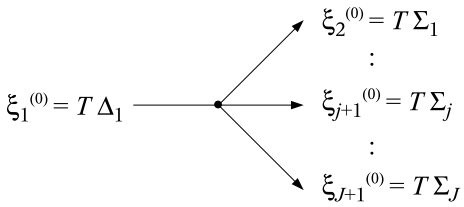


Figure 2.5: A diverge with J outgoing links
The mapping of upstream demands Δ and downstream supplies Σ on GPRC resources ξ is specified in (2.5).

A straightforward continuation of the CTM merge logic is

$$\begin{aligned} \boldsymbol{\varphi}(D) &= \left(\varphi_1(D)\ \ldots\ \varphi_I(D)\ \sum_{i=1}^{I}\varphi_i(D)\right)^T \\ \varphi_i(D) &= \begin{cases} \alpha_i & \{i, I+1\} \subseteq D \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \tag{2.8}$$

where $\{i, I+1\} \subseteq D$ indicates that both the upstream cell i and the only downstream cell provide nonzero resources. For $I = 2$, this reproduces the original CTM merge. Since the total vehicle transmission is only bounded by the available upstream vehicles and the downstream space, flow maximization is ensured.

A generalization of the CTM merge logic to more than two predecessors has previously been referred to as "very complicated" [86]. With the GPRC at hand, this difficulty collapses into specification (2.8).

2.4.3 Diverges

Diverges of the original CTM split the flow from one predecessor cell into exactly two successor cells. The splitting fractions are denoted by $\beta_1 \in [0, 1]$ and $\beta_2 = 1 - \beta_1$. Here, the resource vector $\boldsymbol{\xi} = (\xi_1\ \xi_2\ \xi_3)^T$ is comprised of the single predecessor's available vehicles ξ_1 and the available space ξ_2 and ξ_3 in the successor cells. Allowing for only one non-zero consumption rate vector $\boldsymbol{\varphi}(\{1, 2, 3\}) = (1\ \beta_1\ \beta_2)^T$ implies the assumption of exactly one upstream lane: If a vehicle at the head of the queue on this lane is unable to enter its downstream cell, it completely blocks the diverge. This logic is reasonable for large-scale applications [54, 119]. The resulting total outflow from the predecessor is $\min\{\xi_1, \xi_2/\beta_1, \xi_3/\beta_2\}$, just as for the original CTM.

The simulation of $J \geq 2$ successors for a general diverge, as shown in Figure 2.5, is straightforward by the introduction of an extended resource vector $\boldsymbol{\xi} = (\xi_1\ \xi_2 \ldots \xi_{1+J})^T$ and an according consumption rate vector

$$\boldsymbol{\varphi}(\{1, 2, \ldots, 1+J\}) = (1\ \beta_1 \ldots \beta_J)^T \tag{2.9}$$

for the only non-zero consumption regime $\{1, 2, \ldots, 1+J\}$. For $J = 2$, this yields identical flow transmissions as the original CTM. The flow is again maximized subject to the availability constraints and the additional splitting rule.

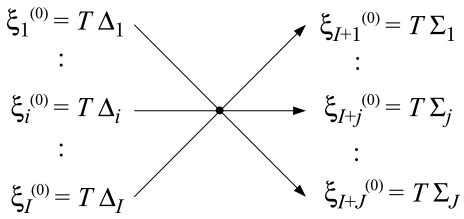


Figure 2.6: A general connection with I ingoing and J outgoing links
The mapping of upstream demands Δ and downstream supplies Σ on GPRC resources ξ is specified in (2.5).

Choosing zero consumption rates for all regimes but $\{1, \ldots, J, 1+J\}$ is necessary to ensure continuity of the flow transmissions with respect to the turning fractions, which is required for the linearization of the model: If traffic could pass the diverge unhindered given an unavailable successor j with $\beta_j = 0$, increasing β_j by an arbitrarily small amount would instantaneously block the diverge. This discontinuity is avoided by letting the diverge block even if $\beta_j = 0$ as soon as successor j becomes unavailable. This restriction can be dropped if continuity is not required and vanishes anyway in the combined micro/macro simulation scheme of the next chapter where all turning fractions are guaranteed to be strictly positive.

2.4.4 General Connections

A general connector is shown in Figure 2.6. Denote by $P = \{1, \ldots, I\}$ the set of its upstream cells, by $S = \{I+1, \ldots, I+J\}$ the set of its downstream cells, and by β_{ij} the prespecified turning fraction from predecessor i towards successor j. Given a predecessor consumption rate $\varphi_i(D)$, the specification of successor oriented consumption rates $\varphi_{ij}(D) = \beta_{ij}\varphi_i(D)$ maintains consistency with diverge logic (2.9). A priority rule equivalent to merge logic (2.8) is ensured by letting $\varphi_i(D) = \alpha_i$ for all available predecessors i as long as the intersection is not blocked by an unavailable successor. The complete resource vector $\boldsymbol{\xi} = (\xi_1 \ldots \xi_I \; \xi_{I+1} \ldots \xi_{I+J})^T$ is then consumed by

$$\begin{aligned} \boldsymbol{\varphi}(D) \quad &= \quad (\varphi_1(D) \ldots \varphi_I(D) \; \varphi_{I+1}(D) \ldots \varphi_{I+J}(D))^T \\ & \qquad i \in P : \varphi_i(D) = \begin{cases} \alpha_i & i \in D, S \subseteq D \\ 0 & \text{otherwise} \end{cases} \\ & \qquad j \in S : \varphi_j(D) = \sum_{i \in P} \beta_{ij} \varphi_i(D). \end{aligned} \tag{2.10}$$

Again, all priorities must be strictly positive. The same statements about zero turning fractions hold as for a diverge. This general connector comprises all previously defined connector types as can be seen from choosing $I = 1$ and/or $J = 1$. Still, it has no immediate counterpart in the CTM. Its logic results as the limiting case of a merge which is connected by an infinitely short link to a diverge whose turning fractions β_j result via $\beta_j = \sum_{i=1}^{I} \beta_{ij} q_i / \sum_{i=1}^{I} q_i$ from the flow composition $q_1, \ldots, q_I$ transmitted by the merge. No additional phenomenological speculations are introduced in this model.

It remains to show that the original CTM's consistency with the KWM is maintained, i.e., that specification (2.10) is still flow-maximizing. In uncongested conditions, the intersection winds down to a linear superposition of I diverges and inherits their properties. In congested conditions, the total flow through the intersection is limited by at least one downstream cell j^* with $\Sigma_{j^*} = \sum_{i=1}^{I} \beta_{ij^*} q_i$, according to (2.9). Assume that $\sum_{i=1}^{I} q'_i > \sum_{i=1}^{I} q_i$ was possible for an altered configuration $q'_1, \ldots, q'_I$ of merging inflows. The downstream diverge logic still requires $\Sigma_{j^*} \geq \sum_{i=1}^{I} \beta_{ij^*} q'_i$, and the merge logic demands $q'_i \geq q_i$ for all $i = 1 \ldots I$ if more downstream space becomes available. Thus, $\Sigma_{j^*} \geq \sum_{i=1}^{I} \beta_{ij^*} q'_i \geq \sum_{i=1}^{I} \beta_{ij^*} q_i = \Sigma_{j^*}$, which implies $q'_i = q_i$ for all i. In consequence, the general intersection inherits the flow-maximizing property of its merge and diverge component.

Specification (2.10) complies with the GPRC's requirements for linearization, as stated in Section 2.3. The relations between demands/supplies and GPRC resources (2.5) and between GPRC resources and flow rates (2.6) are already linear. Combined, this ensures the availability of flow rate sensitivities with respect to demands Δ, supplies Σ, and turning proportions β.

2.5 Simulation Logic

Discrete time network simulation is straightforward if a uniform time step length T is used. Every link with maximum velocity $\hat{v}$ is disassembled into cells of

$$\text{minimum cell length} = T\hat{v}. \tag{2.11}$$

A simulation step (tick) then consists of two parts:

1. Every connector calculates the vehicle transmissions between its adjacent cells.
2. Every cell updates its occupancy according to these transmissions.

The **occupancy** of a cell (link) is defined as the number of vehicle units that are located in that cell (link).

The simulation of a heterogeneous urban network requires relatively small cells in order to model densely meshed regions. This calls for a small T and in turn implies an unnecessarily precise modeling of longer road segments. The use of larger cells running on the same temporal grid somewhat mildens this problem at the cost of a greater numerical dispersion [55, 102, 127]. However, a significant share of urban network computations is incurred by the intersection logic. Thus, a simulation logic that minimizes the number of simulation ticks themselves is needed.

The spatiotemporal dynamics within an isolated link are uniquely defined if an initial density profile as well as feasible upstream inflows and downstream outflows are provided. Given an individually chosen time step length and an appropriate spatial discretization, the standard CTM logic facilitates a KWM-consistent simulation. Since all spatial dynamics are enclosed within the link,

it can be viewed from the outside as a discrete-time, nonlinear, ordinary dynamical system with two inputs (in- and outflows) and two outputs (upstream flow supply and downstream flow demand). The same argument holds for individual cells. Likewise, the intersection model of Section 2.4.4 calculates flows consistently with the KWM. For any chosen time step length, it constitutes a memoryless, discrete-time, nonlinear system with its upstream flow demands and downstream flow supplies as inputs and the resulting flow transmissions as outputs.

Adopting a technical point of view, these systems can immediately be linked. The outputs of systems with a larger time step are held constant when needed as inputs for faster ticking systems, and the outputs of faster ticking systems are integrated/averaged before they are fed into slower ticking systems. Since such holding and averaging affect system dynamics mainly in terms of a delay that is proportional to the involved time step lengths, a reasonable balance between additionally introduced imprecision and computational speedup can be achieved. This is confirmed by the experimental results given in Section 2.5.4.

The remainder of this section details this simulation logic. A cell i (connector c) is denoted as **due** at discrete **simulation time step** k if k is an integer multiple of its **individual time step** length T_i (T_c). The duration of a simulation time step is generally assumed to be 1 second. Two procedures are executed at every simulation time step k:

1. Every cell i that is due according to its individual time step length T_i calculates its supply and demand boundary from its current occupancy and keeps these results constant for the next T_i seconds.
2. Every connector c that is due according to its individual time step length T_c calculates its average flow rates that hold for the next T_c seconds and notifies its adjacent cells of the resulting vehicle transmissions.

Sections 2.5.1, 2.5.2, and 2.5.3 detail these steps.

2.5.1 Cell Boundaries

Every cell i has exactly one preceding and one succeeding connector. Its occupancy during simulation time step k is denoted by $x_i(k) \in [0, \hat{x}_i]$ where $\hat{x}_i$ is its maximum occupancy. While the cell has an individual time step length T_i, it is embedded in a system potentially running at a 1-second time scale. This requires its demand $\Delta_i(k)$ and supply $\Sigma_i(k)$ to be defined at every second. Since these boundaries are static functions only of i's occupancy, it is sufficient to specify x_i in every simulation time step by

$$x_i(rT_i + s) = x_i(rT_i) \quad r \in \mathbb{N},\ s \in \{0, \ldots, T_i - 1\}. \tag{2.12}$$

The original CTM boundary specifications can now be applied:

$$\begin{aligned} \Delta_i(k) &= \min\left\{\hat{q}_i, \frac{\hat{v}_i x_i(k)}{L_i}\right\} \\ \Sigma_i(k) &= \min\left\{\hat{q}_i, \frac{w_i(\hat{x}_i - x_i(k))}{L_i}\right\} \end{aligned} \tag{2.13}$$

where $\hat{q}_i$ denotes the cell's flow capacity (in vehicles per time unit), L_i its length, and w_i its backward wave speed. These equations can approximately be linearized with respect to $x_i(k)$ if at points of non-smoothness the average of left- and right-sided sensitivity is used. Alternative specifications are possible [55, 102].

2.5.2 Connector Flow Rate Update

Every connector c has a set P_c of preceding cells and a set S_c of succeeding cells. Its individual time step length T_c is chosen such that (i) the connector recalculates its flow rates whenever an adjacent cell boundary changes and (ii) the overall computational load is minimized. This is achieved by choosing T_c as the largest common divisor of all adjacent cells' time step durations:

$$T_c = \operatorname*{lcd}_{i \in P_c \cup S_c} \{T_i\}. \tag{2.14}$$

Arbitrary cell time step durations might yield low computational savings because of possibly small T_c values resulting from this equation, so they are constrained to be powers of two. This turns the connector time step length into the minimum of its adjacent cells' time step durations.

2.5.3 Cell State Update

Even if a cell i's state x_i changes only every T_i seconds, its adjacent connectors might run at a higher frequency. On the finest temporal scale, this implies

$$x_i(rT_i + T_i) = x_i(rT_i) + 1\text{ s} \sum_{s=0}^{T_i-1} \left(q_i^{\text{in}}(rT_i + s) - q_i^{\text{out}}(rT_i + s) \right). \tag{2.15}$$

Denote by p_i (s_i) the preceding (succeeding) connector of cell i. Because of (2.14), T_i/T_{p_i} and T_i/T_{s_i} are integer values. This allows for the following simplification:

$$\begin{aligned} x_i(rT_i + T_i) \quad = \quad & x_i(rT_i) \\ & + T_{p_i} \sum_{s=0}^{T_i/T_{p_i}-1} q_i^{\text{in}}(rT_i + sT_{p_i}) \\ & - T_{s_i} \sum_{s=0}^{T_i/T_{s_i}-1} q_i^{\text{out}}(rT_i + sT_{s_i}). \end{aligned} \tag{2.16}$$

Therefore, it is sufficient to notify cell i every $\operatorname{lcd}\{T_{p_i}, T_{s_i}\}$ seconds of possible flow rate changes. This is done independently by its upstream and downstream connector every T_{p_i} and T_{s_i} seconds by transmitting the appropriate addend in (2.16) to the cell. Since the cell's boundaries are held constant for a possibly longer duration according to (2.12) and (2.13), the transmitted vehicles are intermediately cached by the cell. Equation (2.16) is differentiable with respect to in- and outflow rates.

Table 2.1: Link parameters in linear test network

max. density	1 veh / 7.5 m $\approx$ 133 veh/km
flow capacity	2000 veh/h
max. velocity	50 km/h
cell length	50 km/h $\cdot$ 1 s $\approx$ 13.9 m
link length	32 cells/link $\cdot$ 13.9 m $\approx$ 444 m

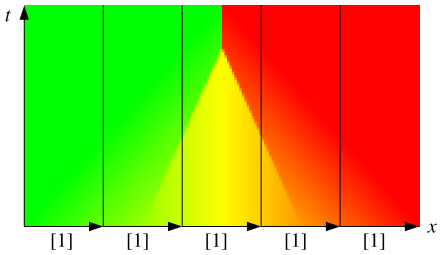
2.5.4 Experimental Investigation of Simulation Precision

A linear test network is considered. It consists of a sequence of 5 identical links the parameters of which are given in Table 2.1. The simulation boundaries resemble the conditions in which the CTM was first investigated [53]: A linear density gradient from zero to maximum density is placed on the network, with zero density at its upstream end and maximum density at its downstream end. No traffic is allowed to enter or leave the network. The simulation is run until a steady state is reached.

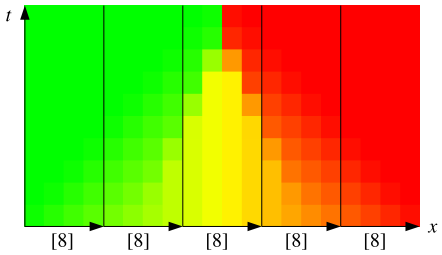
Figure 2.7 shows the resulting space-time plots in various discretization settings. Plot 2.7(a) provides a good approximation to the exact solution. Initially, two shockwaves occur: an upstream shockwave moving at positive velocity and a downstream shockwave moving at negative velocity. They merge in the center of the network and persist as a stationary density discontinuity with all traffic being queued up in the downstream half of the network. For comparison, the simulation results with a much coarser but still homogeneous discretization are shown in plot 2.7(b).

The results with heterogeneous simulation time steps nicely reflect the working of the underlying Godunov method. In every simulation time step, the Godunov scheme solves a Riemann problem at all cell boundaries. Since condition (2.11) ensures that the resulting shockwaves or rarefaction fans do not cross beyond one cell during a single time step, these problems can be solved independently in a computationally efficient way [102, 106]. Placing fast ticking cells next to slower cells explicitly displays these shockwaves, as it can be seen best in plot 2.7(c). While these artifacts are unequivocally owed to the simulation logic on variable time scales, they are put into relation by plot 2.7(d). It shows the same result after it has been averaged on a temporal grid according to the largest involved time step duration. The artifacts are nicely smeared out while the original shockwaves are maintained with a precision that is at least comparable to plot 2.7(b). Analogical statements holds for plots 2.7(e) and 2.7(f).

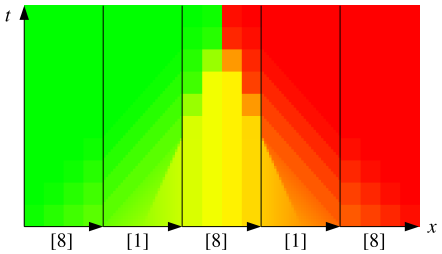
These results indicate that the overall simulation error remains in the order of the largest involved time step duration, as it has been previously hypothesized. Artifacts can occur at the boundaries between slowly and fast ticking cells but can also be removed by a temporal averaging of the simulation output before further processing. No amplification of artifacts is observed. These experiments cannot replace a thorough theoretical investigation. They are, however, considered as sufficient indications that the simulation logic on variable time scales performs well enough to be be applied in the further course of this dissertation.



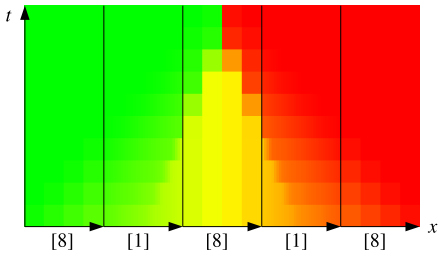
(a) All links have a time step duration of 1 second and consist of 32 cells each.



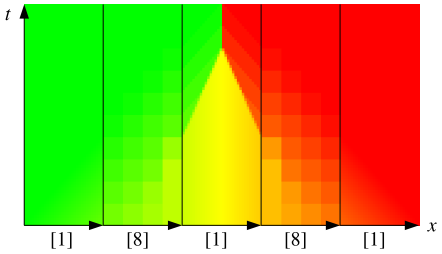
(b) All links have a time step duration of 8 seconds and consist of 4 cells each.



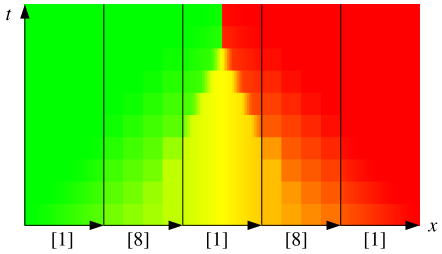
(c) All but the second and fourth link have an 8 second time step duration.



(d) The same data as (c) but averaged on a temporal grid of 8 seconds.



(e) Only the second and fourth link have an 8 second time step duration.



(f) The same data as (e) but averaged on a temporal grid of 8 seconds.

Figure 2.7: Space-time plots with variable spatiotemporal discretizations Colors encode densities as follows: green is zero density, yellow is half of maximum density, and red is maximum density. See also Table 2.1. The parenthesized numbers below the links indicate their individual time step durations.

2.6 Network Discretization

2.6.1 Specification

Specifications of large road networks usually consist of an attributed graph where nodes represent intersections and links represent roads, e.g., [119, 147, 163]. The cell structure of such a network can be generated by the following steps:

1. Choose a maximum simulation time step length $\hat{T}$. This **network time constant** compromises between a high simulation resolution (small $\hat{T}$) and a high computational performance (large $\hat{T}$).
2. For every link a in the network, do:
   (a) Select the individual time step duration T_a of link a as large as possible subject to the following constraints:
   - T_a is strictly positive and not larger than $\hat{T}$.
   - T_a is an integer power of two.
   - It is required that link a can be partitioned into at least two cells of equal length $L_a/2$. Since (2.11) must hold for each of these cells, $T_a \leq L_a/(2\hat{v}_a)$ is required.

   If link a is so short that no feasible T_a exists, increase L_a just until $T_a = 1\,s$ becomes a feasible solution.

   (b) Partition link a into n_a identical cells of length L_a/n_a. In order to minimize dispersion, choose n_a as large as possible without violating condition (2.11). That is, $n_a \leq L_a/(\hat{v}_a T_a)$ must hold. The previous choice of T_a ensures that this yields at least two cells in link a.
3. Place a connector c between every set of adjacent cells, and calculate its individual time step length T_c via (2.14).

The network entrance of traffic is facilitated by entry cells in consistency with the original CTM implementation [40]. Entry cells can hold an arbitrary occupancy, have no upstream connector, and a maximum outflow equal to the entire occupancy that waits in the cell to enter the system. One entry cell is connected to the innermost connector of every link. The existence of such a connector is ensured since every link consists of at least two cells. A specification of the network exit of traffic is postponed to Section 3.1 where multi-commodity traffic is introduced. The allocation of demand entry points to links and not to nodes is chosen in consistency with the MATSim demand specification [119].

2.6.2 Berlin Test Case

The test case of this thesis is modeled after the road network of Greater Berlin, which is illustrated in Figure 2.8. This network consists of 1 083 nodes and 2 459 unidirectional links. It is quite heterogeneous in that the inner-urban area is

Figure 2.8: Major road network of Greater Berlin
The two clippings indicate a locally high network resolution.

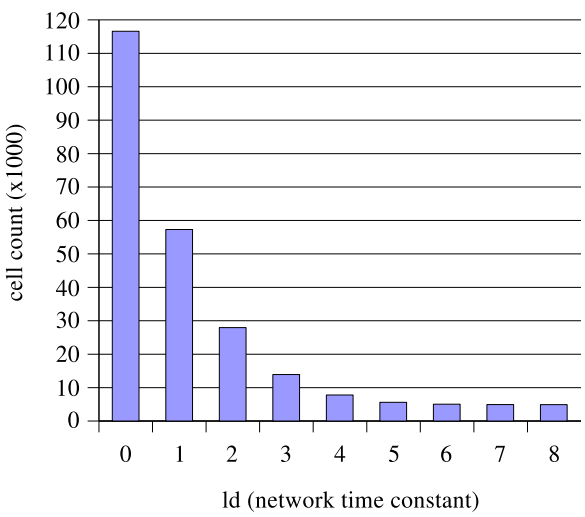


Figure 2.9: Effect of network time constant on cell count
Number of cells over $\log_2(\hat{T})$. Since the network geometry has a limiting effect on the cell sizes, $\hat{T}$ values beyond 2^6 s do not result in a notably increased coarsening.

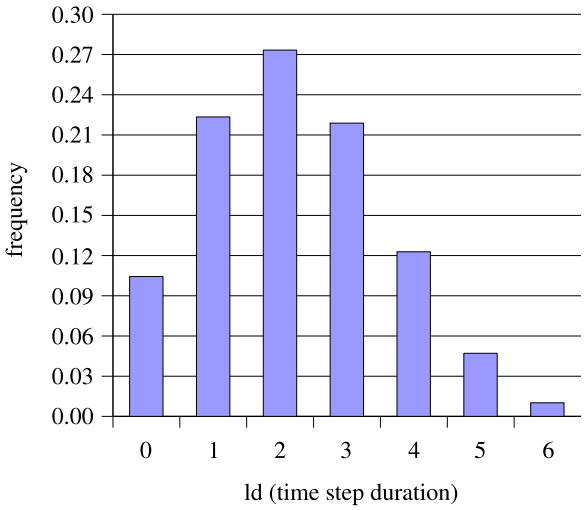


Figure 2.10: Time step duration histogram
Histogram of logarithmic intersection connector time step durations given a network time constant of $\hat{T} = 64$ s.

modeled in relatively high resolution, whereas the surrounding freeway ring is comprised of several links that are many kilometers long.

Figure 2.9 shows the effect of the network time constant $\hat{T}$ on the number of cells in the network. As $\hat{T}$ increases, the number of cells approaches a minimum value of $2 \cdot 2\,459$. This mirrors the above requirement of at least two cells per link. A histogram of intersection connector time step lengths for $\hat{T} = 64\,\mathrm{s}$ is given in Figure 2.10. The high number of intersections with a relatively low time step duration is owed to the finely meshed interurban network, which is precluded from a slower simulation clock. The relation between network time constant and computational performance is investigated in Section 3.1.4.

2.7 State Space Notation

For greatest generality, the remainder of this thesis is decoupled from specific traffic flow modeling assumptions by the following state space representation of the mobility simulation:

$$\begin{aligned} \mathbf{x}^{\mathrm{ms}}(0) &= \mathbf{x}_0^{\mathrm{ms}} \\ \mathbf{x}^{\mathrm{ms}}(k+1) &= \mathbf{f}^{\mathrm{ms}}[\mathbf{x}^{\mathrm{ms}}(k), \boldsymbol{\beta}(k), k]. \end{aligned} \tag{2.17}$$

Vector $\mathbf{x}^{\mathrm{ms}}(k)$ denotes the mobility simulation's **physical state** in time step k. For a spatially discretized first order model, it contains one element for every cell i in the network: $\mathbf{x}^{\mathrm{ms}} = (x_i)$. Single-commodity turning fractions $\boldsymbol{\beta}(k) = (\beta_{ij}(k))$ are provided as exogenous parameters to the model. Vector-valued transition function $\mathbf{f}^{\mathrm{ms}}$ defines the system's evolution through time. It fully encapsulates the specifically chosen mobility simulation. The formal modeling of demand sources and sinks is postponed to the next chapter.

For the subsequent analysis, it is required that at least approximate Jacobians $\partial \mathbf{f}^{\mathrm{ms}}[\ldots, k]/\partial \mathbf{x}^{\mathrm{ms}}(k)$ and $\partial \mathbf{f}^{\mathrm{ms}}[\ldots, k]/\partial \boldsymbol{\beta}(k)$ are available. This condition is fulfilled by the mobility simulation proposed in this chapter since

- cell state update equation (2.16) is linear with respect to in- and outflow rates,
- these flow rates can be linearized with respect to cell boundaries and turning fractions, cf. (2.5), (2.6), and Section 2.4.4, and
- cell boundary specification (2.13) is linearizable with respect to the cell states.

Chapter 3

Microscopic Behavioral Simulation

This chapter prepares a formal link between individual driver behavior and aggregate characteristics of traffic flow.

First, motorist driving decisions are expressed as control measures that act on a state space model of macroscopic traffic dynamics. The resulting formalism is quite general and allows to link different macroscopic mobility simulations and microscopic behavioral models. In particular, it allows to predict the linearized effect of individual driver behavior on global network conditions without repeated simulations.

Second, the decision making process of a driver is formalized in a way that is compatible with the aforementioned micro/macro mobility simulation. This representation comprises a broad variety of possible behavioral simulators. Some more specific modeling approaches are also presented. Apart from their illustrative purpose, they introduce modeling aspects that are referred to in later chapters.

3.1 Coupling Micro- and Macrosimulation

Two different concepts can be encountered in the literature on combined micro/macro mobility simulations.

Hybrid approaches link simulations that work on different degrees of aggregation at well-defined locations in the network [32, 64]. This approach is attractive if the required simulation fidelity varies spatially but does not serve the purpose of this work where a network-wide macroscopic model is needed.

Mesoscopic simulations move individual vehicles based on aggregate laws of motion in order to increase the computational performance while retaining a microscopic representation of behavior [31, 35]. Simulation-based DTA usually employs such models, cf. Section 1.2.2.2 and the references therein. Their counterpart in physics are *smoothed particle hydrodynamics* (SPH) [124, 155].

The approach described here is a mesoscopic model with a distinct macroscopic aspect. In this way, mathematical feasibility (linearization of the macroscopic model) and expressive power (microsimulation of behavior) are combined. High computational performance is maintained by a simulation scheme on variable time scales.

3.1.1 Representation of Behavioral Heterogeneity

Pursuing a strictly macroscopic approach, heterogeneous driver behavior could be captured by splitting traffic volumes into partial flows (*commodities*) with individual behavioral features. For example, destination-bound commodities would exhibit different turning behavior at intersections in order to reach their destinations. The applicability of this approach is limited by the computational cost of tracking partial flows for every commodity on every link in the network.

A mesoscopic simulation easily keeps track of behavioral aspects by attaching them to individual DVUs. A continuation of the mesoscopic method towards somewhat more macroscopic modeling is pursued here. A fully macroscopic representation of the underlying physical model is maintained. The behavioral information is represented by massless **particles** that are dispersed in the macroscopic flow. They drift along with the flow according to its spatiotemporal velocity field. If one maintains the macroscopic multi-commodity point of view, these particles can be interpreted as draws from the commodity distribution of the flow entering the network. Commodity information for any spatiotemporal segment of the network can be recovered by counting the according particles within that segment.

If one such particle is dismissed into the system together with the macroscopic counterpiece of one vehicle, an interpretation as a DVU is obvious. However, the number of particles is not constrained by this and can be chosen as a compromise between behavioral modeling resolution and computational performance.

3.1.2 Particle Movement

3.1.2.1 Specification

The macroscopic traffic flow model is required to specify a local velocity $v_i(k)$ in every cell i in every time step k. The velocity calculation logic employed in all experiments of this thesis is described in Appendix C.

Consider a set of particles $n = 1 \dots N$ (a population of travelers, agents or vehicles) that are floating through the system. Particles have no "mass" insofar as they do not contribute to the macroscopic occupancy in a cell. At the time of a particle's entrance into the network, an appropriate amount of macroscopic flow is also dismissed into the system, resulting in a mass balance between particles and total macroscopic occupancy.

In any time step k of duration T, each particle advances according to the local velocity in its current cell. Particle locations within a cell are continuous variables and their movement is regarded as continuous in time as well: When a

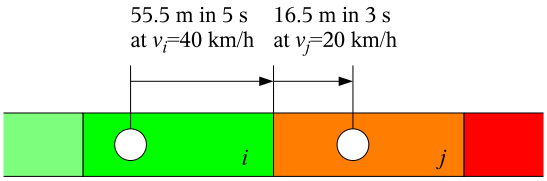


Figure 3.1: Particle movement across cell boundaries
A particle approaches the upstream end of a congested road segment. The time step duration is $T = 8\,\mathrm{s}$. The particle needs 5 s to reach the end of cell i at $v_i = 40$ km/h. During the remaining 3 s, it advances another 16.5 m in cell j at $v_j = 20\,\mathrm{km/h}$.

particle crosses a cell boundary during a single move of duration T, it can freely choose its next cell (if there is more than one downstream cell) and continue with the velocity encountered there until its available move time ends. This procedure is illustrated in Figure 3.1. The particle evaluates all traversed cells' velocities at the start time of its move. In consequence, this simulation scheme is imprecise in the order of a time step length, just as the macroscopic simulation logic itself.

When a particle reaches its destination, it is removed from the system and an appropriate amount of macroscopic flow is also filtered out of the traffic stream passing the exit location.

3.1.2.2 Simulation on Variable Time Scales

The previous chapter describes how a macroscopic simulation can be run with variable time step lengths for different network elements. This approach can be extended to the movement of particles and requires the following completion of the simulation procedure given in Section 2.5, p.41. It is illustrated in Figure 3.2.

1. Every cell i that is due according to its individual time step length T_i calculates its supply and demand boundary from its current occupancy and keeps these results constant for the next T_i seconds.
2. Each particle that currently resides in a cell i that is due is moved forward according to the following rules:
   (a) The particle moves for a duration equal to the time step length T_i of its start cell i. It might cross several cells during this move if cell i has a larger T_i than its downstream cells.
   (b) If the particle has used up its time of movement and has arrived in a cell j with $T_j > T_i$, it continues its move until it has moved for an overall duration of T_j. This continued move never enters another cell because of condition (2.11) and accounts for the expected waiting time $T_j - T_i$ until the particle is again due for movement.

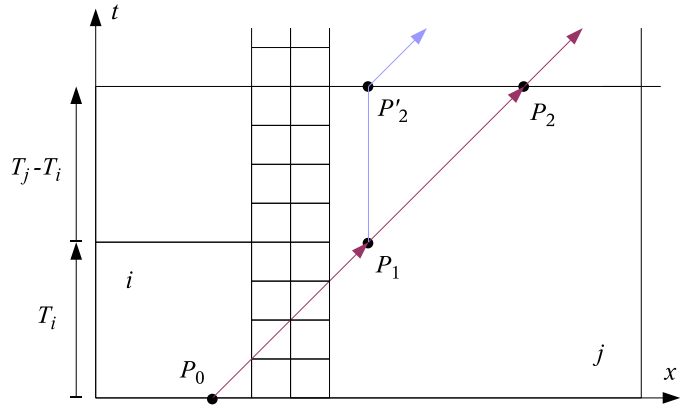


Figure 3.2: Particle movement on variable time scales
A homogeneous velocity field is assumed so that a correct particle trajectory is represented by a straight line in the space-time plot. The considered particle starts its move in cell i at space-time point P_0. During its initial move of duration T_i, it traverses two small intermediate cells and finally arrives in cell j at point P_1. If the move was finished there, it would not be continued until $T_j - T_i$ seconds later from point P'_2 because of cell j's greater time step length T_j. This would be incorrect as the unstraight blue trajectory indicates. The particle has to account for the waiting time on cell j by continuing its move for another $T_j - T_i$ seconds, which results in the linear and therefore correct red trajectory through point P_2.

3. Every connector c that is due according to its individual time step length T_c calculates its average flow rates that hold for the next T_c seconds and notifies its adjacent cells of the resulting vehicle transmissions.

Since the particle still evaluates all traversed cells' velocities at the start time of its move, the resulting imprecisions remain in the order of the largest involved time step duration.

3.1.3 Particle Route Choice

3.1.3.1 Specification

Having stated the influence of macroscopic dynamics on individual particles, the converse problem of synchronizing macroscopic flows with individual particle behavior is considered next.

The route choice of particle n is expressed by a vector $\mathbf{u}_n(k) = (u_{ij,n}(k))$ of **turning move indicators**

$$u_{ij,n}(k) = \begin{cases} 1 & \text{if } n \text{ proceeds from cell } i \text{ to } j \text{ at time step } k \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

An additional state vector $\mathbf{x}^{\text{cnt}}(k) = (x_{ij}(k))$ is introduced. Each element $x_{ij}(k)$ represents the accumulated count of particles having turned from cell i to j until

time step k. The dynamics of these **turning counters** are defined by

$$\begin{aligned} \mathbf{x}^{\mathrm{cnt}}(0) &= \mathbf{0} \\ \mathbf{x}^{\mathrm{cnt}}(k+1) &= \mathbf{x}^{\mathrm{cnt}}(k) + \sum_{n=1}^{N} \mathbf{u}_n(k). \end{aligned} \tag{3.2}$$

The macroscopic turning fractions $\boldsymbol{\beta}(k) = (\beta_{ij}(k))$ can now be specified as a function $\boldsymbol{\beta}(\mathbf{x}^{\mathrm{cnt}}(k)) = (\beta_{ij}(\mathbf{x}^{\mathrm{cnt}}(k)))$ of the turning counters where

$$\beta_{ij}(\mathbf{x}^{\mathrm{cnt}}(k)) = \frac{x_{ij}(k)}{\sum_l x_{il}(k)}. \tag{3.3}$$

This is a maximum likelihood estimator of the turning probabilities if the turning moves follows a stationary multinomial distribution [87]. The resulting estimates can be directly fed into the macroscopic model by a substitution of $\boldsymbol{\beta}$ in (2.17). In order to avoid undefined 0/0 divisions at the beginning of a simulation, the turning counters can be initialized with small positive values instead of all zeros.

While the update equation in (3.2) assumes stationary turning probabilities, a straightforward approach to introduce time dependency is to define an additional forgetting parameter $w \in (0, 1)$ in a modified turning counter update equation

$$\mathbf{x}^{\mathrm{cnt}}(k+1) = w\mathbf{x}^{\mathrm{cnt}}(k) + (1-w)\sum_{n=1}^{N} \mathbf{u}_n(k). \tag{3.4}$$

In the absence of newly observed turning moves, this scheme causes an exponential forgetting of previously observed counts. A useful property of this filter is its infinite memory: Even if no particles arrive at an intersection for a while, the turning counts remain strictly positive and thus ensure well-defined flow splits in (3.3).

One possible problem with (3.4) is the danger of gridlock. If a traffic jam in one of an intersection's downstream cells causes all upstream cells' velocities to drop, it might take a long time until new particles reach the intersection and provide fresh turning move indicators that reflect these drivers' avoidance of the unavailable outgoing cell. An appropriate gridlock resolution logic is described in Appendix D.

A state space representation of the combined system (2.17) and (3.4) can now be given. Defining

$$\mathbf{x}(k) = \begin{bmatrix} \mathbf{x}^{\mathrm{ms}}(k) \\ \mathbf{x}^{\mathrm{cnt}}(k) \end{bmatrix} \tag{3.5}$$

and

$$\mathbf{f}[\mathbf{x}(k), \mathbf{u}_1(k) \dots \mathbf{u}_N(k), k] = \begin{bmatrix} \mathbf{f}^{\mathrm{ms}}[\mathbf{x}^{\mathrm{ms}}(k), \boldsymbol{\beta}(\mathbf{x}^{\mathrm{cnt}}(k)), k] \\ w\mathbf{x}^{\mathrm{cnt}}(k) + (1-w)\sum_{n=1}^{N} \mathbf{u}_n(k) \end{bmatrix}, \tag{3.6}$$

one obtains

$$\mathbf{x}(k+1) = \mathbf{f}[\mathbf{x}(k), \mathbf{u}_1(k) \dots \mathbf{u}_N(k), k]. \tag{3.7}$$

According to the notational conventions of control theory, the turning move indicators $\mathbf{u}_n$ act as control variables in this model. In fact, the individual

driver behavior *steers* the macroscopic traffic flow. $\mathbf{x}$ is subsequently denoted as the **macroscopic state** of the mobility simulation. Note that $\mathbf{x}$ does not account for the microscopic states of individual particles. The combined state transition function $\mathbf{f}$ is linearizable with respect to $\mathbf{x}$ and all $\mathbf{u}_n$ because of the linearizability of its constituting functions (2.17), (3.3), and (3.4). This implies that the effect of an agent's route choice on the macroscopic states can be linearly predicted as the sum of the effects of its turning moves.

The state space model described so far captures mobility only within the network but does not account for vehicle entries and exits. These extensions require the more concise formalization of travel demand given in the second half of this chapter. Regarding linearizability, it can already be stated that the macroscopic effect of a particle's entry or exit can be linearly approximated since an entry or exit move corresponds macroscopically merely to a local occupancy modification.

3.1.3.2 Simulation on Variable Time Scales

If the macroscopic mobility simulation runs on variable time steps, the rows of (3.4) are evaluated at likewise variable frequencies:

$$\begin{aligned} x_{ij}(rT_c + s) &= x_{ij}(rT_c) \quad r \in \mathbb{N},\ s \in \{0, \ldots, T_c - 1\} \\ x_{ij}(rT_c + T_c) &= w_c x_{ij}(rT_c) + (1 - w_c)\frac{1}{T_c}\sum_{s=0}^{T_c-1}\sum_{n=1}^{N} u_{ij,n}(rT_c + s) \end{aligned} \tag{3.8}$$

where T_c is the time step duration of the connector c that is crossed by turning move ij. An individual weight w_c is necessary for every such connector in order to maintain the same degree of averaging for all turning counters.

If the number $\sum_{n=1}^{N} u_{ij,n}(k)$ of microscopically simulated ij turning moves during a single simulation time step is Poissonian with expectation and variance λ_{ij}, the variance of x_{ij} as defined in (3.8) approaches

$$\lim_{r\to\infty} \mathrm{VAR}\{x_{ij}(rT_c)\} = \frac{1 - w_c}{1 + w_c}\frac{\lambda_{ij}}{T_c}. \tag{3.9}$$

A derivation of this equation can be found in Appendix E. The network time constant $\hat{T}$ defined in Section 2.6 is now employed to postulate that a turning counter's variability must be independent of its connector's time step length and, more specifically, identical to

$$\mathrm{VAR}\left\{\frac{1}{\hat{T}}\sum_{s=0}^{\hat{T}-1}\sum_{n=1}^{N} u_{ij,n}(r\hat{T} + s)\right\} = \frac{\lambda_{ij}}{\hat{T}}. \tag{3.10}$$

This variance would result if the turning counters were averaged non-recursively on a temporal grid as coarse as the network time constant. Equating (3.9) and (3.10) yields

$$w_c = \frac{\hat{T} - T_c}{\hat{T} + T_c}. \tag{3.11}$$

An infinite turning counter memory is guaranteed if all T_c are chosen strictly smaller than $\hat{T}$. The working of this specification is illustrated in Figure 3.3.

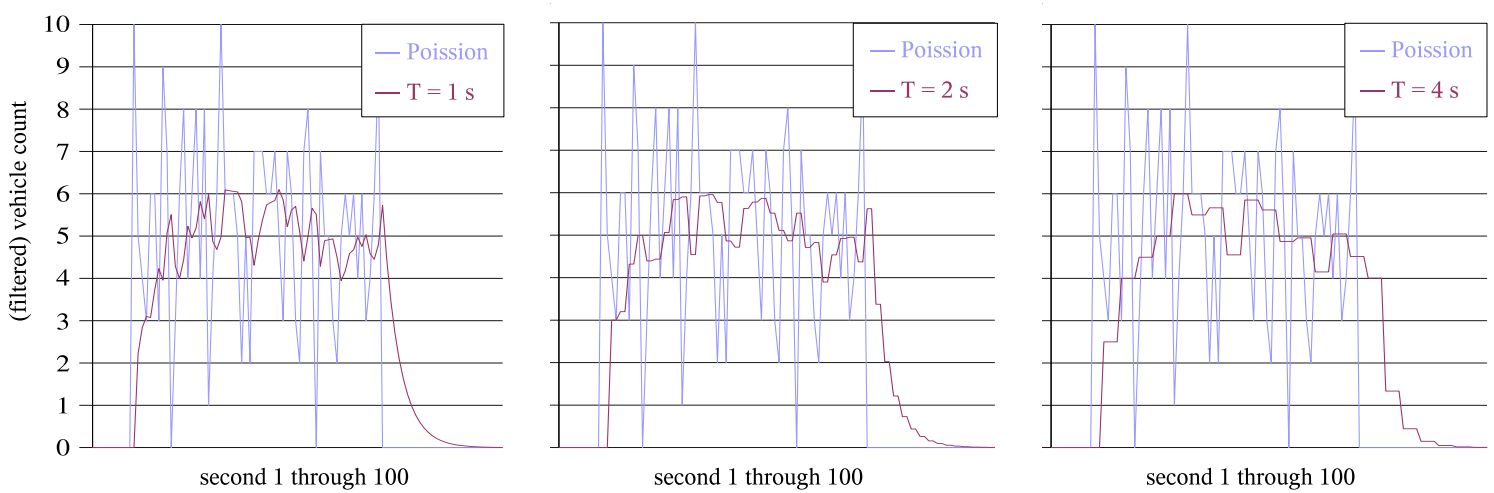


Figure 3.3: Turning counter dynamics

Three turning counters (red) with time step durations of 1, 2, and 4 seconds track a Poissonian signal (blue) for a duration of 100 seconds. The signal's expectation jumps from 0 to 5 after 10 seconds and returns to 0 after another 60 seconds. The network time constant $\hat{T}$ is 8 seconds in all cases. All counters exhibit a similar variability and speed of adaptation.

For a simulation time step length of one second, the requirement of an infinite memory dictates a minimum network time constant of two seconds. Given this inertia, a precise macroscopic tracking of individual vehicles is not possible. However, such a precision is rather undesirable for the purpose of this work. The simulated driver population is an output of MATSim, the mobility simulation of which is a queuing model with relatively limited expressive power but a high computational performance [41]. It accounts for signalized intersections merely by average flow capacity reductions, which results in relatively undisturbed traffic streams. Maintaining this modeling fidelity, a macroscopic reproduction of individual vehicle movements would only introduce additional discretization noise into (3.7) – an utmost undesirable effect since this model is to be linearized.

In a planning context, a network time constant of several minutes is a good choice. It must not be too large since otherwise the macroscopic model eventually looses track of the driver behavior. A reasonable upper bound for the network time constant is the time interval at which traffic information is averaged before it is fed back to the simulated travelers who in turn react to this information by possible turning move changes.

3.1.4 Computational Model Investigation

The micro/macro model's precision and the accelerating effect of the simulation logic on variable time scales are investigated. All experiments are conducted on a 1.7 GHz Pentium 4 machine with 1 GB RAM, using the Sun Java Runtime Environment 5.0 [84].

A synthetic population of 206 353 motorist travelers with complete daily plans is available for the Berlin network introduced in Section 2.6.2 [153]. This is a 10 percent sample of Berlin's true motorist population. Thus, 10 macroscopic vehicle units need to be inserted together with one particle into the simulation. However, since the simulations are run on a thinned out version of the full Berlin network, the use of 2 instead of 10 macroscopic vehicle units per particle already creates realistic congestion patterns.

The following experiments consider the morning rush hour from 6 to 12 am. Figure 3.4 shows the total number of moving vehicles as a function of time. More than 16 000 particles, i.e., 32 000 macroscopic vehicle units, are concurrently simulated during the rush hour peak at approximately 8:30 am.

3.1.4.1 Precision of Micro/Macro Coupling

The microscopic behavior influences the macroscopic flow splits via the turning counter mechanism, whereas the microscopic movements are guided by the macroscopic velocity field. The precision of this micro/macro model synchronization is investigated here.

Figure 3.5 shows the microscopic and macroscopic traffic density trajectories for two selected links of the Berlin network. Macroscopic density is the ratio of macroscopic vehicle units on a link to the link's space capacity. The space

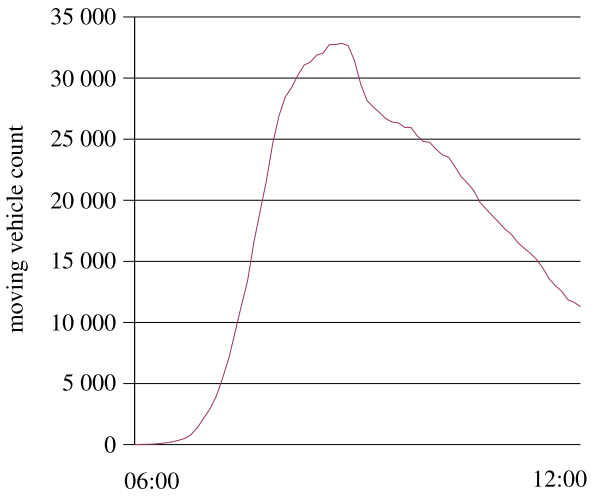


Figure 3.4: Simulated Berlin morning peak
A simulation of the Berlin morning peak between 6 and 12 am. The curve shows the macroscopic number of moving vehicles over time.

capacity of a link is defined as its length times its number of lanes. Microscopic density is calculated here as the quotient between *twice* the microscopic particle count on a link and its space capacity. The factor of two accounts for the fact that one particle represents two vehicle units in the given experimental setting.

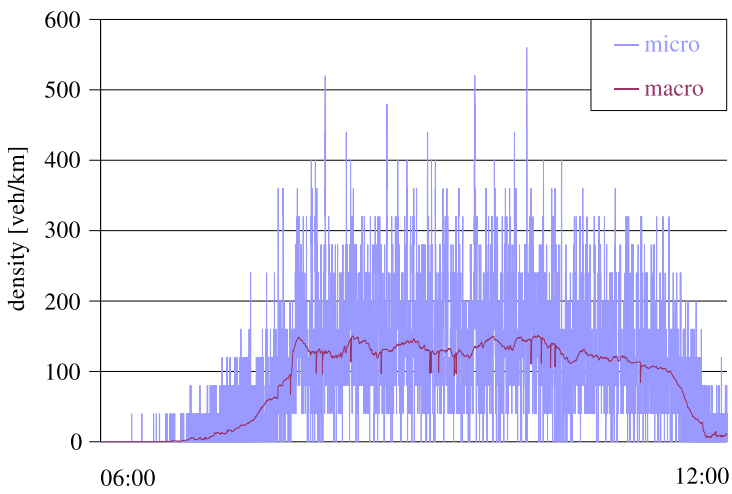
Link (a) is only 25 meters long, whereas link (b) has a length of 1611 meters. This difference is reflected in the much greater variance of the microscopic density on the shorter link. Both macroscopic density trajectories track the microscopic trends with high precision and almost no lag. The strong discretization noise particularly on the shorter link is significantly reduced.

In order to avoid arbitrariness, these links were automatically chosen according to the following criteria: Link (a) exhibits the largest ratio of density to space capacity during the rush hour peak, whereas link (b) carries the largest total amount of vehicle units, i.e., the largest product of density and space capacity, in the same time interval. That is, the first criterion prefers small links, and the second criterion prefers large links. Both criteria favor congested links since uncongested conditions prevail anyway before the rush hour sets in.

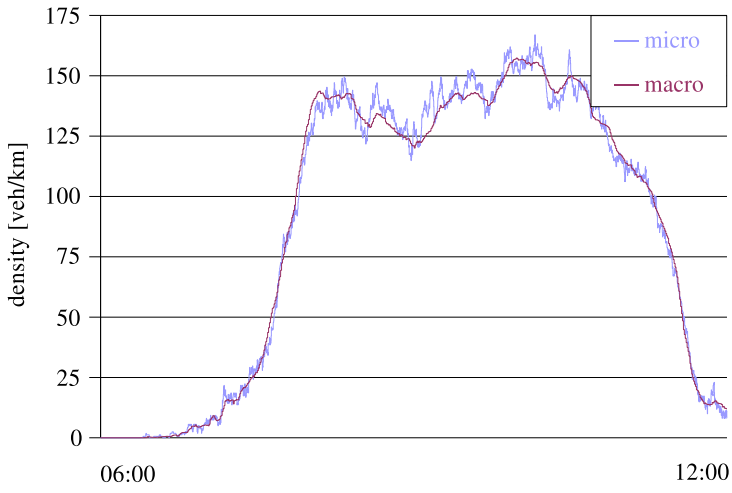
The macroscopic densities beyond 133 veh/km indicate that the gridlock resolution mechanism described in Appendix D actively influences the traffic dynamics. This shows that the purely macroscopic gridlock resolution logic is compatible with the microscopic model components.

The network time constant is chosen as large as 5 minutes. This is justified in light of the 15 minute time bins in which MATSim averages travel times before feeding them back to the simulated travelers in its iterative simulation procedure, cf. Sections 1.2.2.3 and 3.2.2.3.

The difference between this model and a typical mesoscopic approach is emphasized. The presented macroscopic trajectories are not calculated by some kind



(a) Microscopic and macroscopic density trajectory for a short link of 25 m length under heavy congestion. The discrete value domain of the microscopic curve reflects the strong vehicle discretization noise. The macroscopic curve removes most of this noise. Unrealistically high microscopic densities are possible because of the massless particles. The macroscopic curve, however, is within bounds.



(b) Microscopic and macroscopic density trajectory for a 1.6 km long link under heavy congestion. The discretization noise has a weaker effect since a greater number of particles is averaged in the microscopic density calculations. The microscopic signal trend is tracked very well by the macroscopic curve.

Figure 3.5: Precision of micro/macro model synchronization

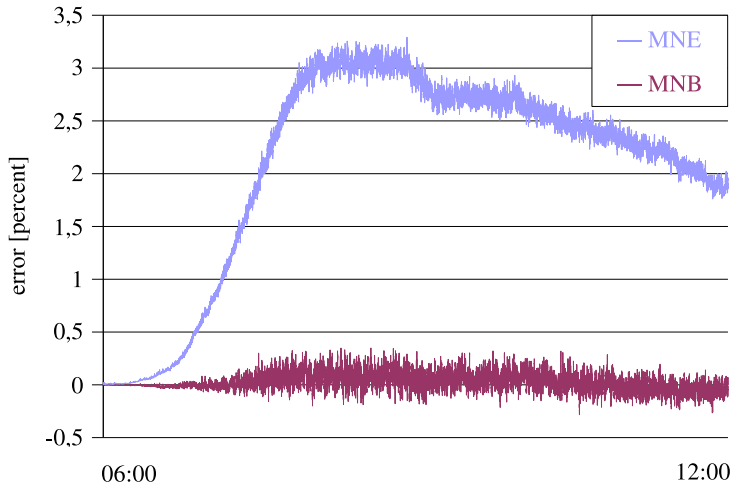


Figure 3.6: Mean normalized bias and error trajectories
Mean normalized bias MNB and mean normalized error MNE as defined in (3.12) and (3.13). The intermediate microscopic excess in MNB of about 1 per mille is negligible and owed to the particle entrance mechanism which puts particles ahead of their macroscopic flow into the system. Likewise, there is a similar undershoot as the particles leave the system ahead of their macroscopic flow at the end of the rush hour.

of microscopic vehicle count averaging. Rather, they implicitly result from continuously tracked turning fractions that guide an appropriate amount of truly macroscopic flow across each link.

A network-wide point of view is adopted by means of the following two characteristics:

$$\mathrm{MNB}(k) = \frac{100}{|A|} \sum_{a \in A} \frac{\varrho_a^{\mathrm{micro}}(k) - \varrho_a^{\mathrm{macro}}(k)}{\hat{\varrho}} \tag{3.12}$$

represents the mean normalized bias where $\varrho_a^{\mathrm{macro}}(k)$ ($\varrho_a^{\mathrm{micro}}(k)$) is the macroscopic (microscopic) vehicle density on link a in time step k, $\hat{\varrho}$ is the macroscopic jam density of 133 veh/km, and A is the set of all links in the network. The second characteristic

$$\mathrm{MNE}(k) = \frac{100}{|A|} \sum_{a \in A} \frac{\left|\varrho_a^{\mathrm{micro}}(k) - \varrho_a^{\mathrm{macro}}(k)\right|}{\hat{\varrho}} \tag{3.13}$$

is the mean normalized error.

Figure 3.6 shows that MNB fluctuates unsystematically around 0 percent. This indicates that the mass balance between microscopic and macroscopic flow is well maintained. The maximum value of approximately 3 percent for MNE is moderate and plausible in consideration of Figure 3.5.

These results show that the micro- and the macro-model are well synchronized despite of their sparse interactions. The resulting macroscopic traffic characteristics exhibit a significantly lower discretization noise than a simple average over the microscopic particles.

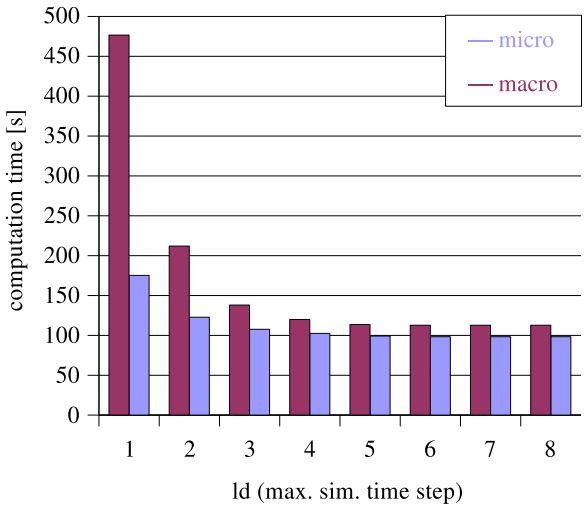


Figure 3.7: Microscopic and macroscopic computation times
Microscopic and macroscopic computation times over $\log_2$ of the greatest allowed time step duration. The simulated time span is 6 hours.

3.1.4.2 Computational Performance

The imprecisions introduced by the simulation scheme on variable time scales are now justified by their countervailing computational benefits. The same morning peak scenario as before is considered.

The computational effort for the micro- and for the macrosimulation is distinguished in the following way. The macrosimulation comprises all processes described in Chapter 2 plus the turning counter tracking described in Section 3.1.3. The microsimulation comprises the additional operations necessary to update the individual particle locations as described in Section 3.1.2. In consequence, the total computational effort is the sum of micro- and macrosimulation.

Figure 3.7 shows the microscopic and macroscopic computation time over $\log_2$ of the greatest allowed simulation time step duration, which is roughly the same as the network time constant $\hat{T}$.[1] The overall number of computations is proportional to the number of network elements and to the frequencies at which these elements are updated. An increased $\hat{T}$ affects both, the element count and the calculation frequency. Thus, the computation times initially decrease quickly with $\hat{T}$ but then stabilize because of the geometrical constraints on the link and node time step durations. Choosing large cells and long time steps does not only reduce the number of macroscopic calculations but also decreases the frequencies at which the microscopic particles are updated.

Figure 3.8 shows the real time ratio, i.e., the ratio of simulated time to the time required to run the simulation. The accomplished maximum value is 90. This

[1] More precisely, the network time constant $\hat{T}$ is slightly larger than the greatest allowed simulation time step duration in order to ensure an infinite turning counter memory, cf. Section 3.1.3.2.

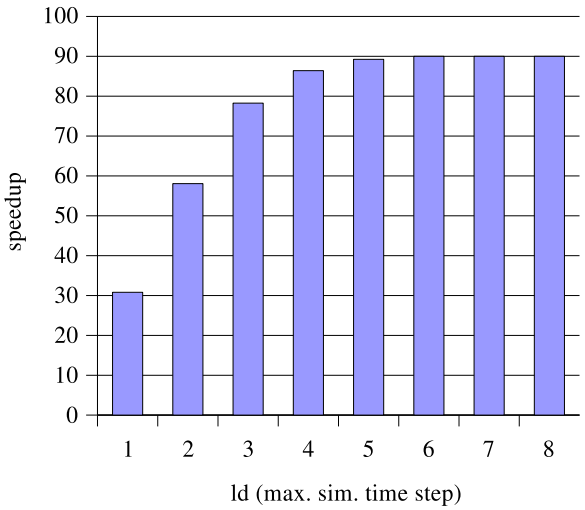


Figure 3.8: Real time ratio
Real time ratio over $\log_2$ of the largest simulation time step duration in the network. These values account for all operations of the simulation system and include a number of supplementary procedures. In consequence, the evaluated running time is slightly larger than the sum of pure micro- and macrosimulation.

shows that the model is ready for real-time simulations of large-scale scenarios. In summary, its computational efficiency is owed to the following properties:

- The model does not require a realistic number of particles. If, for example, only a 10 percent sample of the complete population is loaded on the network, the macroscopic equivalent of 10 vehicles is inserted into the system together with every particle. The chosen sample size must be large enough to properly represent the actual population's behavior but otherwise can be minimized for high computational performance.
- The macroscopic mobility simulation only moves single-commodity flows. No care has to be taken of partial densities as it would be the case if behavioral aspects were represented macroscopically.
- Every link is simulated with a cell size and a time step length that are optimally adjusted to its characteristics.

Altogether, two results obtained in this section are useful independently of a state estimation problem. First, it is shown how a general macroscopic traffic flow model can be employed to simulate microscopic travel behavior. A useful feature of this approach is its ability to remove vehicle discretization noise. Second, the macroscopic simulation logic on variable time scales, cf. Section 2.5, is extended towards this micro/macro coupling scheme and exhibits a high computational performance.

Important for estimation, the linearizability of state space model (3.7) is maintained throughout the entire development. This provides the sensitivity infor-

mation that is subsequently applied to predict the linearized effect of an individual driver's turning move sequence on the global network conditions without repeated simulations.

3.2 Simulation of Drivers' Choices

The first part of this chapter specifies physically observable driver behavior as a sequence of turning moves. In the following, the decisions that precede this behavior are discussed and formalized in a way that allows for a seamless linkage to the previously described micro/macro mobility simulation. The resulting behavioral representation is logically compatible with the estimation algorithm developed in the next chapter and technically compatible with a MATSim-like simulation system. Since this dissertation does not contribute to the field of behavioral modeling itself, the following discussion is kept problem-specific and is not exhaustive from a behavioral modeling point of view.

3.2.1 Choice Formalism

It is assumed that, whenever a traveler is faced with a situation that calls for a decision, this traveler chooses precisely one element from a nonempty set of discrete alternatives. The decision making process itself is structured according to the framework given in [21]:

1. definition of the choice problem,
2. generation of alternatives,
3. evaluation of attributes of alternatives,
4. choice,
5. implementation.

These steps are made precise in the remainder of this section. Note that a reactive decision protocol as defined in Section 1.3.1 may repeat steps 2 and 3 several times before a choice is made.

The discussion omits specific modeling assumptions and algorithmic details that would be necessary for the realization of an applicable behavioral model. This is justified by the intention to provide an estimator that is compatible with a broad range of behavioral models and by the rather technical assumption that the estimator is likely to be attached to an existing traffic simulator, cf. Section 1.3.1. Only a few selected modeling aspects that are referred to in the later developments are discussed at the end of this chapter.

3.2.1.1 Definition of the Choice Problem

Most of the terminology introduced here is consistent with the MATSim system specification given in [149]. However, the underlying conceptions are more universally applicable to the modeling of travel behavior and are not confined to this software.

Plans The activity and traveling intentions of an individual are denoted as her plan. For simplicity, only plans for a single day are considered. Physically, a plan describes a round trip through the transportation network. This round trip comprises a sequence of legs that connect intermediate stops during which activities are conducted. The first and last activity of a plan typically take place at the individual's home location.

Activities are defined in terms of their type (e.g., work, leisure), location (a link in the network), start time, and end time or prespecified duration. Two subsequent activities are connected by a leg. While in general a leg can be associated with different modes (e.g., car, public transport, walking), this thesis considers only individual motorist travelers such that a leg always implies a vehicular movement through the road network. A motorist leg is parameterized by origin and destination link, route (a sequence of links that connects origin and destination), and departure time. Only a desired arrival time can be prespecified since the actual time of arrival depends on the prevailing traffic conditions.

When a traveler chooses her course of action for a given day, she equivalently chooses a plan for that day. It is possible to disaggregate the choice of a plan into a logical or temporal sequence of decisions [27, 99]. The latter method is naturally applicable to within-day replanning, where a traveler continuously reconsiders and adjusts her current plan according to pre- and en-trip collected information. Formally, the choice of a plan segment where some degrees of freedom are fixed is not different from the choice of a full plan, and no such differentiation is made in the following. For example, an en-trip route switching model maintains all activity locations and timings of the present plan. Equivalently, route switching can be represented as the choice of a completely new plan where all degrees of freedom apart from route choice are constrained to be identical to those of the original plan.

Generalized Paths The conception of a plan is now formalized in a way that is amenable to the likewise formal derivation of a behavioral estimator.

A simple route $\mathcal{U}$ connects two subsequent activity locations. It is defined as a (physically feasible) sequence of turning moves

$$\mathcal{U} = \ldots \mathbf{u}(k-1), \mathbf{u}(k), \mathbf{u}(k+1) \ldots = \{\mathbf{u}(k)\}_k \tag{3.14}$$

with $\mathbf{u}(k)$ specified in (3.1). The representation of a route as a sequence of turning moves rather than a sequence of links maintains consistency with the microscopic driver representation specified in the first half of this chapter. It can be thought of as an ordinary edge sequence in an "inverted" network where vertexes represent links and edges represent turning moves, cf. Figure 3.9. A sequence of turning moves uniquely defines a sequence of original links, and vice versa.

The round trip that physically corresponds to an all-day plan is formalized as a (cyclic) path by minor modifications to the inverted network. Every vertex v of the inverted network that represents an activity location is complemented with an additional vertex v' that represents the actual execution of an activity at this location. The start of an activity is then equivalent to a turning move $v \to v'$, and its end can be identified by a $v' \to v$ move. A plan's full sequence

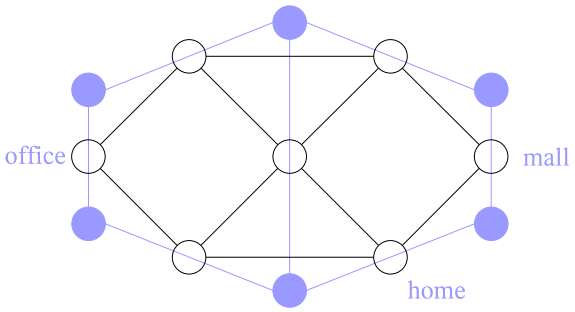


Figure 3.9: Route choice
The original road network is drawn in blue. Three of its links serve as activity locations (office, mall, home). The inverted network for route representation is drawn on top in black. It represents every original link by a vertex and every possible turning move by an edge.

of activities and legs now comprises a single round trip through the inverted network, with cycles at the activity locations. Figure 3.10 provides an example.

This formalism simplifies notation since it allows to represent all physically relevant aspects of a full plan consistently with (3.14) in terms of a **generalized path** $\mathcal{U}$. If only a plan segment is to be represented, its generalized path segment also contains only the corresponding subset of turning moves. Subsequently, the notions of a path and a generalized path will be used synonymically whenever the context allows to distinguish them from a **simple route** that only connects two links in the network.

Traffic flow model (3.7) can be steered by generalized paths instead of simple routes without formal modification. Since the effect of entering and exiting vehicles can be linearly approximated by this model, it is also linearizable with respect to the newly introduced turning moves that represent such entries and exits. This implies that the effect of an agent's plan choice on the macroscopic network conditions can be linearly predicted in the same vein as it has been demonstrated for route choice in Section 3.1.3.1.

Since a generalized path $\mathcal{U}$ is a formal representation of an individual's intentions, it represents an aspect of that individual's **mental state**. Its notation in terms of the typical control symbol "**u**" is maintained here since the largest portion of this thesis deals with the steering effect of driver behavior on macroscopic traffic dynamics. The definition of a full state space model for a combined micro/macro traffic system that includes some kind of mental dynamics is not necessary for the purpose of this dissertation.

3.2.1.2 Generation of Alternatives

The **choice set** of behavioral alternatives available to decision maker n is denoted by C_n. The elements of this set are plans, formally represented by (generalized) paths $\mathcal{U}$. It is reasonable to assume that C_n is significantly smaller than the set of all thinkable plans: The elements in C_n must be compatible with the goals and commitments of a traveler, cf. Section 1.2.2.3. The limited

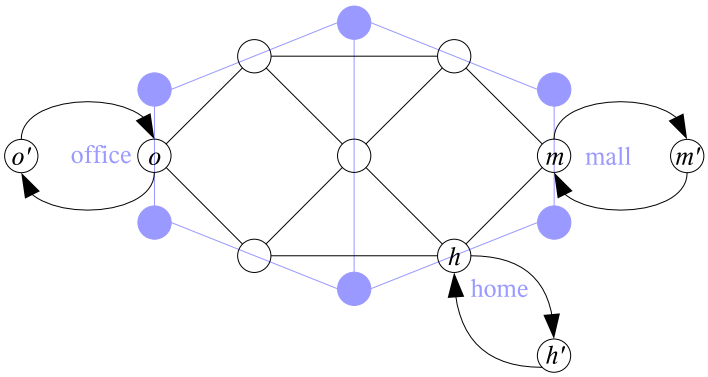


Figure 3.10: Generalized path choice
The same physical network as shown in Figure 3.9. Cycles are added to all possible activity locations. An exemplary plan that consists of the activity sequence home→work→shop→home now consists of one round trip through the inverted network, with cycles at the activity locations. Its equivalent sequence of vertexes is $h', h, \ldots, o, o', o, \ldots, m, m', m, \ldots, h, h'$.

knowledge of the decision maker excludes all unknown options from consideration. Physical, legal, and individual (e.g., financial, constitutional) constraints further reduce the choice set. If a traveler reconsiders only a segment of her current plan, an additional constraint on C_n is that everything but this segment must remain unchanged in all alternative plans.

It is required that a non-empty choice set C_n is available to every agent n in every situation that calls for a decision. This choice set may be specified in two different ways, depending on the deployed decision protocol, cf. Section 1.3.1:

- A reactive decision protocol incrementally constructs a set of considered alternatives given a particular choice situation. Different suchlike sets may be generated in repetitions of otherwise identical conditions because of probabilistic components in the generation procedure. In this case, C_n comprises all possibly generated alternatives.
- In a deliberative decision protocol, the choice set has typically been generated prior to the actual choice situation. That is, C_n is explicitly and deterministically prescribed, even if it was originally generated by a randomized algorithm.

The goal of this work is to treat the decision protocol as much as a black box as possible. The only requirement implied by the above listing is that there exists a nonempty set C_n of alternatives that contains all possible choices of agent n in a given situation. However, an enumeration of this set is not required.

3.2.1.3 Evaluation of Attributes of Alternatives

The **systematic (deterministic) utility** of an alternative, represented by a real-valued number, is a model of the benefits a decision maker expects from

choosing this alternative. It reflects the decision maker's preferences. Utility perception can vary among decision makers, and clearly utility can differ among alternatives. Formally, a systematic (deterministic) utility $V_n(\mathcal{U})$ is associated with every plan $\mathcal{U}$ in the choice set C_n of traveler n.

The utility of a plan is comprised of two components: positive utility for the execution of activities and negative utility (disutility, cost) for travel itself. Typical aspects of route (dis)utility are travel time, distance traveled, number of left-turns, number of signalized intersections, and contact with insecure neighborhoods [18, 20]. The utility of an activity varies depending on the type of activity, its context within the entire plan, and the timing of its execution [43].

If a utility-driven modeling approach is adopted, it is required that the systematic utility for every plan of any agent can be calculated by the utility function shown in Figure 1.1 and that the resulting utility combines all of the aforementioned (dis)utility components in a single number. This evaluation only has to be available on request and on a per-plan basis. It is not required that the choice set is enumerated for a complete evaluation before a choice is made. Furthermore, if the decision protocol sequentially composes a choice, e.g., by incrementally building a plan as a sequence of activities and legs, the utility function may be limited to an evaluation of the according plan components.

3.2.1.4 Choice

The choice of a certain plan (segment) is modeled non-deterministically. The probability that decision maker n chooses plan $\mathcal{U} \in C_n$ is denoted by $P_n(\mathcal{U})$. This choice distribution may be parameterized in an agent-specific way but otherwise is required to depend only on the attributes of the elements in C_n. If the choice model is utility driven, the attributes of a plan must be represented by its utility.

A probabilistic choice logic may represent randomness in human behavior or account for modeling imprecisions [21]. The specific modeling assumptions that underly a particular decision protocol are not relevant for the subsequently developed estimation approach beyond the fact that behavior is uncertain at all. Otherwise, there would be no scope for a behavioral adjustment.

Neither an enumeration of the choice set nor an explicit (e.g., closed-form) representation of the implemented choice distribution need to be available. Only realizations of choices must generated by the behavioral simulation system.

3.2.1.5 Implementation

The implementation of a choice requires its realization in the mobility simulation. However, an agent with an imperfect knowledge of the actual traffic conditions may observe an inconsistency between what it wants to do and what is physically possible. In particular, the generalized path representation of a plan comprises a sequence of turning move indicators that prespecify the timing of every turning move and every entry/exit move in the network. It is unlikely that the (congested) traffic conditions admit precisely this timing.

It therefore is assumed that a plan is *robust* in that it cannot be invalidated by finite changes in travel times. An example of a robust plan is one where (i) the activities have no fixed start time but rather a prespecified duration and (ii) the legs only specify a sequence of links but not the timing of their entry. Consequently, a once chosen plan can always be executed in the mobility simulation without further replanning. The MATSim plans are robust in this regard.

A precise formalization of this situation would require to supplement the mobility simulation (3.7) with another model component that updates the plans $\mathcal{U}_n = \{\mathbf{u}_n(k)\}_k$ for all agents $n = 1 \dots N$ in every simulation time step k such that their consistency with the physical situation is maintained. However, since the actually implemented mobility simulation does not require the generalized path abstraction at all, the fictitious existence of such a model component merely maintains *formal* consistency whenever it is stated that "$\mathcal{U}_1 \dots \mathcal{U}_N$ are loaded on the network" or "$\mathcal{U}_1 \dots \mathcal{U}_N$ are fed into the mobility simulation".

The generalized paths $\mathcal{U}_1 \dots \mathcal{U}_N$ uniquely specify both the intended and the implemented driver behavior. Therefore, no formal differentiation between these aspects is subsequently made.

3.2.2 Specific Modeling Assumptions

The structural outline given above is made precise in terms of two fairly different modeling approaches.

Random utility models (RU models, RUMs) constitute a broadly applicable class of choice models that are based on reasonable behavioral assumptions and sound mathematical inference. The simple mathematical structure of certain RUMs is exploited in the derivation of a behavioral estimator.

MATSim's behavioral model basically relies on a dynamical systems assumption about human learning. Since the resulting model behavior is defined rather implicitly through this learning process, and since the dynamics of this process are not yet well-understood, MATSim constitutes a particularly challenging model for a behavioral estimator.

3.2.2.1 Random Utility Models

RUMs constitute the mainstay of travel behavior modeling, and a specific implementation of the decision protocol is likely to be based on RU theory [21, 22]. The RU modeling assumptions are outlined below.

It is assumed that a decision maker n always chooses the alternative of greatest perceived utility from her prespecified choice set C_n. The systematic utility $V_n(\mathcal{U})$ constitutes only an imperfect model of her true utility perception. In order to reflect this imprecision, a random error component $\varepsilon_{\mathcal{U},n}$ is added to the systematic utility of every alternative $\mathcal{U}$. The probability $P_n(\mathcal{U})$ that $\mathcal{U}$ is chosen thus equals the probability that the random utility of $\mathcal{U}$ is greatest among all alternatives:

$$P_n(\mathcal{U}) = \Pr(V_n(\mathcal{U}) + \varepsilon_{\mathcal{U},n} \geq V_n(\mathcal{V}) + \varepsilon_{\mathcal{V},n}, \forall \mathcal{V} \in C_n). \tag{3.15}$$

Closed-form expressions for these choice probabilities can be obtained for certain joint distributions of the error components. But even if no such closed form can be found, a simulation of choices that are consistent with (3.15) is possible. The procedure requires (i) to draw a disturbance from the joint error distribution for all alternatives, possibly through a simulation procedure as described below, and (ii) to deterministically choose the alternative of greatest disturbed utility.

3.2.2.2 Models of Route Choice

The two major modeling approaches to route choice have already been addressed in Section 1.2.2.2: Either route (re)planning is realized by the calculation of a best path, or a route is chosen probabilistically from a prespecified choice set.

Behaviorally, the calculation of a best path is an idealization. It implies global network knowledge and an optimal choice mechanism given a certain objective function such as trip travel time. The effective calculation of a best path requires route cost to be additive in link cost which ignores existing evidence for nonlinear cost perception. Probabilistic route choice allows for greater realism. A choice set of routes can be generated in a way that is consistent with a driver's (usually limited) knowledge of available alternatives. There is no limitation of link-additive costs. The random choice component properly reflects behavioral and modeling uncertainties [148].

Computationally, best path has an edge over probabilistic choice. Routing problems have been intensively studied in computational science and efficient solution algorithms are available for problems with link-additive cost [83]. In contrast, probabilistic choice implies some computational overhead. Choice set generation itself is a nontrival task [20, 148]. Every agent's individual choice set has to be stored and processed during simulation, and every alternative needs to be evaluated for the simulation of a single choice. Contrarily, the efficiency of best path algorithms is owed to their avoidance of path enumeration [130].

The realism of probabilistic choice and the efficiency of routing algorithms can be combined. Since best path routing is a cost minimization procedure, it can be applied to model a decision maker's rational choice given a simulated error of utility perception. This coincides with the aforementioned simulation procedure for RUMs. In this context, it is interesting to inspect a variation of the route choice model implemented in the MATSim planning simulation. MATSim models the day-to-day evolution of driver behavior as a continuous learning process. Speaking only in terms of routes, a certain fraction of drivers is allowed to recalculate new routes at the beginning of every simulated day. These routes are generated based on previously simulated link traversal costs by a time-dependent best path algorithm. The simultaneous execution of all routes results in experienced costs that are likely to differ from those costs based on which the new routes were calculated. This implicitly simulates a perceptional error that is identical for all replanning agents and equal to the difference between the actually experienced costs and the costs assumed during replanning. This logic even avoids the explicit generation of perceptional errors but is not derived from RU theory.

The *path size logit* (PS-logit) model defines closed-form route choice probabilities. Its derivation from RU theory can be found in [67]. This model is

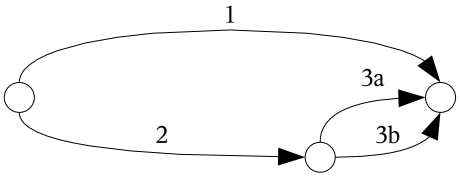


Figure 3.11: Three routes example
A simple route choice example with three alternative routes $\mathcal{A}$ (comprised of link 1), $\mathcal{B}$ (comprised of link sequence 2 →3a), and $\mathcal{C}$ (comprised of links 2→3b). The length of link 1 is l, that of links 3a and 3b is dl, and that of link 2 is $l - dl$.

presented here since its particular structure allows for some formal manipulations that greatly simplify the behavioral estimation problem. PS-logit specifies the probability that individual n chooses route $\mathcal{U} \in C_n$ by

$$\begin{aligned} P_n(\mathcal{U}) &= \frac{e^{\mu V_n(\mathcal{U}) + \ln \mathrm{PS}_n(\mathcal{U})}}{\sum_{\mathcal{V} \in C_n} e^{\mu V_n(\mathcal{V}) + \ln \mathrm{PS}_n(\mathcal{V})}} \\ &= \frac{\mathrm{PS}_n(\mathcal{U}) e^{\mu V_n(\mathcal{U})}}{\sum_{\mathcal{V} \in C_n} \mathrm{PS}_n(\mathcal{V}) e^{\mu V_n(\mathcal{V})}}. \end{aligned} \tag{3.16}$$

It is instructive to start the discussion with all PS parameters set to one. Then, specification (3.16) collapses into the *multinomial logit* (MNL) model, the arguably simplest and most popular RUM. The positive scale parameter μ controls to what degree routes of higher systematic utility are preferred. If $\mu \to 0$, all routes are chosen with equal probability, whereas $\mu \to \infty$ deterministically selects a route of maximum utility.

In a route choice context, the major drawback of MNL is its inability to model situations with overlapping routes. This is most easily demonstrated by an example. Figure 3.11 shows a simple four-link network. Three routes $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ connect the leftmost to the rightmost node. All routes have equal utility $\bar{V}$ such that MNL invariably predicts a uniform route split $(P(\mathcal{A})\ P(\mathcal{B})\ P(\mathcal{C})) = (1/3\ 1/3\ 1/3)$. This is not realistic because routes $\mathcal{B}$ and $\mathcal{C}$ have a large overlap and therefore are likely to be perceived as a single alternative. Behaviorally reasonable route splits thus approach $(1/2\ 1/4\ 1/4)$ as the overlap of $\mathcal{B}$ and $\mathcal{C}$ gets larger.

PS-logit corrects the MNL model by specifying

$$\mathrm{PS}_n(\mathcal{U}) = \sum_{a \in \Gamma_\mathcal{U}} \frac{l_a}{L_\mathcal{U}} \frac{1}{\sum_{\mathcal{V} \in C_n} \delta_{a\mathcal{V}}} \tag{3.17}$$

where $\Gamma_\mathcal{U}$ is the set of all links in route $\mathcal{U}$, l_a is the length of link a, $L_\mathcal{U}$ is the length of route $\mathcal{U}$, and $\delta_{a\mathcal{V}}$ is one if link a is contained in route $\mathcal{V}$ and zero otherwise. That is, $\sum_{\mathcal{V} \in C_n} \delta_{a\mathcal{V}}$ counts how many routes in C_n contain link a. Each addend in (3.17) represents the contribution of a single link to the path size of route $\mathcal{U}$, and $\mathrm{PS}(\mathcal{U})$ measures to what degree route $\mathcal{U}$ is perceived as a distinct alternative. It is one if $\mathcal{U}$ has no overlap with other routes, and it approaches zero the greater $\mathcal{U}$'s overlap with other routes becomes. A perfect overlap of routes $\mathcal{B}$

and $\mathcal{C}$ in the above example yields path sizes (PS($\mathcal{A}$) PS($\mathcal{B}$) PS($\mathcal{C}$)) = (1 1/2 1/2) that generate the behaviorally reasonable route splits (1/2 1/4 1/4) when inserted in (3.16).

The purposeful nature of these examples is emphasized. Alternative utility correction terms and path size definitions have been proposed in the literature [38, 67] as well as alternative RU models that are not limited to the simple structure of (3.16) [18, 20, 148].

3.2.2.3 Models of Plan Choice

Even with realistic restrictions on possible activity sequences, locations, and timings, and with a likewise restricted route choice set, the combinatorial number of available plans quickly becomes intractable. For a single day, roughly 10^{17} alternative behavioral patterns per traveler are estimated in [27]. It is not realistic to assume that travelers possess the computational resources to process such a choice set. However, they do make a decision in some way, and therefore it appears justified to simulate plan choice by simplifying heuristics that resemble human decision making [71].

This approach is also chosen in the MATSim planning simulation. A traveler's plan is "scored" by a utility function that comprises positive addends for activity execution and negative addends representing travel costs [43]. Every simulated traveler strives to maximize its score by explorative day-to-day learning. This is realized as a simplified classifier system [149]: A small set of (typically five) alternative plans is memorized by an agent. Every simulated day, one of these plans is executed and the experienced score is memorized. Occasionally, a new plan is generated, executed, and the worst plan is discarded. New plans are generated by variations of old ones. Routes are recalculated as best paths based on previously observed link traversal costs [130], and activity timings are chosen by a variety of heuristics such as random search [10], reinforcement learning [44], and evolutionary algorithms [43, 120].

Plan selection itself is implemented as a simple RU model. However, the continuous choice set evolution by explorative learning prevents a straightforward RU interpretation and also complicates a mapping on the structural system requirements that are presupposed for estimation. There are three difficulties.

1. The plan choice set is variable. If it was fixed after a limited number of iterations, the simulation until that point could be regarded only as a fairly heavyweight choice set generation process. However, the limited number of memorized plans in such a setting (rather a technological problem) could raise an issue of behavioral variability.

2. Plan choice is not based on deterministic utilities but on continuously updated scores. While score expectations are technically easy to estimate by recursive averaging, their very existence requires that the simulation converges towards a stationary distribution of network conditions. This property is yet to be established [132].

3. A newly generated plan is immediately selected for execution. This is necessary since a plan's score can only be identified through simulation.

Still, this leads to a not yet clarified coincidence of choice set generation and choice itself. Again, an occasionally stabilized choice set would resolve this issue.

This is not to say that these aspects of MATSim are incompatible with the proposals of this dissertation. Rather, they require the more specialized treatment given later in Section 6.4.5.

MATSim's learning-based approach is a specific instance in a broad model range proposed in the field of activity based demand modeling, e.g., [27, 98, 99, 172], and the structural outline given in Section 3.2.1 is likely to apply to a greater variety of demand models. Still, the MATSim-related development of this work naturally suggests a presentation in terms of this system.

Concluding, the second part of this chapter formalizes a behavioral simulation system but leaves the behavioral model itself unspecified for the most part. This presentation is not given as an end in itself. The next chapter identifies what behavioral estimates are possible in this setting.

Chapter 4

Estimation

The previous two chapters describe a simulation system that consists of two components: a mobility simulation and a representation of human behavior. The specific properties of these components are now exploited in the formulation and solution of a traffic state estimation problem.

As outlined in the introduction, the task is to use spatially and temporally incomplete sensor information to reconstruct spatially and temporally complete system state information. Examples for sensors are loop detectors that measure flow rates at road cross-sections [91], ground- or airborne cameras that identify traffic densities on road segments [62, 77, 150], and floating cars that measure link velocities [156]. Only aggregate measurements are considered. While the importance of advanced traffic monitoring technologies such as vehicle re-identification systems is likely to increase in the future, they are not yet in broad application.

Macroscopically, the system states to be reconstructed are represented by state vector sequence

$$\mathcal{X} = \{\mathbf{x}(k)\}_k \tag{4.1}$$

of traffic flow model (3.7). This model unfolds deterministically given an initial state $\mathbf{x}(0) = \mathbf{x}_0$ and a driver population's behavior $\mathcal{U}_1 \ldots \mathcal{U}_N$. Since $\mathcal{U}_1 \ldots \mathcal{U}_N$ comprise all aspects of the individual drivers' *mental states* that are necessary to define all *macroscopic states* $\mathcal{X}$ in the model, the *state estimation problem* becomes to identify control sequences $\mathcal{U}_1 \ldots \mathcal{U}_N$ that steer $\mathcal{X}$ towards most likely values given the available measurements and the behavioral a priori knowledge.

The mapping from individual driver behavior on macroscopic system states is nonlinear. The proposed estimator deals with this difficulty by repeated linearizations of the macroscopic model. Since the model is dynamical, this requires to calculate system state sensitivities through simulated time. In result, the linearized effect of a single driver's decision in any time step k on the macroscopic states in any later time step $k + \Delta k$ can be predicted. Given a distance measure between true and simulated traffic conditions, these sensitivities then provide directional information for behavioral adjustments. Conceptually, this approach has a counterpart for example in meteorology, where the linearized version of a dynamical weather model is denoted as its "adjoint model". The

spatiotemporal sensitivities it provides are used to iteratively improve the full model's consistency with real world observations, e.g., for the purpose of short-term weather forecasting [63].

The remainder of this chapter is organized in four parts.

First, the problem of how to steer the behavior of simulated travelers by systematic manipulation of their utility perception is investigated in Section 4.1. Apart from being of practical interest itself, this section prepares a number of technical results that simplify the subsequent presentation. This includes the aforementioned linearization logic.

Second, a first heuristic estimator is proposed in Section 4.2. It applies the previously developed method to steer agents towards a plausible reproduction of available sensor data. However, this approach is not yet based on a solid statistical foundation.

Third, a Bayesian formulation of the estimation problem is given in Section 4.3. Starting with a conceptually straightforward but computationally cumbersome formulation, various simplifications are adopted that allow for a flexible balance between mathematical precision and computational efficiency.

Fourth, Section 4.4 illustrates the theoretical developments with a small example. A test case of realistic size is postponed to Chapter 5.

4.1 Steering Agent Behavior

The problem is investigated of how to influence the behavior of simulated travelers by changing their perception of systematic utility. The objective according to which agent behavior is to be influenced is represented by a once differentiable function

$$\Phi(\mathcal{X}) = \sum_{k=1}^{K} \varphi[\mathbf{x}(k), k] \tag{4.2}$$

that maps the macroscopic system states in simulation time steps 1 through K on a real number. An improved fulfillment of the objective is reflected by an increase of this function.

This problem statement is related to that of a *dynamic system optimal traffic assignment*. The latter seeks to identify a traffic pattern that minimizes the average cost experienced by all travelers. It is behaviorally not realistic since it implies that travelers cooperate in their efforts to minimize cost, but it is a good measure to estimate the greatest effectiveness of a traffic system or to identify optimal control strategies [35, 121].

Since the problem considered here is not to attain a strict system optimum but rather a compromise between individual driver objectives and global objective (4.2), and since only limited measures to affect agent behavior are available, the notion of a system optimal traffic assignment is avoided. The results obtained here only improve a microscopic assignment with respect to a global objective.

4.1.1 Modified Utility Perception

The agents' behavior is to be influenced by a modification of their systematic utility evaluation. Because of the decision protocol's probabilistic nature, cf. Section 3.2.1, there is no guarantee that a single choice based on such a modified utility does indeed improve the global objective. However, it is reasonable to assume that, once the effect of agent behavior on the global objective is identified, a utility modification that favors advantageous generalized paths also leads to choice distributions that improve the global objective on average. Unless otherwise noted, the notion of a path now represents an arbitrary behavioral pattern ranging from a single route to an all-day plan.

The problem of steering agent behavior is therefore posed as an ordinary assignment problem with modified systematic utility

$$W_n(\mathcal{U}) = V_n(\mathcal{U}) + \Phi(\mathcal{X}(\mathcal{U}_1 \ldots \mathcal{U}_{n-1}, \mathcal{U}, \mathcal{U}_{n+1} \ldots \mathcal{U}_N))/\mu \tag{4.3}$$

for every agent n and path $\mathcal{U} \in C_n$. That is, agent n evaluates Φ as a function of its individual path choice with the behavior of all other agents being fixed. The strictly positive parameter μ determines the weight of individual utility when compared to the global objective. Its choice is left to the analyst.

This problem statement is given yet independently of an estimation problem and requires no such interpretation. Since the subsequently developed method to steer simulated travelers holds promise for applications that go beyond traffic state estimation, its specific deployment for estimation purposes is postponed to Section 4.2.

A straightforward implementation of the above would require the following:

1. "Unsteered" population behavior $\mathcal{U}_1 \ldots \mathcal{U}_N$ is given.
2. For each agent $n = 1 \ldots N$, do:
   (a) Replace V_n by W_n according to (4.3).
   (b) Draw $\mathcal{U}'_n$ from C_n based on $W_n(\mathcal{U})$.
3. "Steered" population behavior is $\mathcal{U}'_1 \ldots \mathcal{U}'_N$.

The following subsections operationalize this procedure.

4.1.2 Linearization of Global Objective Function

Every evaluation of $W_n(\mathcal{U})$ requires an evaluation of $\Phi(\mathcal{X}(\ldots\mathcal{U}\ldots))$ and therefore a run of the entire mobility simulation. Since Φ is evaluated separately by all agents that make decisions based on their modified utility $W_n(\mathcal{U})$, a straightforward implementation of this approach is computationally intractable. This problem can be circumvented if the mapping from individual path choice $\mathcal{U}$ on Φ is linearized. Given $\mathcal{U}' = \mathcal{U} + \Delta\mathcal{U}$, this linearization essentially is $W(\mathcal{U}') \approx V(\mathcal{U}') + \Phi(\mathcal{X}(\ldots\mathcal{U}\ldots)) + \Delta\mathcal{U} \cdot d\Phi/d\mathcal{U}$. It will turn out that it is feasible to compute the sensitivities $d\Phi/d\mathcal{U}$ simultaneously for all agents. In

consequence, it is possible to linearly predict the effect of behavioral variations $\Delta\mathcal{U}$ on the global objective function Φ for all agents with just one run of the mobility simulation.

The linearization must account for the coupling between $\mathcal{U}$ and $\mathcal{X}$ through dynamical system constraint (3.7) that represents the mobility simulation. This difficulty can be dealt with by well-known methods from control theory [101, 138, 145]. A self-contained exposition is given in the following.

Denote

$$\Phi(k) = \sum_{\kappa=k}^{K} \varphi[\mathbf{x}(\kappa), \kappa] \tag{4.4}$$

for $k = 1 \dots K$. This is the remaining contribution to $\Phi(\mathcal{X})$ from time step k on. It can be recursively written as

$$\Phi(k) = \begin{cases} \varphi[\mathbf{x}(k), k] + \Phi(k+1) & k = 1 \dots K-1 \\ \varphi[\mathbf{x}(K), K] & k = K. \end{cases} \tag{4.5}$$

As a first step, sensitivities with respect to states are computed by

$$\frac{d\Phi(k)}{d\mathbf{x}(k)} = \begin{cases} \dfrac{\partial\varphi[\mathbf{x}(k), k]}{\partial\mathbf{x}(k)} + \dfrac{d\Phi(k+1)}{d\mathbf{x}(k)} & k = 1 \dots K-1 \\ \dfrac{\partial\varphi[\mathbf{x}(K), K]}{\partial\mathbf{x}(K)} & k = K. \end{cases} \tag{4.6}$$

Since the interplay between variables in different time steps is fully defined by state equation (3.7),

$$\frac{d\Phi(k+1)}{d\mathbf{x}(k)} = \left(\frac{\partial\mathbf{f}[\mathbf{x}(k), \mathbf{u}_1(k) \dots \mathbf{u}_N(k), k]}{\partial\mathbf{x}(k)}\right)^T \frac{d\Phi(k+1)}{d\mathbf{x}(k+1)} \tag{4.7}$$

holds for $k < K$, where $\mathbf{x}(k+1) = \mathbf{f}[\dots]$ is used.

Now, sensitivities with respect to control variables $\mathbf{u}_1(k) \dots \mathbf{u}_N(k)$ result from

$$\frac{d\Phi(\mathcal{X})}{d\mathbf{u}_n(k)} = \left(\frac{\partial\mathbf{f}[\mathbf{x}(k), \mathbf{u}_1(k) \dots \mathbf{u}_N(k), k]}{\partial\mathbf{u}(k)}\right)^T \frac{d\Phi(k+1)}{d\mathbf{x}(k+1)}. \tag{4.8}$$

Here, $\partial\varphi[\mathbf{x}(k), k]/\partial\mathbf{u}_n(k)$ disappears since $\mathbf{u}_n(k)$ influences no state earlier than $\mathbf{x}(k+1)$. $\partial\mathbf{f}[\dots]/\partial\mathbf{u}(k)$ denotes the partial derivative of $\mathbf{f}[\dots]$ with respect to any $\mathbf{u}_n(k)$, which is independent of n. This independence allows to entirely omit the n subscript in Φ's sensitivities and to subsequently write $d\Phi(\mathcal{X})/d\mathbf{u}(k)$ instead of $d\Phi(\mathcal{X})/d\mathbf{u}_n(k)$, and it allows to compute all sensitivities for all agents simultaneously.

In summary, $d\Phi(\mathcal{X})/d\mathbf{u}(k)$ is obtained in a two-pass-procedure:

1. Using (4.7), solve (4.6) recursively for $k = K \dots 1$. Moving backwards through time introduces a "far sightedness" into the calculations that is necessary to predict the influence of present state variations on future system states.

2. Determine the influence of control variables by (4.8) for $k = 0 \ldots K-1$. Since this expression is identical for all agents, it needs to be evaluated only once for the entire population.

One obtains the following linearization of $\Phi(\mathcal{X})$ with respect to $\mathcal{U}_1 \ldots \mathcal{U}_N$:

$$\Phi(\mathcal{X}(\mathcal{U}_1 \ldots \mathcal{U}_N)) \approx \Phi(\mathcal{X}^0) + \sum_{k=0}^{K-1} \left(\frac{d\Phi(\mathcal{X}^0)}{d\mathbf{u}(k)} \right)^T \sum_{n=1}^{N} (\mathbf{u}_n(k) - \mathbf{u}_n^0(k)) \tag{4.9}$$

where $\mathbf{u}_n^0(k)$ is the control vector of traveler n in time step k around which the linearization takes place and $\mathcal{X}^0$ is the resulting macroscopic state sequence. Defining the sensitivity sequence

$$\Lambda = \left\{ \frac{d\Phi(\mathcal{X}^0)}{d\mathbf{u}(k)} \right\}_k \tag{4.10}$$

and the "inner product"

$$\langle \Lambda, \mathcal{U} \rangle = \sum_k \left(\frac{d\Phi(\mathcal{X}^0)}{d\mathbf{u}(k)} \right)^T \mathbf{u}(k), \tag{4.11}$$

(4.9) can be rewritten as

$$\Phi(\mathcal{X}(\mathcal{U}_1 \ldots \mathcal{U}_N)) \approx \sum_{n=1}^{N} \langle \Lambda, \mathcal{U}_n \rangle + \text{const} \tag{4.12}$$

where the constant addend contains all terms independent of $\mathcal{U}_1 \ldots \mathcal{U}_N$. The elements of Λ are sensitivities of the global objective function with respect to individual turning moves, and as such they serve as coefficients that are multiplied with the turning move indicators contained in the populations' path set $\mathcal{U}_1 \ldots \mathcal{U}_N$.

Macroscopic traffic dynamics are linear in good approximation with respect to a single agent's behavior since individual control variables $u_{ij,n}(k) \in \{0, 1\}$ are small compared to actual turning counts in a congested network. Thus, for a single agent, a linearization yields a reasonable approximation to the nonlinear problem, and

$$\begin{aligned} W_n(\mathcal{U}) &= V_n(\mathcal{U}) + \Phi(\mathcal{X}(\mathcal{U}_1 \ldots \mathcal{U}_{n-1}, \mathcal{U}, \mathcal{U}_{n+1} \ldots \mathcal{U}_N))/\mu \\ &\approx V_n(\mathcal{U}) + \langle \Lambda, \mathcal{U} \rangle / \mu + \text{const} \end{aligned} \tag{4.13}$$

holds with good precision. The constant addend is identical for all alternatives available to an agent. Since it is reasonable to assume that the preferences of a decision maker are not influenced by a constant shift in the utilities of all alternatives,[1]

$$W_n(\mathcal{U}) = V_n(\mathcal{U}) + \langle \Lambda, \mathcal{U} \rangle / \mu \tag{4.14}$$

defines as from now the **modified utility** of agent n's option $\mathcal{U} \in C_n$. Using the same Λ for all agents reflects the fact that the sensitivity of Φ to a turning

[1] This is always true for RUMs, cf. (3.15).

move (sequence) is independent of which agent is actually moving. Here, the elements of Λ constitute (up to a scaling coefficient μ) utility corrections for every single turning move in the network, and the modified utility of a specific path is identified by adding up these corrections along that path. This can be seen most clearly if $\langle \Lambda, \mathcal{U} \rangle$ is fully expanded:

$$\langle \Lambda, \mathcal{U} \rangle = \sum_k \sum_{ij} \frac{d\Phi(\mathcal{X}^0)}{du_{ij}(k)} u_{ij}(k). \tag{4.15}$$

Only such components of Λ are summed up in $\langle \Lambda, \mathcal{U} \rangle$ that correspond to turning moves that are actually represented by path $\mathcal{U}$ through non-zero turning move indicators. In light of this, Λ is denoted either as a sequence of sensitivities or of utility corrections, depending on the context.

The above linearization procedure is considerably accelerated if the underlying mobility simulation runs on variable time scales as proposed in Section 2.5. Since the mobility simulation's sensitivities vary on the same temporal grid as its macroscopic states, the overall number of sensitivity evaluations is reduced in the same order as the number of flow transmissions during a simulation.

The importance of this computationally still expensive linearization becomes clear in comparison with a simplistic approximation. Assume that the macroscopic system state $\mathcal{X}$ is composed of vehicle occupancies on all road segments in all time steps. Then, the effect of a vehicle's path choice $\mathcal{U}$ might appear predictable by simply increasing the occupancy of every link in $\mathcal{U}$ for the duration of this link's traversal time. In a way, this does predict the effect of $\mathcal{U}$ on $\mathcal{X}$ and thus on Φ without any linearization. Still, it does not capture the global effect of driver behavior in congested conditions. A vehicle that tries to enter a congested link is slowed down, and in turn it slows down all vehicles behind it. That is, it also affects upstream links that are not contained in its path. A full linearization of traffic flow dynamics accounts for these interdependencies and thus is superior in all but trivially uncongested traffic conditions.

4.1.3 Consistent Linearization for Many Agents

The linearization of Φ relies on the relatively small influence of a single traveler on the global traffic situation. This argument does not hold if an entire population is considered since any utility correction Λ that is obtained by a linearization around a certain state trajectory $\mathcal{X}^0$ may result in a population reaction $\mathcal{U}_1 \ldots \mathcal{U}_N$ that causes a significantly different network state trajectory $\mathcal{X}$ and thus invalidates the underlying linearization.

For a non-stochastic planning or telematics simulation, a utility correction Λ is consistent if the population behavior given this Λ generates network states $\mathcal{X}$ such that a repeated linearization of Φ reproduces the original Λ values, cf. Figure 4.1. Formally, a fixed point of the combined map "sim(ulation), followed by lin(earization)" is required: $\Lambda = \mathrm{lin} \circ \mathrm{sim}(\Lambda)$.

Since there are stochastic elements in the simulation, its outcome $\mathcal{X}$ given a specific Λ is stochastic as well, and the reproducibility of Λ calls for a likewise stochastic interpretation. One may assume that only a randomly distorted map

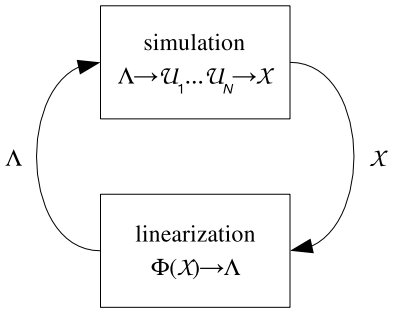


Figure 4.1: Fixed point of utility corrections
Consistent utility corrections Λ are attained if a linearization of Φ around simulation outcome $\mathcal{X}$ results in the same Λ corrections that have previously been applied in the simulation.

$\text{lin} \circ \text{sim}(\Lambda) + \mathcal{E}$ can be evaluated where $\mathcal{E}$ is a zero mean disturbance of the same dimension as Λ. Since no algorithm is known that definitely converges to a deterministic Λ fixed point in suchlike noisy conditions for the whole range of possibly implemented simulation mappings, and since not even the existence of such a fixed point is ascertained, a pragmatic course of action is taken: The existence of a fixed point is merely assumed, and an elementary stochastic approximation (SA) method is employed for its identification [26].[2] This particular method is chosen here because of its simplicity and clarity. Possible algorithmical improvements are indicated in Section 6.4.1.3.

The proposed SA approach is outlined in Algorithm 2. It assumes an iterative simulation logic, which is equally applicable to a SUE-based planning model and to a telematics model of spontaneous and imperfectly informed drivers. The conceptual difference is that a SUE decision protocol typically utilizes all information from the most recent network loading, whereas a telematics decision protocol generates every elementary decision within a plan only based on that subset of this information that could have actually been gathered up to the considered point in simulated time [26]. A full implementation of this algorithm is experimentally investigated in the next chapter.

4.1.4 Behavioral Justification

Since the modified utility deviates from the originally modeled agent perception, any behavior that is based on the modified utility is not reasonable in itself. A path $\mathcal{U}$ that is chosen by traveler n based on a modified utility function W_n only is consistent with the behavioral model if n's utility perception is indeed represented by W_n instead of the original V_n. Thus, the method's applicability depends on the possibility to reinterpret utility perception itself. Three fields where this is possible are identified below:

- The method is developed with behavioral traffic state estimation in mind and is applicable for this purpose. Given a specification of Φ that reflects

[2] A self-contained convergence proof for the SA method can be found in [69]. However, its requirements cannot be established in the setting considered here.

Algorithm 2 Steering a population of agents

1. Initialization.
   (a) Set iteration counter $m = 0$.
   (b) Fill $\bar{\Lambda}^{(m)}$ (estimate of Λ fixed point) with all zeros.
2. Simulation.
   (a) For all $n = 1 \ldots N$, do: Use $W_n(\mathcal{U}) = V_n(\mathcal{U}) + \langle \bar{\Lambda}^{(m)}, \mathcal{U} \rangle / \mu$ instead of $V_n(\mathcal{U})$ in the decision protocol when drawing $\mathcal{U}_n^{(m)}$.
   (b) Load $\mathcal{U}_1^{(m)} \ldots \mathcal{U}_N^{(m)}$ on the network and obtain $\mathcal{X}^{(m)}$.
3. Linearize $\Phi(\mathcal{X}^{(m)})$ and obtain $\Lambda^{(m)}$.
4. Update $\bar{\Lambda}^{(m+1)} = \dfrac{m}{m+1} \bar{\Lambda}^{(m)} + \dfrac{1}{m+1} \Lambda^{(m)}$.
5. If another iteration is desired:
   (a) Increase m by one.
   (b) Goto step 2.

the quality of measurement reproduction, the resulting W_n is interpreted as an estimate of individual n's most likely utility perception given these measurements. Here, the original V_n constitutes a model-based a priori assumption that is corrected by the estimation procedure such that Φ is improved. The belief in the behavioral prior information is reflected by weight parameter μ. A discussion of possible ambiguities in this interpretation is given in Section 4.4.3.

- Φ may also represent a general utility of system operations. Applying the above procedure, the resulting Λ coefficients define a toll on all turning moves in the network. An agent n which chooses its path based on the resulting W_n strives to maximize a weighted combination of individual and system utility. Clearly, a physically implementable toll must meet a number of additional constraints that are beyond the scope of this thesis.
- An iterative planning simulation requires large amounts of computation time. If a specification of Φ was found that (i) reflects the degree of such a simulation's convergence and (ii) has a vanishing influence upon convergence, it may help to reduce the number of required iterations until an equilibrium is reached. Here, utility perception is modified only during the transient phase of an iterative algorithm but not in its outcome. Still, this application is of rather hypothetical nature since no such version of Φ is proposed in this dissertation.

In all cases, W_n constitutes a modified utility perception of driver n that is in one way or the other consistent with the original assumption of utility-driven be-

havior, and this modification is generated such that a problem-specific instance of Φ is improved.

4.2 Heuristic Estimation

A similarity measure between simulated and observed sensor data is chosen as the global objective function Φ, and the agents are steered towards an increase of this function.

4.2.1 Modeling of Aggregate Traffic Measurements

A likelihood function suggests itself to quantify a model's measurement fit. In this subsection, the likelihood of aggregate traffic measurements is formally related to individual agent behavior.

Macroscopic state space model (3.7) is supplemented with an output equation

$$\mathbf{y}(k) = \mathbf{g}[\mathbf{x}(k), \boldsymbol{\epsilon}(k)] \tag{4.16}$$

that maps system state $\mathbf{x}(k)$ by a once differentiable function $\mathbf{g}$ on output vector $\mathbf{y}(k)$ of macroscopic observables. The latter may include flows, velocities, and densities generated by sensors such as inductive loops, floating cars, and traffic surveillance cameras. The influence of various sources of error on these observations is accounted for by random disturbance vector $\boldsymbol{\epsilon}(k)$ that turns $\mathbf{y}(k)$ into a random variable itself. Equation (4.16) defines $\mathbf{y}(k)$'s probability density function (p.d.f.)

$$p(\mathbf{y}(k)|\mathbf{x}(k)) = \int \delta(\mathbf{y}(k) - \mathbf{g}[\mathbf{x}(k), \boldsymbol{\epsilon}])p(\boldsymbol{\epsilon})d\boldsymbol{\epsilon} \tag{4.17}$$

where δ is the Dirac function and $p(\boldsymbol{\epsilon})$ is the known p.d.f. of $\boldsymbol{\epsilon}$. A lower-case p generally denotes a p.d.f., whereas an upper-case P represents a discrete probability. Subsuming the above expression in terms of trajectories $\mathcal{Y} = \{\mathbf{y}(k)\}_k$ and $\mathcal{X} = \{\mathbf{x}(k)\}_k$ yields

$$p(\mathcal{Y}|\mathcal{X}) = \prod_k p(\mathbf{y}(k)|\mathbf{x}(k)) \tag{4.18}$$

where stochastic independence between outputs at different time steps is assumed. This is, so far, the not unexpected result that all spatiotemporal measurements can be probabilistically described if all spatiotemporal system states $\mathcal{X}$ are known – no behavioral information is needed directly.

Nevertheless, the states $\mathcal{X}$ are indirectly caused by the population behavior $\mathcal{U}_1 \ldots \mathcal{U}_N$. This allows to define the behavioral likelihood $l(\mathcal{U}_1 \ldots \mathcal{U}_N|\mathcal{Y})$ given the measurements $\mathcal{Y}$ as a function of $\mathcal{U}_1 \ldots \mathcal{U}_N$:

$$l(\mathcal{U}_1 \ldots \mathcal{U}_N|\mathcal{Y}) = p(\mathcal{Y}|\mathcal{X}(\mathcal{U}_1 \ldots \mathcal{U}_N)). \tag{4.19}$$

This function is linearizable with respect to $\mathcal{U}_1 \ldots \mathcal{U}_N$ if the p.d.f. of $\mathcal{Y}$ given $\mathcal{X}$ is differentiable with respect to $\mathcal{X}$. Frequently, the (likewise linearizable) log-likelihood function

$$L(\mathcal{U}_1 \ldots \mathcal{U}_N|\mathcal{Y}) = \ln l(\mathcal{U}_1 \ldots \mathcal{U}_N|\mathcal{Y}) \tag{4.20}$$

is also referred to.

Others than link-related measurements are possible. Since the state vector of model (3.7) contains smoothed turning counts, observations of these can be directly incorporated in the output equation. The additional value of such measurements is pointed out in the literature review of Section 1.2.1.

4.2.2 Steering Agents Towards the Measurements

Maximum likelihood estimation is the arguably most popular approach to statistical parameter identification, e.g., [140]. It is an established method for the identification of OD matrices from traffic counts [162], and its application for agent-based behavioral estimation is complicated in the same way as traditional OD matrix estimation: The available number of link-related measurements is usually much smaller than the number of parameters to be identified – the problem is extremely under-determined.

Typically, a prior OD matrix is integrated in the likelihood function as a supplementary measurement that resolves this under-determinedness. Since no such prior is available here, a different and statistically less rigorous approach is pursued. Algorithm 2 is employed, with its general objective function defined as the measurement log-likelihood, i.e.,

$$\Phi(\mathcal{X}(\mathcal{U}_1 \ldots \mathcal{U}_N)) = L(\mathcal{U}_1 \ldots \mathcal{U}_N|\mathcal{Y}). \tag{4.21}$$

The resulting overall objective function (4.3) of any agent n is the weighted sum $V_n(\mathcal{U}) + \Phi(\mathcal{X}(...\mathcal{U}...))/\mu$ of its individual utility function and the log-likelihood. The weighting parameter μ determines the importance of the behavioral prior information represented by the original utility perception. If μ is chosen very large, the likelihood term vanishes and the agent acts in a way that is fully prespecified by its original utility function. The smaller μ gets the more weight is put on the likelihood and the more the agent adjusts its behavior towards an increase of the likelihood. While μ is used here as a mere weighting parameter, the Bayesian problem reformulation given in the next section enables its interpretation as a behavioral model parameter.

Specifically, if mutually independent normal measurement distributions are assumed, (4.21) yields a global objective function

$$\Phi(\mathcal{X}) = -\sum_a \sum_k \frac{(y_a(k) - g_a[\mathbf{x}(k)])^2}{2\sigma_a^2} \tag{4.22}$$

where $y_a(k)$ is the sensor information available for link a in time step k, $g_a[\mathbf{x}(k)]$ is its simulated expectation, and σ_a^2 is its variance.[3] This is the arguably sim-

[3] The log-likelihood of mutually independent measurements $y_a(k)$ is $L(\mathcal{U}_1 \ldots \mathcal{U}_N|\mathcal{Y}) = \sum_{ak} \ln p(y_a(k)|\mathbf{x}(k))$. Assuming $y_a(k) = g_a[\mathbf{x}(k)] + \varepsilon_a(k)$, a normally distributed $\varepsilon_a(k)$ with zero expectation and variance σ_a^2 implies $p(y_a(k)|\mathbf{x}(k)) \propto \exp[-(y_a(k) - g_a[\mathbf{x}(k)])^2/2\sigma_a^2]$. Consequently, $L(\mathcal{U}_1 \ldots \mathcal{U}_N|\mathcal{Y}) = -\sum_{ak}(y_a(k) - g_a[\mathbf{x}(k)])^2/2\sigma_a^2 + \text{const}$.

plest approach to the behavioral estimation problem: Define a quadratic distance measure between observed and simulated traffic characteristics, choose a "reasonable" weight parameter μ, and let the general method to steer agent behavior push the simulation towards a reduction of this error function.

The particular assumption of independent normal measurements yields an objective function (4.22) of greatest simplicity. Still, different distributional assumptions are feasible. In particular, correlated measurements with a known covariance structure can be accounted for in terms of a multivariate (normal) distribution.

Providing a modified utility that comprises a weighted sum of individual utility perception and measurement log-likelihood to the decision protocol does not result in an overall maximum likelihood estimator for two reasons: (i) The individual utility addend permits no interpretation as a log-likelihood component, and (ii) the decision protocol draws a choice instead of deterministically maximizing the modified utility. For these reasons, a more systematic derivation of a statistical estimator is given in the following.

4.3 Bayesian Estimation

Section 4.1 prepares a general tool to steer simulated travelers. This tool facilitates the proposal of a first heuristic estimator in Section 4.2. Here, the estimation problem is reconsidered in a statistically more rigorous setting. The presentation starts with a conceptually straightforward but computationally cumbersome formulation. Several simplifications are then adopted that significantly increase the computational feasibility and result in the proposal of two operational estimators. Ultimately, the heuristic estimator is rediscovered, this time, however, with a better understanding of its properties and limitations.

It has been stated before that aggregate measurements $\mathcal{Y}$ alone do not provide sufficient information for a unique estimate of population behavior $\mathcal{U}_1 \dots \mathcal{U}_N$ since usually there are many behavioral combinations that generate the same observations. Here, this problem is resolved by the incorporation of additional behavioral information in a Bayesian setting. In order to build on a solid foundation, the Bayesian estimator is designed from scratch. While some previously developed results such as the linearization of a log-likelihood function in dynamical conditions are reused in this section, no constitutional dependency on the heuristic estimator itself is allowed for.

4.3.1 General Formulation of Estimator

An arbitrary implementation of the decision protocol is assumed. It draws choices $\mathcal{U} \in C_n$ according to an individual choice distribution $P_n(\mathcal{U})$ for every agent $n = 1 \dots N$. Only realizations of this distribution can be observed, cf. Section 3.2.1.4. $\mathcal{U}$ may still represent any of the behavioral dimensions described in Section 3.2.1.1, ranging from a single route to an all-day plan. Given mutually independent traveler decisions, the **behavioral prior** for the whole population

is defined as

$$P(\mathcal{U}_1 \ldots \mathcal{U}_N) = \prod_{n=1}^{N} P_n(\mathcal{U}_n). \tag{4.23}$$

The assumption of mutually independent choices is to be understood in the context of the iterative simulation logic outlined in Section 4.1.3 in that (4.23) describes the population's plan choice distribution in a particular iteration of the simulator given the network conditions only from the previous iteration(s).

The available measurements $\mathcal{Y}$ parameterize a likelihood $l(\mathcal{U}_1 \ldots \mathcal{U}_N|\mathcal{Y})$ of the population's path choice as specified in (4.19). Bayes' theorem allows to combine these two sources of information into a **behavioral posterior**

$$P(\mathcal{U}_1 \ldots \mathcal{U}_N|\mathcal{Y}) = \frac{l(\mathcal{U}_1 \ldots \mathcal{U}_N|\mathcal{Y})P(\mathcal{U}_1 \ldots \mathcal{U}_N)}{\sum_{\mathcal{V}_1 \in C_1} \cdots \sum_{\mathcal{V}_N \in C_N} l(\mathcal{V}_1 \ldots \mathcal{V}_N|\mathcal{Y})P(\mathcal{V}_1 \ldots \mathcal{V}_N)}, \tag{4.24}$$

where the denominator results from

$$p(\mathcal{Y}) = \sum_{\mathcal{V}_1 \in C_1} \cdots \sum_{\mathcal{V}_N \in C_N} p(\mathcal{Y}|\mathcal{V}_1 \ldots \mathcal{V}_N)P(\mathcal{V}_1 \ldots \mathcal{V}_N). \tag{4.25}$$

The estimation objective is to have the population choose its behavior according to the posterior (4.24) instead of the prior (4.23). This can be enforced if draws are taken from the prior but are rejected with a certain probability that depends on the measurements. Denote by $\phi(\mathcal{U}_1 \ldots \mathcal{U}_N)$ the probability to accept a draw $\mathcal{U}_1 \ldots \mathcal{U}_N$ from the prior. If this probability is specified by

$$\begin{aligned} \phi(\mathcal{U}_1 \ldots \mathcal{U}_N) &= l(\mathcal{U}_1 \ldots \mathcal{U}_N|\mathcal{Y})/D \\ D &\geq \max_{\mathcal{V}_1 \in C_1 \ldots \mathcal{V}_N \in C_N} l(\mathcal{V}_1 \ldots \mathcal{V}_N|\mathcal{Y}), \end{aligned} \tag{4.26}$$

then the following accept/reject procedure draws from the posterior:

1. Draw candidate choices $\mathcal{U}_1 \ldots \mathcal{U}_N$ from the prior (4.23).
2. With probability $1 - \phi(\mathcal{U}_1 \ldots \mathcal{U}_N)$, discard the candidates and goto 1.
3. The first accepted $\mathcal{U}_1 \ldots \mathcal{U}_N$ constitute a draw from the posterior (4.24).

The correctness of this simple algorithm is shown by straightforward manipulations. Noting that the overall probability of a rejection is

$$\phi_{\text{reject}} = 1 - \sum_{\mathcal{V}_1 \in C_1} \cdots \sum_{\mathcal{V}_N \in C_N} \phi(\mathcal{V}_1 \ldots \mathcal{V}_N)P(\mathcal{V}_1 \ldots \mathcal{V}_N), \tag{4.27}$$

the probability that $\mathcal{U}_1 \ldots \mathcal{U}_N$ is the first accepted draw is

$$\begin{aligned} & \sum_{d=0}^{\infty} \phi_{\text{reject}}^d \phi(\mathcal{U}_1 \ldots \mathcal{U}_N)P(\mathcal{U}_1 \ldots \mathcal{U}_N) \\ = \; & \frac{\phi(\mathcal{U}_1 \ldots \mathcal{U}_N)P(\mathcal{U}_1 \ldots \mathcal{U}_N)}{1 - \phi_{\text{reject}}} \\ = \; & \frac{\phi(\mathcal{U}_1 \ldots \mathcal{U}_N)P(\mathcal{U}_1 \ldots \mathcal{U}_N)}{\sum_{\mathcal{V}_1 \in C_1} \cdots \sum_{\mathcal{V}_N \in C_N} \phi(\mathcal{V}_1 \ldots \mathcal{V}_N)P(\mathcal{V}_1 \ldots \mathcal{V}_N)} \\ = \; & P(\mathcal{U}_1 \ldots \mathcal{U}_N|\mathcal{Y}). \end{aligned} \tag{4.28}$$

The behavioral posterior can thus be generated by suppressing certain draws from the prior. Somewhat coarsely expressed: (i) The simulation is run many times with different random seeds, (ii) a large portion of these runs is "thrown away", based on the above rejection criterion, and (iii) the remaining runs are draws from an accurate Bayesian combination of the behavioral prior and the measurements.

Although appealing because of its simplicity, this approach is in this form computationally intractable in all but trivial cases. There are two major problems:

1. It is computationally infeasible to evaluate all possible $l(\mathcal{U}_1 \ldots \mathcal{U}_N|\mathcal{Y})$ values beforehand since every such evaluation requires a full network loading in order to map $\mathcal{U}_1 \ldots \mathcal{U}_N$ on a macroscopic state sequence $\mathcal{X}$ that enters the likelihood via (4.19). However, these evaluations are required in order to guarantee a feasible denominator for the acceptance probabilities (4.26). Furthermore, the need for a choice set enumeration implies that the estimation logic is aware of this set, which constitutes an unwanted dependency of the estimator on modeling details.

2. Even if the acceptance probabilities' denominator is replaced by an estimate in order to mitigate problem 1, a single draw from the posterior might still require a substantial number of mobility simulation runs since every draw from the prior needs to be loaded on the network at least once and since it cannot be guaranteed that an "accept" occurs after a fixed number of draws from the prior.

In light of these difficulties, simplifying assumptions that speed up the simulation of the posterior are highly desirable even at the cost of some loss in accuracy. Two suchlike simplified estimators are proposed in the following two sections.

4.3.2 Operational Accept/Reject Estimator

The Bayesian estimator is considerably simplified if the full likelihood is replaced by an approximation. In Section 4.1.2, a general function Φ of the macroscopic system states is linearized with respect to the population's path choice. Proceeding in this respect similarly to the heuristic estimator of Section 4.2.2, this result is now utilized to linearize the measurement log-likelihood. Let

$$\Phi(\mathcal{X}(\mathcal{U}_1 \ldots \mathcal{U}_N)) = L(\mathcal{U}_1 \ldots \mathcal{U}_N|\mathcal{Y}). \tag{4.29}$$

A linearization of Φ yields the approximation

$$L(\mathcal{U}_1 \ldots \mathcal{U}_N|\mathcal{Y}) \approx \sum_{n=1}^{N} \langle \Lambda, \mathcal{U}_n \rangle + \text{const} \tag{4.30}$$

with the Λ coefficients defined in (4.10) through (4.12). The resulting likelihood approximation is

$$l(\mathcal{U}_1 \ldots \mathcal{U}_N|\mathcal{Y}) \approx \text{const} \cdot \prod_{n=1}^{N} e^{\langle \Lambda, \mathcal{U}_n \rangle}. \tag{4.31}$$

A substitution of this and the behavioral prior (4.23) in the behavioral posterior (4.24) yields

$$P(\mathcal{U}_1 \dots \mathcal{U}_N|\mathcal{Y}) \approx \frac{\prod_{n=1}^{N} e^{\langle \Lambda, \mathcal{U}_n \rangle} P_n(\mathcal{U}_n)}{\sum_{\mathcal{V}_1 \in C_1} \cdots \sum_{\mathcal{V}_N \in C_N} \prod_{n=1}^{N} e^{\langle \Lambda, \mathcal{V}_n \rangle} P_n(\mathcal{V}_n)}. \quad (4.32)$$

The denominator of this expression requires some attention. It is a sum over *all* possible combinations of behavioral patterns $\mathcal{V}_1 \dots \mathcal{V}_N$ in the population, whereas the $e^{\langle \cdots \rangle}$ terms result from a linearization around a *particular* macroscopic state sequence. The feasibility of this approximation results from the observation that, even if individuals exhibit variable behavior, the resulting macroscopic traffic patterns are relatively concentrated in state space. All deterministic traffic assignment efforts rely on this assumption. Thus, the majority of behavioral draws results in traffic patterns over which a linearization can be justified. Behavioral patterns $\mathcal{V}_1 \dots \mathcal{V}_N$ that generate physical states far away from this domain are assumed to have such low probabilities $\prod_{n=1}^{N} P_n(\mathcal{V}_n)$ that the according addends in the denominator can be neglected.

Applying the distributive law to (4.32), one obtains

$$\begin{aligned} P(\mathcal{U}_1 \dots \mathcal{U}_N|\mathcal{Y}) &\approx \frac{\prod_{n=1}^{N} e^{\langle \Lambda, \mathcal{U}_n \rangle} P_n(\mathcal{U}_n)}{\prod_{n=1}^{N} \sum_{\mathcal{V}_n \in C_n} e^{\langle \Lambda, \mathcal{V}_n \rangle} P_n(\mathcal{V}_n)} \\ &= \prod_{n=1}^{N} \frac{e^{\langle \Lambda, \mathcal{U}_n \rangle} P_n(\mathcal{U}_n)}{\sum_{\mathcal{V}_n \in C_n} e^{\langle \Lambda, \mathcal{V}_n \rangle} P_n(\mathcal{V}_n)}. \end{aligned} \quad (4.33)$$

The linearization is beneficial in two ways. First, the population's joint posterior (4.33) is decomposed into a product of individual posteriors that can be evaluated agent by agent. These individual posteriors are subsequently denoted by

$$P_n(\mathcal{U}|\mathcal{Y}) = \frac{e^{\langle \Lambda, \mathcal{U} \rangle} P_n(\mathcal{U})}{\sum_{\mathcal{V} \in C_n} e^{\langle \Lambda, \mathcal{V} \rangle} P_n(\mathcal{V})}. \quad (4.34)$$

Second, only a single run of the mobility simulation (plus one calculation of the Λ coefficients) is needed to parameterize these posteriors for all agents in the population.

The accept/reject procedure can now be applied to every agent individually. The acceptance probability for path $\mathcal{U}$ from agent n's choice set is defined as

$$\begin{aligned} \phi_n(\mathcal{U}) &= e^{\langle \Lambda, \mathcal{U} \rangle} / D_n \\ D_n &\geq \max_{\mathcal{V} \in C_n} e^{\langle \Lambda, \mathcal{V} \rangle}, \end{aligned} \quad (4.35)$$

but otherwise the method remains unchanged. The only simplifying assumption made here is that the log-likelihood can be linearized with sufficient precision. Since this linearization is likely to be different given either the behavioral prior or the posterior, an iterative approach similar to the fixed point search of Algorithm 2 is appropriate: Starting from the behavioral prior, successively improved linearizations are generated from iteration to iteration until a stable state is reached where the estimator draws from the behavioral posterior based

Algorithm 3 Accept/reject estimator

1. Initialization.
   (a) Set iteration counter $m = 0$.
   (b) Fill $\bar{\Lambda}^{(m)}$ (estimate of Λ fixed point) with all zeros.
2. Simulation.
   (a) For all $n = 1 \dots N$, do:
      i. Draw candidate choice $\mathcal{U}_n^{(m)}$ from n's behavioral prior.
      ii. With probability $1 - \phi_n(\mathcal{U}_n^{(m)})$ (where $\bar{\Lambda}^{(m)}$ is substituted for Λ in (4.35)), discard the candidate and goto 2(a)i.
      iii. Retain the first accepted choice $\mathcal{U}_n^{(m)}$.
   (b) Load $\mathcal{U}_1^{(m)} \dots \mathcal{U}_N^{(m)}$ on the network and obtain $\mathcal{X}^{(m)}$.
3. Linearize $\Phi(\mathcal{X}^{(m)})$ and obtain $\Lambda^{(m)}$.
4. Update $\bar{\Lambda}^{(m+1)} = \frac{m}{m+1}\bar{\Lambda}^{(m)} + \frac{1}{m+1}\Lambda^{(m)}$.
5. If another iteration is desired:
   (a) Increase m by one.
   (b) Goto step 2.

on a linearization that in turn is most appropriate given this very posterior. This approach is subsequently denoted as the **accept/reject (AR) estimator**. It is summarized in Algorithm 3. Again, only a basic SA fixed point search procedure is deployed for greatest clarity.

The type of behavior to be estimated and the prior implemented by the decision protocol are arbitrary. Since a choice set enumeration is only required to provide a lower bound for the acceptance probabilities' denominator defined in (4.35), it can be avoided if this denominator is treated as a tuning parameter: Choosing a large value is likely to comply with the (unknown) lower bound but also to result in low acceptance probabilities and increased computational cost. Vice versa, a smaller denominator yields faster but also increasingly imprecise estimates. The loss in precision can be appraised by observing the frequency at which infeasible probabilities greater one occur in (4.35) that need to be truncated. This provides a practically attractive balancing mechanism between estimation precision and computational efficiency, which does not rely on a choice set enumeration.

Computational difficulties remain if a behavioral draw is expensive, e.g., because it involves some kind of optimization procedure, such as a (randomized) best path calculation. One alternative would be not to discard unwanted draws but to duplicate desired ones and to use these in a number of repeated choice situations. However, since this would introduce possibly unwanted serial correlations, it is at odds with the intention to develop a transparent estimation

layer. A computationally more efficient yet not as broadly applicable estimator is presented next.

4.3.3 Operational Utility-Modification Estimator

The behavioral posterior (4.34) for a single agent constitutes the starting point of this development. It is restated here for ease of reference:

$$P_n(\mathcal{U}|\mathcal{Y}) = \frac{e^{\langle\Lambda,\mathcal{U}\rangle} P_n(\mathcal{U})}{\sum_{\mathcal{V}\in C_n} e^{\langle\Lambda,\mathcal{V}\rangle} P_n(\mathcal{V})}. \tag{4.36}$$

The PS-logit model prepared in Section 3.2.2.2 is now used as a distributional assumption about the prior choice probabilities, i.e.,

$$P_n(\mathcal{U}) = \frac{\mathrm{PS}_n(\mathcal{U}) e^{\mu V_n(\mathcal{U})}}{\sum_{\mathcal{V}\in C_n} \mathrm{PS}_n(\mathcal{V}) e^{\mu V_n(\mathcal{V})}}. \tag{4.37}$$

Recall that the PS coefficients account for path overlap in a route choice context. If they are omitted, a plain MNL model results. A substitution of (4.37) in (4.36) yields

$$P_n(\mathcal{U}|\mathcal{Y}) = \frac{\mathrm{PS}_n(\mathcal{U}) e^{\mu(V_n(\mathcal{U})+\langle\Lambda,\mathcal{U}\rangle/\mu)}}{\sum_{\mathcal{V}\in C_n} \mathrm{PS}_n(\mathcal{V}) e^{\mu(V_n(\mathcal{V})+\langle\Lambda,\mathcal{V}\rangle/\mu)}}. \tag{4.38}$$

This posterior is structurally identical to its prior. Only the addition of $\langle\Lambda,\mathcal{U}\rangle/\mu$ to $V_n(\mathcal{U})$ is different. This allows to force a decision protocol that implements a PS-logit prior to immediately draw from the posterior only by adding a correction term $\langle\Lambda,\mathcal{U}\rangle/\mu$ to every alternative $\mathcal{U}$'s systematic utility. The PS coefficients need not be known to the estimator for the generation of these corrections. Consequently, this approach is feasible for all priors that exhibit the functional form of the PS-logit model, even if the PS coefficients result from a different specification than given in (3.17). Such priors are said to be "of PS-logit structure". Note that this includes the plain MNL model.

This approach is subsequently denoted as the **utility-modification (UM) estimator**. Its requirements are more restrictive than those of the AR estimator since a decision protocol of PS-logit structure needs to be available. However, if such a behavioral prior is given, the UM estimator and the AR estimator yield equivalent results since both rely on the same linearization-based approximation (4.36) of the posterior. In this case, the UM estimator is to be preferred over the AR estimator since it is computationally more efficient in that it rejects no draws from the prior but immediately draws from the posterior.

Section 4.2's estimation heuristic coincides structurally with the UM estimator: In either case, the modified utility is defined by (4.14), and the Λ coefficients are identically generated by a linearization of the measurement log-likelihood function. The heuristic's weight coefficient μ coincides with the scale parameter of the PS-logit prior. For completeness, the UM estimator is specified in Algorithm 4.

Algorithm 4 Utility-modification estimator

1. Apply Algorithm 2 with the global utility function Φ defined by (4.21) as the measurement log-likelihood function.
2. This estimator has the following properties.
   (a) It is identical to the heuristic estimator of Section 4.2.
   (b) If the behavioral prior is of PS-logit structure, this estimator is equivalent to the AR estimator specified in Algorithm 3.

4.3.4 Applicability of Heuristic Estimator

Technically, the UM estimator can be applied in conjunction with an arbitrary utility-driven behavioral prior for the estimation of anything from routes to all-day plans. In such a general setting, it coincides with the heuristic estimator of Section 4.2. This analysis identifies the *conceptual* limitations of such an approach and thus clarifies the applicability of the heuristic estimator itself.

Assume that decision maker n disposes of a choice set C_n and that prespecified utilities $V_n^0(\mathcal{U})$ for every $\mathcal{U} \in C_n$ are given. Based on these utilities, the decision protocol draws from well-defined but to the estimator unknown choice probabilities $P_n^0(\mathcal{U})$. These choice probabilities can be perfectly reproduced by a model of PS-logit structure if the PS coefficients are re-defined as

$$\mathrm{PS}_n(\mathcal{U}) = \frac{P_n^0(\mathcal{U})}{e^{\mu V_n^0(\mathcal{U})}}. \tag{4.39}$$

The resulting choice probabilities are

$$P_n(\mathcal{U}) = \frac{P_n^0(\mathcal{U}) e^{\mu(V_n(\mathcal{U}) - V_n^0(\mathcal{U}))}}{\sum_{\mathcal{V} \in C_n} P_n^0(\mathcal{V}) e^{\mu(V_n(\mathcal{V}) - V_n^0(\mathcal{V}))}} \tag{4.40}$$

such that $V_n(\mathcal{U}) = V_n^0(\mathcal{U})$ results in $P_n(\mathcal{U}) = P_n^0(\mathcal{U})$ for all $\mathcal{U} \in C_n$. Loosely speaking, any behavioral prior can be approximated up to 0th order in this way. The adequacy of this approximation for others than the prespecified utilities only depends on the approximated prior's elasticities, i.e., the way relative utility changes induce relative changes in the choice probabilities.

The elasticities of the PS-logit choice probabilities with respect to deterministic utilities are structurally identical to those of the MNL model:

$$\frac{\partial P_n(\mathcal{U})}{\partial V_n(\mathcal{V})} \frac{V_n(\mathcal{V})}{P_n(\mathcal{U})} = \begin{cases} \mu V_n(\mathcal{U})(1 - P_n(\mathcal{U})) & \mathcal{U} = \mathcal{V} \\ -\mu V_n(\mathcal{V}) P_n(\mathcal{V}) & \text{otherwise.} \end{cases} \tag{4.41}$$

In particular, if alternative $\mathcal{V}$ becomes more (less) attractive, its increased (decreased) choice probability reduces (increases) the choice probabilities of all other alternatives $\mathcal{U} \neq \mathcal{V}$ by the same relative amount.

Recall that the UM estimator functions without explicit knowledge of the PS coefficients. This implies that an application of the UM estimator can be justified

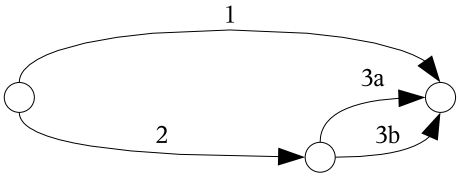


Figure 4.2: Three routes example, repeated
A simple route choice example with three alternative routes $\mathcal{A}$ (comprised of link 1), $\mathcal{B}$ (comprised of link sequence 2 →3a), and $\mathcal{C}$ (comprised of links 2→3b).

by approximation (4.40) even if the P_n^0 and V_n^0 values that (re-)define the PS coefficients in (4.39) are unknown. However, it is required that the elasticities of the prior choice distribution are sufficiently well captured by (4.41). Since the UM estimator's working coincides with that of Section 4.2's heuristic estimator, identical limitations hold for that heuristic.

4.4 Illustrative Example

The proposed estimators are illustrated with a simple example. For clarity, only a route choice problem is considered, and stationary conditions are assumed instead of a full dynamical model.

4.4.1 Scenario Description

The example network of Section 3.2.2.2 is reconsidered. It is repeated in Figure 4.2. A choice set of three routes $\mathcal{A}, \mathcal{B}$, and $\mathcal{C}$ connects the origin node at the very left to the destination node at the very right. The systematic utility of all routes is identically and invariably $\bar{V}$. The assumption of a constant systematic utility is adequate either in uncongested conditions or in a telematics setting where drivers are a priori unaware of actually prevailing network conditions. (An example with an underlying equilibrium assumption is given in the next chapter.)

Since routes $\mathcal{B}$ and $\mathcal{C}$ have almost perfect overlap, a behaviorally reasonable route split is $(P(\mathcal{A})\ P(\mathcal{B})\ P(\mathcal{C}))=(1/2\ 1/4\ 1/4)$. However, for the purpose of this example, a plain MNL model that does not account for route overlap is chosen as the behavioral prior:

$$P(\mathcal{U}) \propto e^{\mu \bar{V}},\ \mathcal{U} = \mathcal{A}, \mathcal{B}, \mathcal{C}, \tag{4.42}$$

where $\mu, \bar{V} = 1$ in all numerical experiments. This results in prior route splits

$$(P(\mathcal{A})\ P(\mathcal{B})\ P(\mathcal{C})) = (1/3\ 1/3\ 1/3). \tag{4.43}$$

The model is microscopic in that every departing driver $n = 1 \ldots N$ individually chooses a route. Since stationary conditions are assumed, a traveler's turning

move "sequence" $\mathcal{U}_n = \{\mathbf{u}_n\}$ and the resulting state "sequence" $\mathcal{X} = \{\mathbf{x}\}$ only consist of a single vector each:

$$\begin{aligned} \mathbf{u}_n &= (u_{\mathcal{A},n}\ u_{\mathcal{B},n}\ u_{\mathcal{C},n})^T && (4.44) \\ \mathbf{x} &= (x_{\mathcal{A}}\ x_{\mathcal{B}}\ x_{\mathcal{C}})^T. && (4.45) \end{aligned}$$

The elements of $\mathbf{u}$ indicate a driver's initial turn into route $\mathcal{A}$, $\mathcal{B}$ or $\mathcal{C}$: $\mathbf{u} = (1\ 0\ 0)^T$ represents the choice of route $\mathcal{A}$, $\mathbf{u} = (0\ 1\ 0)^T$ stands for route $\mathcal{B}$, and $(0\ 0\ 1)^T$ indicates route $\mathcal{C}$. Since no traffic flow dynamics are modeled, the network states are defined as the total route volumes

$$\mathbf{x} = \sum_{n=1}^{N} \mathbf{u}_n. \tag{4.46}$$

A single flow sensor is located on route $\mathcal{A}$. Its output y is modeled by the measurement equation

$$y = x_{\mathcal{A}} + \epsilon \tag{4.47}$$

where ϵ is a normal error with zero mean and σ^2 variance. The resulting log-likelihood (4.20) of population route choice $\mathcal{U}_1 \ldots \mathcal{U}_N$ given measurement "sequence" $\mathcal{Y} = \{y\}$ is

$$\begin{aligned} L(\mathcal{U}_1 \ldots \mathcal{U}_N | \mathcal{Y}) &= -\frac{(y - x_{\mathcal{A}})^2}{2\sigma^2} \\ &= -\frac{\left(y - \sum_{n=1}^{N} u_{\mathcal{A},n}\right)^2}{2\sigma^2}. \end{aligned} \tag{4.48}$$

A linearization of this function with respect to individual route choice is easier than in the general case of Section 4.1.2 since no dynamical constraints are involved. Maintaining the formalism of that section, $\Phi(\mathcal{X}(\mathcal{U}_1 \ldots \mathcal{U}_N))$ is defined to be $L(\mathcal{U}_1 \ldots \mathcal{U}_N | \mathcal{Y})$, Φ is linearized, and (4.10) yields a "sequence"

$$\Lambda = \left\{((y - x^0_{\mathcal{A}})/\sigma^2\ 0\ 0)^T\right\} \tag{4.49}$$

of Φ's sensitivities evaluated at a state "sequence" $\mathcal{X}^0 = \{\mathbf{x}^0\}$. According to (4.11), the approximate effect of a single agent that chooses route $\mathcal{A}$, $\mathcal{B}$ or $\mathcal{C}$ on the log-likelihood is

$$\begin{aligned} \langle \Lambda, \mathcal{A} \rangle &= (y - x^0_{\mathcal{A}})/\sigma^2 \\ \langle \Lambda, \mathcal{B} \rangle &= 0 \\ \langle \Lambda, \mathcal{C} \rangle &= 0. \end{aligned} \tag{4.50}$$

These expressions account for the effect of adding an agent to a route but ignore the effect of removing it from its previously chosen route. This is feasible because, once the effect of route choice is linearized, removing an agent from its original route does not change the linear effect of its reassignment to a new route. Since every choice implies that any previous choice is discarded, only the newly made choice is relevant for estimation. Formally, the effect of discarding an outdated choice is subsumed in the constant addend of (4.12).

4.4.2 Accept/Reject Estimator

The choice set $\{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$ is known and sampling from the prior (4.42) is easy, so the AR estimator can be applied without difficulty. Since all agents have identical choice sets, the acceptance probabilities (4.35) are likewise identical for all agents:

$$\begin{aligned} \phi(\mathcal{A}) &= e^{\langle \Lambda, \mathcal{A} \rangle}/D = e^{(y - x^0_{\mathcal{A}})/\sigma^2}/D \\ \phi(\mathcal{B}) &= e^{\langle \Lambda, \mathcal{B} \rangle}/D = 1/D \\ \phi(\mathcal{C}) &= e^{\langle \Lambda, \mathcal{C} \rangle}/D = 1/D \\ D &= \max\{e^{(y - x^0_{\mathcal{A}})/\sigma^2}, 1\}. \end{aligned} \tag{4.51}$$

That is, draws of route $\mathcal{A}$ are preferred over those of routes $\mathcal{B}$ and $\mathcal{C}$ if the exponent in $\phi(\mathcal{A})$ is positive, and they are suppressed if it is negative. Since a positive exponent indicates that less vehicles than measured are simulated on route $\mathcal{A}$ and a negative exponent indicates that too many simulated vehicles choose this route, the AR mechanism functions like a controller that works against the measurement error.

The acceptance probabilities of routes $\mathcal{B}$ and $\mathcal{C}$ are equal. This reflects the lack of measurement information that could justify a preference for either route. The equal acceptance probabilities in conjunction with the constant deterministic utilities also imply that the prior ratio of the choice probabilities for $\mathcal{B}$ and $\mathcal{C}$ is not affected by estimation. (If, however, the deterministic utilities were a function of the route volumes, the decision protocol may react to a change in estimated traffic conditions with a likewise changed ratio of $\mathcal{B}$'s and $\mathcal{C}$'s choice probabilities.)

An adopted version of Algorithm 3 that accounts for the simplified mobility simulation and the homogeneous driver population of this example is given below.

1. Initialization.
   (a) Set iteration counter $m = 0$.
   (b) Fill $\bar{\Lambda}^{(m)}$ (estimate of Λ fixed point) with all zeros.
2. Simulation.
   (a) Calculate acceptance probabilities $\phi^{(m)}(\mathcal{U})$ for $\mathcal{U} = \mathcal{A}, \mathcal{B}, \mathcal{C}$ (where $\bar{\Lambda}^{(m)}$ is substituted for Λ in (4.51)).
   (b) For $n = 1 \ldots N$, do:
      i. Draw candidate route $\mathcal{U}_n^{(m)}$ from the prior (4.43).
      ii. With probability $1 - \phi^{(m)}(\mathcal{U}_n^{(m)})$, discard the candidate and goto step 2(b)i.
      iii. Retain the first accepted choice $\mathcal{U}_n^{(m)}$.
   (c) As a stationary surrogate for a full network loading, use (4.46) to map $\mathcal{U}_1^{(m)} \ldots \mathcal{U}_N^{(m)}$ on $\mathcal{X}^{(m)}$.

3. Linearize the log-likelihood function by (4.49) and obtain $\Lambda^{(m)}$.

4. Update $\bar{\Lambda}^{(m+1)} = \frac{m}{m+1}\bar{\Lambda}^{(m)} + \frac{1}{m+1}\Lambda^{(m)}$.

5. If another iteration is desired:

   (a) Increase m by one.

   (b) Goto step 2.

For simulative investigations, a total demand of $N = 1000$ drivers is generated, and a single measurement $y_{\mathcal{A}} = 500$ is assumed on route $\mathcal{A}$. This value is what one would expect on average if a model was used that realistically accounts for route overlap by distributing the demand according to $(P(\mathcal{A})\ P(\mathcal{B})\ P(\mathcal{C}))$=(1/2 1/4 1/4).

The estimation convergence of 100 AR iterations for different measurement variances $\sigma^2 = 1000$, 100, and 10 is illustrated in Figure 4.3. The realistic volumes of 500 vehicles on route $\mathcal{A}$ and 250 vehicles on routes $\mathcal{B}$ and $\mathcal{C}$ are reproduced better with decreasing σ^2. An improved measurement reproduction comes at the cost of a lengthened settling time until the estimator draws from an apparently stable posterior. This is owed to the log-likelihood's increased steepness that complicates the identification of a fixed point. The ratio of route $\mathcal{B}$ and $\mathcal{C}$'s share is not influenced by the estimation, as it has been previously hypothesized.

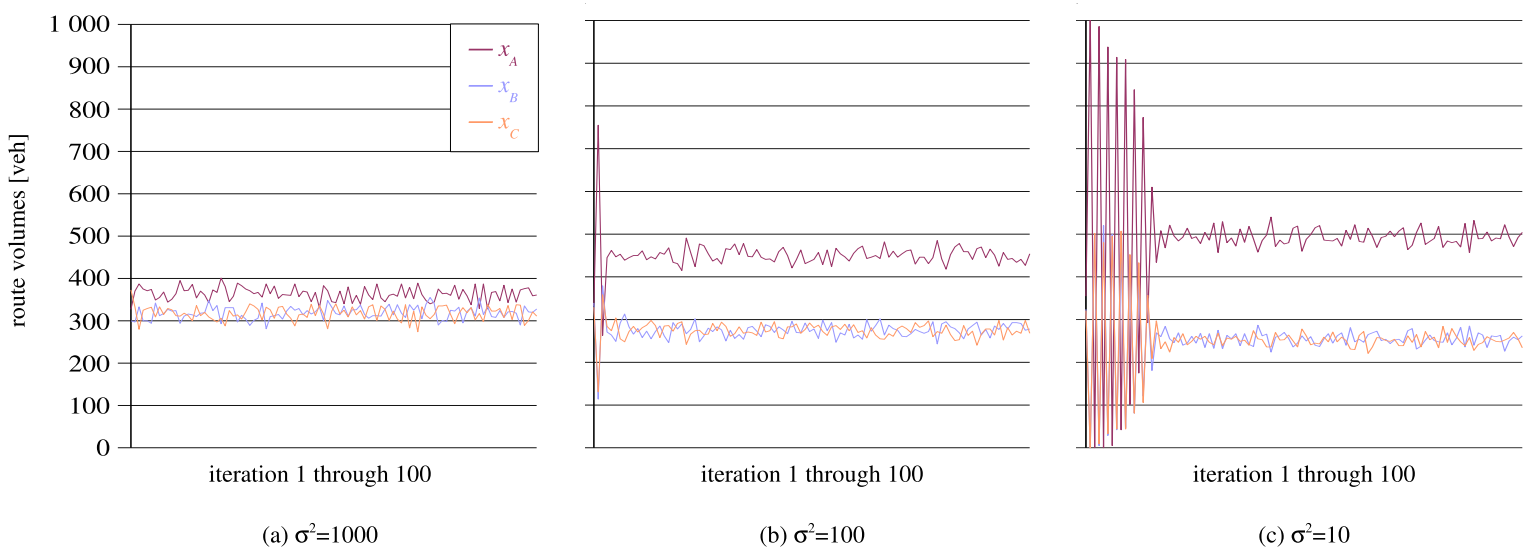
The percentage of accepted draws is 92%, 74%, and 64% for $\sigma^2 = 1000$, 100, and 10. The smaller the measurement variance the more pronounced the difference between prior and posterior and the more draws from the prior need to be rejected to generate the posterior. The number of draws required by the AR estimator generally increases the more the likelihood contradicts the prior.

4.4.3 Utility-Modification Estimator

The UM estimator specified in Algorithm 4 is employed. The same experimental setting as for the AR estimator is chosen, and the same adjustments are made in order to account for the simplified nature of this example. Since every draw based on the modified utilities is accepted, the computational overhead of the AR estimator is avoided. Furthermore, since the MNL prior route choice distribution (4.42) is of PS-logit structure, the resulting estimates are draws from an identical posterior distribution as for the AR estimator. Their illustration is therefore omitted.

In this simple example, the utility corrections generated by the UM estimator allow to reconstruct the PS coefficients that are disregarded in the plain MNL prior (4.42): Given $P(\mathcal{U}) \propto e^{\mu\bar{V}}$, the UM estimator generates a posterior $P(\mathcal{U}|\mathcal{Y}) \propto e^{\langle\Lambda,\mathcal{U}\rangle}e^{\mu\bar{V}}$, cf. (4.36). Comparing this to a hypothetical PS-logit prior $P(\mathcal{U}) \propto \mathrm{PS}(\mathcal{U})e^{\mu\bar{V}}$ that properly accounts for route overlap, one notices that $e^{\langle\Lambda,\mathcal{U}\rangle}$ can indeed be considered as an estimate of $\mathrm{PS}(\mathcal{U})$.

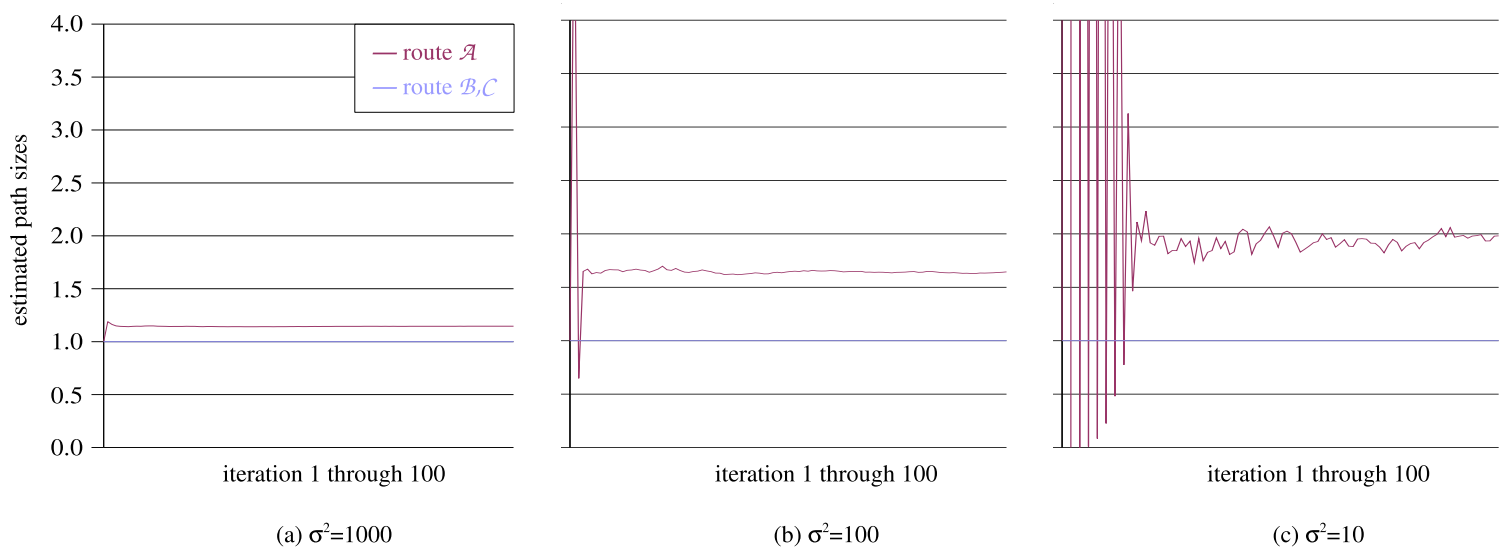
Figure 4.4 plots $e^{\langle\bar{\Lambda}^{(m)},\mathcal{U}\rangle}$ for $\mathcal{U} = \mathcal{A}, \mathcal{B}, \mathcal{C}$ over the iteration counter m. Apparently, these values converge towards $(e^{\langle\bar{\Lambda}^{(\infty)},\mathcal{A}\rangle}\ e^{\langle\bar{\Lambda}^{(\infty)},\mathcal{B}\rangle}\ e^{\langle\bar{\Lambda}^{(\infty)},\mathcal{C}\rangle}) = (2\ 1\ 1)$ for



(a) σ^2=1000 (b) σ^2=100 (c) σ^2=10

Figure 4.3: Measurement fit

Estimated route volumes over the iteration counter for various measurement variances. An increasing belief in the measurement results in a closer reproduction of the true route splits but also in a lengthened settling time.



(a) σ^2=1000 (b) σ^2=100 (c) σ^2=10

Figure 4.4: Estimated path sizes

Trajectories of path size estimates $e^{\langle \bar{\Lambda}^{(m)}, \mathcal{U} \rangle}$ for $\mathcal{U} = \mathcal{A}, \mathcal{B}, \mathcal{C}$ over iteration counter m. For decreasing σ^2, these estimates approach values that are proportional to the real path sizes based on which the utilized measurement was generated.

small measurement variances. This is a merely scaled version of the path size coefficients (PS($\mathcal{A}$) PS($\mathcal{B}$) PS($\mathcal{C}$)) = (1 1/2 1/2) that were derived for this scenario in Section 3.2.2.2. These path sizes yield the plausible route choice probabilities ($P(\mathcal{A})$ $P(\mathcal{B})$ $P(\mathcal{C})$)=(1/2 1/4 1/4) based on which the utilized measurement was generated.

It was hypothesized in Section 4.1.4 that an estimated utility modification captures those systematic features of an alternative that are not included in its original utility. However, in the present example, systematic utility is perfectly modeled, and the UM estimator only accounts for the overlap of routes $\mathcal{B}$ and $\mathcal{C}$. This shows, given a RUM-based decision protocol, that the correction terms only represent unmodeled systematic utilities if all correlations in the utility errors are properly modeled. Otherwise, unmodeled correlations may also be accounted for by the estimator. In general, a distinct interpretation of the results is impossible. Still, this observation does not impair the correctness of the estimated posterior distributions themselves.

Concluding, this chapter provides a number of methods for the estimation of individual-level motorist behavior. All methods have the same Bayesian origin but differ in their adopted simplifications. A small example clarifies the proposed algorithms. A large test case is investigated in the next chapter.

Chapter 5

Test Case

This chapter investigates the applicability of the proposed estimation approach to a synthetic scenario of practically relevant size. It focuses on computational feasibility and logical correctness. Since various simplifications are necessary to implement the test case, its limitations likewise confine the scope of these investigations. However, the results clearly establish that the estimator exhibits sufficient precision, robustness, and computational performance to be studied in more realistic settings and in conjunction with more sophisticated modeling components.

5.1 Experimental Overall Setting

5.1.1 Scenario Description

A scenario consists of two components: (i) invariable settings that describe the structural features of this test case and (ii) a particular choice of variable settings.

5.1.1.1 Invariable Settings

All experiments utilize the Berlin network described in Section 2.6.2. The respective driver population is introduced in Section 3.1.4. Behavioral estimation for a 206 353-agent population on a 2 459-link network is a nontrivial problem. All experiments are constrained to the time span from 6 to 9 am. This interval exhibits the most variable traffic conditions because of the morning rush hour.

Since only placeholder components for the behavioral simulator are available, the sole degree of freedom considered here is route choice. That is, all behavioral aspects apart from route choice are retained unchanged in the original plans generated by MATSim. This setting is motivated in two ways. First, MATSim's basic approach to route choice is relatively simple to simulate but at the same time non-trivial from an estimation point of view, cf. Section 3.2.2.2. Second, route choice can be generalized to plan choice by minor modifications to the

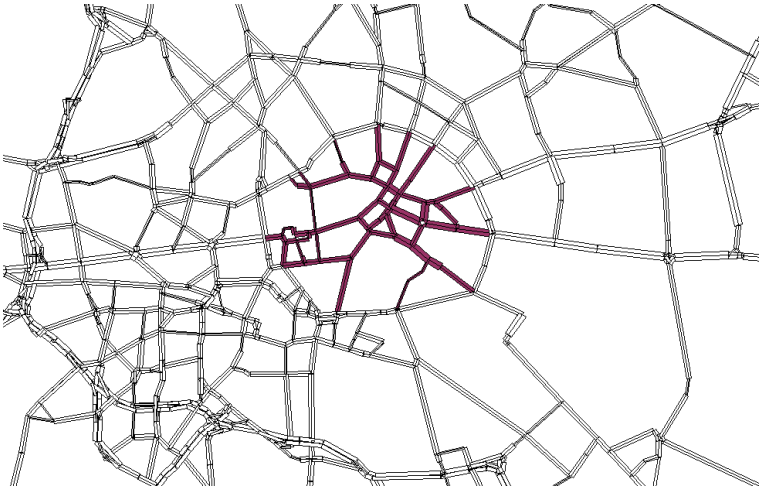


Figure 5.1: Inner-urban part of Berlin
A time-independent toll of 0.24 EUR/km is charged on the colored links.

original network, cf. Section 3.2.1.1. This suggests that an effective route choice estimator is likely to be applicable in a more general setting as well.

In all experiments, a time-independent toll of 0.24 EUR/km is charged in the city center shown in Figure 5.1, and no toll is charged outside of this area. The unitless utility of a route $\mathcal{U}$ is

$$V_n(\mathcal{U}) = \left(-\mathrm{tt}(\mathcal{U}) - \frac{\mathrm{toll}(\mathcal{U})}{\mathrm{VOT}_n} \right) / \, 1 \text{ s} \tag{5.1}$$

where $\mathrm{tt}(\mathcal{U})$ is the travel time on route $\mathcal{U}$, $\mathrm{toll}(\mathcal{U})$ is the toll accumulated along route $\mathcal{U}$, and VOT_n is individual n's value of time in EUR/h. For comparison, the effect of a 0.24 EUR/km toll is equivalent to a travel time increase by once the free-flow travel time given a 12 EUR/h VOT and a 50 km/h speed limit.

5.1.1.2 Variable Settings

Combining the invariable settings given above with a particular VOT defines a **scenario**. For simplicity, it is assumed that all drivers within one scenario have an identical value of time, i.e., $\mathrm{VOT}_n = \mathrm{VOT}$, $n = 1 \ldots N$. Clearly, this setting disregards a multi-agent model's prominent advantage of capturing a heterogeneous driver population. However, the purpose of these experiments is not to re-iterate the well-known features of a multi-agent simulation but to investigate an estimator's performance in controlled conditions. A homogeneous VOT simplifies the setup of the experiments and their interpretation. Since VOT is an agent-specific parameter that is entirely transparent to the estimator, no conceptual difficulty exists in estimating the behavior of a population that is heterogeneous in this regard. Finally, no VOT information is contained in the synthetic population available for this dissertation anyway because the current MATSim implementation provides no such information.

Depending on the particular modeling assumptions, a **planning scenario** and a **telematics scenario** can be distinguished consistently with the terminology of Section 1.1.3: If drivers are aware of a recently implemented toll but not yet of the resulting changes in traffic conditions, the hitherto prevailing equilibrium conditions are invalidated and a transient phase emerges. This scenario can only be represented by a telematics simulation that does not rely on a (S)UE assumption. If drivers are aware of the toll but also have learned the resulting changes in traffic patterns, the transient phase stabilizes again. This scenario can be addressed by a planning simulation the equilibrium assumption of which is approximately satisfied here.

5.1.2 Simulation and Estimation Logic

The following two subsections elaborate on the applied simulation and estimation logic. The simulator is described first. Since the estimator "wraps around" an existing simulation system, cf. Figure 1.2, the simulator is entirely independent of the subsequently selected estimation approach.

5.1.2.1 Simulation

Traffic flow dynamics are represented by the mobility simulation described in Chapters 2 and 3.1. For behavioral simulation, the simple logic outlined in Section 3.2.2.2 is applied with minor modifications. Basically, 10 percent of all agents recalculate a new route in every iteration. Only pre-trip route (re)planning is considered.[1] The implemented decision protocol executes choice set generation and choice in a deliberative manner, cf. Section 1.3.1.

Whenever an agent starts a trip, it has one already generated route $\mathcal{U}$ at hand. This is either the route chosen in the previous iteration or, at the initial iteration, the route provided in the MATSim plans file. The agent also is aware of the most recently observed travel times. An alternative route is generated by *randomly* choosing a VOT from the set $\{6, 12, 18, \infty\}$ (all in EUR/h) and running a time-dependent best path algorithm that maximizes the resulting generalized utility specified in (5.1). The infinite VOT serves as a notational proxy for a no-toll case since it effectively eliminates the toll addend from the utility. The newly calculated route is denoted by $\mathcal{V}$. This yields a choice set of two elements: the original route $\mathcal{U}$ and the new route $\mathcal{V}$.

The agent then selects from $\{\mathcal{U}, \mathcal{V}\}$ the route of higher utility based on the simulated scenario's *actual* VOT and the most recently observed traffic conditions. Since the traffic conditions vary from iteration to iteration, this choice may not be optimal in hindsight.

This model is chosen because of its similarity to the original MATSim route replanning logic. Altogether, a single iteration of this simple DTA simulator consists of two steps, and repeated executions of these iterations constitute a simulation run:

[1] The sole consideration of pre-trip replanning keeps the modeling simple. The estimator itself is applicable to en-trip replanning as well, cf. Section 4.3.1.

1. For all agents $n = 1 \ldots N$, do: With probability 0.9, maintain n's route. Otherwise, generate an alternative route based on a randomly generated VOT and the most recently observed travel times, and select the better one of these two alternatives according to the scenario's actual VOT.

2. Load all agents on the network.

This procedure can be applied to simulate both a planning and a telematics scenario. The planning scenario assumes that drivers learn from iteration to iteration. If one looks at **relaxed** iterations only, i.e., such iterations where traffic conditions have attained a stable distribution, then an alternative interpretation is that the situation of interest is one where drivers are aware of global traffic conditions. This is realized if route-replanning is based on the previous iteration's travel times. For a telematics scenario, however, it is necessary to run iterations while drivers remain on their initial level of knowledge. This knowledge is generated beforehand by running many iterations of a relaxed planning simulation and saving the travel times of every iteration. These travel times are then used by replanning travelers in the iterated telematics simulation.

Even this simple simulator exhibits fairly complex dynamics. Since an elaborate analysis of these dynamics is beyond the scope of this dissertation, the notion of a "relaxed simulation" that reaches "stable network conditions" is to be understood informally and only in a given experimental context. Consequently, all "convergence" statements regarding the subsequently described simulation-based estimator are of likewise experimental nature.

5.1.2.2 Estimation

The estimator adjusts a **prior scenario** to measurements that are observations from a **true scenario**. (Measurement generation is described further below.) The prior and the true scenario only differ in their VOT. The true scenario represents a **synthetic reality** that would in a real-world application be replaced by reality itself.

At this stage of research, a real-world test case would rather obscure than clarify the estimator's working since (i) no guidelines for its application are yet available, (ii) uncontrollable error sources would complicate an interpretation of the estimation results, and (iii) only a simulated reality is perfectly observable for a comparison to its estimated counterpart. Furthermore, merely an outdated Berlin network and driver population are available since the MATSim research efforts shifted towards the city of Zurich around the beginning of 2007. This change occurred too late to be traced by this research.

The UM estimator is applied in all experiments. This is required by the implicit nature of the behavioral model. As explained in Section 3.2.2.2, route recalculations based on a previous iteration's travel times model a perceptional error that does not become observable until the next network loading is executed. Since this error is generated in hindsight, there is no variability within a single choice situation. The AR estimator is generally not applicable to this type of

"best response" simulation.[2] Furthermore, since no PS-logit route choice model is used, only a heuristic application of the UM estimator is possible. This also puts its robustness with regard to a behavioral prior that is not guaranteed to be of PS-logit structure to test, cf. Section 4.3.4.

Since the UM estimator is technically equivalent to the heuristic estimator of Section 4.2.2, the following presentation is given in terms of the latter. The heuristic estimator adds a global utility function Φ to the individual utility of every agent, where Φ is a similarity measure between simulated and observed sensor data. More precisely, the estimator replaces any driver n's original utility perception $V_n(\mathcal{U})$ as defined in (5.1) by a modified utility $W_n(\mathcal{U}) = V_n(\mathcal{U}) + \langle\Lambda,\mathcal{U}\rangle/\mu$ where the second addend is a linearized and scaled version of Φ. In all subsequent experiments, Φ is specified by

$$\Phi(\mathcal{X}) = -\sum_a \sum_k \frac{(y_a(k) - g_a[\mathbf{x}(k)])^2}{2\sigma^2} \tag{5.2}$$

where $y_a(k)$ is a measurement on sensor-equipped link a in time step k and $g_a[\mathbf{x}(k)]$ is its simulated counterpart. An interpretation of this function as the log-likelihood of mutually independent normal measurements with identical variances σ^2 is possible but, in light of the overall heuristic setting, not mandatory.

Φ is effectively scaled by σ^{-2}. Since this multiplication can be applied either before or after the linearization, it is assumed that the Λ values result from a linearization of $\Phi(\mathcal{X}) = -\sum_{ak}(y_a(k) - g_a[\mathbf{x}(k)])^2/2$ and that the σ^2 parameter is accounted for afterwards:

$$W_n(\mathcal{U}) = V_n(\mathcal{U}) + \frac{\langle\Lambda,\mathcal{U}\rangle}{\mu\sigma^2}. \tag{5.3}$$

Only the product of μ and σ^2 is relevant to the estimation problem. Since it reflects the belief in the prior information represented by the original utility perception $V_n(\mathcal{U})$, it is subsequently represented by a prior weight

$$w_{\text{prior}} = \sqrt{\mu\sigma^2}. \tag{5.4}$$

For interpretation, given a unit scale parameter μ, w_{prior} is equivalent to a normal measurement's standard deviation. An experimental parameter tuning approach is adopted for its selection. This also is likely to be the course of action in a real-world application [171].

The estimation logic approaches a fixed-point of the Λ values by means of the SA algorithm described in Section 4.1.3. This procedure iterates between a linearization of (5.2) and an iteration of the traffic simulator. That is, in every iteration of the estimator, 10 percent of all departing agents replan based on the most recently obtained utility corrections, a single network loading is run, and the utility corrections are immediately updated. The complete estimation logic is given below:

[2] Speaking in terms of the particularly chosen model: The route choice set is generated based on a randomized VOT once per iteration, but it is fixed throughout that iteration. That is, repeated best response choices within a single iteration invariably yield the same result.

1. Initialization.
   (a) Set iteration counter $m = 0$.
   (b) Fill $\bar{\Lambda}^{(m)}$ (estimate of Λ fixed point) with all zeros.
2. Simulation.
   (a) For all $n = 1 \ldots N$, do with probability 0.1:
      i. Choice set generation. Generate an alternative route based on a randomly generated VOT and the most recent travel times.
      ii. Choice. Evaluate $W_n(\mathcal{U}) = V_n(\mathcal{U}) + \langle \bar{\Lambda}^{(m)}, \mathcal{U} \rangle / w^2_{\text{prior}}$ instead of $V_n(\mathcal{U})$ when selecting $\mathcal{U}_n^{(m)}$. $V_n(\mathcal{U})$ is evaluated based on the prior scenario's actual VOT.
   (b) Load $\mathcal{U}_1^{(m)} \ldots \mathcal{U}_N^{(m)}$ on the network and obtain $\mathcal{X}^{(m)}$.
3. Linearize $\Phi(\mathcal{X}^{(m)})$ and obtain $\Lambda^{(m)}$.
4. Update $\bar{\Lambda}^{(m+1)} = \dfrac{m}{m+1}\bar{\Lambda}^{(m)} + \dfrac{1}{m+1}\Lambda^{(m)}$.
5. If another iteration is desired:
   (a) Increase m by one.
   (b) Goto step 2.

Note that the choice set generation is based on the *original* utility V_n and a *randomized* VOT, whereas the choice is based on the *modified* utility W_n and the prior scenario's *actual* VOT. This ensures that every once in a while the choice set contains a route that is consistent with the true scenario's VOT.[3] The question thus becomes in how far the estimator, given the above set of behavioral alternatives but only a limited number of measurements, can pull the system away from the "wrong" VOT of the prior scenario towards the "correct" VOT of the true scenario.

If there are no measurements, the Λ coefficients are invariably zero and the above algorithm merely repeats steps 2a and 2b. That is, it functions as a simulator that, upon stabilization in relaxed conditions, produces a sequence of draws from the behavioral prior distribution. As measurements become available, nonzero Λ values result, and the estimator stabilizes in different relaxed conditions. Every iteration then generates a draw from the behavioral posterior given the particular prior scenario and the available measurements from the true scenario.

Technically, the estimation problem is to identify a fixed point of the Λ coefficients. Since the mapping from Λ on itself is effectively from Λ on $\mathcal{X}$ on Λ, cf. Figure 4.1, the existence of a Λ fixed point indicates the existence of a $\mathcal{X}$ fixed point, and vice versa. This justifies the exclusive evaluation of the readily interpretable system states $\mathcal{X}$ to monitor the estimator's convergence, as it is described in the next section.

[3] There is no guarantee that running a best path algorithm directly on modified link utilities ever produces a likewise realistic alternative. Section 5.4 elaborates on this matter.

Figure 5.2: Exemplary sensor locations
50 automatically selected sensor locations. One flow sensor is located in the center of each colored link.

5.1.3 Sensor and Validation Data

The estimator utilizes a limited amount of flow measurements as sensor data. The estimation results are validated based on network-wide occupancy information.

5.1.3.1 Sensor Data

Flow measurements, i.e., traffic counts at road cross-sections per time interval, are used in all experiments as synthetically generated **sensor data**. The term **measurement data** is equivalently used. All such data is averaged in 5 minute time bins.

For every estimation experiment, 50 sensor locations are selected based on a comparison of the traffic conditions in the according prior and true scenario. The locations are automatically chosen by a simple tool that prefers links on which the average flow difference between both scenarios is largest and at the same time seeks to maintain independent measurement locations. Sensor locations are chosen for all scenarios individually in order to provide equally advantageous preconditions for better comparability. An example of suchlike generated locations is given in Figure 5.2. The true traffic conditions utilized by this procedure are of course unknown in a real-world application, where, however, prespecified sensor locations can be expected to be available.

The mapping from driver behavior on traffic flows is nonlinear. In particular, the intermediate mapping from traffic densities on flow rates is ambiguous in that every non-maximum flow can be explained by two different densities, cf. Section 2.2. Since the estimation is based on repeated linearizations, such nonlinearities increase the danger of local convergence. Therefore, an additional

source of information is employed. Even a simple single-loop detector does not only measure flow rates but also the fraction of time it is covered by a vehicle. This information is likely to be too noisy to provide immediately useful traffic density information, but it does allow to distinguish free and congested traffic conditions [49]. The estimator uses this information in its linearization step where it recognizes that in uncongested conditions the log-likelihood of any measurement is only sensitive to the upstream traffic situation and in congested conditions it is only sensitive to the downstream situation.

5.1.3.2 Validation Data

One may argue that an appraisal of the estimation quality should be directly based on routes. However, since every agent may choose any cycle-free route towards its destination, it is unlikely that an estimated and a true route coincide. In principle, the measure of route overlap proposed in [148] is applicable here. Still, the continuous variability of the simulated traffic conditions and of the resulting routes complicates such a comparison, and a more viable validation approach is at hand: In the considered model, simulated flows result deterministically from macroscopic system states, which in turn are consequences of microscopic driver behavior, cf. Section 3.1.3. Macroscopic link occupancies thus constitute intermediate states that are easy to process and interpret.[4] Since the route choice model is based on travel times which are deterministically dependent on macroscopic link states, an estimator that reproduces link states well is likely to also generate realistic routes. Particularly, the behavioral model placeholder is by design sufficiently restricted to unequivocally ascribe any systematic change in aggregate traffic conditions to the behavioral aspect of toll-avoidance.

In consequence, network-wide occupancy information, i.e., the average number of vehicle units on every link in every 5-minute time bin, is used as the **validation data** based on which global traffic conditions are compared. More general experiments are likely to also call for more powerful behavioral monitoring tools, which constitutes a research question in its own right.

5.1.3.3 Quantitative Error Measures

The notion of a **run** is subsequently used as a generic term for both a simulation run and an estimation run. The difference of a run to a **reference data set** is evaluated in terms of a root mean square error measure

$$\mathrm{RMS}_z^{(m)}[\mathrm{run}] = \sqrt{\frac{\sum_{a\in A}\sum_k \left(z_a^{(m)}(k) - z_a^{\mathrm{ref}}(k)\right)^2}{K|A|}} \tag{5.5}$$

where $z_a^{(m)}(k)$ is the considered traffic characteristic (flow, occupancy) of the studied run in iteration m on link a in time bin k, and $z_a^{\mathrm{ref}}(k)$ is the according reference value. K is the total number of time bins and A is the set of links for

[4] The hitherto used notion of occupancy as the number of vehicle units located in a cell or link is not to be confused with the common notion of an inductive loop's occupancy as the fraction of time it is covered by a vehicle. The latter is not employed here.

which traffic characteristics are evaluated. [run] is a shortcut for the evaluated run. Unique reference data sets are used in all planning experiments and in all telematics experiments respectively. Whenever the dependency of RMS on iteration counter m is omitted, the last RMS value in a prespecified sequence of iterations is referred to.

It is frequently required to compare a run's (recursively) averaged characteristics

$$\bar{z}_a^{(m)}(k) = \begin{cases} z_a^{(m)}(k) & m < m_0 \\ \dfrac{1}{m - m_0 + 1} \displaystyle\sum_{m'=m_0}^{m} z_a^{(m')}(k) & m \geq m_0 \end{cases} \tag{5.6}$$

to the reference data, where m_0 is always chosen large enough to ensure that the considered run reaches a stable distribution of network conditions before the averaging starts. This allows for the definition of an additional error measure

$$\mathrm{RMSA}_z^{(m)}[\mathrm{run}] = \sqrt{\frac{\sum_{a \in A} \sum_k \left(\bar{z}_a^{(m)}(k) - z_a^{\mathrm{ref}}(k)\right)^2}{K|A|}}, \tag{5.7}$$

where the only difference to RMS_z is that $z_a^{(m)}(k)$ is now replaced by the average value $\bar{z}_a^{(m)}(k)$.

The following particular error measures are used.

- The **measurement error** RMSA_q is an instance of (5.7) that represents the deviation of an estimation run from its measurement data set. That is, the reference data used here is identical to the measurement data used for estimation. Consequently, only the flow rates at the prespecified 50 sensor locations are evaluated. Note that the measurement error is basically a scaled version of $\sqrt{-\Phi}$, cf. (5.2). Its unit is veh/h, which is subsequently omitted for brevity.
- The **validation error** $\mathrm{RMS(A)}_x$ is an instance of (5.5) or (5.7) that represents the deviation of a simulation run or an estimation run from its validation data set. At this, it compares the occupancies on all links in the network. Its unit is veh, which also is subsequently omitted.

5.2 Planning Experiments (Equilibrium Situation)

A planning-like setting is considered first. SUE conditions are modeled by providing global knowledge about the previous iteration's traffic conditions to all replanning agents in the iterative DTA procedure described in Section 5.1.2.1. All experiments use sensor data from a true scenario that is based on one particular VOT, whereas the prior scenario assumed by the estimator is based on a different VOT.

The experiments given here examine the logical correctness and overall precision of the estimator. Since computational performance is not of primary concern in an offline planning application, its investigation is postponed to Section 5.3 where a telematics case study in simulated online conditions is described.

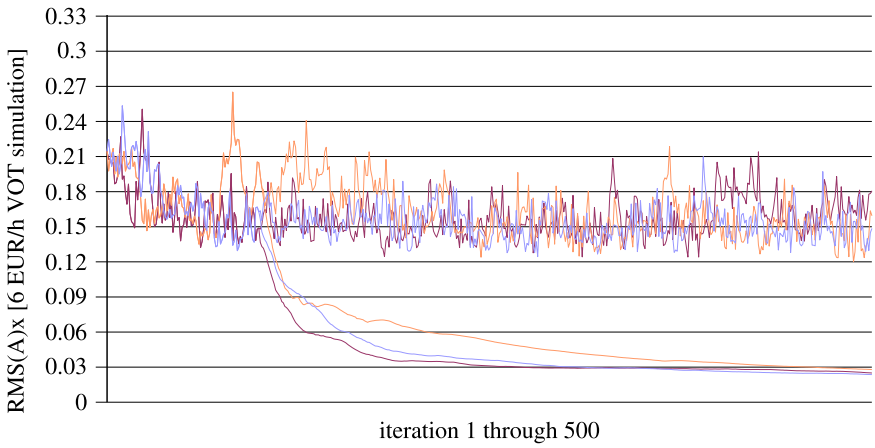


Figure 5.3: RMS_x and $RMSA_x$ [6 EUR/h VOT simulation]
Three simulation runs of 500 iterations each are conducted in order to investigate the stability of the 6 EUR/h VOT scenario. The fluctuating RMS_x values effectively represent the Euclidean distance between the reference data and the simulated occupancies of a particular iteration. The recursive state averaging is turned on after 100 iterations such that a smooth $RMSA_x$ curve branches off each RMS_x curve.

5.2.1 Scenario Generation

Given the above overall settings, one planning simulation is run for a scenario without toll, and three further simulations are run for toll-scenarios with VOTs of 6, 12, and 18 EUR/h. Each simulation is executed for 500 iterations. These initial runs are subsequently denoted as the "no-toll" and the "6 (12,18) EUR/h VOT" **reference simulations** of their respective planning scenarios. Link flows and occupancies are averaged over the last 400 iterations of each reference simulation according to (5.6). These average values constitute the reference data sets for all RMS and RMSA error measures given in the subsequent planning experiments, cf. Section 5.1.3.3.

5.2.1.1 Investigation of Scenario Stability

To test the robustness of this set-up, another three simulations are run for every scenario. They are compared to their respective reference scenario by tracking the validation errors RMS_x and $RMSA_x$ over 500 iterations, as shown in Figures 5.3 through 5.6.

All experiments start with an identical plans file. This results in different transients during the first iterations. Since these transients represent no relaxed network conditions, the recursive state averaging is turned on not before iteration 100 where a $RMSA_x$ curve branches off each RMS_x curve. Since this branching in conjunction with much smoother dynamics is characteristic for all $RMSA_x$ curves, they are not explicitly labeled in the plots.

The $RMSA_x$ curves approach small values when compared to their RMS_x counterparts. This indicates that all simulations for a particular VOT attain similar

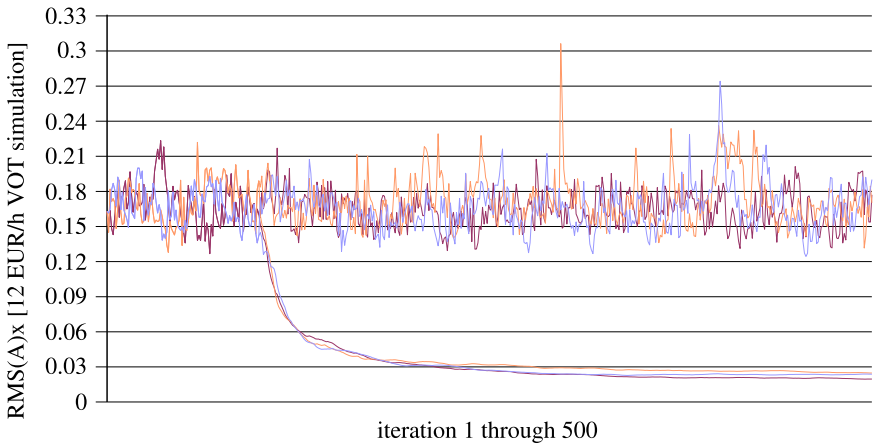


Figure 5.4: RMS_x and RMSA_x [12 EUR/h VOT simulation]
Three simulations of the 12 EUR/h VOT scenario. See Figure 5.3 for further explanations.

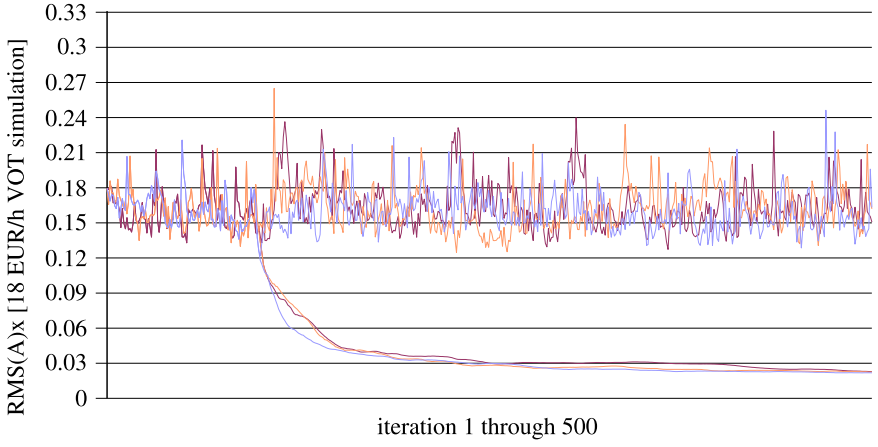


Figure 5.5: RMS_x and RMSA_x [18 EUR/h VOT simulation]
Three simulations of the 18 EUR/h VOT scenario. See Figure 5.3 for further explanations.

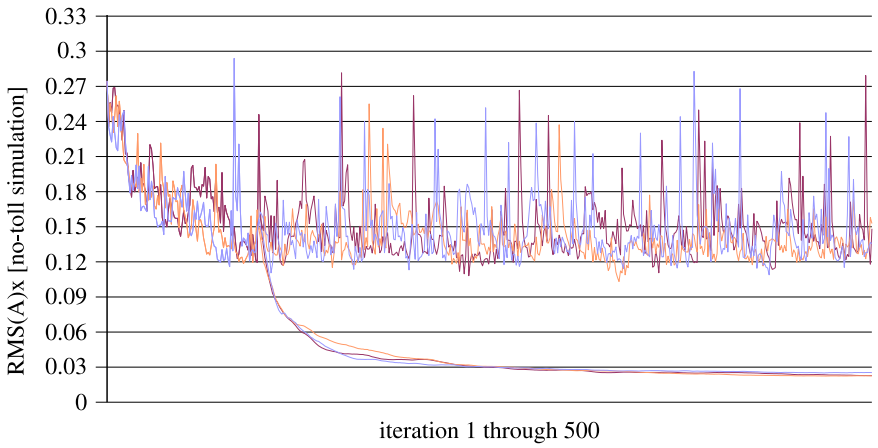


Figure 5.6: RMS_x and RMSA_x [no-toll simulation]
Three simulations of the no-toll scenario. See Figure 5.3 for further explanations.

average system states. The RMS_x curves stabilize at a constant degree of variability. A visual inspection shows a positive auto-correlation within each curve. This results from the simulation logic that invariably copies 90 percent of all routes from one iteration to the next. Altogether, the network states exhibit a fluctuating and possibly cycling behavior. Since no systematic drift is observed, convergence towards a stable state distribution cannot be disproved.

All RMS curves are located above their RMSA counterparts. However, this observation does not prove a systematic difference between the average system states and the single-iteration draws. It rather is a consequence of the chosen error measures, and the same RMS vs. RMSA constellation would result even if the relaxed system states were perfectly normally distributed: The surface of an $(n+1)$-dimensional sphere with radius r is proportional to r^n. The probability that a single network state is simulated r distance units away from its expectation therefore results from an integration of its p.d.f. over a domain the size of which is proportional to r^n. Since the reference data used in RMS(A) consists of average network characteristics that approximate this expectation, a small RMS value is as unlikely to occur as a small r value, whereas vanishing RMSA values merely result from the law of large numbers.

5.2.1.2 Measurement and Validation Data Generation

An accurate generation of the synthetic measurements for a single day requires to take one relaxed iteration of the true scenario, to extract the flow data at all sensor locations, and to randomly disturb this data according to a distributional assumption about the measurement error. Based on this information, the (planning) estimator is run with the goal to reproduce the true distribution of traffic conditions. In consequence, an "exhaustive" validation procedure must compare two full distributions of traffic conditions.

Within the scope of this work, distributions are compared in terms of their

expectations. The similarity of an estimated and a true distribution of network conditions can thus be quantified by an RMSA error measure. This error measure is a random variable itself since it depends on the particular draw of measurement data $\mathcal{Y}$ that is used for estimation, i.e., $\mathrm{RMSA} = \mathrm{RMSA}(\mathcal{Y})$. A reliable appraisal of the estimation quality would therefore require to generate a large number of measurement data sets $\mathcal{Y}$ and to run the estimator for each of these sets individually. An expected error $\mathrm{E}\{\mathrm{RMSA}(\mathcal{Y})\}$ could then be identified by averaging $\mathrm{RMSA}(\mathcal{Y})$ over all experiments.

Since strong variability can be observed in the simulations, many computationally demanding experiments would be needed to identify the estimator's expected performance.[5] Even if this effort was shouldered, the validity of the resulting assessment would be limited by that of the deployed model placeholders. These reservations motivate a less rigorous yet computationally more viable approach.

A single, most representative measurement data set is used for each true scenario. The stability analysis of Section 5.2.1.1 shows that repeated simulations of a particular scenario converge to similar average network states. The initially generated reference data sets for each scenario are therefore used as sensor and validation data in all planning experiments. Averaging the data instead of averaging the evaluation results is equivalent to the clearly idealized assumption that $\mathrm{E}\{\mathrm{RMSA}(\mathcal{Y})\} \approx \mathrm{RMSA}(\mathrm{E}\{\mathcal{Y}\})$ is a feasible approximation.

No additive sensor noise is simulated since only its zero expectation appears in $\mathrm{E}\{\mathcal{Y}\}$. This underlines the idealized experimental setting since the true level of sensor noise will in reality certainly impair the estimation performance. However, since there is no guarantee that the average of many physically possible system states is itself physically feasible, a systematic error may be introduced. These aspects must be accounted for when interpreting the estimation results.

This simplification may even be realistic in a setting where the sensor data available for planning purposes has been averaged over many days. However, the effective motivation for this approach is to limit the degrees of freedom that need to be experimentally investigated. One should recall that the purpose of these experiments is to demonstrate the estimator's logical correctness. Once this is achieved, sufficient conceptual background is provided in Chapter 4 for more extensive investigations in likewise more realistic experimental settings.

5.2.1.3 Comparison of Scenarios

Figure 5.7 provides an impression of the difference between the synthetic reality on the one hand and the prior scenario assumed later during estimation on the other hand. It contains six scatterplots that compare the flow and occupancy data of the 12 EUR/h VOT reference simulation to the 6 EUR/h VOT, the 18 EUR/h VOT, and the no-toll reference simulation.

The first column compares the reference flow rates and the second column compares the reference occupancies. All scatterplots contain data points for *all* links

[5] Recent experimental results milden this concern. However, since these results were obtained too late to be accounted for in this dissertation, they are only indicated in this and a few subsequent footnotes.

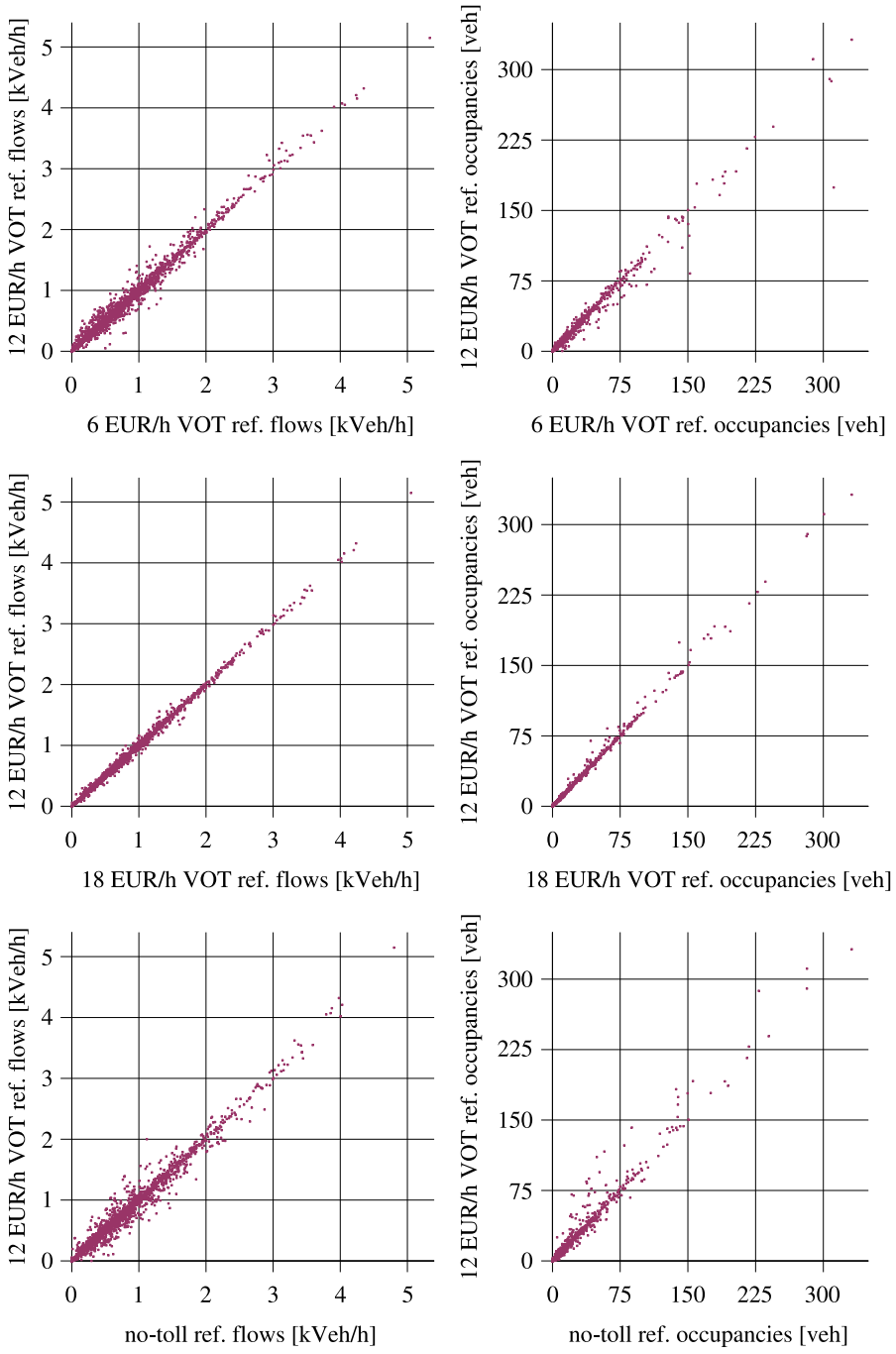


Figure 5.7: Scatterplots for comparison of planning reference simulations
The scatterplots compare data from the 12 EUR/h VOT planning reference simulation (on the ordinate) to the other planning reference simulations (on the abscissa). The first column compares flow rates and the second column compares occupancies. All scatterplots contain data points for *all* links in the network. The data points apply to the simulation time interval from 8:30 to 8:35 and represent average values over 400 iterations.

in the network. That is, the flow scatterplots contain more information than the RMSA_q measurement error, which only accounts for data at sensor locations. The measurement error indicates to what degree the estimator is able to reconstruct available sensor data, whereas the scatterplots allow for a network-wide comparison of traffic conditions.

All data points apply to the simulation time interval from 8:30 to 8:35. At first glance, the deviations appear moderate in consideration of the broad range of VOTs. However, recall that all reference data sets are averaged over 400 iterations. An inspection of the simulation dynamics in Figures 5.3 through 5.6 shows that variability is much larger without averaging. Using average data allows to ascribe all perceptible deviations in the scatterplots to systematic causes.

The flow scatterplots in the left column give an impression of the amount of information effectively available to the estimator. The stronger the flow deviations between two scenarios the more useful are flow measurements to adjust one scenario to another. Vice versa, if two scenarios differ only slightly in their flows, the estimator has only little information at hand. In all plots, the flows exhibit no distinct bias in that they are scattered unsystematically around the main diagonal. The reason for this is that route choice is the only behavioral degree of freedom: Every driver who bypasses the downtown area invariable drives through the "inverse" of that area, and vice versa, such that the flows can merely be reallocated among links.

The second column contains occupancy scatterplots. This type of data also defines the RMSA_x validation error. The degree of variability among different scenarios follows the same order as for the flows.[6] However, systematic differences between the scenarios can now be observed. Since the toll is not designed to maximize traffic throughput, it causes increased congestion outside the city center. This effect becomes more pronounced for smaller VOTs, which model a greater behavioral sensitivity to the toll. The nonlinear congestion effects are reflected in unsymmetrical plots: The positive effect of the toll (less vehicles downtown) is not as pronounced as its negative counterpart (more vehicles on the bypass roads). Such an effect can be justified if there are other motives than congestion relief for the introduction of the toll. One should keep in mind that this is a synthetic scenario with no ambition to evaluate road pricing strategies themselves.

5.2.2 Experimental Results

12 EUR/h is a reasonable a priori guess for an average VOT. The estimator therefore adjusts a 12 EUR/h VOT prior scenario to the reference measurements of a true no-toll scenario, a true 6 EUR/h VOT scenario, and a true 18 EUR/h VOT scenario. Every estimation run starts with a plans file that is

[6] A prominent outlier at coordinates (312/175) in the "6 vs 12 EUR/h VOT" occupancy scatterplot can be observed. This is the western segment of "Frankfurter Allee", leading immediately into the toll zone. It has 3 lanes and is almost 3 kilometers long. The lower the value of time the more drivers try to divert at at the downstream end of this road into the increasingly congested bypasses and cause the observed spillback.

drawn from the 12 EUR/h prior distribution. That is, in the absence of measurements, the estimator immediately draws from the prior, and if measurements are available, all transients towards the posterior can be unequivocally ascribed to the measurements. Experiments with various prior weights w_{prior} as defined in (5.4) are conducted in order to investigate the estimator's robustness against suboptimal parameter settings. Three estimation runs are evaluated in every configuration in order to increase the statistical reliability of the results.[7]

5.2.2.1 Description of Results

Figure 5.8 shows the resulting error measures over different w_{prior} values for sensor data generated from the 6 EUR/h VOT, the 18 EUR/h VOT, and the no-toll reference scenario. These settings are subsequently denoted as "no-toll estimation" and "6(18) EUR/h VOT estimation". Measurement errors RMSA_q are given in the first column and validation errors RMSA_x are shown in the second column. For comparison, error measures for the 12 EUR/h VOT reference simulation and for the additional three simulation runs conducted in the stability analysis of Section 5.2.1.1 are also given in each diagram. They are equivalent to running the estimator without sensor input. For ease of comparison, they are re-drawn over every considered w_{prior} value in red color. The three estimation results per w_{prior} value are drawn in blue. All experiments are run for 250 iterations. Flow and occupancy averaging is started after a settling time of 50 iterations.

All results are fairly stable in that there is limited variability among repeated runs. Often enough, the dots lie on top of each other and cannot be distinguished. Reproducible convergence is a desirable and not at all self-evident feature for a nonlinear estimator. In these experiments, it can be observed with good precision. However, this result is at least partially owed to the use of a representative measurement data set in all experiments for a particular true scenario. Another general observation is that the occupancy error levels are relatively small. This is a consequence of the network-wide point of view which accounts for many links in the periphery that are hardly affected by the toll.

The first column of Figure 5.8 shows that the measurement error RMSA_q decreases monotonously with w_{prior}. This is plausible: the smaller the belief in the behavioral model the more weight is put on measurement reproduction. The results differ in the previously hypothesized way in that a large difference between flows in the prior and the true scenario provides substantial information that can be facilitated for estimation, whereas smaller flow differences result in a less focused search: The 12 EUR/h VOT prior scenario is most different from the no-toll scenario, less different from the 6 EUR/h VOT scenario, and least different from the 18 EUR/h VOT scenario. Accordingly, the greatest estimation improvements over a plain simulation of the prior are 86%, 63%, and 58%, respectively.

[7] All results apart from the performance benchmarks of Section 5.3.3.3 are obtained on a computing cluster where the nodes are equipped with AMD 2.6 GHz Opteron processors and have at least 2 GB of RAM. On such a node, the computing time of an estimation run as described in this section is in the order of one day.

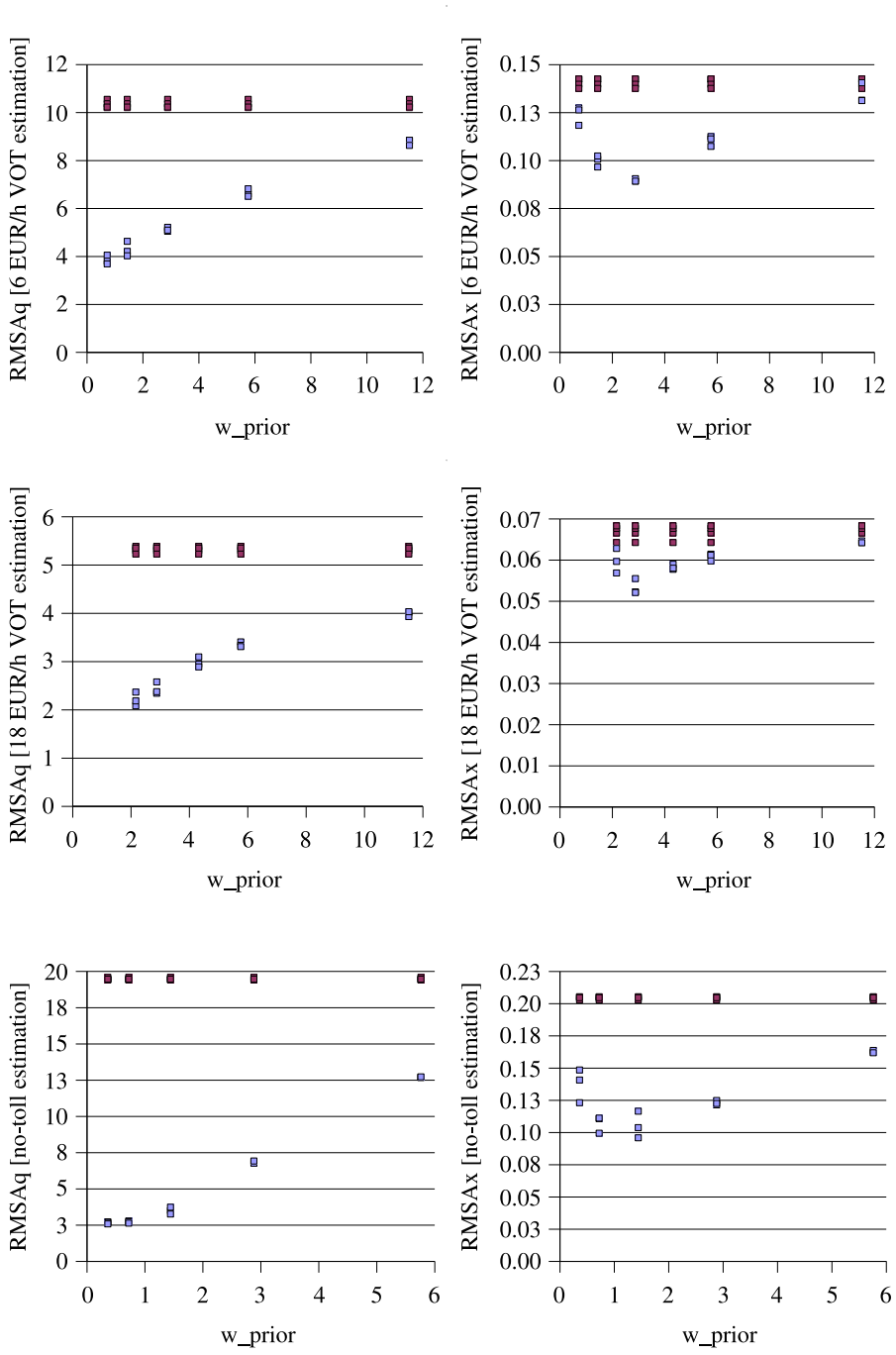


Figure 5.8: Result overview for planning experiments
The left column shows measurement errors RMSA_q and the right columns shows validation errors RMSA_x over different w_{prior} values for a true 6 EUR/h VOT scenario, a true 18 EUR/h VOT scenario, and a true no-toll scenario. The three estimation results per w_{prior} value are represented by blue dots. For comparison, the error measures for four plain simulations of the 12 EUR/h VOT prior scenario are represented by red dots. All experiments are run for 250 iterations. Flow and occupancy averaging started after a settling time of 50 iterations.

The second column of Figure 5.8 shows a non-monotonous relation between w_{prior} and the validation error RMSA_x. As w_{prior} grows, the measurements' influence vanishes and the estimation quality gracefully deteriorates towards that of a plain simulation . However, as w_{prior} decreases, a minimum value of RMSA_x is invariably encountered, after which a further decrease of w_{prior} results in an increased validation error. The attained minimum RMSA_x value reflects the estimator's ability to spatiotemporally extrapolate the available flow measurements. The RMSA_x improvements follow the same order as the RMSA_q results. When compared to the 12 EUR/h VOT prior scenario, the estimator achieves a 48% improvement for the true no-toll scenario at $w_{\text{prior}} = 0.72$ or 1.44, a 36% improvement for the true 6 EUR/h VOT scenario at $w_{\text{prior}} = 2.88$, and even for the subtle true 18 EUR/h VOT scenario a 20% improvement can be observed at $w_{\text{prior}} = 2.88$. The last improvement is particularly noteworthy since fairly little difference between the 12 and the 18 EUR/h VOT scenario can be identified in Figure 5.7 at all. This indicates that the estimator is quite precise in that it recognizes even such subtle differences. Recall that all of these extrapolation results are obtained using only 50 measurement locations out of altogether 2 459 links.

Figures 5.9 and 5.10 provide flow and occupancy scatterplots that result from the best configuration in each experimental setting. Here and subsequently, the "best" configuration corresponds to the w_{prior} value that yields the smallest validation error on average. From the according three estimation runs, the second best is chosen for illustration. The first column of each figure repeats the data obtained during the preparatory simulations, cf. Figure 5.7, and the second column shows the corresponding estimation results. All data points are averaged over many relaxed iterations such that all differences between left and right column can be ascribed to a systematic effect of the estimator. Overall, the visual impression affirms the quantitative error measures. Recall that the previously given RMSA_q values only account for the 50 sensor locations, whereas the flow scatterplots contain data points for all links in the network.

5.2.2.2 Discussion of Results

Three explanations can be given for the increased validation errors at small w_{prior} values. The first is over-fitting. Even if the representative measurements are not corrupted by sensor noise, their averaging may result in an inconsistency with the dynamics of the underlying nonlinear traffic flow model.[8] The second explanation is under-determinedness in combination with nonlinear dynamics. There may be many global traffic situations that reproduce the measurements equally well. As the behavioral model's effect vanishes with decreasing w_{prior}, insufficient behavioral information is available as a guidance towards a plausible solution, and the estimator gets locally stuck. This effect is possible even though the flow sensors provide supplementary information about free and congested traffic conditions since this data is still insufficient to uniquely define the traffic conditions in the further surroundings of a sensor. Finally, a small w_{prior} effectively acts like a large gain on the log-likelihood function, and the steepness of this function can have a negative effect on the convergence of the underlying

[8] The recent experimental results confirm this hypothesis.

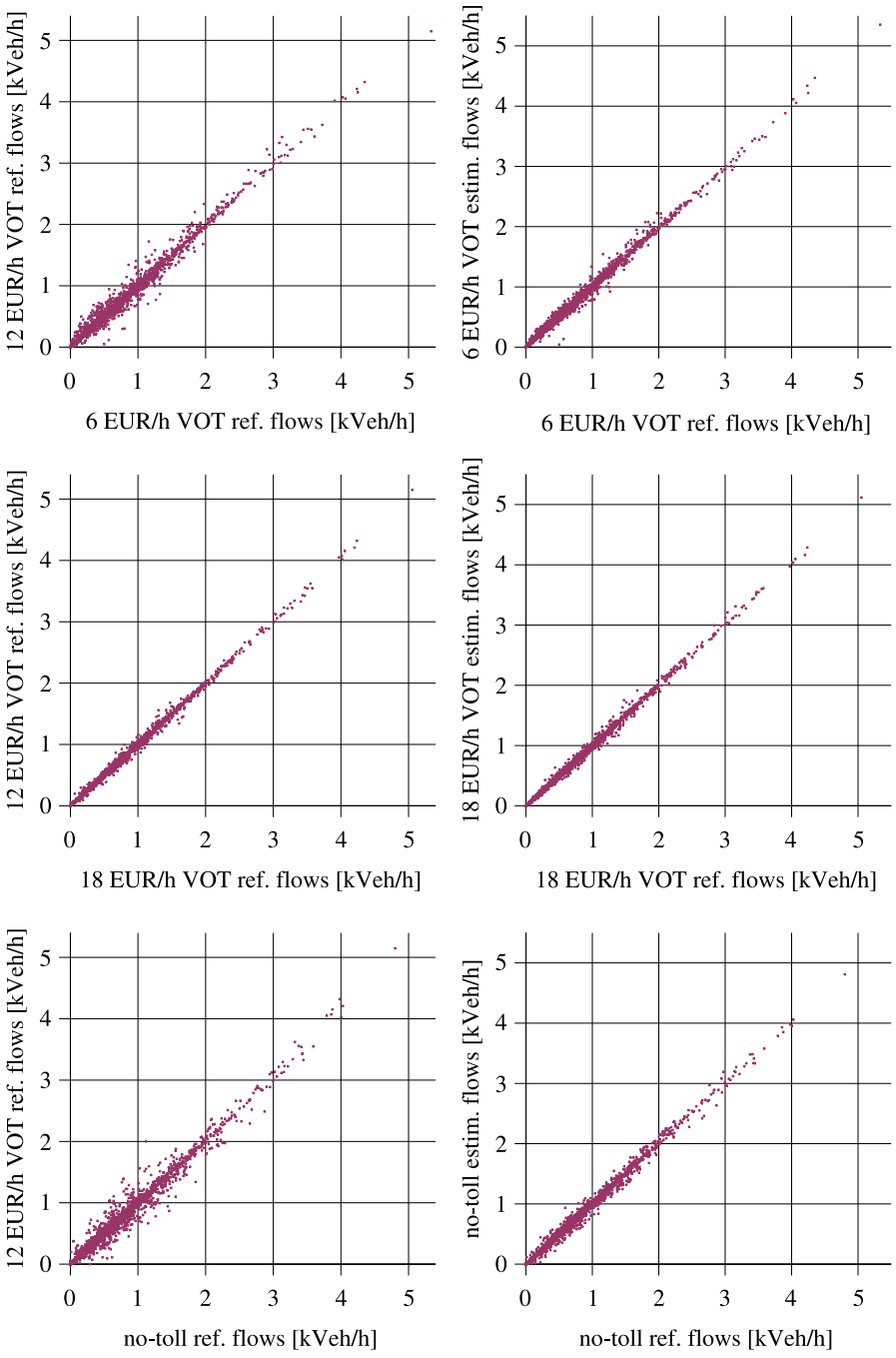


Figure 5.9: Comparison of true and estimated flows (planning)
The first column repeats the preparatory flow scatterplots of Figure 5.7. The second column shows the according estimation results where the reference flows (on the abscissa) are compared to their estimated counterparts (on the ordinate). That is, every row contains one scatterplot that compares a particular true scenario to the prior scenario, and it contains another scatterplot that compares the true scenario to the estimation result. These plots already represent average values such that all differences between left and right column can be ascribed to a systematic effect of the estimator.

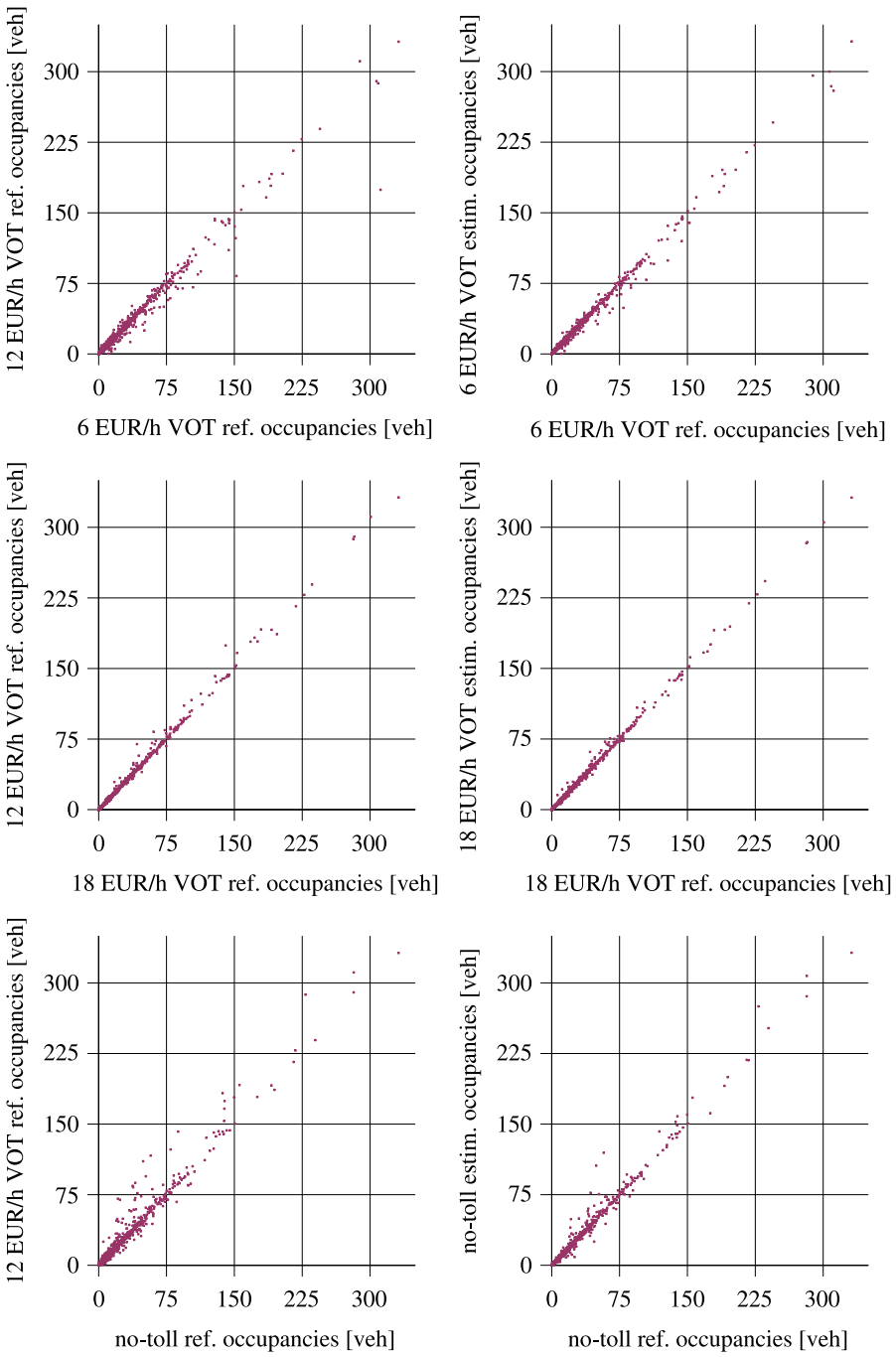


Figure 5.10: Comparison of true and estimated occupancies (planning)
The first column repeats the preparatory occupancy scatterplots of Figure 5.7. The second column shows the according estimation results where the reference occupancies (on the abscissa) are compared to their estimated counterparts (on the ordinate). See Figure 5.9 for further explanations.

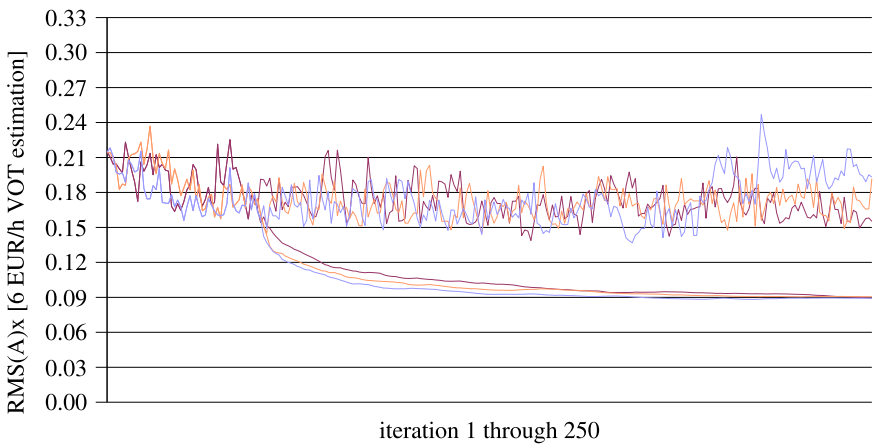


Figure 5.11: RMS_x and RMSA_x [6 EUR/h VOT estimation]
Validation errors over 250 iterations for the three best experiments with a true 6 EUR/h VOT scenario. RMS_x effectively represents the Euclidean distance of the 6 EUR/h VOT reference occupancies to the estimation results of a particular iteration. The recursive state averaging is turned on after 50 iterations such that a smooth RMSA_x curve branches off each RMS_x curve.

SA fixed point search algorithm. In either case, a trustworthy behavioral model that calls for a sufficiently large w_{prior} avoids the problem.

Rephrasing this observation in more general terms, a good state reproduction depends crucially on data and modeling quality, which cannot be compensated for by the estimation logic itself. The measurements need to contain sufficient information for a spatiotemporal extrapolation, and the behavioral simulator must be structurally correct in that it generates choices that are compatible with the measurements.

Overall, the achieved measures of estimation quality must be considered in light of the idealized setting in which they were obtained. The use of representative measurement data that is free of sensor errors is an idealization. In a real-world application, the over-fitting of certainly existing measurement errors must be avoided. This is likely to require larger w_{prior} values than used here and would consequently yield a reduced measurement and validation data fit. However, it can be concluded that the estimator performs structurally correct and that the estimation results in a specific application will mainly depend on the available data and modeling quality.

5.2.2.3 Estimation Dynamics

Finally, a closer look at the estimation dynamics is provided in Figures 5.11 through 5.13 for the 6 EUR/h estimation, the 18 EUR/h estimation, and the no-toll estimation. Each figure shows all three RMS_x and RMSA_x trajectories for the respective best w_{prior} configuration over 250 iterations. Most RMS_x trajectories oscillate fairly stable in the temporally auto-correlated manner known

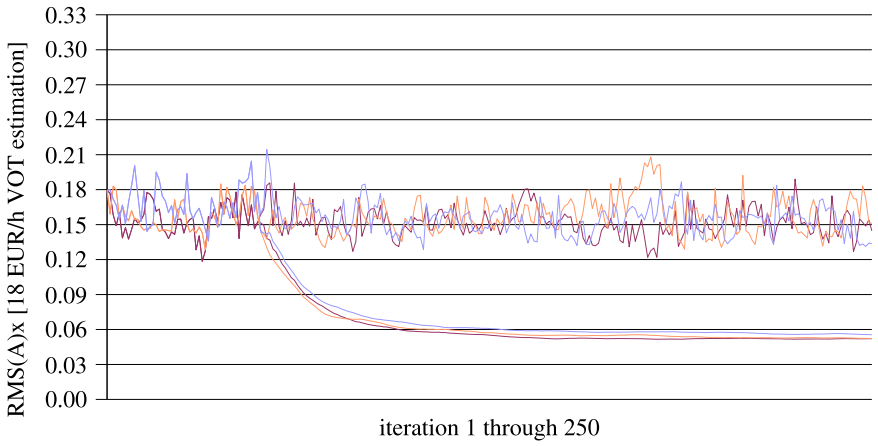


Figure 5.12: RMS_x and RMSA_x [18 EUR/h VOT estimation]
Validation errors over 250 iterations for the three best experiments with a true 18 EUR/h VOT scenario. See Figure 5.11 for further explanations.

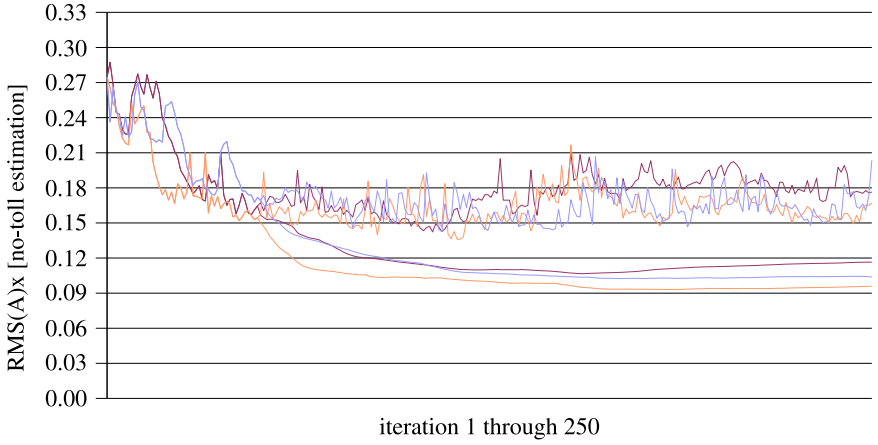


Figure 5.13: RMS_x and RMSA_x [no-toll estimation]
Validation errors over 250 iterations for the three best experiments with a true no-toll scenario. See Figure 5.11 for further explanations.

from the preparatory simulation runs. The eventual outliers, particularly the blue curve in Figure 5.11, may be due to a yet imperfectly relaxed posterior distribution. However, similar periods of "disarranged" dynamics can also be found in the preparatory simulations, where no estimation was involved.

All RMSA_x curves stabilize well in the available 250 iterations. Their speed and reliability of convergence increases as the prior and the true scenario become more similar. The 18 EUR/h VOT estimation converges fastest, the 6 EUR/h VOT estimation is somewhat slower yet still very reliable, and the no-toll estimation exhibits the least consistent convergence behavior. This may result from the fact that the more distant prior and true scenario are the longer the estimator's way through state space becomes. In nonlinear conditions, the chance of "branching off" towards different local solutions is likely to increase as this way gets longer.

Altogether, the estimator consistently generates distinct state reconstruction improvements. It extracts the relevant information out of limited flow measurements even for very subtle differences between prior and true scenario. Its ability to function in the planning-like setting given here shows its applicability in conjunction with a non-deterministic, equilibrium-based dynamic traffic simulator.

5.3 Telematics Experiments (Non-Equilibrium Situation)

The second half of this chapter applies the proposed estimator in conjunction with a telematics model that replaces the hitherto assumed SUE conditions by an assumption of imperfectly informed drivers. This has a significant influence on the traffic conditions when compared to the planning scenario, and the estimator has, even under strict running time constraints, a substantially more distinct effect in this setting.

Experiments are conducted in offline and simulated online conditions, cf. Section 1.1.3. In offline conditions, a set of beforehand collected measurement data is processed "en block". In a telematics context, this is useful for the ex post analysis of a particular day. The online estimator runs in a rolling horizon mode where the estimation of the traffic state for a certain point in time has only measurements from earlier times available. This setting is characteristic for a continuous traffic monitoring problem. The experiments in simulated online conditions allow to investigate the estimator's real time capabilities and to conclude about the scenario size its current implementation can handle.

5.3.1 Rolling Horizon Estimation

A rolling horizon logic is implemented that runs the estimator in simulated online conditions. The time period of investigation still is 6 to 9 am. While one iteration of an offline estimator facilitates all measurements from this interval at once, online conditions imply that the measurements become available bit by bit as the simulated real time proceeds.

The online estimation starts at 6:30 simulated real time. Only measurements until this moment are available. The estimator iteratively adjusts the simulated driver behavior to these measurements according to the by now established estimation logic of Section 5.1.2.2. During this first **estimation period**, only a simulation from 6:00 to 6:30 is iteratively adjusted. After a prespecified number of iterations, the simulated real time is advanced to 6:35, the most recent simulation is continued until 7:00 to evaluate the estimator's predictive capabilities, the measurements from 6:30 to 6:35 become available, and the next estimation period from 6:05 to 6:35 begins. All driver behavior until 6:05 is now fixed according to the last iteration of the previous estimation period.

It is noteworthy that such a simulation logic is attractive not only for telematics purposes in online conditions. Being able to iterate critical time intervals more frequently than others allows to deploy computational resources in a more focused way. This also appears useful during the first iterations of a planning simulation where the system is far away from an equilibrium. An eventual sequence of "full" planning iterations eliminates the accrued tendency of local convergence. The danger of imperfect convergence also needs to be accounted for in online estimation and calls for the more elaborate discussion given next.

In rolling horizon estimation, behavior is adjusted only within a limited estimation period that ends at or shortly before the current point in time. As time proceeds, this estimation period is also shifted. In the subsequent period, all driver decisions that have fallen out of the estimation time window are kept fixed at their last values. This is necessitated by the estimation window's constant length, which in turn is enforced by the real time requirement of a constant calculation time per estimation period. Since the estimator continues to adjust behavior to measurements, it may change agent decisions within the given estimation period in an attempt to compensate for imperfect estimates at earlier times.

The problem of suboptimal rolling horizon estimation has already been investigated for traffic monitoring problems with aggregate models [23]. Since an individual-level analysis is pursued here, a behaviorally more descriptive point of view is adopted. The question arises to what degree it is feasible to substitute the behavior of different travelers when matching sensor data without accumulating incorrect behavioral estimates from one estimation period to the next. Feasibility is not to be confused with individual-level realism – no real traveler accounts for what others do and compares it to traffic counts. It rather means that the clearly suboptimal behavioral predictions for agents that compensate for imperfect estimates of earlier periods still result in future traffic conditions that are more realistic than an a priori guess without estimation. For example, distorting the behavior of a few travelers at a critical time and location in the network might prevent an unrealistic gridlock in the simulation. This also prevents the likewise unrealistic reactions of many other agents to this gridlock. In consequence, agents that replan in later estimation periods do so in more realistic conditions and thus with more realistic results – even if no measurements are accounted for in these later periods.

It is worthwhile to adopt a more formal view on this matter. The behavioral posterior

$$P(\mathcal{U}_1 \dots \mathcal{U}_N | \mathcal{Y}) \propto l(\mathcal{U}_1 \dots \mathcal{U}_N | \mathcal{Y}) P(\mathcal{U}_1 \dots \mathcal{U}_N) \tag{5.8}$$

differs from its prior $P(\mathcal{U}_1 \ldots \mathcal{U}_N)$ only because of the information contained in the measurement likelihood $l(\mathcal{U}_1 \ldots \mathcal{U}_N | \mathcal{Y})$, cf. (4.19) and (4.24). Fixing the behavior of some agents at unreasonable values degrades the estimation quality by means of this likelihood.

This effect can be substantially mildened by the behavioral simulator itself. A chameleonic behavioral prior that admits even highly unrealistic actions with a low yet non-zero probability is likely to be inapplicable in conjunction with a suboptimal estimator. If, in suboptimal conditions, the likelihood is badly approximated, the choice probabilities of implausible actions may be excessively increased. However, if the behavioral model simply does not generate implausible actions, i.e., if implausible choices are selected with zero probability, no Bayesian estimator can ever generate a positive choice probability by mere multiplication in fundamental relation (5.8). The behavioral model placeholder used here is robust in this regard since it generates alternative routes only based on reasonable VOT variations. Its simplicity prevents it from ever generating a "strange" route that may even be selected during estimation because of a poor likelihood approximation.

A computational implication of these observations relates to the fact that the estimator linearizes the log-likelihood. If the likelihood is imprecise, there is little meaning in running a large number of iterations per estimation period in order to finally draw from a posterior that is based on an utmost precise linearization of the according log-likelihood. The experiments of Section 5.3.3.2 provide more insight into this issue.

5.3.2 Scenario Generation

5.3.2.1 Simulation of Imperfectly Informed Drivers

The first day after the implementation of the toll is simulated. In this setting, drivers are aware of typical travel times without toll and of the toll itself. However, they have not yet learned the alterations in traffic conditions that result from other travelers' changed behavior in response to the toll. Suchlike imperfectly informed drivers are simulated in the following way.

1. A planning simulation without toll is run. When the simulation attains relaxed conditions, time-dependent travel times for all links are written to file over a long sequence of iterations. The travel time distribution captured by these files is used in all subsequent experiments as a representation of drivers' **memory** of the no-toll situation.

2. When running the telematics simulation, this sequence of files is provided to pre-trip replanning travelers instead of the last iteration's travel times. The travelers base their routing decisions on this memory, plus the (known) toll. This allows to run the simulation in an iterative manner and to maintain variability in the traffic conditions while avoiding a learning effect that results if actually simulated travel times are fed back for replanning.

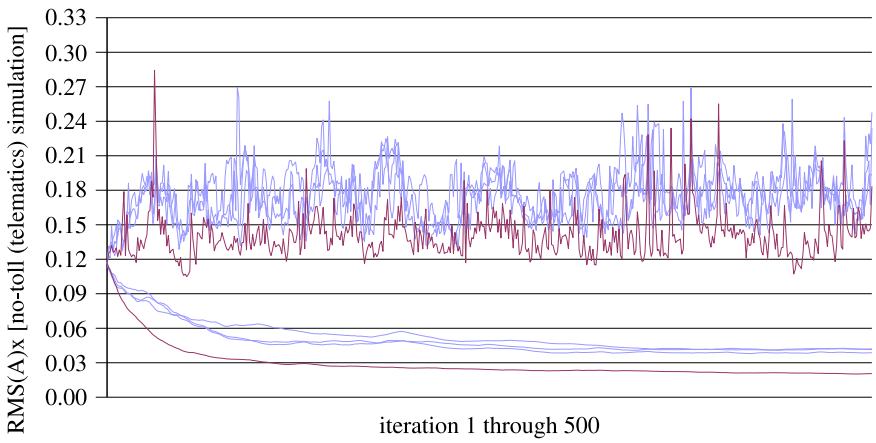


Figure 5.14: RMS(A)$_x$ [no-toll planning/telematics simulation]
The red curves show RMS(A)$_x$ [no-toll planning simulation] and the blue curves show RSM(A)$_x$ [no-toll telematics simulation] over 500 iterations. The validation data from the no-toll reference planning scenario is used as reference data in all error measures. Since the simulations start with an already relaxed plans file, the recursive state averaging is turned on from the very first.

For estimation, the overall logic of Section 5.1.2.2 is maintained, only that replanning is now based on the previously generated driver memory. The only structural difference between a prior and a true telematics scenario is a different VOT. Since every estimation starts with a plans file that is drawn from a stable simulation of its respective prior scenario, all transients during estimation reflect the transition from the prior to the estimated posterior distribution.

5.3.2.2 Investigation of Scenario Stability

Figure 5.14 shows, in red color, the RMS$_x$ and RMSA$_x$ curves for 500 iterations of a planning simulation in the no-toll case when compared to the reference data for that scenario. Since these iterations start from an already relaxed plans file, the recursive state averaging is turned on from the very first. Three further curve pairs are drawn in blue. They result from an identical set-up as the first run, only that the travel times on which replanning is based are now taken from the memory files that were written during the first simulation.

Using the memory files results in an increased variability of the traffic conditions. This can be seen from the greater variability of the blue RMSA$_x$ curves, which indicates that the network states are drawn from a wider distribution than in the initial simulation. The higher overall levels of the blue RMSA$_x$ curves also show that a moderate additional error is introduced. The higher level of the blue RMS$_x$ curves results from the combination of both effects. However, all blue RMS$_x$ curves exhibit a similar structure. This shows that, even if the telematics logic has a side effect on the simulation dynamics, this effect is fairly stable.

The source of the difference between the original simulation and the telematics simulations is that the replanning agents are selected at random in every

iteration. That is, even if the available information itself is identical in all simulations, different travelers at different locations and with different destinations react to it. The resulting deviations in the traffic conditions are not accounted for by the replanning agents. This can be seen as an increased perceptional error, which, in the given setting, also increases the variability of the resulting traffic conditions.

5.3.2.3 Measurement and Validation Data Generation

The previous section shows that the dynamics of telematics simulations are even less well-behaved than their planning counterparts such that the argumentation of Section 5.2.1.2 applies here with even stronger emphasis.

Consequently, representative measurement and validation data sets are again generated by averaging. That is, a telematics **reference simulation** is run for the no-toll scenario and for the 6,12, and 18 EUR/h VOT scenario.[9] Flows and occupancies are averaged over 400 stable iterations of each simulation. These average values constitute the measurement and validation data sets used as the reference data in all subsequent evaluations and RMS(A) error measures.

There is a conceptual difference in the validation of a planning and a telematics estimator. In a planning application, the goal is to estimate a posterior that is similar to the true *distribution* of traffic states (from which a draw is realized every day). In a telematics setting, reality consists of a single day only. Consequently, a telematics posterior must represent the knowledge about a single *realization* of traffic conditions only. This difference is disregarded in the simplified setting considered here since only a single, representative reference data set is used to validate the planning and the telematics estimator respectively.

5.3.2.4 Comparison of Scenarios

Figure 5.15 compares flows and occupancies of the 12 EUR/h VOT (telematics) reference simulation to the 6 EUR/h VOT reference simulation, the 18 EUR/h VOT reference simulation, and the no-toll reference simulation. Again, all data points are 400-iteration averages, and, again, they apply to the simulated time interval from 8:30 until 8:35 am.

The 12 EUR/h VOT scenario deviates remarkably from the no-toll scenario but does not differ much from the other simulations with a non-zero toll. This is a result of the lacking equilibrium assumption: At the first day of the toll's implementation, the presumably most advantageous route choice for most drivers that so far have traversed the toll area is now to avoid it but to bypass it as sharply as possible in order to minimize the increase in travel time. This, however, causes an unforeseeable congestion on the roads that immediately encircle the toll zone. The no-toll scenario is the only scenario in which this congestion does not occur.

[9] The no-toll telematics reference simulation differs somewhat from the no-toll planning reference simulation because of the file-based driver memory in the telematics simulation logic.

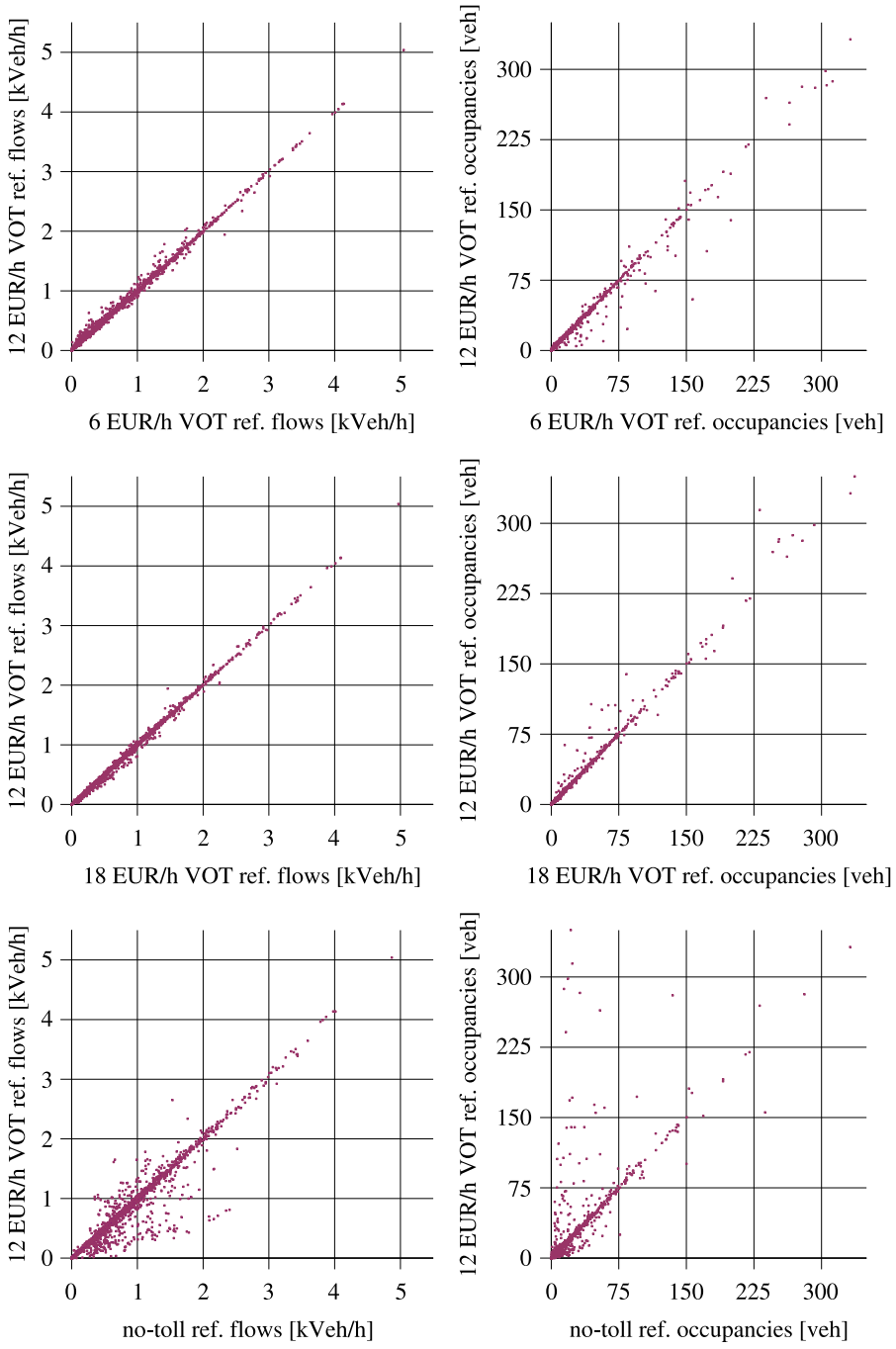


Figure 5.15: Scatterplots for comparison of telematics reference simulations
The scatterplots compare data from the 12 EUR/h VOT telematics reference simulation (on the ordinate) to the other telematics reference simulations (on the abscissa). The first column compares flow rates and the second column compares occupancies. All scatterplots contain data points for *all* links in the network. The data points apply to the simulation time interval from 8:30 to 8:35. All data points represent average values over 400 iterations.

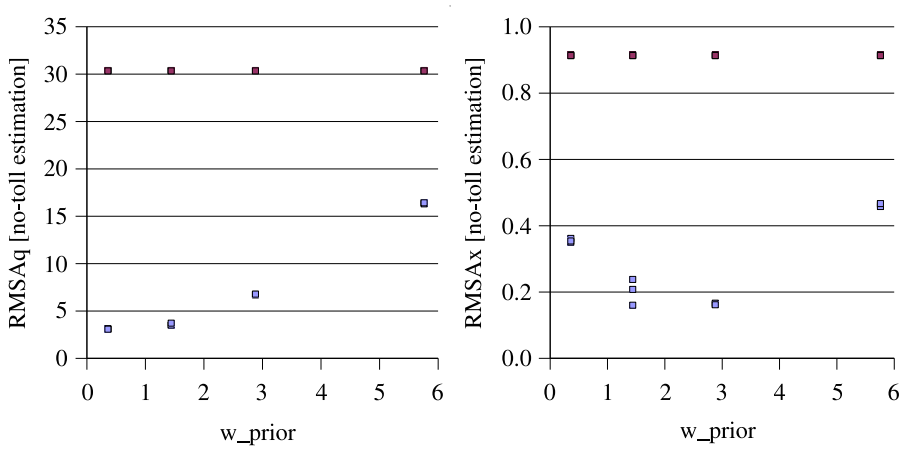


Figure 5.16: Result overview for telematics offline experiments
The left diagram shows measurement errors RMSA_q and the right diagram shows validation errors RMSA_x over different w_{prior} values for a 12 EUR/h VOT prior scenario and a true no-toll scenario. The three estimation errors per w_{prior} value are represented by blue dots. For comparison, the error measures for three plain simulations of the prior scenario are represented by red dots. All experiments are run for 250 iterations. Flow and occupancy averaging is started after a settling time of 50 iterations.

Since the estimator's ability to track rather subtle deviations is already demonstrated in the planning experiments, only the no-toll scenario is subsequently used as the synthetic reality. This implies that the real drivers effectively ignore the toll's effect. Keeping in mind that only the first day after the installation of the toll is simulated, such a behavior may either result from unawareness or from curiosity about the involved technical installations. Again, the purpose of these experiments is to sound the capabilities of the estimator, not to discuss road pricing issues themselves.[10]

5.3.3 Experimental Results

In all telematics experiments, the estimator adjusts a 12 EUR/h VOT prior scenario to measurements that are obtained from a true no-toll scenario.

5.3.3.1 Offline Estimation

To begin with, the rolling horizon mode is not facilitated and a sequence of offline estimations is run over the entire 6 to 9 am time period. Figure 5.16 shows the resulting error measures over different w_{prior} values. The measurement error RMSA_q is given on the left, and the validation error RMSA_x is given on

[10] The recent experimental results indicate that the estimator works equally well if the prior scenario and the synthetic reality are exchanged. Such a setting, where the real reaction to the toll is much stronger that a priori expected, could result from an overreaction of the drivers to the toll.

the right. The results of three plain simulation runs of the 12 EUR/h VOT prior scenario are represented by red dots, and the three estimation results per w_{prior} value are drawn in blue. All experiments are run for 250 iterations. The recursive state averaging turned on after a settling period of 50 iterations.

Both, the simulation and the estimation results are very stable; most dots lie on top of each other. This even greater stability than in the planning case despite of the greater difference between the prior and the true scenario is ascribed to the simpler simulation logic that now dispenses with the equilibrium-generating travel time feedback between subsequent iterations. The estimator generates remarkable improvements. For $w_{\text{prior}} = 2.88$, it improves RMSA_q by 78% and RMSA_x by 82% over a plain simulation of the prior scenario. The severe congestion of the 12 EUR/h VOT prior scenario that does not occur in the simulated reality is successfully prevented by the estimator. The flows and occupancies of the best estimation run (selected according to the same criterion as in the planning experiments) are opposed to the reference data for the true scenario in the scatterplots of Figure 5.17. Since these data points are averaged over many iterations, their differences leave no doubt about the estimator's systematic and beneficial influence.[11]

5.3.3.2 Online Estimation in Rolling Horizon Mode

The same estimation problem as before is now tackled in rolling horizon mode. With a real-time application in mind, an evaluation of the estimator performance in terms of average system states that are obtained over hundreds of iterations is now inappropriate. Therefore, only the RMS_x validation error is subsequently evaluated. A temporally disaggregate point of view is adopted by considering each estimation period individually. Predictive capabilities are also investigated.

A rolling horizon application challenges the estimator more than the previous offline experiments because of the different use of the travel time memory files. An identical memory file sequence is used for measurement generation and for offline estimation. The rolling horizon estimator still uses the same files but loads a new file in every iteration of every estimation period. Since these files are now applied in a temporal context that is different from the setting in which the measurements were generated, any advantage the estimator may have had during offline estimation is now precluded.

A prior weight of $w_{\text{prior}} = 2.88$ is maintained in all runs since this setting achieved the best results in the preparatory offline experiments. Figure 5.18 provides separate results for every 30-minute estimation period ending at 7 through 9 am. The blue bars represent (from left to right) the RMS_x validation errors obtained at the end of 5, 10, 20, 30, 40, and 50 iterations per estimation period. They are drawn on top of red validation error bars that result from plain rolling horizon simulations with respective iteration numbers. These simulations follow an identical logic as the estimator, only that the measurements are not accounted for.

[11] Results of comparable quality were recently obtained in a setting where the sensor data is not averaged over many iterations but where it is taken from a single iteration of the telematics simulation that generates the synthetic reality.

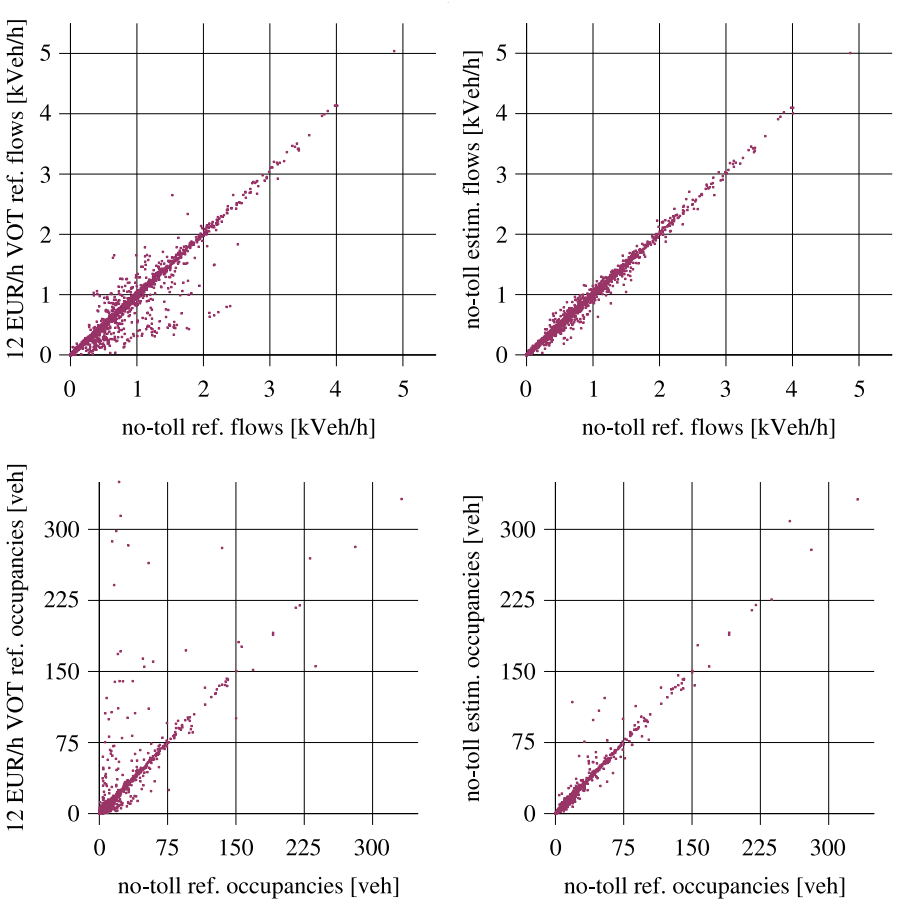


Figure 5.17: Comparison of true and estimated flows/occupancies (telematics) The first row contains flow scatterplots, and the second row shows occupancy scatterplots. The first column repeats the "no-toll vs. 12 EUR/h VOT" scatterplots of Figure 5.15. The second column shows the according estimation results where the reference data (on the abscissa) is compared to its estimated counterpart (on the ordinate). That is, every row contains one scatterplot that compares the true no-toll scenario to the 12 EUR/h VOT prior scenario, and it contains another scatterplot that compares the true scenario to the estimation result. These plots already represent average values such that all difference between left and right column can be ascribed to a systematic effect of the estimator.

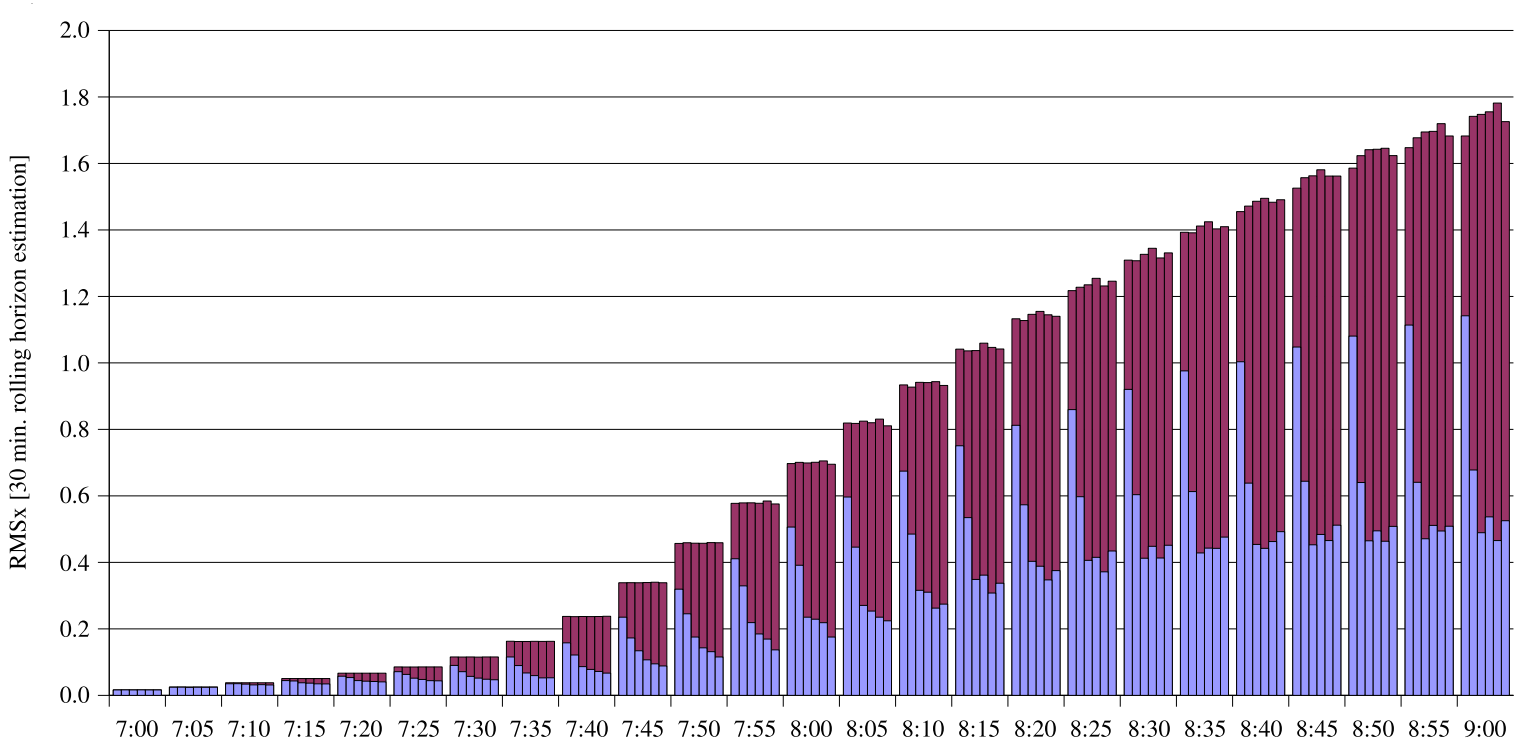


Figure 5.18: RMS_x [30 min. rolling horizon estimation]
The blue bars represent (from left to right) validation error measures RMS_x obtained after 5, 10, 20, 30, 40, and 50 iterations per estimation period. They are drawn on top of red error bars that result from plain rolling horizon simulations with respective iteration numbers.

The estimation and simulation errors rise over time as the traffic volumes increase in the morning rush hour. The plain simulation errors do not systematically depend on the number of iterations since the deployed initial plans file already results from a stable telematics simulation. A pronounced difference between simulation and estimation can be observed as the congestion around the toll zone becomes severe in the prior scenario. Overall, the estimator reduces RMS_x by up to 70% in the later periods. Conducting only 5 or 10 iterations per estimation period results in lower improvements when compared to 20 iterations and more. However, running beyond 20 iterations yields only marginal improvements.

Figure 5.19 shows the same setup of validation errors as before, only that now the average prediction errors over a 0 to 30 minute time interval are given. This and the previous diagram match temporally in the following way: An estimation error drawn, e.g., over the 8:30 label is generated at this particular time and thus applies to the interval from 8:00 to 8:30. A prediction result that is drawn over the 8:30 label is generated at 8:00 for a 30 minute prediction window and consequently applies to the same interval. A comparison of both figures yields the expected diagnosis that the estimation quality is generally higher than the prediction quality. However, an estimation-based prediction is clearly better than a plain simulation. Again, the prediction results for 5 and 10 iterations per estimation period are inferior when compared to those with 20 iterations and more. The computational effort of executing more than 20 iterations per estimation period does not result in significantly improved predictions. Overall, the estimator reduces the RMS_x prediction error by 50% to 60% in the later time periods.

Figures 5.20, 5.21, and 5.22 provide separate RMS_x plots for the prediction intervals from 5 to 10, 15 to 20, and 25 to 30 minutes ahead in time. Here, the time labels simply indicate when the prediction is made. The quality deteriorates gracefully as the prediction time increases, starting from a 60% to 65% improvement for 5 to 10 minutes, attaining 55% to 60% for 15 to 20 minutes, and yielding around 50% even for the 25 to 30 minute prediction. This remarkably sustained improvement can be traced back to the rather restricted behavioral degrees of freedom a simulated traveler faces. It also benefits from the fact that only pre-trip replanning is accounted for such that a once estimated decision is maintained for the entire duration of a trip. Finally, the deterministic traffic dynamics certainly have a positive influence on predictability. However, even after all these words of reservation, the results show clearly that a rolling horizon estimation and prediction for this particular scenario is near-optimal if 20 iterations per 5-minute estimation period are allowed for.

5.3.3.3 Computational Performance

The current implementation of the estimator accomplishes 6 iterations per 5-minute interval in the given scenario. That is, near-optimal results require another estimation speedup of 3 to 4. Given the considered problem's size, this is an encouraging result. After all, one iteration consists of a 30 minute traffic simulation during the morning rush hour, comprises a behavioral model that relies on time-dependent best path calculations, and conducts a complete

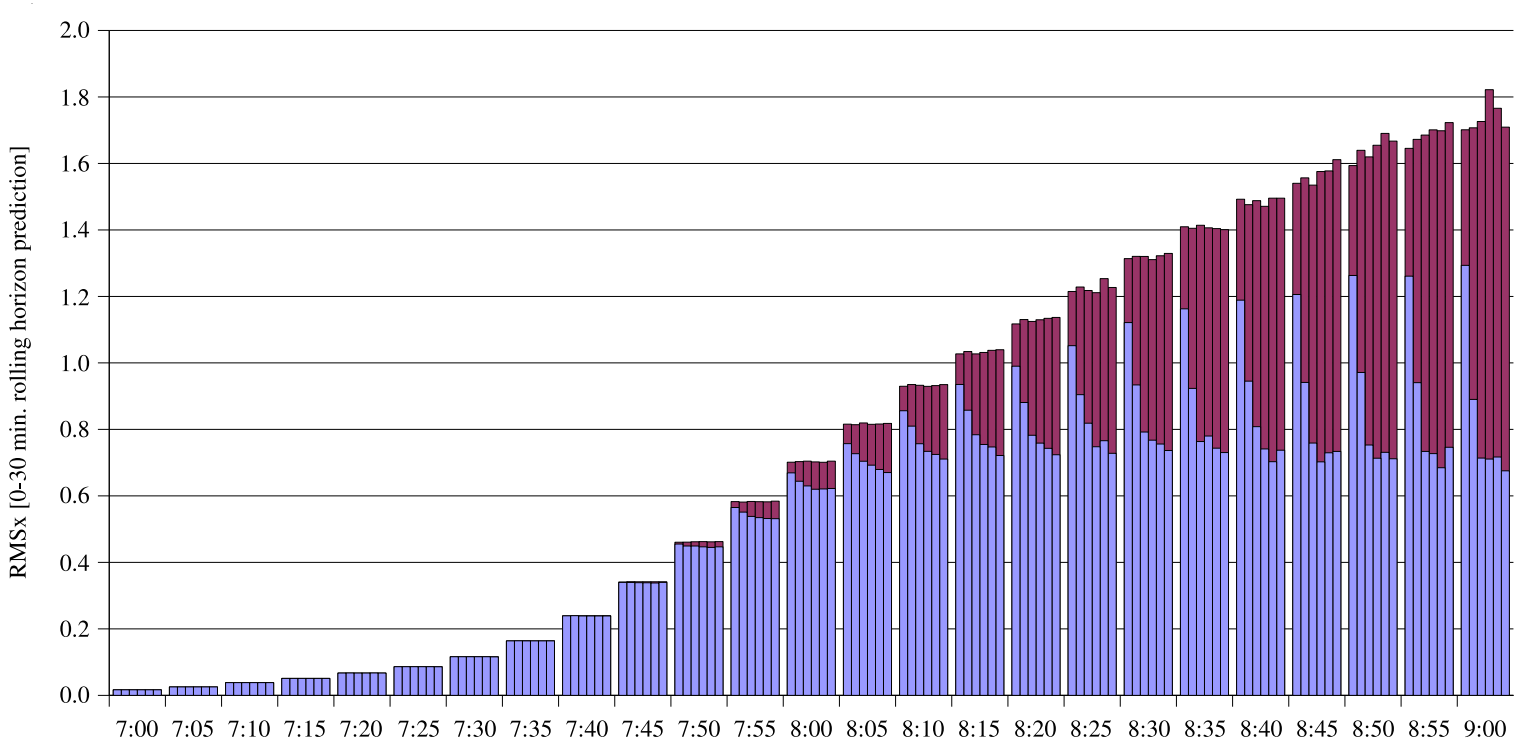


Figure 5.19: RMS_x [0-30 min. rolling horizon prediction]
The blue bars represent (from left to right) 0-30 minute prediction error measures RMS_x obtained after 5, 10, 20, 30, 40, and 50 iterations per estimation period. They are drawn on top of red error bars that result from plain rolling horizon simulations with respective iteration numbers.

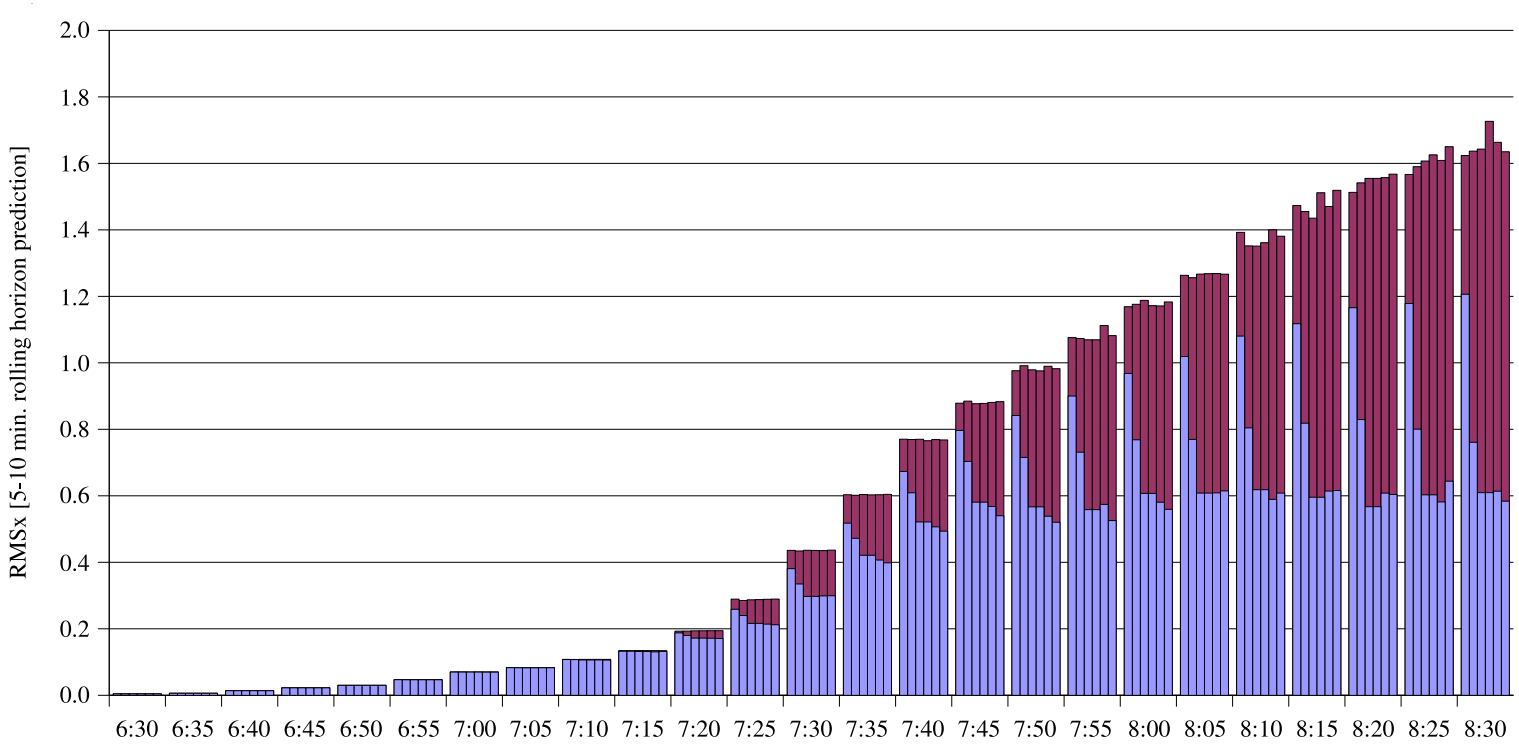


Figure 5.20: RMS_x [5-10 min. rolling horizon prediction]
The blue bars represent (from left to right) 5-10 minute prediction error measures RMS_x obtained after 5, 10, 20, 30, 40, and 50 iterations per estimation period. They are drawn on top of red error bars that result from plain rolling horizon simulations with respective iteration numbers.

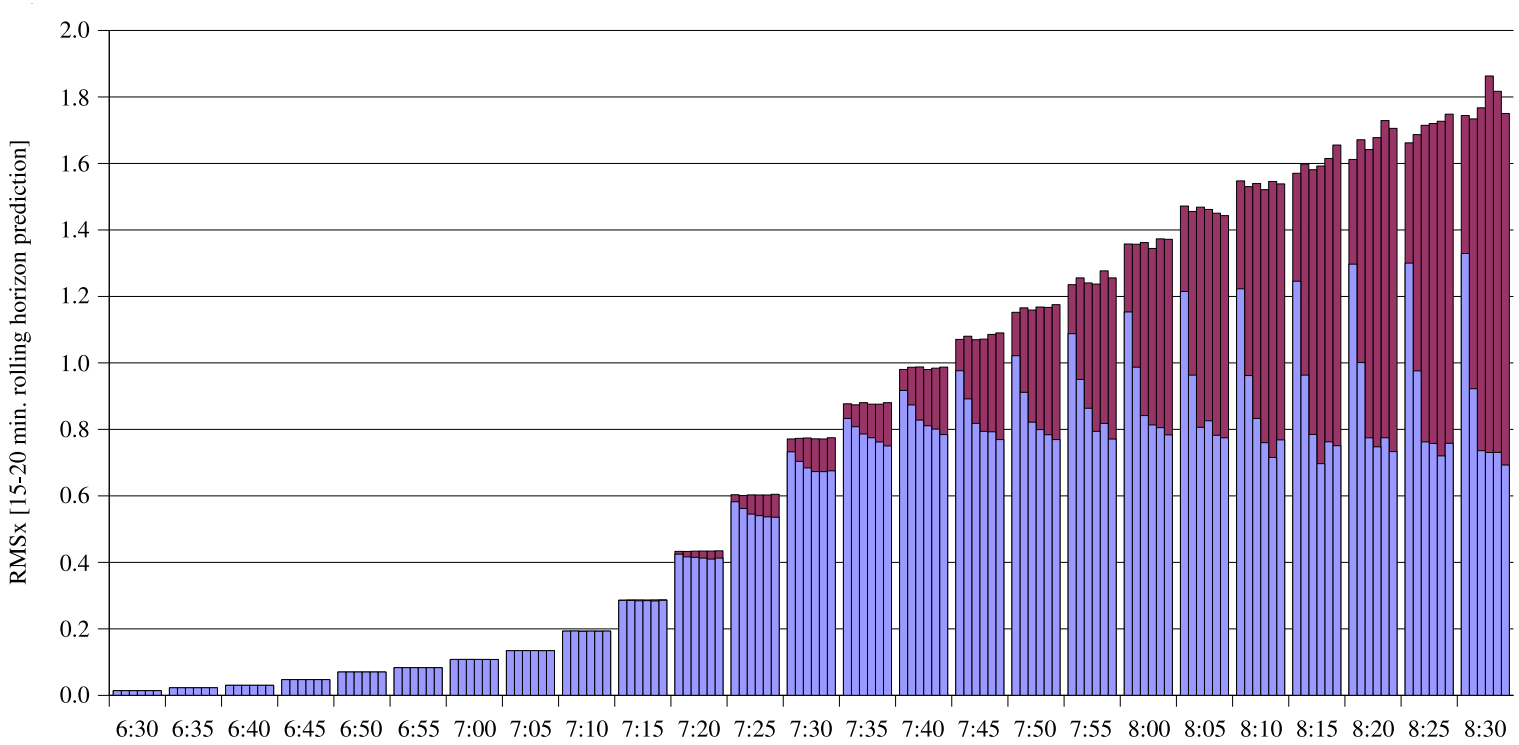


Figure 5.21: RMS_x [15-20 min. rolling horizon prediction]

The blue bars represent (from left to right) 15-20 minute prediction error measures RMS_x obtained after 5, 10, 20, 30, 40, and 50 iterations per estimation period. They are drawn on top of red error bars that result from plain rolling horizon simulations with respective iteration numbers.

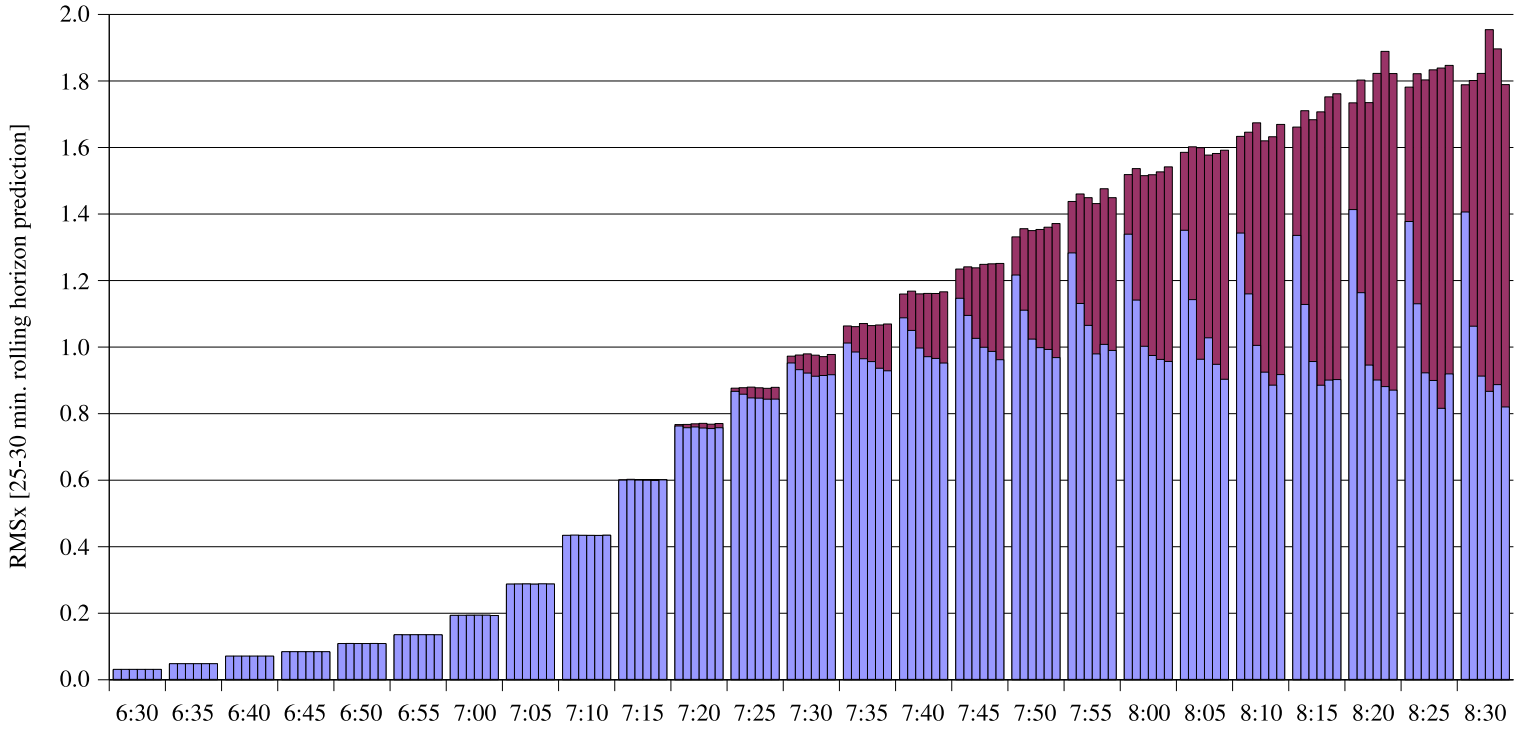


Figure 5.22: RMS_x [25-30 min. rolling horizon prediction]

The blue bars represent (from left to right) 25-30 minute prediction error measures RMS_x obtained after 5, 10, 20, 30, 40, and 50 iterations per estimation period. They are drawn on top of red error bars that result from plain rolling horizon simulations with respective iteration numbers.

spatiotemporal linearization of the resulting traffic dynamics. Even with only 6 iterations per 5 minutes, the estimator yields substantial improvements when compared to the prior scenario, which, however, is likely to benefit from the simple behavioral model as explained in Section 5.3.1.[12]

The computing times are obtained on a 3.2 GHz Pentium 4 stand-alone machine with 2 GB of RAM. File i/o constitutes a major bottleneck in the currently single-threaded implementation of the estimator. A large fraction of this file i/o results from the necessity to calculate sensitivities of macroscopic system dynamics backwards through simulated time, cf. Section 4.1.2. This requires to store all macroscopic states during the simulation and to process them backwards during the linearization. Even if the sparsity of this data because of the simulation scheme on variable time scales is accounted for, cf. Section 2.5, this adds up to 3.2 MB of binary data per minute of simulation. Since the resulting 4 608 MB for a whole day exceed the available RAM of most machines deployed in this work, the data is written to hard disk in 5-minute chunks of 16 MB during the simulation. These files are then reloaded for the linearization. This allows to estimate the given scenario on a machine with 2 GB of memory. However, for a limited estimation period of only 30 minutes, the data could be kept in RAM as well. Therefore, the approximate 25% of running time that are spent waiting for file i/o are omitted when measuring the estimator's computational performance.

Altogether, the estimator achieves significant improvements in a telematics setting. Even if the available scenario is somewhat too large to allow for near-optimal results in real-time conditions, feasible problems have the same order of magnitude: Since the computational effort rises at least linearly with the network and population size, a 600+ link scenario with 50 000+ agents is immediately approachable by the current implementation in real time.[13] A more extensive preprocessing of the Berlin network illustrated in Figure 2.8 that merges the many detailed intersections into single nodes might already suffice to run this very scenario in real-time.

5.4 Further Discussion

The demonstrated estimator does not depend on a choice set enumeration. This suggests its application for choice set generation itself. Since only best path calculations are used in the present example, why not run these calculations directly based on the modified utilities instead of first making a well-informed guess about possible routing alternatives and only then choosing a route based on these modifications? To make a long story short: Choice set generation is a modeling problem, and traffic counts alone do not provide sufficient information to substitute for the structural information contained in such a model. However,

[12] The recent experiments in which the sensor data is not averaged over many iterations converge in roughly half as many iterations but stabilize at somewhat higher error levels. Apparently, the estimator spends significant amounts of time in the experiments given here trying to extrapolate contradictory measurements that result from the averaging over many iterations.

[13] The recent results allow for a 1 200+ link scenario with 100 000+ agents.

this neither implies that traffic counts are useless for choice set generation nor that the proposed estimator is categorically unsuited for this purpose.

The considered behavioral model generates its choice set by running a best path algorithm that minimizes travel times which are generated by the mobility simulation. These travel times exhibit a particular correlation structure that results from the simulated traffic dynamics. This very property enables the generation of variable routes only based on best path calculations without ever resorting to the explicit simulation of a perceptional error by drawing from a multidimensional travel time distribution with an explicitly known covariance matrix.

In contrast, the estimator only disposes of local measurement information and processes this information in a likewise local (linearization-based) manner. If only few sensors are available, the measurement data is sparsely distributed over the network. In order to infer a driver's global utility perception from this information, a model is required that captures the network-wide correlation of travel times. In the given simulation system, this correlation is not accounted for by the time-dependent best path algorithm itself but results from the simulated travel times based on which this algorithm is run. If sparse utility corrections are added to these travel times during the choice set generation, routes result that locally account for the correction terms but globally still adhere to the correlation structure of the a priori assumed travel times. If suchlike generated routes differ sufficiently from those that actually caused the measurements, the estimator can only select among inappropriate prior routes and newly generated routes that are likewise unrealistically structured. The result is local convergence to a poor solution.

A visual inspection of routes that are generated based on estimated utility corrections has been conducted. Their interpretation is difficult since such routes invariably account for both travel times and utility corrections. However, a distinct increase in zig-zagging as one might expect in consequence of the local utility corrections cannot be observed. Still, even plausibly looking routes can consist of turning move sequences that are implausible given a certain correlation pattern of the travel times. Within the scope of this work, it is concluded that a more rigorous analysis of the simulation-based best-path route choice model itself is necessary before its implications for the estimation can be clarified. Recall that this particular model is only implemented as placeholder and that the estimator is not constrained to its deployment.

Again, the above discussion addresses a modeling problem. The estimator is not unable to provide useful information for choice set generation; it just is unable to solve the generally impossible task of inferring a network-wide utility pattern from arbitrarily few observations. If a model was at hand to meaningfully complete locally estimated utility corrections, choice set generation could be supported by measurements. This type of model would represent a rather common aspect of travelers' information processing. For example, a radio message regarding a single construction site is likely to motivate a driver to circumnavigate the surroundings of this site as well since experience teaches that the resulting obstructions are not concentrated at the single location indicated on the radio. That is, the driver is aware of correlations in the network conditions.

One might argue that full sensor coverage should allow for choice set generation

without further modeling support. However, this also would require to account for measurement correlations in the likelihood function. This is avoided here by choosing sparse sensor locations. Since travel times are one particular type of link-related measurements, the problem of correlation modeling would not be solved but only be shifted in a different context. In addition, full sensor coverage cannot be expected in real-world conditions.

A meaningful interpretation of the local utility corrections in the closing example of Chapter 4 was possible because of its simple structure. In the more general setting considered here, such an interpretation suffers from the same problems as the direct application of utility corrections for route generation: Every single turning move's utility correction is only meaningful given the behavioral model that is used for its identification. That is, the utility corrections are meaningful on route level – with the route generated based on simulated travel times with a particular correlation structure – but not necessarily on turning-move level. The behavioral model represents the global context that cannot be captured by local utility corrections. This interplay of modeling and estimation does not invalidate the estimator's ability to function with an arbitrary implementation of the behavioral simulator. It does, however, necessitate an interpretation of the estimation results in terms of the particular behavioral model based on which they are obtained.

Summarizing, this chapter demonstrates the proposed estimator's applicability in conjunction with a fully dynamical planning or telematics simulator and verifies its computational feasibility for a scenario of practically relevant size.

Chapter 6

Summary and Outlook

This chapter summarizes the present dissertation, highlights its key findings, and gives an outlook on further research topics.

6.1 Recapitulation of Work

The goal of this research is (i) to develop a behavioral traffic state estimator for a multi-agent simulation and (ii) to demonstrate its applicability to a scenario of practically relevant size. Since a model-based estimation approach is chosen, experimental investigations call for executable models of reasonable performance and realism. This applies to both the behavioral and the physical simulator.

The development of a macroscopic traffic flow model in Chapter 2 results in a computationally efficient mobility simulation that is applicable to general networks and has linearizable dynamics. Its computational performance also contributes to an efficient solution of the estimation problem itself. The model is encapsulated in a general state space representation and thus can be replaced by a different implementation, if required.

This macroscopic mobility simulation is combined with a microscopic driver representation in a mathematically tractable way by the mixed micro/macro simulation logic presented in the first half of Chapter 3. This logic links any macroscopic mobility simulation that takes flow splits as input parameters to any microscopic behavioral model that generates individual-level turning decisions at intersections and network entry/exit points. The representation of arbitrary mobility patterns in terms of such turning decisions is demonstrated in the second half of Chapter 3.

These modeling efforts establish a linearizable relation between individual driver behavior and aggregate traffic characteristics. Based on this technically pivotal result, a number of behavioral estimators is developed in Chapter 4. First, a heuristic approach is presented. It is based on a more generally applicable method to steer simulated travelers such that a general objective function of macroscopic system states is increased. For estimation purposes, this objective

function is chosen as the log-likelihood of the available aggregate sensor data, and the agents are steered towards a fulfillment of the measurements.

Second, a statistically more rigorous reconsideration of the estimation problem is given, and two operational Bayesian estimators are developed: (i) The accept/reject estimator functions without further assumptions about the behavioral prior. Its takes an increased number of draws from this prior and retains only a subset of these draws. This subset is representative for the behavioral posterior. (ii) The utility-modification estimator adds a correction term to the systematic utility of every evaluated alternative. Given a particular form of the behavioral prior, the simulation system then draws immediately from the behavioral posterior. The heuristic estimator is found to coincide technically with the UM estimator and can thus be re-analyzed in the Bayesian setting.

The development of these estimators is aimed at but not tailored to an application in conjunction with the MATSim simulation software. Since MATSim was in a transitional period of re-implementation during this work, stable interfaces could not be set up and MATSim's emerging modeling capabilities could not be facilitated. In hindsight, this is not considered as a disadvantageous situation. Since no predetermined simulator implementation was at hand, no flexibility was given away by restricting the developments towards a particular system design. At the time of this writing, an application in conjunction with MATSim is conceptually and technologically feasible. Guidelines for this undertaking are given in Section 6.4.5. Still, the estimators' applicability to systems different from MATSim is not hindered by a confinement to this particular software.

Experimental results are presented in Chapter 5. Since the proposed estimation system is of substantial complexity, it is advisable to obtain a good understanding of its working by an initially synthetic test case that allows for greatest experimental control. It is demonstrated that the method is able to adjust individual-level behavior based on a limited amount of traffic counts such that a significantly improved picture of the global traffic situation is obtained. The method is found to be computationally capable of dealing with scenarios of practically relevant size and to be applicable in both a planning and a telematics setting. The simple behavioral model placeholder implemented for experimental purposes is found to constitute a major limitation of continuative investigations, and the need for advanced behavioral modeling is accentuated.

Additional real world experiments would go beyond the scope of this work. The expected effort to prepare and implement such a test case is substantial [129]. The synthetic experiments given here level the ground for this undertaking. Guidance on how to proceed towards real-world experiments is provided in Section 6.4.1.

6.2 Research Contributions

The key results of this work are highlighted in this section. The listing is confined to novel contributions to the state of the art.

1. Development of a macroscopic mobility simulation with the following features:

- phenomenological consistency with the cell-transmission model,
- simulation of nodes with an arbitrary number of upstream and downstream links,
- approximate linearization of traffic flow dynamics with respect to cell occupancies (system states) and turning fractions (exogenous parameters),
- fast execution by a simulation logic that runs all network elements on individual time scales.

2. Development of a combined micro/macro mobility simulation with the following features:

   - compatibility with broad classes of macroscopic traffic flow models and microscopic driver representations,
   - linearizability in that the effect of any driver's behavior on the global network conditions can be linearly predicted,
   - computational efficiency in that only a sample of the microscopic driver population is required for simulation,
   - computational efficiency by compatibility with the macroscopic simulation logic on variable time scales,
   - removal of most vehicle discretization noise from the macroscopic traffic characteristics.

3. Formalization of the physical aspects of a partial or whole-day plan as a sequence of turning moves on a slightly expanded network such that the linearizability of the global network conditions with respect to individual plan choice is maintained.

4. Development of a general method to steer microscopic agent behavior such that a general objective function of macroscopic traffic conditions is improved.

5. Development of two operational behavioral estimators with the following common features:

   - estimation of fully disaggregate behavior from aggregate traffic measurements and prior behavioral knowledge,
   - compatibility with a purely simulation-based representation of the behavioral prior information,
   - no requirement of a choice set enumeration,
   - computational efficiency that allows for an application to large scenarios.

6. In particular, development of the following distinct estimators:

   - an accept/reject estimator that takes an increased number of draws from an arbitrary behavioral prior and retains only a subset of these draws that is representative for the behavioral posterior,

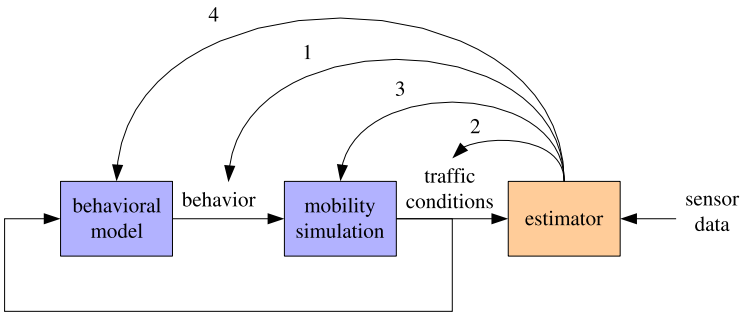


Figure 6.1: Estimated quantities
Two state estimation problems and two parameter identification problems are illustrated in this figure: (1) estimation of behavior (mental states), (2) estimation of traffic conditions (physical states), (3) identification of physical model parameters, (4) identification of behavioral model parameters.

- a utility-modification estimator that corrects the systematic utility of every evaluated alternative such that, given a certain structure of the behavioral prior, the simulation system draws immediately from the behavioral posterior. A heuristic application of this estimator for different or unknown priors is possible.

7. Experimental investigations in a synthetic yet fully dynamical setting with the following conclusions:

   - Given only a limited amount of traffic counts, the global correctness of (i) a SUE planning simulation and (ii) a (rolling-horizon) telematics simulation is consistently and significantly improved by the proposed estimator;
   - the method is capable of handling online estimation problems of practically relevant size in real time;
   - since aggregate traffic measurements contain only limited information, a structurally correct behavioral model is essential for good estimator performance.

6.3 Classification of Results

As a transition to some of the further research topics, Figure 6.1 illustrates the simulation system in terms of only two components, the behavioral model and the mobility simulation. The lower feedback loop indicates that not only behavior influences traffic conditions, but also traffic conditions affect behavior. The estimator compares simulated and real traffic conditions and adjusts the simulation system based on this comparison.

Four different types of adjustment are identified in this figure. Number 1, estimation of behavior, is treated in this dissertation: The estimation of a plan set $\mathcal{U}_1 \ldots \mathcal{U}_N$ comprises all aspects of the individual drivers' mental states that are

necessary to define all macroscopic states $\mathcal{X}$ in the mobility simulation. This estimation approach relies on (i) a deterministic mobility simulation and (ii) an available parameterization of the underlying behavioral and physical model components.

A relaxation of these assumptions leads to the three further estimation tasks indicated in Figure 6.1. They are: (2) estimation of non-deterministic physical system states, (3) parameter identification for the mobility simulation, and (4) parameter identification for the behavioral model. Items (2) and (3) are discussed in Section 6.4.2, and item (4) is considered in Section 6.4.4.

6.4 Further Research Topics

Various directions for future research are thinkable in continuation of this dissertation. This section structures these topics and provides guidance on further developments.

6.4.1 Towards a Real-World Application

This work was conducted with a real-world application in mind and consequently accounts for typical data requirements, performance issues, and modes of operation. The following matters need to be addressed in the preparation of a real-world test case.

6.4.1.1 Model Calibration and Validation

Model-based state estimation crucially depends on structural model correctness. Only a good understanding of reality allows to meaningfully inter- and extrapolate the information contained in limited measurements. This statement equally applies to the physical and the behavioral model components.

The proposed mobility simulation exhibits several novel features: general intersections, variable time scales, and the combined micro/macro simulation logic. These developments were necessary to realize an estimator prototype that is applicable to general scenarios of realistic size. While the synthetic nature of the presented experiments circumvents the need to calibrate and validate the physical model, additional effort in this regard is necessary before a real-world application can be attempted. Since the macroscopic mobility simulation is encapsulated within a general state space representation, it may even be replaced by an entirely different model that is more applicable in a particular setting.

As to behavioral modeling, a structurally correct behavioral simulator must be externally provided. RUMs are particularly applicable here because of their sophisticated calibration and validation procedures. However, the estimator itself is indifferent to the applied model's degree of mathematization, and a simple rule-based model is technically just as feasible for estimation as a full-blown RUM.

6.4.1.2 Measurement Sources and Sensor Types

The experimental investigations of this work focus on flow measurements because of their predominant role in traffic monitoring. However, the general formalism presented in Section 4.2.1 allows to utilize a greater variety of sensor data. As noted there, any aggregate measurement that is a function of the state of a link or a turning counter can directly be fed into the estimation procedure. If the measurements are not statistically independent, their covariance structure needs to be identified before the behavioral estimator can be applied.

Some advanced data sources are addressed below. While they are not accounted for in this dissertation, the fully disaggregate behavioral modeling assumption is at least structurally adequate for their future consideration.

Any vehicle that is equipped with a GPS receiver can serve as a traffic sensor. If its spatiotemporal trajectory is mapped on a representation of the underlying network, a wealth of disaggregate information becomes available that is well suited for the calibration of a behavioral model [67]. This type of information may also be available at a more aggregate level. For example, GPS-equipped taxis typically report their current position to a dispatch center every few minutes. This data can be transformed into local velocity information, e.g., [156], which in turn can be utilized by the proposed estimator. Unlike traffic counts from inductive loops, such floating car data is available at variable locations. It also requires different distributional assumptions about the derived velocity information: A slowly driving vehicle might do so for several reasons and thus is only an imperfect indicator of dense traffic. On the other hand, a quickly advancing vehicle is a reliable indicator of uncongested traffic conditions.

Vehicle re-identification systems provide similar information at a coarser level. The time span between two detections of a vehicle is the sum of all link travel times along an unobserved route that connects the two identification points and, furthermore, includes the duration of all intermediate stops. In consequence, additional modeling assumptions regarding at least route choice are necessary to relate this type of information to the link- or turning move-related states of a macroscopic mobility simulation [4, 183].

6.4.1.3 Performance Tuning

The currently implemented estimator already tackles online problems of nontrivial size. However, further performance tuning is possible.

Algorithmically, the estimation requires to identify a fixed point of a nonlinear and stochastic mapping that comprises a complete traffic simulator, cf. Section 4.1.3. Only a basic SA procedure is utilized in this work, and advanced fixed point search algorithms should be considered for this purpose. The research on the "consistent anticipatory route guidance generation problem" has produced a number of promising results in this regard [26, 51, 52].

Operationally, the estimator is not yet optimized. Its implementation reflect its experimental nature that focuses on flexibility and robustness. Once a particular mode of operation is specified, this implementation should be fine-tuned and

stripped of computational ballast. For example, the currently realized rolling-horizon estimator runs the same SA logic as used in offline operations independently in every estimation period, cf. Section 5.3.1. However, the results of one estimation period contain valuable information for the subsequent estimation periods. This information should be accounted for in a more fine-tuned implementation.

6.4.2 Combined Behavioral and Physical Estimation

So far, it is assumed that the mobility simulation is modeled without error. A possible relaxation of this assumption is outlined in this section.

Uncertain traffic flow dynamics are modeled by adding a temporally uncorrelated zero-mean random disturbance vector $\boldsymbol{\eta}(k)$ to state equation (2.17):

$$\mathbf{x}^{\mathrm{ms}}(k+1) = \mathbf{f}^{\mathrm{ms}}[\mathbf{x}^{\mathrm{ms}}(k), \boldsymbol{\beta}(k), \boldsymbol{\eta}(k), k] \tag{6.1}$$

where $\mathbf{x}^{\mathrm{ms}}$ is the mobility simulation's physical state vector and $\boldsymbol{\beta}$ represents the single-commodity turning fractions. Equation (6.1) replaces the deterministic traffic flow model component of the mixed micro/macro state space model (3.7). The relation between $\mathbf{x}^{\mathrm{ms}}$ and the available measurements $\mathbf{y}$ is represented by the likewise randomly disturbed output equation

$$\mathbf{y}(k) = \mathbf{g}[\mathbf{x}^{\mathrm{ms}}(k), \boldsymbol{\epsilon}(k)], \tag{6.2}$$

which corresponds to (4.16) without loss of generality. The two above equations can be linearized. Given a parameterization $\{\boldsymbol{\beta}(k)\}_k$, they constitute a nonlinear, dynamical system that is amenable to the macroscopic state estimation techniques reviewed in Section 1.2.

All behavioral estimators of this thesis disregard the stochastic error $\boldsymbol{\eta}$ in (6.1). Without exception, they contain a step in which "$\mathcal{U}_1 \ldots \mathcal{U}_N$ are loaded on the network and $\mathcal{X}$ is obtained", cf. Algorithms 2 through 4. That is, the behavioral estimation problem is solved given a particular mapping of the behavior $\mathcal{U}_1 \ldots \mathcal{U}_N$ on the macroscopic states $\mathcal{X}$.

The $\boldsymbol{\beta}$ parameters in (6.1) result from the behavior of individual particles in the mixed micro/macro mobility simulation of Section 3.1. This particle behavior is fully determined by a plan set $\mathcal{U}_1 \ldots \mathcal{U}_N$. The network loading step can therefore be replaced by a physical state estimator that formally operates exclusively on the model specifications (6.1) and (6.2) with an externally provided $\{\boldsymbol{\beta}(k)\}_k$ parameterization that is internally generated by an execution of $\mathcal{U}_1 \ldots \mathcal{U}_N$. The physical estimator utilizes the same sensor data $\mathcal{Y} = \{\mathbf{y}(k)\}_k$ as the behavioral estimator.

Consequently, the behavioral estimation problem is still solved given a particular mapping of the behavior $\mathcal{U}_1 \ldots \mathcal{U}_N$ on the macroscopic states $\mathcal{X}$, only that this mapping now incorporates a physical state estimation procedure. This also enables the tracking of time-dependent physical model parameters by an appropriate extension of the macroscopic state vector, e.g., [3, 175]. The straightforwardness of this approach is owed to the minimal interface between the microscopic and the macroscopic modeling components.

6.4.3 Combined Telematics and Planning Estimation

Mutual benefits can be expected if a telematics and a planning estimator are applied concertedly. Two possibilities to realize such a coupling are outlined in this section. In either case, it is assumed that an online estimator generates results on a daily basis that are used to improve the outcome of a planning simulation. This enables the latter to provide improved behavioral priors for the next day's online estimation problem.

The ability to provide an improved prior does not imply that a suchlike adjusted planning simulation can also be applied to predict structurally different scenarios, where, for example, infrastructural changes are considered. This ability would require not only to estimate what choices are made by the travelers in a given scenario but also to identify the underlying behavioral parameters that trigger these choices. This section only considers the problem of how to adjust a planning simulation for purposes of "incremental" online traffic monitoring. The behavioral parameter estimation problem is discussed in subsequent Section 6.4.4.

6.4.3.1 Fusion of Λ Coefficients

The difference between a behavioral prior and an estimated posterior is fully captured by the Λ coefficients. The most straightforward approach to facilitate the coefficients Λ^d obtained by the online estimator at a certain day d is to incorporate them in baseline coefficients $\bar{\Lambda}$ that are used as starting values in the next day's online estimation problem. These baseline coefficients can also be accounted for in a planning model if updated prior information is to be simulated. A similar procedure can be found in the context of OD matrix estimation where a within-day estimated OD matrix is used to update a planning OD matrix, cf. Section 1.2.2.2. Possible update methods are recursive averaging [7] and Kalman filtering. The latter assumes that $\bar{\Lambda}$ follows a random walk and that one noisy measurement Λ^d of $\bar{\Lambda}$ becomes available per day [183].

6.4.3.2 Choice Set Modifications

Choice set generation is a computationally demanding step that is likely to be performed at least in part offline. In online operations, computational considerations might require a relatively small choice set per agent that in consequence needs to be chosen with particular care. If the online estimator has selected a certain plan rather infrequently, this indicates that this plan is unlikely to belong to the considered traveler's choice set and thus should be replaced by a more reasonable alternative. This allows for an incremental offline choice set adjustment that should also result in an improved online estimation performance.

6.4.4 Behavioral Parameter Estimation

The proposed estimator also holds promise to provide information about parameters that underlie the estimated choices, i.e., to address parameter estimation

problem (4) in Figure 6.1. Two such approaches are discussed in this section.[1]

6.4.4.1 Estimation of Population Parameters

A synthetic population needs to be created before an agent-based simulation of traffic is possible, cf. Section 1.2.2.3. Typically, its generation relies on a sequence of sampling procedures where agent parameters are drawn from beforehand specified distributions that apply to homogeneous subsets of the population [9]. For example, the activity patterns for all male workers of an urban population may be drawn from a single distribution, the work locations for all employees that live in a certain traffic zone may be drawn from yet another distribution, and so forth.

Since the distributions that underlie this generation procedure are themselves estimates of imperfect precision, aggregate traffic measurements may help to improve the realism of the synthetic population. Since this implies that the sensor data is used to adjust structural features of the multi-agent model, the resulting population should be applicable in a wider variety of scenarios that may considerably differ from the conditions in which the measurements are obtained. An application of the proposed behavioral state estimator for this purpose is described hereafter.

A subset $M \subseteq \{1 \dots N\}$ of the synthetic population is considered. This subset is homogeneous with respect to the distribution $P_M(\theta)$ of a certain population parameter $\theta \in \Theta$ where Θ is a discrete and permissibly non-ordinal domain. Disregarding the sensor data, a single draw of this parameter is assigned to every individual $n \in M$. All plans of an agent in M are thus parameterized directly or indirectly by this value. When the simulation is run, the agent learns individually optimal behavioral patterns, and when the iterations have stabilized, the agent exhibits a reasonable plan choice distribution given its particular θ value.

Assume that there is uncertainty about the true distribution of θ. Since M is homogeneous with respect to this distribution, it is feasible to provide every agent in M with two instead of one parameter values, say θ_1 and θ_2, and to

[1] A unified Bayesian formulation of both parameter estimation problems considered in this section was found shortly after the submission of this dissertation. Let the decision protocol be parameterized with an individual-level parameter vector $\boldsymbol{\theta}_n$ for every agent $n = 1 \dots N$, denote the individually parameterized choice distributions by $P_n(\mathcal{U}_n|\boldsymbol{\theta}_n)$, and assume that a prior p.d.f. $p(\boldsymbol{\theta}_n)$ is available for the parameters. In complete analogy to the derivation given in Section 4.3.2, an individual-level posterior

$$p_n(\mathcal{U}_n, \boldsymbol{\theta}_n|\mathcal{Y}) = \frac{e^{\langle\Lambda,\mathcal{U}_n\rangle} P_n(\mathcal{U}_n|\boldsymbol{\theta}_n)p(\boldsymbol{\theta}_n)}{\sum_{\mathcal{V}\in C_n} e^{\langle\Lambda,\mathcal{V}\rangle} \int P_n(\mathcal{V}|\boldsymbol{\theta}')p(\boldsymbol{\theta}')d\boldsymbol{\theta}'}$$

of agent n's joint choice and parameter distribution given the measurements can be formulated. The following version of the AR estimator draws from this posterior:

1. Draw $\boldsymbol{\theta}_n$ from $p(\boldsymbol{\theta}_n)$.
2. Draw $\mathcal{U}_n$ from $P_n(\mathcal{U}_n|\boldsymbol{\theta}_n)$.
3. Accept $(\mathcal{U}_n, \boldsymbol{\theta}_n)$ with the original acceptance probability $\phi_n(\mathcal{U}_n)$ defined in (4.35). Otherwise, goto 1.

Note that this estimator is equally applicable to the identification of discrete-valued parameters.

parameterize one half of its plans with θ_1 and the other half with θ_2. The resulting parameter occurrences still follow the original distribution $P_M(\theta)$ in that the probability that an individual in M gets assigned two particular parameters θ_1 and θ_2 is $P_M(\theta_1)P_M(\theta_2)$.

The estimator now adjusts the population's behavior to the sensor data $\mathcal{Y}$. The resulting choice frequency of any particular θ value in M is

$$P_M(\theta|\mathcal{Y}) \approx \frac{1}{R|M|} \sum_{r=1}^{R} \sum_{n \in M} \mathcal{I}(\mathcal{U}_n^r \sim \theta) \tag{6.3}$$

where $r = 1 \ldots R$ iterations are considered, $\mathcal{U}_n^r$ is the plan selected by individual n in iteration r, and $\mathcal{I}(\mathcal{U} \sim \theta)$ is one if plan $\mathcal{U}$ is parameterized with θ and zero otherwise. This simulated posterior distribution of θ given the measurements can be applied to re-sample the parameters of the population subset M and to re-run the estimation. This is repeated until consistency of the prior and the posterior parameter distribution is attained.

A precaution is necessary to avoid biases in this approach. If there is no sensor data, the estimator is reduced to a plain simulator, and the result of such a simulation is that every individual in M discards the θ value of inferior subjective performance. If, for example, θ represents a leisure location and all else is equal, the plans that contain the more distant leisure location are discarded because they implicate longer travel times. That is, the plan selection mechanism itself generates a drift in the parameter distribution.

A remedy to this problem is to split the plan set of every individual according to the different θ values. Every agent in M now has two choice sets C_n^1 (all elements of which are parameterized with θ_1) and C_n^2 (parameterized with θ_2) of equal size. When making a decision, the agent first chooses a choice set with uniform probability and then selects a plan from that set according to its behavioral model. In result, the agent exhibits a dual behavior. This should not introduce systematic side effects in the simulation since the whole subpopulation's parameterization is still consistent with $P_M(\theta)$. If now the AR estimator is applied, all resulting changes in the θ selection frequencies can be attributed exclusively to the sensor data. The UM estimator is not applicable here since it has no influence on the uniform distribution used for choice set selection.

6.4.4.2 Estimation of RUM Parameters

Typically, the deterministic utility of a RUM is linear in parameters:

$$V_n(\mathcal{U}) = \boldsymbol{\theta}^T \mathbf{x}_{\mathcal{U},n} + k_{\mathcal{U}} \tag{6.4}$$

where $\mathbf{x}_{\mathcal{U},n}$ is a vector that represents the features of decision maker n and of alternative $\mathcal{U} \in C_n$, and $\boldsymbol{\theta}$ is a vector of real-valued parameters. The alternative-specific constant $k_{\mathcal{U}}$ captures all choice-relevant aspects of $\mathcal{U}$ that are independent of $\mathbf{x}_{\mathcal{U},n}$.

The UM estimator of Section 4.3.3 affects estimated behavior via additive utility corrections:

$$\begin{aligned} W_n(\mathcal{U}) &= V_n(\mathcal{U}) + \langle \Lambda, \mathcal{U} \rangle / \mu \\ &= \boldsymbol{\theta}^T \mathbf{x}_{\mathcal{U},n} + k_{\mathcal{U}} + \langle \Lambda, \mathcal{U} \rangle / \mu. \end{aligned} \tag{6.5}$$

That is, the UM estimator effectively adjusts the alternative-specific constants of an underlying RUM. The predictive power of suchlike adjusted RUMs depends on the stability of the alternative-specific constants across different scenarios.

If the $\boldsymbol{\theta}$ parameters themselves admit improvements, an incorporation of sensor data into the RUM calibration procedure is a desirable goal. RUM parameters are typically identified by maximum likelihood estimation [21], which requires a likelihood function $l(\boldsymbol{\theta}|\mathcal{Y}) = p(\mathcal{Y}|\boldsymbol{\theta})$ to be available. Noting that the sensor data $\mathcal{Y}$ is not directly dependent on $\boldsymbol{\theta}$, one obtains

$$\begin{aligned} l(\boldsymbol{\theta}|\mathcal{Y}) &= \sum_{\mathcal{U}_1 \in C_1} \dots \sum_{\mathcal{U}_N \in C_N} p(\mathcal{Y} \mid \mathcal{U}_1 \dots \mathcal{U}_N) P(\mathcal{U}_1 \dots \mathcal{U}_N|\boldsymbol{\theta}) \\ &= \mathrm{E}\left\{ l(\mathcal{U}_1 \dots \mathcal{U}_N|\mathcal{Y}) \,|\boldsymbol{\theta}\right\}. \end{aligned} \tag{6.6}$$

That is, the likelihood of $\boldsymbol{\theta}$ given the sensor data can be expressed as the expectation of the available likelihood $l(\mathcal{U}_1 \dots \mathcal{U}_N|\mathcal{Y}) = p(\mathcal{Y} \mid \mathcal{U}_1 \dots \mathcal{U}_N)$, cf. Section 4.2.1, given that the population's plan choice distribution is parameterized by $\boldsymbol{\theta}$. A Monte Carlo approximation of this expectation is possible:

$$\mathrm{E}\left\{ l(\mathcal{U}_1 \dots \mathcal{U}_N|\mathcal{Y}) \,|\boldsymbol{\theta}\right\} \approx \frac{1}{R} \sum_{r=1}^{R} l(\mathcal{U}_1^r \dots \mathcal{U}_N^r|\mathcal{Y}) \tag{6.7}$$

where R is the number of draws and $\mathcal{U}_n^r$ is the plan chosen by individual n in simulation r given parameter $\boldsymbol{\theta}$. Parameter estimation based on a suchlike simulated likelihood is possible in principle [166], but it is computationally extremely demanding since every draw requires a full run of the traffic simulator. An interesting question is to what degree a linearization-based approximation of the network loading procedure can help to accelerate this process.

6.4.5 Integration with MATSim

6.4.5.1 Conceptual Aspects

Section 3.2 characterizes a behavioral simulation system that is applicable in conjunction with the proposed estimator. It is observed there that the following properties of the MATSim planning simulation are not immediately compatible with this specification:

1. variable plan choice set,
2. continuously updated (learned) plan utilities (scores),
3. immediate execution of a newly generated plan.

Problems 1 and 3 are resolved collectively. An invariable choice set results if an agent is assumed not only to select from its currently memorized plans but also also from all other plans that can possibly be generated by the MATSim replanning mechanisms described in Section 3.2.2.3. The overall probability that a new plan is generated in a given iteration is denoted by P_{new}. Accordingly, the selection probability of any existing plan is $1 - P_{\text{new}}$ times its choice probability

without plan generation, whereas the selection probability of any newly generated plan is P_{new} times its probability of generation. Thus, every agent disposes of a well-defined (albeit possibly very large) choice set, and a choice probability for each element in this set exists. Since neither the explicit availability of these probabilities nor an enumeration of the choice set is required, an application of the AR is conceptually feasible at every *single* MATSim iteration. However, since the generation of new plans is not utility-driven, the UM estimator is not applicable here.

Item 2 is related to the strong correlation between *subsequent* MATSim iterations. Travel behavior is not simulated based on systematic utilities that are averaged over a long time horizon but relies more strongly on the most recent iterations: The scores of executed plans are updated by a recursive filter that has an infinite but exponentially decaying memory. The route recalculations utilize only the most recent iteration's travel times. Thus, even after a large number of iterations, a situation in which the traffic conditions of subsequent iterations fluctuate uncorrelatedly around stable average values is unlikely to occur. This effect can also be observed throughout the experiments given in Chapter 5.

The estimation procedure, however, fundamentally relies on the Λ coefficients that represent the sensitivities of the measurement log-likelihood to the driver behavior. These sensitivities are averaged over many iterations, cf. Section 4.1.3, and the resulting averages may stabilize even if the overall system exhibits a cyclic behavior, as it is likely to occur in MATSim. Since this implies a systematic deviation between the actually occurring sensitivities and their average values, a declined estimator performance may result. However, no general statement about MATSim's dynamics can be made at this point.

The AR estimator repeats a single choice situation several times. It requires that repeated draws are independent and identically distributed. This estimator is not impaired by the correlation between subsequent MATSim iterations as long as the behavioral distribution of an agent is invariable within a single iteration. MATSim evolves as a Markov process, with its state being defined through the current agent memory (in terms of available plans) and the last iteration's traffic conditions (used for the generation of new plans). In every single iteration, the AR estimator corrects the transition probabilities of this process in a most plausible way. Thus, it is reasonable to expect that the resulting iteration dynamics of MATSim are likewise improved.

The estimator's conceptual ability to function even in conjunction with this rather untypical model of dynamical traffic evolution indicates its flexibility and independence of a specific system design. The following section exemplifies the technical steps that are necessary to assert the above hypotheses in practice.

6.4.5.2 Technical Aspects

Several exemplary Java code snippets are provided that represent the arguably simplest way to attach the estimator to the MATSim system as implemented in October 2007. For simplicity, only the selection of full plans is considered and the code is stripped of all conceptually irrelevant elements. Of course, various alternative implementations that achieve the same effect are thinkable.

For the purpose of this presentation, it is sufficient to specify an agent by a `Person` interface that provides access to the set of its available `Plan` instances.

```
interface Person {
    Set getPlans();
}
```

The utility function is an implementation of a `ScoringFunction` interface that maps a `Plan` on a utility value as perceived by a particular `Person`.

```
interface ScoringFunction {
    double getScore(Plan p, Person n);
}
```

The decision protocol is represented by a `PlanSelector` class that implements a `selectPlan(Person, ScoringFunction)` function. This function returns a single draw from the `Person`'s `Plan` set.

```
class PlanSelector {
    Plan selectPlan(Person n, ScoringFunction sF) {
        Plan result;
        // Choice logic implemented here.  Examples:
        // * access choice set via n.getPlans();
        // * evaluate a plan p via sF.getScore(p, n);
        return result;
    }
}
```

An application of the UM estimator requires to modify the implemented `ScoringFunction`. An appropriate technique is to implement a wrapper class `UMScoringFunction` around the original `ScoringFunction` and to pass this wrapper instead of the original implementation to the `PlanSelector`.

```
class UMScoringFunction implements ScoringFunction {
    ScoringFunction sF;
    UMScoringFunction(ScoringFunction sF) {
        this.sF = sF;
    }
    double getScore(Plan p, Person n) {
        return sF.getScore(p, n) + ⟨Λ,𝒰⟩/μ;
        // 𝒰 is turning move sequence of Plan p.
        // ⟨Λ,𝒰⟩/μ addend is defined in (4.14).
    }
}
```

The AR estimator requires a modification of the plan selection logic itself. This can be realized by function overriding. A subclass `ARPlanSelector` is derived from `PlanSelector`, the `selectPlan(..)` function is overridden, and the original `PlanSelector` is replaced by an instance of the `ARPlanSelector`.

```
class ARPlanSelector extends PlanSelector {
    Plan selectPlan(Person n, ScoringFunction sF) {
        Plan result;
        do {
            result = super.selectPlan(n, sF);
        } while (Math.random() >= φ_n(U));
        // U is turning move sequence of Plan result.
        // φ_n(U) is acceptance probability (4.35).
        return result;
    }
}
```

Both the `UMScoringFunction` and the `ARPlanSelector` need references to the Λ coefficients for the calculation of utility corrections and acceptance probabilities. The linearization logic that generates these coefficients is part of the macroscopic mobility simulation. In conjunction with MATSim, the easiest way of accessing this data is via files: In every iteration, the behavioral simulation system writes out a file that contains the selected plans of all agents. The mobility simulation then loads these plans, executes them, and in turn writes out the Λ coefficients plus all further data that is required for agent replanning. This basic implementation suggests itself for first experimental investigations. The programming effort of a tighter coupling by direct function calls would mainly pay off in terms of an increased execution speed because of the avoided file i/o.

6.4.6 Structural Model Refinements

6.4.6.1 Physical Simulation

The micro/macro coupling logic does not differentiate among vehicle types. Within limits, this is possible by a specification of different macroscopic "sizes" for passenger cars, trucks, buses, and so forth. Continuative modeling may also differentiate the dynamics of different vehicle classes within the macroscopic mobility simulation. This is likely to require a representation of multi-commodity flows within the macroscopic model component [33].

Inner-urban traffic flow is dominated by signaling. While the employed mobility simulation does not account for this aspect, the modeling of signalized intersections has already been demonstrated in conjunction with a cell-transmission model [1]. This requires a network model at the granularity of individual lanes in order to avoid unrealistic spill-backs at simulated intersections that in reality have turning pockets. In such a setting, it might prove useful to switch off the exponential turning counter forgetting mechanism (3.4) for the duration of a red phase.

There is an important issue regarding adaptive signaling. Adaptive controls may switch strategies based on threshold values and thus may introduce discontinuities in the mobility simulation: A small behavioral change of a single driver that causes a sensor output to exceed a threshold value might change the entire control strategy and thus might have a large effect on the macroscopic system states. However, since adaptive signaling is sensor driven, the according sensor data can be made available to the estimator as well. This allows to reproduce

the true control strategy without error, either by a reconstruction of its logic in the simulator or by a direct observation of the real signaling. Since suchlike simulated signaling is a perfect image of reality, no adaptivity is necessary within the mobility simulation such that its continuity with respect to plan choice is preserved.

6.4.6.2 Behavioral Simulation

Flexibility as to different behavioral implementations is a main objective of this work, and few limitations are imposed on a refined behavioral simulator.

Switching from single-day plans to weekly plans discloses new potentials for mid-term forecasting. Since weekly plans introduce a logical relation between travel behavior at subsequent days, single-day plan estimates provide information about upcoming behavior that can be facilitated for prediction and, in particular, as an improved prior for the next day's estimation problem.

Traffic monitoring is not conducted as an end in itself. In online operations, a traffic prediction that is based on the most recent traffic state estimate can be utilized to provide various information services to travelers. However, if this guidance is not carefully chosen, the resulting driver reactions might invalidate the underlying prediction. This anticipatory guidance generation problem is decoupled from the state estimation problem since all disseminated information is known up to the present point in time at which the online estimation ends. In consequence, the estimator only requires a behavioral model that properly accounts for the most recently generated guidance, but it is indifferent with respect to the particular nature of this guidance [19].

Appendix A

Implementation of GPRC Integer Sets

The GPRC requires many integer set operations. Since all set implementations provided by the Java Collections Framework [85] rely on object representations of their elements, they carry a formidable overhead if only primitive types are required. This appendix describes a set implementation that is tailored towards the GPRC.

A GPRC integer set contains elements from a small value domain $1 \ldots I + J$ where I (J) is the number of upstream (downstream) links of the considered intersection. Equivalently, a value domain $0 \ldots I + J - 1$ is assumed here in order to allow for an array-based implementation that starts counting at zero.

The subsequently provided Java code fragments constitute the basis of a class `NSet`.

```
public class NSet {
    // code fragments here
}
```

This class contains a primitive and two array members of integer type.

```
private int size;
private final int[] values;
private final int[] indices;
```

`size` holds the number of entries in a given instance of `NSet`. The first `size` fields of the `values`-array contain these entries. If `indices[x]` equals `-1`, then `x` is not contained in the set. Otherwise, `indices[x]` contains the index of `x` in `values`, that is, `values[indices[x]]==x` if `x` is contained in the set. During construction, both arrays are initialized according to the maximum size `maxSize` allowed for this set.

```
public NSet(int maxSize) {
    size = 0;
    values = new int[maxSize];
    indices = new int[maxSize];
    for (int i = 0; i < maxSize; i++)
        indices[i] = -1;
}
```

This data structure has a constant memory requirement of $2(I+J)+1$ integers. The following three functions provide access to the content of this set. Parameter range checks are omitted for clarity.

```
public boolean contains(int value) {
    return (indices[value] != -1);
}

public void add(int value) {
    if (!contains(value)) {
        indices[value] = size;
        values[size] = value;
        size++;
    }
}

public void remove(int value) {
    if (contains(value)) {
        size--;
        final int removedIndex = indices[value];
        if (removedIndex != size && size > 0) {
            final int movedValue = values[size];
            values[removedIndex] = movedValue;
            indices[movedValue] = removedIndex;
        }
        indices[value] = -1;
    }
}
```

If only these three functions were required, a single boolean array that simply indicates the existence of an entry would be roughly twice as efficient. However, an iteration over the elements of such a set would require to access every array entry in order to check if the according marker is set. The following implementation of the *iterator design pattern* [70] provides a more efficient solution. It is just as fast as looping only through the first `size` elements of an array. This is particularly advantageous if there are relatively few entries in the data structure.

```
public NSet.Iterator iterator() {
    return new NSet.Iterator();
}
```

```
public class Iterator {

    private int index;

    private Iterator() {
        index = 0;
    }

    public boolean hasNext() {
        return index < size;
    }

    public int next() {
        return values[index++];
    }
}
```

The implementation of `Iterator` as an inner class of `NSet` is a common Java technique that supports data encapsulation.

Appendix B

Sensitivity Analysis for the GPRC

This appendix provides calculation schemes for $\partial\boldsymbol{\xi}^{(M)}/\partial\boldsymbol{\xi}^{(0)}$ and $\partial\boldsymbol{\xi}^{(M)}/\partial\boldsymbol{\beta}$ where $\boldsymbol{\xi}^{(0)}(\boldsymbol{\xi}^{(M)})$ is the GPRC's initial (final) state vector and $\boldsymbol{\beta} = (\beta_q)$ is a vector of constant consumption rate parameters with an available Jacobian $\partial\boldsymbol{\varphi}(\ldots)/\partial\boldsymbol{\beta}$. The notational overlap of $\boldsymbol{\beta}$ with the turning fractions β_{ij} of Section 2.4 is intended but not required. The complete notation for the GPRC can be found in Section 2.3.

The subsequent analysis builds on the following preliminaries:

- If state index j is the only element in $B^{(m)}$, then the duration $\theta^{(m)}$ of step m is $\theta^{(m)} = \xi_j^{(m)}/\varphi_j(D^{(m)})$ such that a small variation $\delta\xi_j^{(m)}$ of resource j at the beginning of step m implies a likewise small variation $\delta\theta^{(m)}$ of $\theta^{(m)}$:

$$B^{(m)} = \{j\} \quad \Rightarrow \quad \delta\theta^{(m)} = \delta\xi_j^{(m)}/\varphi_j(D^{(m)}). \tag{B.1}$$

- The consumption rate of any resource must be monotonously increasing with the number of nonzero resources:

$$\varphi_i(D \cup \{j\}) \geq \varphi_i(D) \quad \forall i, j. \tag{B.2}$$

  A resource is denoted as **blocked** if it is nonzero but has a zero consumption rate. The monotonicity property implies that (i) available and previously non-blocked resources cannot block from the addition of resources to D and (ii) once blocked resources stay blocked since D only gets reduced during a run of the GPRC.

- The state of a blocked resource i has no influence on the resource consumption rates:

$$\varphi_i(D \cup \{i\}) = 0 \quad \Rightarrow \quad \boldsymbol{\varphi}(D\backslash\{i\}) = \boldsymbol{\varphi}(D \cup \{i\}). \tag{B.3}$$

Algorithm 5 GPRC sensitivity calculation logic

1. Initialize $\partial\boldsymbol{\xi}^{(0)}/\partial\boldsymbol{\xi}^{(0)}$ and $\partial\boldsymbol{\xi}^{(0)}/\partial\boldsymbol{\beta}$. See Section B.1.
2. At the end of every GPRC step $m = 0, 1, \ldots$, do:
   (a) Calculate $\partial\boldsymbol{\xi}^{(m+1/2)}/\partial\boldsymbol{\xi}^{(0)}$ and $\partial\boldsymbol{\xi}^{(m+1/2)}/\partial\boldsymbol{\beta}$. See Section B.2 and Algorithm 6.
   (b) Calculate $\partial\boldsymbol{\xi}^{(m+1)}/\partial\boldsymbol{\xi}^{(0)}$ and $\partial\boldsymbol{\xi}^{(m+1)}/\partial\boldsymbol{\beta}$. See Section B.3, B.4, and Algorithm 7.
3. Complete $\partial\boldsymbol{\xi}^{(M)}/\partial\boldsymbol{\xi}^{(0)}$ and $\partial\boldsymbol{\xi}^{(M)}/\partial\boldsymbol{\beta}$. See Section B.5.

Approximations of $\partial\boldsymbol{\xi}^{(M)}/\partial\boldsymbol{\xi}^{(0)}$ and $\partial\boldsymbol{\xi}^{(M)}/\partial\boldsymbol{\beta}$ are built incrementally while the GPRC runs through $m = 0 \ldots M$. For notational convenience, these approximations are denoted by the same symbols as the exact partial derivatives. Every step m is again split in two segments of equal length $\theta^{(m)}/2$, which necessitates two sensitivity updates in every step m and the notion of an "intermediate" step $m + 1/2$. This somewhat inflates the presentation but is necessary to handle situations where several resources run dry simultaneously. Algorithm 5 provides an overview. The remainder of this appendix describes the details of this logic.

B.1 Initialization of Sensitivities

This is straightforward: $\partial\boldsymbol{\xi}^{(0)}/\partial\boldsymbol{\xi}^{(0)} = I$ (identity matrix) implies that resources cannot have interacted before the process has started, and $\partial\boldsymbol{\xi}^{(0)}/\partial\boldsymbol{\beta} = 0$ (all zero matrix) states that the consumption rate parameters $\boldsymbol{\beta}$ cannot have had an influence before the consumption has taken place.

B.2 Calculation of $\partial\boldsymbol{\xi}^{(m+1/2)}/\partial\boldsymbol{\xi}^{(0)}$ and $\partial\boldsymbol{\xi}^{(m+1/2)}/\partial\boldsymbol{\beta}$

If $j \in D^{(m)}$, resource j is strictly positive at $m + 1/2$. A variation $\delta\xi_j^{(m)}$ cannot cause any intermediate regimes but only punches through to $\xi_j^{(m+1/2)} = \xi_j^{(m)} - \frac{\theta^{(m)}}{2}\varphi_j(D^{(m)})$, resulting in $\delta\xi_j^{(m+1/2)} = \delta\xi_j^{(m)}$, as illustrated in Figure B.1(a). A variation $\delta\beta_q^{[m,m+1/2]}$ of consumption rate parameter β_q that occurs exclusively during $[m, m+1/2]$ generates $\delta\xi_j^{(m+1/2)} = -\frac{\theta^{(m)}}{2}\frac{\partial\varphi_j(D^{(m)})}{\partial\beta_q}\delta\beta_q^{[m,m+1/2]}$, as shown in Figure B.1(b).

If $j \notin D^{(m)}$, resource j is originally zero during step m, which makes it indifferent to consumption rate variations and only allows for a positive variation $\delta\xi_j^{(m)} > 0$. If $\varphi_j(D^{(m)} \cup \{j\}) = 0$, (B.3) ensures that j does not interact with other resources such that the variation only punches through to $\xi_j^{(m+1/2)}$, resulting in $\delta\xi_j^{(m+1/2)} = \delta\xi_j^{(m)}$, see Figure B.1(c).

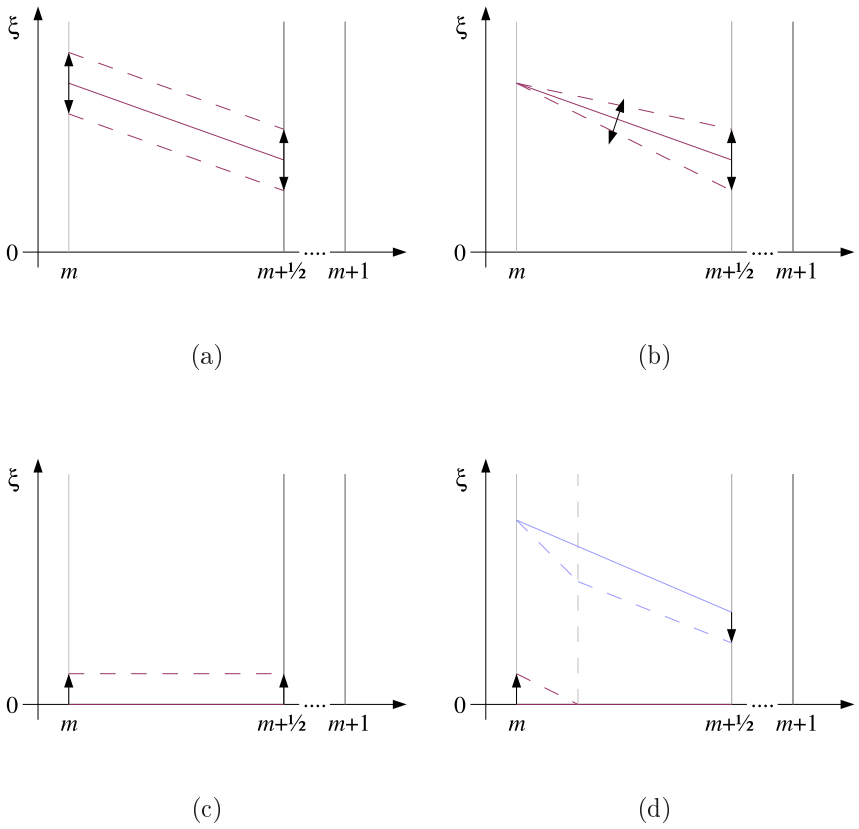


Figure B.1: Resource variations for first half of GPRC sensitivity calculation
All diagrams show resource trajectories over "GPRC time". Within each diagram, the left arrow represents the causative variation, and the right arrow represents the induced variation. Varied resources are drawn in red, and influenced resources (if any) are drawn in blue. Original trajectories are solid, and their variations are dashed.

If $j \notin D^{(m)}$ and $\varphi_j(D^{(m)} \cup \{j\}) > 0$, resource j runs dry again after its variation and a new regime $D' = D^{(m)} \cup \{j\}$ occurs at the very beginning of step m. D' is limited by $B' = \{j\}$ such that (B.1) can be used to obtain its duration $\delta\theta' = \delta\xi_j^{(m)}/\varphi_j(D')$. During $\delta\theta'$, all resources $i \in D^{(m)}$ are reduced by consumption rates $\varphi_i(D')$ instead of $\varphi_i(D^{(m)})$. Equation (B.2) ensures that these resources do not block because of j's addition, which guarantees continuity. This varies $\xi_i^{(m+1/2)}$ by $\delta\xi_i^{(m+1/2)} = (\varphi_i(D^{(m)}) - \varphi_i(D'))\delta\theta'$, see Figure B.1(d).

Summarized, the effects of variations $\delta\xi_j^{(m)}$ and $\delta\beta_q^{[m,m+1/2]}$ until step $m + 1/2$ are:

$$\frac{\delta\xi_i^{(m+1/2)}}{\delta\xi_j^{(m)}} = \begin{cases} \mathcal{I}(i=j) & j \in D^{(m)} \vee \varphi_j(D') = 0 \\ \varphi_i(D^{(m)}) - \varphi_i(D') & i \in D^{(m)} \wedge j \notin D^{(m)} \\ \varphi_j(D') & \ldots \wedge \varphi_j(D') > 0 \\ 0 & \text{otherwise} \end{cases} \tag{B.4}$$

$$\frac{\delta\xi_i^{(m+1/2)}}{\delta\beta_q^{[m,m+1/2]}} = \begin{cases} -\frac{\theta^{(m)}}{2}\frac{\partial\varphi_i(D^{(m)})}{\partial\beta_q} & i \in D^{(m)} \\ 0 & \text{otherwise} \end{cases} \tag{B.5}$$

where $\mathcal{I}(A)$ is one if A is true and zero if A is false. The full sensitivities until step $m + 1/2$ can now recursively be evaluated via

$$\frac{\partial\xi_i^{(m+1/2)}}{\partial\boldsymbol{\xi}^{(0)}} = \sum_j \frac{\delta\xi_i^{(m+1/2)}}{\delta\xi_j^{(m)}}\frac{\partial\xi_j^{(m)}}{\partial\boldsymbol{\xi}^{(0)}} \tag{B.6}$$

$$\frac{\partial\xi_i^{(m+1/2)}}{\partial\boldsymbol{\beta}} = \frac{\delta\xi_i^{(m+1/2)}}{\delta\boldsymbol{\beta}^{[m,m+1/2]}} + \sum_j \frac{\delta\xi_i^{(m+1/2)}}{\delta\xi_j^{(m)}}\frac{\partial\xi_j^{(m)}}{\partial\boldsymbol{\beta}}. \tag{B.7}$$

A calculation scheme for these Jacobians is given in Algorithm 6.

B.3 Calculation of $\partial\boldsymbol{\xi}^{(m+1)}/\partial\boldsymbol{\xi}^{(0)}$

If $j \in D^{(m+1)}$, resource j is strictly positive at step $m+1$ so that any variation $\delta\xi_j^{(m+1/2)}$ only punches through to $\xi_j^{(m+1)}$. Figure B.1(a) captures a similar situation. If $j \notin D^{(m)}$, it originally has run dry before regime $D^{(m)}$. A (positive) variation $\delta\xi_j^{(m+1/2)}$ can only occur if a positive variation $\delta\xi_j^{(m)}$ has caused the resource to block. As stated before, this implies that j will stay blocked without influencing other resources, so the variation $\delta\xi_j^{(m+1/2)}$ only punches through to $\xi_j^{(m+1)}$, similarly to Figure B.1(c). These cases can be combined in that $\delta\xi_j^{(m+1)} = \delta\xi_j^{(m+1/2)}$ holds for $(j \in D^{(m+1)} \vee j \notin D^{(m)}) \equiv j \notin B^{(m)}$.

If $j \in B^{(m)}$, then $\varphi_j(D^{(m)})$ must have been greater 0, and therefore $\xi_j^{(m+1/2)} > 0$ can be varied in both directions. A positive variation $\delta\xi_j^{(m+1/2)}$ only punches through to $\xi_j^{(m+1)}$, see Figure B.2(a). Given a negative variation $\delta\xi_j^{(m+1/2)}$, a new regime $D'' = D^{(m)} \backslash \{j\}$ occurs directly before the end of step m, as illustrated in Figure B.2(b). The new regime D'' is limited only by $B'' = \{j\}$, so (B.1)

Algorithm 6 First half of GPRC sensitivity calculation

for all $j \in D^{(m)}$, do {

$$\frac{\partial \xi_j^{(m+1/2)}}{\partial \boldsymbol{\xi}^{(0)}} = \frac{\partial \xi_j^{(m)}}{\partial \boldsymbol{\xi}^{(0)}}$$

$$\frac{\partial \xi_j^{(m+1/2)}}{\partial \boldsymbol{\beta}} = \frac{\partial \xi_j^{(m)}}{\partial \boldsymbol{\beta}} - \frac{\theta^{(m)}}{2} \frac{\partial \varphi_j(D^{(m)})}{\partial \boldsymbol{\beta}}$$

}

for all $j \notin D^{(m)}$, do {

$\boldsymbol{\varphi'} = \boldsymbol{\varphi}(D^{(m)} \cup \{j\})$

if $(\varphi'_j = 0)$ {

$$\frac{\partial \xi_j^{(m+1/2)}}{\partial \boldsymbol{\xi}^{(0)}} = \frac{\partial \xi_j^{(m)}}{\partial \boldsymbol{\xi}^{(0)}}$$

$$\frac{\partial \xi_j^{(m+1/2)}}{\partial \boldsymbol{\beta}} = \frac{\partial \xi_j^{(m)}}{\partial \boldsymbol{\beta}}$$

} else {

$$\frac{\partial \xi_j^{(m+1/2)}}{\partial \boldsymbol{\xi}^{(0)}} = \mathbf{0}$$

$$\frac{\partial \xi_j^{(m+1/2)}}{\partial \boldsymbol{\beta}} = \mathbf{0}$$

for all $i \in D^{(m)}$, do {

$$\frac{\delta \xi_i^{(m+1/2)}}{\delta \xi_j^{(m)}} = \frac{\varphi_i(D^{(m)}) - \varphi'_i}{\varphi'_j}$$

$$\frac{\partial \xi_i^{(m+1/2)}}{\partial \boldsymbol{\xi}^{(0)}} += \frac{\delta \xi_i^{(m+1/2)}}{\delta \xi_j^{(m)}} \frac{\partial \xi_j^{(m)}}{\partial \boldsymbol{\xi}^{(0)}}$$

$$\frac{\partial \xi_i^{(m+1/2)}}{\partial \boldsymbol{\beta}} += \frac{\delta \xi_i^{(m+1/2)}}{\delta \xi_j^{(m)}} \frac{\partial \xi_j^{(m)}}{\partial \boldsymbol{\beta}}$$

}

}

}

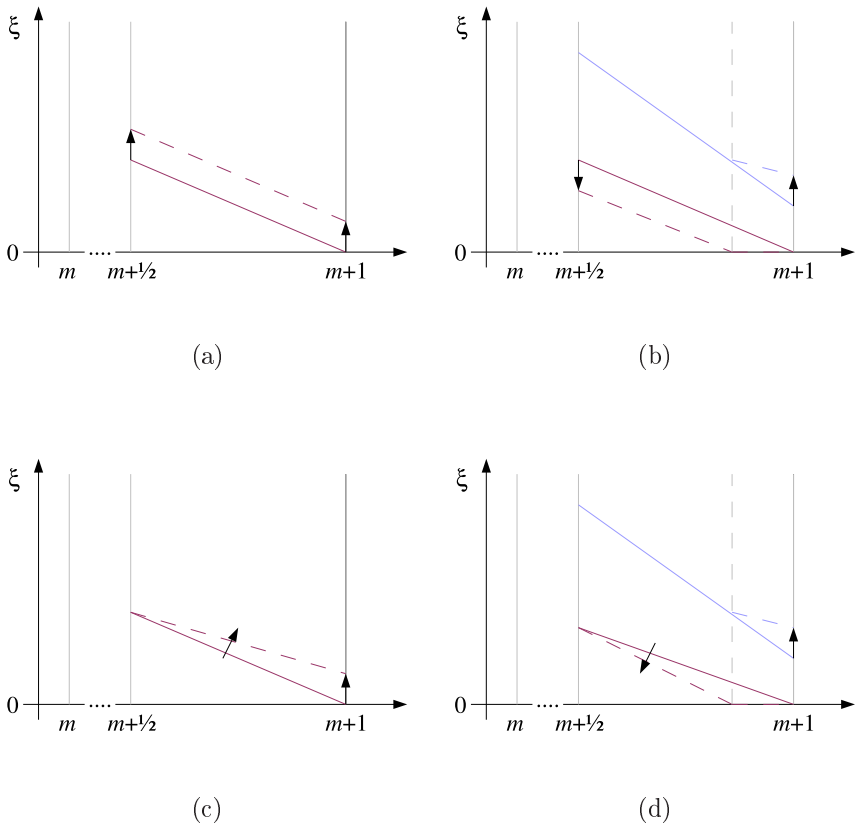


Figure B.2: Resource variations for second half of GPRC sensitivity calculation
All diagrams show resource trajectories over "GPRC time". Within each diagram, the left arrow represents the causative variation, and the right arrow represents the induced variation. Varied resources are drawn in red, and influenced resources (if any) are drawn in blue. Original trajectories are solid, and their variations are dashed.

can be used to obtain its duration $\delta\theta'' = -\delta\xi_j^{(m+1/2)}/\varphi_j(D^{(m)})$. (The negative sign in this expression is owed to the fact that $\delta\xi_j^{(m+1/2)}$ reduces $\theta^{(m)}$ and that $\delta\theta''$ is the negative of this reduction.) During $\delta\theta''$, all states $i \in D^{(m)}$, $i \neq j$, are reduced by consumption rates $\varphi_i(D'')$ instead of $\varphi_i(D^{(m)})$. This varies the subsequent $\xi_i^{(m+1)}$ by $\delta\xi_i^{(m+1)} = (\varphi_i(D^{(m)}) - \varphi_i(D''))\delta\theta''$. If a suchlike affected i belongs to $B^{(m)}$ itself, (B.2) ensures that $\varphi_i(D^{(m)}) \geq \varphi_i(D^{(m)}\backslash\{j\})$ such that $\delta\xi_i^{(m+1)} \geq 0$ results from a negative variation $\delta\xi_j^{(m+1/2)} < 0$. This eliminates the possibility of additional regime occurrences at the end of D''.

Averaging the sensitivities for positive and negative variations $\delta\xi_j^{(m+1/2)}$, one obtains

$$\frac{\delta\xi_i^{(m+1)}}{\delta\xi_j^{(m+1/2)}} = \begin{cases} 1 & i = j \notin B^{(m)} \\ 1/2 & i = j \in B^{(m)} \\ \dfrac{\varphi_i(D'') - \varphi_i(D^{(m)})}{2\varphi_j(D^{(m)})} & i \neq j \wedge i \in D^{(m)} \wedge j \in B^{(m)} \\ 0 & \text{otherwise.} \end{cases} \tag{B.8}$$

This allows to calculate the full sensitivities via

$$\frac{\partial\xi_i^{(m+1)}}{\partial\boldsymbol{\xi}^{(0)}} = \sum_j \frac{\delta\xi_i^{(m+1)}}{\delta\xi_j^{(m+1/2)}} \frac{\partial\xi_j^{(m+1/2)}}{\partial\boldsymbol{\xi}^{(0)}}. \tag{B.9}$$

B.4 Calculation of $\partial\boldsymbol{\xi}^{(m+1)}/\partial\boldsymbol{\beta}$

If $j \in D^{(m+1)}$, resource j is strictly positive at step $m+1$ so that any variation $\delta\beta_q^{[m+1/2,m+1]}$ of parameter β_q during $[m+1/2, m+1]$ only affects to $\xi_j^{(m+1)}$. This yields $\delta\xi_j^{(m+1)} = -\dfrac{\theta^{(m)}}{2}\dfrac{\partial\varphi_j(D^{(m)})}{\partial\beta_q}\delta\beta_q^{[m+1/2,m+1]}$, similarly to the effect illustrated in Figure B.1(b). If $j \notin D^{(m)}$, it is insensitive to consumption rate variations.

If $j \in B^{(m)}$, resource j can be affected by a variation $\delta\beta_q^{[m+1/2,m+1]}$. If this variation causes a decrease $\delta\varphi_j^{[m+1/2,m+1]} < 0$ of j's consumption rate, $\xi_j^{(m+1)}$ increases by $\delta_j^{(m+1)} = -\dfrac{\theta^{(m)}}{2}\dfrac{\partial\varphi_j(D^{(m)})}{\partial\beta_q}\delta\beta_q^{[m+1/2,m+1]}$, see Figure B.2(c). Given a positive $\delta\varphi_j^{[m+1/2,m+1]}$, resource j is consumed faster, which causes a regime $D'' = D\backslash\{j\}$ to occur immediately before $m+1$. The duration of D'' is

$$\begin{aligned} \delta\theta'' &= -\frac{\partial}{\partial\beta_q}\left(\frac{\xi_j^{(m+1/2)}}{\varphi_j(D^{(m)})}\right)\delta\beta_q^{[m+1/2,m+1]} \\ &= \frac{\xi_j^{(m+1/2)}}{\varphi_j^2(D^{(m)})}\frac{\partial\varphi_j(D^{(m)})}{\partial\beta_q}\delta\beta_q^{[m+1/2,m+1]} \;\Big|\; \frac{\xi_j^{(m+1/2)}}{\varphi_j(D^{(m)})} = \frac{\theta^{(m)}}{2} \\ &= \frac{\theta^{(m)}}{2\varphi_j(D^{(m)})}\frac{\partial\varphi_j(D^{(m)})}{\partial\beta_q}\delta\beta_q^{[m+1/2,m+1]}, \end{aligned} \tag{B.10}$$

see Figure B.2(d). The effect of D'' is identical to that described in the previous section.

Averaging the sensitivities for positive and negative variations $\delta\beta_q^{[m+1/2,m+1]}$, one obtains

$$\begin{aligned}\frac{\delta\xi_i^{(m+1)}}{\delta\beta_q^{[m+1/2,m+1]}} &= -\begin{cases}\dfrac{\theta^{(m)}}{2}\dfrac{\partial\varphi_i(D^{(m)})}{\partial\beta_q} & i\in D^{(m+1)}\\ \dfrac{\theta^{(m)}}{4}\dfrac{\partial\varphi_i(D^{(m)})}{\partial\beta_q} & i\in B^{(m)}\\ 0 & \text{otherwise}\end{cases}\\ &\quad\ldots-\begin{cases}\dfrac{\theta^{(m)}}{2}\displaystyle\sum_{\substack{j\in B^{(m)}\\ j\neq i}}\frac{\delta\xi_i^{(m+1)}}{\delta\xi_j^{(m+1/2)}}\frac{\partial\varphi_j(D^{(m)})}{\partial\beta_q} & i\in D^{(m)}\\ 0 & \text{otherwise,}\end{cases}\end{aligned} \tag{B.11}$$

where (B.8) could be reused because of the identical effect of D'' in this and the previous section.

A calculation of the full sensitivities is now possible via

$$\frac{\partial\xi_i^{(m+1)}}{\partial\boldsymbol{\beta}} = \frac{\delta\xi_i^{(m+1)}}{\delta\boldsymbol{\beta}^{[m+1/2,m+1]}} + \sum_j \frac{\delta\xi_i^{(m+1)}}{\delta\xi_j^{(m+1/2)}}\frac{\partial\xi_j^{(m+1/2)}}{\partial\boldsymbol{\beta}}. \tag{B.12}$$

A logic for the synchronous calculation of the second half of the state and parameter sensitivities is given in Algorithm 7.

B.5 Completition of Sensitivities

When the process has terminated at step M, the sensitivity calculations are completed by a last run of Algorithm 6 in order to account for resource variations around $m = M$. Beyond M, all resources are either blocked or zero and require no further sensitivity updates.

Algorithm 7 Second half of GPRC sensitivity calculation

for all i, do {
  if ($i \in B^{(m)}$) {
    $$\frac{\partial \xi_i^{(m+1)}}{\partial \boldsymbol{\xi}^{(0)}} = \frac{1}{2}\frac{\partial \xi_i^{(m+1/2)}}{\partial \boldsymbol{\xi}^{(0)}}$$
    $$\frac{\partial \xi_i^{(m+1)}}{\partial \boldsymbol{\beta}} = \frac{1}{2}\frac{\partial \xi_i^{(m+1/2)}}{\partial \boldsymbol{\beta}} - \frac{\theta^{(m)}}{4}\frac{\partial \varphi_i(D^{(m)})}{\partial \boldsymbol{\beta}}$$
  } else {
    $$\frac{\partial \xi_i^{(m+1)}}{\partial \boldsymbol{\xi}^{(0)}} = \frac{\partial \xi_i^{(m+1/2)}}{\partial \boldsymbol{\xi}^{(0)}}$$
    $$\frac{\partial \xi_i^{(m+1)}}{\partial \boldsymbol{\beta}} = \frac{\partial \xi_i^{(m+1/2)}}{\partial \boldsymbol{\beta}}$$
    if ($i \in D^{(m)}$)
      $$\frac{\partial \xi_i^{(m+1)}}{\partial \boldsymbol{\beta}} -= \frac{\theta^{(m)}}{2}\frac{\partial \varphi_i(D^{(m)})}{\partial \boldsymbol{\beta}}$$
  }
}
for all $j \in B^{(m)}$, do {
  $\boldsymbol{\varphi''} = \boldsymbol{\varphi}(D^{(m)} \backslash \{j\})$
  for all $i \in D^{(m)}, i \neq j$, do {
    $$\frac{\delta \xi_i^{(m+1)}}{\delta \xi_j^{(m+1/2)}} = \frac{\varphi_i'' - \varphi_i(D^{(m)})}{2\varphi_j(D^{(m)})}$$
    $$\frac{\partial \xi_i^{(m+1)}}{\partial \boldsymbol{\xi}^{(0)}} += \frac{\delta \xi_i^{(m+1)}}{\delta \xi_j^{(m+1/2)}}\frac{\partial \xi_j^{(m+1/2)}}{\partial \boldsymbol{\xi}^{(0)}}$$
    $$\frac{\partial \xi_i^{(m+1)}}{\partial \boldsymbol{\beta}} += \frac{\delta \xi_i^{(m+1)}}{\delta \xi_j^{(m+1/2)}}\left(\frac{\partial \xi_j^{(m+1/2)}}{\delta \boldsymbol{\beta}} - \frac{\theta^{(m)}}{2}\frac{\partial \varphi_j(D^{(m)})}{\partial \boldsymbol{\beta}}\right)$$
  }
}

Appendix C

Calculation of Cell Velocities

The CTM calculates flow rates directly from cell occupancies. The elementary relationship $q = v\varrho$ is used to determine cell velocity v from flow q and density ϱ.

Consider a cell that holds a density ϱ at the beginning of its next time step of duration T. The cell's length is L, and its maximum velocity is $\hat{v}$. The macroscopic simulation logic provides in- and outflow rates q^{in} and q^{out} (per lane) that persist for the duration of the next time step. The resulting density change is $(q^{\text{in}} - q^{\text{out}})T/L$. A substitution of the average density $\varrho + 0.5(q^{\text{in}} - q^{\text{out}})T/L$ and the average flow $0.5(q^{\text{in}} + q^{\text{out}})$ in $v = q/\varrho$ yields

$$v = \frac{(q^{\text{in}} + q^{\text{out}})}{2\varrho + (q^{\text{in}} - q^{\text{out}})T/L}. \tag{C.1}$$

Two further modifications are necessary to make this formula operational.

First, this logic fails for an empty network because of an undefined 0/0 division. This can be avoided by the introduction of small addends $\delta\varrho > 0$ and $\delta q = \hat{v}\delta\varrho$ in

$$v = \frac{(q^{\text{in}} + q^{\text{out}}) + \hat{v}\delta\varrho}{2\varrho + (q^{\text{in}} - q^{\text{out}})T/L + \delta\varrho}. \tag{C.2}$$

This yields $v = \hat{v}$ for an empty network. For larger occupancies, the modification's influence vanishes quickly.

Second, the resulting velocity is not limited by $\hat{v}$. Assume that $\varrho = 0 \Rightarrow q^{\text{out}} = 0$ and $\delta\varrho \to 0$. This yields $v = L/T \geq \hat{v}$ according to (2.11). Therefore,

$$v = \min\left\{\hat{v}, \frac{(q^{\text{in}} + q^{\text{out}}) + \hat{v}\delta\varrho}{2\varrho + \delta\varrho + (q^{\text{in}} - q^{\text{out}})T/L}\right\}. \tag{C.3}$$

The truncation only has an effect during transient dynamics. In stationary conditions with $q^{\text{in}} = q^{\text{out}} = q$, the velocity becomes $v = q/\varrho$, which cannot exceed $\hat{v}$ of the fundamental diagram from which q is obtained as a function of ϱ.

All experiments of this dissertation are based on velocity definition (C.3). Section 3.1.4.1 shows that the resulting vehicle movements are well-synchronized with the macroscopic flow.

Appendix D

Gridlock Resolution

Gridlock is a known problem in traffic simulations that also occurs in reality [56, 152]. Since the models employed in this thesis are relatively simple and only roughly calibrated, it is hypothesized that a simulated gridlock is likely to result from modeling imprecisions and thus needs to be resolved within the simulation. For this purpose, a simple modification to the traffic flow dynamics of Chapter 2 is subsequently described.

A minimum velocity $v^{\min}$ that is smaller than the free flow speed of any link is chosen. A reasonable value is the walking speed of 4 km/h, which implies that taking a car yields some time savings over walking. Preventing velocities below $v^{\min}$ bounds the network clearance time, thus resolves any gridlock in finite duration, and reduces the risk of gridlock occurrence by limiting queue spillovers. The minimum velocity is enforced by two modifications of the simulation logic. The following presentation assumes a single-lane cell. For multiple lanes, flow rates must be accordingly scaled.

First, the upper flow constraint of every cell's demand function is replaced by a function that increases linearly with slope $v^{\min}$, as illustrated in Figure D.1. This still complies with the demand/supply logic of the KWM since concavity is maintained. Phenomenologically, it also has little effect since all supply functions still have a horizontal flow limit.

Second, it is ensured for every cell i with a current density ϱ_i that its outflow q_i^{out} is not smaller than $v^{\min}\varrho_i$. This is equivalent to an "enforced" demand $\Delta^{\min}(\varrho) = v^{\min}\varrho$ that is pushed downstream whatever the congestion level is. The modified upper bound of the demand function ensures that the enforced demand never exceeds the original demand.

The second modification is not consistent with the KWM. The lower velocity bound implies that beyond a certain density flow is an increasing function of density even in congested conditions. Consequently, densities above jam density are possible. Although the resulting "fundamental diagram" of Figure D.1 has no counterpiece in reality, the resulting traffic dynamics give a satisfactory impression. The densities in most cells of the network stay in the feasible part of the fundamental diagram. An increased flow that is squeezed through critical sections is observed mainly at bottlenecks and roundabouts. These local

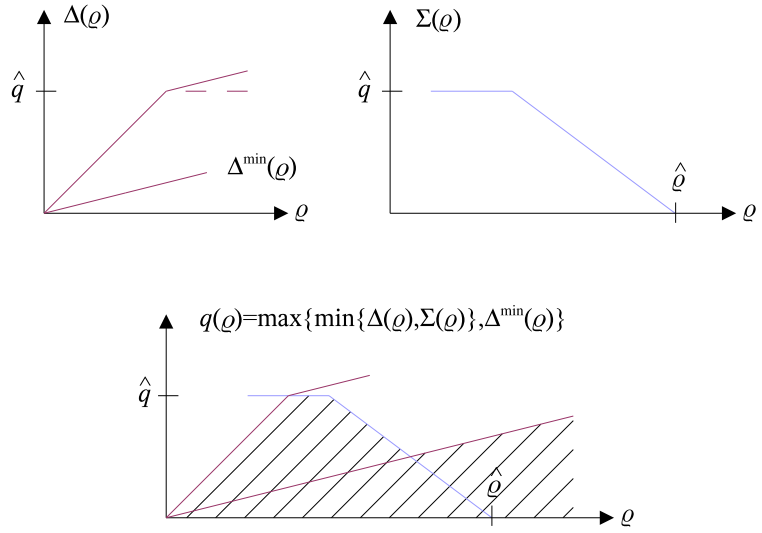


Figure D.1: Modified fundamental diagram

Effect of gridlock resolution on the fundamental diagram of a homogeneous road. The upper flow constraint of the demand function $\Delta(\varrho)$ is bent upwards at the slope of the enforced demand $\Delta^{\min}(\varrho)$ such that these two lines do not intersect. The minimum operation that originally combines demand and supply is supplemented by a lower flow bound that takes effect only at high densities.

flow modifications avoid the unrealistically heavy spillbacks that may cause a domino effect of gridlock throughout the network.

Since all involved functions are continuous, the gridlock-resolved traffic flow dynamics can still be linearized.

Appendix E

Stationary Limit of Turning Counter Variance

This appendix derives (3.9) in Section 3.1.3.2.

First, the variance of the left- and right-hand side of turning counter state equation (3.8) is noted:

$$\begin{aligned} x_{ij}(rT_c + T_c) &= w_c x_{ij}(rT_c) + (1 - w_c)\frac{1}{T_c}\sum_{s=0}^{T_c-1}\sum_{n=1}^{N} u_{ij,n}(rT_c + s) \\ \Rightarrow \mathrm{VAR}\{x_{ij}(rT_c + T_c)\} &= w_c^2 \mathrm{VAR}\{x_{ij}(rT_c)\} \\ &+ \frac{(1 - w_c)^2}{T_c^2}\mathrm{VAR}\left\{\sum_{s=0}^{T_c-1}\sum_{n=1}^{N} u_{ij,n}(rT_c + s)\right\}. \quad \text{(E.1)} \end{aligned}$$

Assuming that $\sum_{n=1}^{N} u_{ij,n}(k)$ is Poissonian with expectation and variance λ_{ij}, the stationary limit of a turning counter's variance results from the following manipulations:

$$\begin{aligned} \mathrm{VAR}\{x_{ij}(rT_c + T_c)\} &= w_c^2 \mathrm{VAR}\{x_{ij}(rT_c)\} + \frac{(1 - w_c)^2}{T_c}\lambda_{ij} \\ \Rightarrow \lim_{r\to\infty} \mathrm{VAR}\{x_{ij}(rT_c + T_c)\} &= w_c^2 \lim_{r\to\infty} \mathrm{VAR}\{x_{ij}(rT_c)\} + \frac{(1 - w_c)^2}{T_c}\lambda_{ij} \\ \Rightarrow \lim_{r\to\infty} \mathrm{VAR}\{x_{ij}(rT_c)\} &= \frac{1 - w_c}{1 + w_c}\frac{\lambda_{ij}}{T_c}. \quad \text{(E.2)} \end{aligned}$$

Bibliography

[1] E. Almasri and B. Friedrich. Online offset optimisation in urban networks based on cell transmission model. In *Proceedings of the 5th European Congress on Intelligent Transport Systems and Services*, Hannover, Germany, June 2005.

[2] R. Ansorge. What does the entropy condition mean in traffic flow theory. *Transportation Research Part B*, 24(2):133–143, 1990.

[3] C. Antoniou. *On-line Calibration for Dynamic Traffic Assignment*. PhD thesis, Massachusetts Institute of Technology, 2004.

[4] C. Antoniou, M. Ben-Akiva, and H. N. Koutsopoulos. Dynamic traffic demand prediction using conventional and emerging data sources. *IEE Proceedings Intelligent Transport Systems*, 153(1):97–104, 2006.

[5] C. Antoniou, H.N. Koutsopoulos, and G. Yannis. An efficient non-linear Kalman filtering algorithm using simultaneous pertubation and applications in traffic estimation and prediction. In *Proceedings of the 10th IEEE Intelligent Transportation Systems Conference*, pages 217–222, Seattle, USA, September/October 2007.

[6] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.

[7] K. Ashok. *Estimation and Prediction of Time-Dependent Origin-Destination Flows*. PhD thesis, Massachusetts Institute of Technology, 1996.

[8] V. Astarita, K. Er-Rafia, M. Florian, M. Mahut, and S. Velan. A comparison of three methods for dynamic network loading. *Transportation Research Record*, 1771:179–190, 2001.

[9] M. Balmer, K.W. Axhausen, and K. Nagel. Agent-based demand-modeling framework for large-scale microsimulations. *Transportation Research Record*, 1985:125–134, 2006.

[10] M. Balmer, B. Raney, and K. Nagel. Adjustment of activity timing and duration in an agent-based traffic flow simulation. In H.J.P. Timmermans, editor, *Progress in Activity-Based Analysis*, pages 91–114. Elsevier, 2005.

[11] R.J. Beckman et al. TRANSIMS–release 1.0 – The Dallas-Fort Worth case study. Los Alamos unclassified report (LA-UR) 97-4502, Los Alamos National Laboratory, Los Alamos, NM, http://transims.tsasa.lanl.gov, accessed 2008, 1997.

[12] M.G.H. Bell. The estimation of origin-destination matrices by constrained generalised least squares. *Transportation Research Part B*, 25(1):13–22, 1991.

[13] M.G.H. Bell. The real time estimation of origin-destination flows in the presence of platoon dispersion. *Transportation Research Part B*, 25(2/3):115–125, 1991.

[14] M.G.H. Bell. Stochastic user equilibrium assignment in networks with queues. *Transportation Research Part B*, 29(2):125–137, 1995.

[15] M.G.H. Bell and S. Grosso. Estimating path flows from traffic counts. In W. Brilon, F. Huber, M. Schreckenberg, and H. Wallentowitz, editors, *Traffic and Mobility: Simulation–Economics–Environment*, pages 85–102. Springer, 1999.

[16] M.G.H. Bell, W.H.K. Lam, and Y. Iida. A time-dependent multi-class path flow estimator. In J.-B. Lesort, editor, *Proceedings of the 13th International Symposium on Transportation and Traffic Theory*, pages 173–193, Lyon, France, July 1996. Pergamon.

[17] M.G.H. Bell, C.M. Shield, F. Busch, and G. Kruse. A stochastic user equilibrium path flow estimator. *Transportation Research Part C*, 5(3/4):197–210, 1997.

[18] M. Ben-Akiva and M. Bierlaire. Discrete choice methods and their applications to to short-term travel decisions. In R. Hall, editor, *Handbook of Transportation Science*, pages 5–34. Kluwer, 1999.

[19] M. Ben-Akiva, M. Bierlaire, D. Burton, H.N. Koutsopoulos, and R. Mishalani. Network state estimation and prediction for real–time transportation management applications. *Networks and Spatial Economics*, 1:293–318, 2001.

[20] M. Ben-Akiva, M.S. Ramming, and S. Bekhor. Route choice models. In M. Schreckenberg and R. Selten, editors, *Human Behaviour and Traffic Networks*, pages 23–45. Springer, 2004.

[21] M.E. Ben-Akiva and S.R. Lerman. *Discrete Choice Analysis*. MIT Press series in transportation studies. The MIT Press, 1985.

[22] M. Bierlaire. Discrete choice models. In M. Labbe, G. Laporte, K. Tanczos, and Ph. Toint, editors, *Vol. 166 of NATO ASI Series, Series F: Computer and Systems Sciences*, Operations Research in Traffic and Transportation Management, pages 203–227. Springer, 1998.

[23] M. Bierlaire and F. Crittin. An efficient algorithm for real-time estimation and prediction of dynamic od table. *Operations Research*, 52(1):116–127, 2004.

[24] P. Bonsall. Information systems and other intelligent transport system innovations. In D.A. Hensher and K.J. Button, editors, *Handbook of Transport Modelling*, pages 481–496. Elsevier, 2000.

[25] J. Bottom, M. Ben-Akiva, M. Bierlaire, I. Chabini, H. Koutsopoulos, and Q. Yang. Investigation of route guidance generation issues by simulation with DynaMIT. In A. Ceder, editor, *Proceedings of the 14th International Symposium on Transportation and Traffic Theory*, pages 577–600. Pergamon, Jerusalem, Israel, July 1999.

[26] J.A. Bottom. *Consistent Anticipatory Route Guidance*. PhD thesis, Massachusetts Institute of Technology, 2000.

[27] J.L. Bowman and M. Ben-Akiva. Activity-based travel forecasting. In *Activity-Based Travel Forecasting Conference: Summary, Recommendations and Compendium of Papers*, New Orleans, Louisiana, June 1996. http://tmip.fhwa.dot.gov/clearinghouse/, accessed 2008.

[28] E. Brockfeld and P. Wagner. Validating microscopic traffic flow models. In *Proceedings of the 9th IEEE Intelligent Transportation Systems Conference*, pages 1604–1608, Toronto, Canada, September 2006.

[29] C. Buisson, J.P. Lebacque, and J.B. Lesort. STRADA. A discretized macroscopic model of vehicular traffic flow in complex networks based on the Godunov scheme. In *Symposium on Modelling, Analysis and Simulation, held at CESA 1996 IMACS Multiconference*, volume 2, pages 976–981, Lille, France, July 1996.

[30] C. Buisson, J.P. Lebacque, J.B. Lesort, and H. Mongeot. The STRADA model for dynamic assignment. In *Proceedings of the 1996 ITS Conference*, Orlando, USA, 1996.

[31] W. Burghout. Mesoscopic simulation models for short-term prediction. PREDIKT project report CTR2005:03, Centre for Traffic Research (CTR) of Royal Institute of Technology (KTH), Swedish National Road and Transport Research Institute (VTI), and University of Linköping, http://www.infra.kth.se/ctr/publications/ctr2005_03.pdf, accessed 2008, 2005.

[32] W. Burghout, H. Koutsopoulos, and I. Andreasson. Hybrid mesoscopic-microscopic traffic simulation. *Transportation Research Record*, 1934:218–225, 2005.

[33] C. Caligaris, S. Sacone, and S. Siri. Freeway traffic modeling: extension to different vehicle classes and numerical analysis. In *Proceedings of the 10th IEEE Intelligent Transportation Systems Conference*, pages 325–330, Seattle, USA, September/October 2007.

[34] E. Cascetta. Estimation of trip matrices from traffic counts and survey data: a generalised least squares estimator. *Transportation Research Part B*, 18(4/5):289–299, 1984.

[35] E. Cascetta. *Transportation Systems Engineering: Theory and Methods*. Kluwer Academic Publishers, 2001.

[36] E. Cascetta, D. Inaudi, and G. Marquis. Dynamic estimators of origin-destination matrices using traffic counts. *Transportation Science*, 27:363–373, 1993.

[37] E. Cascetta and S. Nguyen. A unified framework for estimating or updating origin/destination matrices from traffic counts. *Transportation Research Part B*, 22(6):437–455, 1988.

[38] E. Cascetta, A. Nuzzolo, F. Russo, and A. Vitetta. A modified logit route choice model overcoming path overlapping problems. Specification and some calibration results for interurban networks. In J.-B. Lesort, editor, *Proceedings of the 13th International Symposium on Transportation and Traffic Theory*, pages 697–711, Lyon, France, July 1996. Pergamon.

[39] E. Cascetta and N.N. Posterino. Fixed point approaches to the estimation of o/d matrices using traffic counts on congested networks. *Transportation Science*, 35(2):134–147, 2001.

[40] R. Cayford, W.-H. Lin, and C.F. Daganzo. The NETCELL simulation package: technical description. California PATH research report UCB-ITS-PRR-97-23, University of California, Berkeley, 1997.

[41] N. Cetin, A. Burri, and K. Nagel. A large-scale agent-based traffic microsimulation based on queue model. In *Proceedings of the 3rd Swiss Transport Research Conference*, Monte Verita/Ascona, March 2003.

[42] E. Chin-Ping Chang. Improving traffic estimation and prediction through dynamic traffic assignment development. In *Proceedings of the 2004 IEEE International Conference on Networking, Sensing and Control*, pages 1313–1316, Taipei, Taiwan, March 2004.

[43] D. Charypar and K. Nagel. Generating complete all-day activity plans with genetic algorithms. *Transportation*, 32(4):369–397, 2005.

[44] D. Charypar and K. Nagel. Q-learning for flexible learning of daily activity plans. *Transportation Research Record*, 1935:163–169, 2005.

[45] R. Chrobok, A. Pottmeier, J. Wahle, and M. Schreckenberg. Traffic forecast using a combination of on-line simulation and traffic data. In M. Fukui, Y. Sugiyama, M. Schreckenberg, and D.E. Wolf, editors, *Traffic and Granular Flow '01*, pages 345–350. Springer, 2003.

[46] Roland Chrobok. *Theory and Application of Advanced Traffic Forecast Methods*. PhD thesis, Universität Duisburg-Essen, Germany, 2005.

[47] L. Chu, H.X. Liu, J.-S. Oh, and W. Recker. A calibration procedure for microscopic traffic simulation. In *Proceedings of the 6th IEEE Intelligent Transportation Systems Conference*, volume 2, pages 1574–1579, Shanghai, October 2003.

[48] S.L. Cohen. An approach to calibration and validation of traffic simulation models. In *Proceedings of the 83. Annual Meeting of the Transportation Research Board*, Washington, DC, USA, 2004.

[49] B. Coifman. Improved velocity estimation using single loop detectors. *Transportation Research Part A*, 35(10):863–880, 2001.

[50] M. Cremer and H. Keller. A new class of dynamic methods for the identification of origin-destination flows. *Transportation Research Part B*, 21(2):117–132, 1987.

[51] F. Crittin and M. Bierlaire. A generalization of secant methods for solving nonlinear systems of equations. In *Proceedings of the 3rd Swiss Transport Research Conference*, Monte Verita/Ascona, March 2003.

[52] F. Crittin and M. Bierlaire. Solving the anticipatory route guidance generation problem using a generalization of secant methods. In *Proceedings of the 3rd Swiss Transport Research Conference*, Monte Verita/Ascona, March 2003.

[53] C.F. Daganzo. The cell transmission model: a dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B*, 28(4):269–287, 1994.

[54] C.F. Daganzo. The cell transmission model, part II: network traffic. *Transportation Research Part B*, 29(2):79–93, 1995.

[55] C.F. Daganzo. A finite difference approximation of the kinematic wave model of traffic flow. *Transportation Research Part B*, 29(4):261–276, 1995.

[56] C.F. Daganzo. Queue spillovers in transportation networks with a route choice. *Transportation Science*, 32(1):3–11, 1998.

[57] A. De Palma and F. Marchal. Real cases applications of the fully dynamic METROPOLIS tool-box: an advocacy for large-scale mesoscopic transportation systems. *Networks and Spatial Economics*, 2:347–369, 2002.

[58] ITS program of the US department of transport web site. http://www.its.dot.gov, accessed 2008.

[59] R. Dowling, A. Skarbadonis, J. Halkias, G.M. Hale, and G. Zammit. Guidlines for calibration of microsimulation models: framework and applications. In *Proceedings of the 83. Annual Meeting of the Transportation Research Board*, Washington, DC, USA, 2004.

[60] DynaMIT web site. http://mit.edu/its/dynamit.html, accessed 2008.

[61] DYNASMART web site. http://www.dynasmart.com, accessed 2007.

[62] I. Ernst, M. Hetscher, K. Thiessenhusen, M. Ruhe, A. Börner, and S. Zuev. New approaches for real time traffic data acquisition with airborne systems. In *Proceedings of the 84. Annual Meeting of the Transportation Research Board*, Washington, DC, USA, 2005.

[63] R.M. Errico. What is an adjoint model? *Bulletin of the American Meteorological Society*, 78(11):2577–2591, 1997.

[64] S. Espie, D. Gattuso, and F. Galante. A hybrid traffic model coupling macro and behavioral micro simulation. In *Proceedings of the 85. Annual Meeting of the Transportation Research Board*, Washington, DC, USA, 2006.

[65] J. Esser and K. Nagel. Iterative demand generation for transportation simulations. In D. Hensher and J. King, editors, *The Leading Edge of Travel Behavior Research*, pages 659–681. Pergamon, 2001.

[66] O. Feldman and M.J. Maher. The optimisation of traffic signals using a cell transmission model. In *Proceedings of the 9th Meeting of the EURO Working Group on Transportation*, pages 503–507, Bari, Italy, June 2002.

[67] E. Frejinger and M. Bierlaire. Capturing correlation with subnetworks in route choice models. *Transportation Research Part B*, 41(3):363–378, 2007.

[68] B. Friedrich. Traffic monitoring and control in metropolitan areas. In *Proceedings of the 2nd International Symposium Networks for Mobility*, Stuttgart, Germany, 2004.

[69] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Electrical Science Series. Academic Press, New York and London, 1972. Chapter 7: Successive Parameter Estimation.

[70] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns*. Addison–Wesley Professional Computing Series. Addison–Wesley, 1994.

[71] T. Gärling. The feasible infeasibility of activity scheduling. In M. Schreckenberg and R. Selten, editors, *Human Behaviour and Traffic Networks*, pages 231–250. Springer, 2004.

[72] B.D. Greenshields. A study of traffic capacity. In *Proceedings of the Annual Meeting of the Highway Research Board*, volume 14, pages 448–477, 1935.

[73] H. Haj Salem, J. Chrisoulakis, M. Papageorgiou, N. Elloumi, and P. Papadakos. The use of metacor tool for integrated urban and interurban traffic control. evaluation in corridor peripherique, Paris. In *Proceedings of the Vehicle Navigation and Information Systems Conference*, pages 645–650, Yokohama, Japan, August/September 1994.

[74] R.W. Hall. Non-recurrent congestion: how big is the problem? Are traveler information systems the solution? *Transportation Research Part C*, 1(1):89–103, 1993.

[75] A. Hegyi, D. Girimonte, R. Babuska, and B. De Schutter. A comparison of filter configurations for freeway traffic state estimation. In *Proceedings of the 9th IEEE Intelligent Transportation Systems Conference*, pages 1029–1034, Toronto, Canada, September 2006.

[76] D. Helbing. A section-based queuing-theoretical model for congestion and travel time analysis in networks. *Journal of Physics A: Mathematical and General*, 36:L593–L598, 2003.

[77] S. Hinz, D. Lenhart, and J. Leitloff. Detection and tracking of vehicles in low framerate aerial image sequences. In *ISPRS Workshop, High Resolution Earth Imaging for Geospatial Information*, Hannover, Germany, May/June 2007.

[78] S.P. Hoogendoorn and P.H.L. Bovy. State-of-the-art of vehicular traffic flow modelling. *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, 215(4):283–303, 2001.

[79] J. Illenberger. Agent-based modeling of route switching behavior. Student research project, Berlin Institute of Technology, 2007.

[80] J. Illenberger, G. Flötteröd, and K. Nagel. Enhancing MATSim with capabilities of within-day re-planning. In *Proceedings of the 10th IEEE Intelligent Transportation Systems Conference*, pages 94–99, Seattle, USA, September/October 2007.

[81] Intergovernmental Panel on Climate Change. Climate change 2007: the physical science basis, summary for policymakers. Technical report, http://www.ipcc.ch/, accessed 2008, 2007.

[82] Intergovernmental Panel on Climate Change web site. http://www.ipcc.ch, accessed 2008.

[83] R.R. Jacob, M.V. Marathe, and K. Nagel. A computational study of routing algorithms for realistic transportation networks. *ACM Journal of Experimental Algorithms*, 6, 1999.

[84] Java web site. http://java.sun.com, accessed 2008.

[85] Java collections framework web site. http://java.sun.com/j2se/1.5.0/docs/guide/collections/overview.html, accessed 2008.

[86] W.L. Jin and H.M. Zhang. On the distribution schemes for determining flows through a merge. *Transportation Research Part B*, 37(6):521–540, 2003.

[87] M.C. Jones and S.K. Vines. Choosing the smoothing parameter for unordered multinomial data. *Sociedad de Estadistica e Investigacion Operativa Test*, 7(2):411–424, 1998.

[88] S.J. Julier and J.K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Proceedings of the 11th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls*, pages 182–193, Orlando, Florida, USA, April 1997.

[89] S.J. Julier and J.K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[90] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME – Journal of Basic Engineering*, 82D:35–45, 1960.

[91] J. Kamata and T. Oda. Detectors for road traffic. In M. Papageorgiou, editor, *Concise Encyclopedia of Traffic and Transportation Systems*, pages 96–101. Pergamon, 1991.

[92] O. Kaumann, K. Froese, R. Chrobok, J. Wahle, L. Neubert, and M. Schreckenberg. Online simulation of the freeway network of NRW. In D. Helbing, H.J. Hermann, M. Schreckenberg, and D.E. Wolf, editors, *Traffic and Granular Flow '99*, pages 351–356. Springer, 2000.

[93] H. Keller and G. Ploss. Real-time identification of o-d network flows from counts for urban traffic control. In N.H. Gardner and N.H.M. Wilson, editors, *Proceedings of the 10th International Symposium on Transportation and Traffic Theory*, pages 267–284, Cambridge, Massachusetts, USA, July 1987. Elsevier.

[94] B. Kerner. Verkehrsprognoseverfahren für ein Verkehrsnetz mit verkehrsgeregelten Netzknoten (Traffic prognosis method for a network with traffic controlled intersections). Patentschrift (patent specification) DE 199 40 957 C2, 2001.

[95] B. Kerner. *The Physics of Traffic*. Springer, 2004. Part IV: Engineering Applications.

[96] B.S. Kerner. Tracing and forecasting of congested patterns for highway traffic management. In *Proceedings of the 4th IEEE Intelligent Transportation Systems Conference*, pages 88–93, Oakland, CA, USA, August 2001.

[97] S.-J. Kim, W. Kim, and L.R. Rilett. Calibration of micro-simulation models using non-parametric statistical techniques. In *Proceedings of the 84. Annual Meeting of the Transportation Research Board*, Washington, DC, USA, 2005.

[98] R. Kitamura. An evaluation of activity-based travel analysis. *Transportation*, 15:9–34, 1988.

[99] R. Kitamura. Applications of models of activity behavior for activity based demand forecasting. In *Proceedings of the Activity-Based Travel Forecasting Conference*, pages 119–150, New Orleans, LA, USA, June 1996.

[100] F. Klügl, A. Bazzan, and S. Ossowski, editors. *Applications of Agent Technology in Traffic and Transportation*. Birkhäuser, 2005.

[101] A. Kotsialos, M. Papageorgiou, C. Diakaki, Y. Pavlis, and F. Middelham. Traffic flow modeling of large-scale motorway networks using the macroscopic modeling tool METANET. *IEEE Transactions on Intelligent Transportation Systems*, 3(4):282–292, 2002.

[102] J.P. Lebacque. The Godunov scheme and what it means for first order traffic flow models. In J.-B. Lesort, editor, *Proceedings of the 13th International Symposium on Transportation and Traffic Theory*, Lyon, France, July 1996. Pergamon.

[103] D.-H. Lee, X. Yang, and P. Chandrasekar. Parameter calibration for PARAMICS using genetic algorithm. In *Proceedings of the 80. Annual Meeting of the Transportation Research Board*, Washington, DC, USA, 2001.

[104] S. Lee. A cell transmission based assignment-simulation model for integrated freeway/surface street systems. Master thesis, Ohio State University, 1996.

[105] M. Lenzen, C. Dey, and C. Hamilton. Climate change. In D.A. Heshner and K.J. Button, editors, *Handbook of Transport and the Environment*, pages 37–60. Elsevier, 2003.

[106] R.J. LeVeque. *Numerical Methods for Conservation Laws*. Lectures in Mathematics: ETH Zürich. Birkhäuser, 1992.

[107] B. Li and B. De Moor. Recursive estimation based on the equality-constrained optimization for intersection origin-destination matrices. *Transportation Research Part B*, 33(3):203–214, 1999.

[108] M.J. Lighthill and J.B. Witham. On kinematic waves II. a theory of traffic flow on long crowded roads. *Proceedings of the Royal Society A*, 229:317–345, 1955.

[109] H.P. Lo, N. Zhang, and W. H. K. Lam. Estimation of an origin-destination matrix with random link choice proportions: a statistical approach. *Transportation Research Part B*, 30(4):309–324, 1996.

[110] H.P. Lo, N. Zhang, and W.H.K. Lam. Decomposition algorithm for statistical estimation of od matrix with random link choice proportions from traffic counts. *Transportation Research Part B*, 33(5):369–385, 1999.

[111] F. Logi, M. Ullrich, and H. Keller. Traffic estimation in Munich: practical problems and pragmatical solutions. In *Proceedings of the 4th IEEE Intelligent Transportation Systems Conference*, pages 416–421, Oakland, CA, USA, August 2001.

[112] S. Lorkowski and P. Wagner. Parameter calibration of traffic models in microscopic online simulations. In *Proceedings of the 84. Annual Meeting of the Transportation Research Board*, Washington, DC, USA, 2005.

[113] M. Maher. Inferences on trip matrices from observations on link volumes: a Bayesian statistical approach. *Transportation Research Part B*, 17(6):435–447, 1983.

[114] M.J. Maher, X. Zhang, and D. Van Vliet. A bi-level programming approach for trip matrix estimation and traffic control problems with stochastic user equilibrium link flows. *Transportation Research Part B*, 35(1):23–40, 2001.

[115] H. S. Mahmassani. Dynamic network traffic assignment and simulation methodology for advanced system management applications. *Networks and Spatial Economics*, 1(3/4):267–292, 2001.

[116] S. Marinosson, R. Chrobok, A. Pottmeier, J. Wahle, and M. Schreckenberg. Simulation framework for the autobahn traffic in North Rhine-Westphalia. In S. Bandini, B.B. Chopard, and M. Tomassini, editors, *Proceedings of the 5th International Conference on Cellular Automata for Research and Industry*, pages 315–324, Geneve, Switzerland, 2002.

[117] I. Matschke and B. Friedrich. Dynamic od estimation using additional information from traffic signal lights timing. In *Proceedings of the 4th Triennial Symposium on Transportation Analysis*, Sao Miguel, Azores, Portugal, June 2001.

[118] I. Matschke, B. Friedrich, and K. Heinig. Data fusion technique in the context of traffic state estimation. In *Proceedings of the 5th Triennial Symposium on Transportation Analysis*, Le Gosier, Guadaloupe, French West Indies, June 2004.

[119] MATSim web site. http://www.matsim.org, accessed 2008.

[120] K. Meister, M. Balmer, K.W. Axhausen, and K. Nagel. planomat: a comprehensive scheduler for a large-scale multi-agent transportation simulation. In *Proceedings of the 11th International Conference on Travel Behaviour Research*, Kyoto, August 2006.

[121] D.K. Merchant and G.L. Nemhauser. A model and an algorithm for the dynamic trafic assignment problems. *Transportation Science*, 12(3):183–199, 1978.

[122] L. Mihaylova and R. Boel. A particle filter for freeway traffic estimation. In *Proceedings of the 43th IEEE Conference on Decision and Control*, pages 2106–2111, Atlantis, Paradise Island, Bahamas, December 2004.

[123] Leitprojekte Mobilität in Ballungsräumen (pilot project "mobility in urban centers") web site. http://www.tuvpt.de/abgeschlossene-projekte/mobilitaet-in-ballungsraeumen.html, accessed 2008.

[124] R.A. Gingold J.J. Monaghan. Smoothed particle hydrodynamics - theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375–389, 1977.

[125] L. Munoz, X. Sun, R. Horowitz, and L. Alvarez. Traffic density estimation with the cell transmission model. In *Proceedings of the American Control Conference*, pages 3750–3755, Denver, Colorado, June 2003.

[126] L. Munoz, X. Sun, R. Horowitz, and L. Alvarez. A piecewise-linearized cell transmission model and parameter calibration methodology. In *Proceedings of the 85. Annual Meeting of the Transportation Research Board*, Washington, DC, USA, 2006.

[127] L. Munoz, X. Sun, D. Sun, and G. Gomez nd R. Horowitz. Methodological calibration of the cell transmission model. In *Proceedings of the American Control Conference*, pages 798–803, Denver, Colorado, June 2004.

[128] K. Nagel and K.W. Axhausen. Microsimulation. In D. Hensher and J. King, editors, *The Leading Edge of Travel Behavior Research*, pages 239–246. Pergamon, 2001.

[129] K. Nagel and G. Flötteröd. State estimation for traffic simulation. application for DFG funding, granted in May 2007, 2006.

[130] K. Nagel and F. Marchal. Computational methods for multi-agent simulations of travel behaviour. In K.W. Axhausen, editor, *Moving Through Nets: The Physical and Social Dimensions of Travel. Selected Papers from the 10th International Conference on Travel Behaviour Research*, pages 131–188. Elsevier, 2007.

[131] K. Nagel and P. Nelson. A critical comparison of the kinematic-wave model with observational data. In H.S. Mahmassani, editor, *Proceedings of the 16th International Symposium on Transportation and Traffic Theory*, pages 145–163, Maryland, USA, July 2005. Elsevier.

[132] K. Nagel, M. Rickert, P.M. Simon, and M. Pieck. The dynamics of iterated transportation simulations. In *Proceedings of the 3rd Triennial Symposium on Transportation Analysis*, San Juan, Puerto Rico, 1998.

[133] Y. Nie and D.-H. Lee. An uncoupled method for the equilibrium-based linear path flow estimator for origin-destination trip matrices. *Transportation Research Record*, 1783:72–79, 2002.

[134] Y. Nie, H.M. Zhang, and W.W. Recker. Inferring origin-destination trip matrices with a decoupled GLS path flow estimator. *Transportation Research Part B*, 39(6):497–518, 2005.

[135] N.L. Nihan and G.A. Davis. Recursive estimation of origin-destination matrices from input/output counts. *Transportation Research Part B*, 21(2):149–163, 1987.

[136] K. Nökel and M. Schmidt. Parallel DYNEMO: meso-scopic traffic flow simulation on large networks. *Networks and Spatial Economics*, 2(4):387–403, 2002.

[137] OLSIM web site. http://www.autobahn.nrw.de, accessed 2008.

[138] M. Papageorgiou. *Optimierung: statische, dynamische, stochastische Verfahren für die Anwendung*. Oldenbourg, 1996. exhausted.

[139] M. Papageorgiou. Schwierigkeiten bei der Anwendung von Optimierungsmethoden im Verkehrswesen (Difficulties in the application of optimization methods in transport engineering). Technical report, OptiV project, http://www.optiv.de, accessed 2008, 2006.

[140] A. Papoulis and S.U. Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill Series in Electrical and Computer Engineering. McGraw-Hill, 2002.

[141] B. Park and H. Qi. Development and evaluation of simulation model calibration procedure. In *Proceedings of the 84. Annual Meeting of the Transportation Research Board*, Washington, DC, USA, 2005.

[142] B. Park and J.D. Schneeberger. Microscopic simulation model calibration and validation: a case study of VISSIM for a coordinated actuated signal system. *Transportation Research Record*, 1856:185–192, 2003.

[143] California PATH program web site. http://www.path.berkeley.edu, accessed 2008.

[144] H.J. Payne. Models of freeway traffic and control. In *Mathematical Models of Public Systems*, volume 1, pages 51–61. Simulation Council, La Jolla, CA, USA, 1971.

[145] J.B. Pearson and R. Sridhar. A discrete optimal control problem. *IEEE Transactions on Automatic Control*, 11(2):171–174, 1966.

[146] S. Peeta and A.K. Ziliaskopoulos. Foundations of dynamic traffic assignment: the past, the present and the future. *Networks and Spatial Economics*, 1(3/4):233–265, 2001.

[147] PTV AG. Benutzerhandbuch VISUM 9.2 (user manual), 2004.

[148] M.S. Ramming. *Network Knowledge and Route Choice*. PhD thesis, Massachusetts Institute of Technology, 2002.

[149] B. Raney and K. Nagel. An improved framework for large-scale multi-agent simulations of travel behavior. In P. Rietveld, B. Jourquin, and K. Westin, editors, *Towards better performing European Transportation Systems*, pages 305–347. Routledge, 2006.

[150] P. Reinartz, T. Krauss, M. Pötsch, H. Runge, and S. Zuev. Traffic monitoring with serial images from airborne cameras. In *ISPRS Workshop: High-Resolution Earth Imaging for Geospatial Information*, Hannover, May 2005.

[151] P.I. Richards. Shock waves on highways. *Operations Research*, 4:42–51, 1956.

[152] M. Rieser and K. Nagel. Network breakdown "at the edge of chaos" in multi-agent traffic simulations. *The European Physical Journal B – Condensed Matter and Complex Systems*, 63(3):321–327, 2008.

[153] M. Rieser, K. Nagel, U. Beuck, M. Balmer, and J. Rümenapp. Truly agent-oriented coupling of an activity-based demand generation with a multi-agent traffic simulation. In *Proceedings of the 86. Annual Meeting of the Transportation Research Board*, Washington, DC, USA, 2007.

[154] C. Rommel. Automatic feedback control applied to microscopically simulated traffic. Master thesis, University of Uppsala, 2007.

[155] S. Rosswog and P. Wagner. Car–SHP: a Lagrangian particle scheme for the solution of the macroscopic traffic flow equations. In D. Helbing, H.J. Herrmann, M. Schreckenberg, and D.E. Wolf, editors, *Traffic and Granular Flow '99: Social, Traffic, and Granular Dynamics*, pages 401–406. Springer, 2000.

[156] R.-P. Schäfer, K.-U. Thiessenhusen, and P. Wagner. A traffic information system by means of real-time floating-car data. In *Proceedings of the ITS World Congress*, Chicago, USA, October 2002.

[157] H.D. Sherali, A.Narayan, and R. Sivanandan. Estimation of origin-destination trip-tables based on a partial set of traffic link volumes. *Transportation Research Part B*, 37(9):815–836, 2003.

[158] H.D. Sherali and T. Park. Estimation of dynamic origin-destination trip tables for a general network. *Transportation Research Part B*, 35(3):217–235, 2001.

[159] H.D. Sherali, R. Sivanandan, and A.G. Hobeika. A linear programming approach for synthesizing origin-destination trip tables from link traffic volumes. *Transportation Research Part B*, 28(3):213–233, 1994.

[160] S.E. Shladover. PATH at 20 – history and major milestones. In *Proceedings of the 9th IEEE Intelligent Transportation Systems Conference*, pages 584–592, Toronto, Canada, September 2006.

[161] H. W. Sorensen. Least-squares estimation: from Gauss to Kalman. *IEEE Spectrum*, 7:63–68, July 1970.

[162] H. Spiess. A maximum likelihood model for estimating origin-destination models. *Transportation Research Part B*, 21(5):395–412, 1987.

[163] SUMO web site. http://sourceforge.net/projects/sumo, accessed 2008.

[164] X. Sun, L. Munoz, and R. Horowitz. Highway traffic state estimation using improved mixture Kalman filters for effective ramp metering control. In *Proceedings of the 42th IEEE Conference on Decision and Control*, pages 6333–6338, Maui, Hawaii, USA, 2003.

[165] C.M.J. Tampere and L.H. Immers. An extended Kalman filter application for traffic state estimation using CTM with implicit mode switching and dynamic parameters. In *Proceedings of the 10th IEEE Intelligent Transportation Systems Conference*, pages 209–216, Seattle, USA, September/October 2007.

[166] K.E. Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, 2003.

[167] M. Treiber and D. Helbing. An adaptive smoothing method for traffic state identification from incomplete information. In H. Emmerich, B. Nestler, and M. Schreckenberg, editors, *Interface and Transport Dynamics: Computational Modelling*, pages 343–360. Springer, 2003.

[168] H. van Zuylen and L. G. Willumsen. The most likely trip matrix estimated from traffic counts. *Transportation Research Part B*, 14(3):281–293, 1980.

[169] Forschungsschwerpunkt Verkehrsmanagement 2010 (research focus "traffic management 2010") web site. http://www.vm2010.de, accessed 2008.

[170] VMZ Berlin web site. http://www.vmzberlin.de, accessed 2008.

[171] P. Vortisch. *Modellunterstützte Messwertpropagierung zur Verkehrslageschätzung in Stadtstrassennetzen (Model-Supported Propagation of Measured Values for Real-Time Traffic State*

Estimation in Urban Road Networks). PhD thesis, Universität Karlsruhe (TH), 2005. English version available at http://www.ptvamerica.com/docs/ModelSupportedProp.pdf, accessed 2008.

[172] P. Vovsha, M. Bradley, and J.L. Bowman. Activity-based travel forecasting models in the United States: progress since 1995 and prospects for the future. In *Proceedings of the EIRASS Conference on Progress in Activity-Based Analysis*, Maastricht, The Netherlands, May 2004.

[173] M. Vrtic, K.W. Axhausen, M.G. H. Bell, S. Grosso, and W. Matthews. Methoden zu Erstellung und Aktualisierung von Wunschlinienmatrizen im motorisierten Individualverkehr – methods for estimating and updating of origin-destination matrices from traffic counts. Technical Report 2000/379, Bundesamt für Strassen, UEVK, Bern, 2004.

[174] J. Wahle, R. Chrobok, A. Pottmeier, and M. Schreckenberg. A microscopic simulator for freeway traffic. *Networks and Spatial Economics*, 2(4):371–386, 2002.

[175] Y. Wang and M. Papageorgiou. Real-time freeway traffic state estimation based on extended Kalman filter: a general approach. *Transportation Research Part B*, 39(2):141–167, 2005.

[176] Y. Wang, M. Papageorgiou, and A. Messmer. An adaptive freeway traffic state estimator and its real-data testing part I: basic properties. In *Proceedings of the 8th IEEE Intelligent Transportation Systems Conference*, pages 531–536, Vienna, Austria, September 2005.

[177] Y. Wang, M. Papageorgiou, and A. Messmer. An adaptive freeway traffic state estimator and its real-data testing part II: adaptive capabilities. In *Proceedings of the 8th IEEE Intelligent Transportation Systems Conference*, pages 537–542, Vienna, Austria, September 2005.

[178] Y. Wang, M. Papageorgiou, and A. Messmer. RENAISSANCE: a real-time freeway network traffic surveillance tool. In *Proceedings of the 9th IEEE Intelligent Transportation Systems Conference*, pages 839–844, Toronto, Canada, September 2006.

[179] Y. Wen, R. Balakrishna, M. Ben-Akiva, and S. Smith. Online deployment of dynamic traffic assignment: architecture and run-time management. *IEE Proceedings Intelligent Transport Systems*, 153(1):76–84, 2006.

[180] L.G. Willumsen. Origin-destination matrix: static estimation. In M. Papageorgiou, editor, *Concise Encyclopedia of Traffic and Transportation Systems*, pages 315–322. Pergamon, 1991.

[181] H. Yang. Heuristic algorithms for the bilevel origin/destination matrix estimation problem. *Transportation Research Part B*, 29(4):231–242, 1995.

[182] H. Yang, T. Sasaki, and Y. Iida. Estimation of origin-destination matrices from link traffic counts on congested networks. *Transportation Research Part B*, 26(6):417–434, 1992.

[183] X. Zhou. *Dynamic Origin-Destination Demand Estimation and Prediction for Off–Line and On–Line Dynamic Traffic Assignment Operation.* PhD thesis, University of Maryland, College Park, 2004.