Bayesian calibration of dynamic traffic simulations

Gunnar Flötteröd* Michel Bierlaire† Kai Nagel‡

October 28, 2008

Report TRANSP-OR 081028
Transport and Mobility Laboratory
School of Architecture, Civil and Environmental Engineering
Ecole Polytechnique Fédérale de Lausanne
transp-or.epfl.ch

*Transport Systems Planning and Transport Telematics Laboratory, Berlin Institute of Technology & Transport and Mobility Laboratory, École Polytechnique Fédérale de Lausanne, floetteroed@vsp.tu-berlin.de (corresponding author)

†Transport and Mobility Laboratory, École Polytechnique Fédérale de Lausanne, michel.bierlaire@epfl.ch

‡Transport Systems Planning and Transport Telematics Laboratory, Berlin Institute of Technology, nagel@vsp.tu-berlin.de

Abstract

We present an operational framework for the calibration of demand models for dynamic traffic simulations. Our focus is on disaggregate simulators that represent every traveler individually. We calibrate, at a likewise individual level, arbitrary choice dimensions within a Bayesian framework, where the analyst's prior knowledge is represented by the dynamic traffic simulator itself and the measurements are comprised of sensor data such as traffic counts.

The approach is equally applicable to an equilibrium-based planning model and to a telematics model of spontaneous and imperfectly informed drivers. It is based on consistent mathematical arguments, yet applicable in a purely simulation-based environment, and, as our experimental results show, capable of estimating practically relevant scenarios in real-time.

1 Introduction

There is a broad consensus about the adequacy of microsimulators to the modeling of urban transportation systems, and a wide scope of suchlike simulation systems has been put forward, e.g., (Ben-Akiva et al., 2001; Mahmassani, 2001; Raney and Nagel, 2006; Waddell et al., 2007). The arguably most prominent advantage of microsimulators is their superior expressiveness because of their in principle arbitrarily fine-grained model structure. However, increasing the fidelity of a model also increases its degrees of freedom, which calls for more interactions to be modeled and more parameters to be identified. That is, the potentially greater expressiveness of a microsimulator is faced with a likewise increased need for modeling and calibration. Typically, the calibration of a (nontrivial) model is cast in a statistical framework and is carried out by some numerical procedure. The mathematical convenience of the model under consideration, e.g., in terms of continuity, differentiability, normality or ergodicity, defines the computational feasibility of this approach. A microsimulator easily reaches a level of detail at which most of these features are lost.

In this article, we present a mathematically consistent and computationally efficient framework for the calibration of microsimulation-based travel demand models in the context of dynamic traffic assignment (DTA). Specifically, we show how to calibrate a microscopic motorist demand simulator

from aggregate measurements of traffic flows (and, with some additional modeling effort, also of densities or velocities) that are obtained at a limited set of network locations. The problem is solved in a Bayesian setting, where the a priori assumption about every individual's choice distribution is combined with the available measurements' likelihood into an estimated posterior choice distribution. The method is entirely simulation-based in that it only requires a simulation system to represent the behavioral prior distribution and only generates realizations from the behavioral posterior distribution. Further demand parameters can be derived from these realizations. The approach is applicable both in stochastic equilibrium conditions and in non-equilibrium conditions. We present experimental results that demonstrate the method's applicability to systems with thousands of network links and hundreds of thousands of travelers.

The calibration of both DTA simulators and disaggregate demand models has received much attention in the literature. However, we are not aware of any work that estimates individual-level travel behavior within a DTA simulation system from aggregate sensor data on a practically relevant scale. All of the subsequently reviewed approaches consider either simplified or partial versions of this problem.

The most frequently adopted method for demand calibration from traffic counts is origin-destination (OD) matrix estimation. An OD matrix models the demand of a given time interval in terms of flows from every origin to every destination of a traffic system. The originally static problem was to estimate such a matrix from traffic counts, given a linear assignment mapping of demand on link flows. Various methods such as entropy maximization and information minimization (van Zuylen and Willumsen, 1980), Bayesian estimation (Maher, 1983), generalized least squares (Bell, 1991; Bierlaire and Toint, 1995; Cascetta, 1984), and maximum likelihood estimation (Spiess, 1987) were proposed to solve this task. Nonlinear assignment mappings were incorporated by a bilevel-approach that iterates between the nonlinear assignment and a linearized estimation problem (Maher et al., 2001; Yang, 1995; Yang et al., 1992) until a fixed point of this mutual mapping is reached (Bierlaire and Crittin, 2006; Cascetta and Posterino, 2001). The combined estimation of OD matrices in subsequent time slices was demonstrated in (Cascetta et al., 1993), and many originally static methods were applied to dynamical problems in this vein, e.g.,

(Ashok, 1996; Bierlaire, 2002; Sherali and Park, 2001; Zhou, 2004).

Since a time-dependent OD matrix maps (origin, destination, departure time) tuples on demand levels, it represents destination and departure time choice on an aggregate level. Route choice, however, constitutes no additional degree of freedom but is a function of demand that is defined through the DTA system's modeling assumptions. Path flow estimators (PFEs) overcome this confinement.

The seminal PFE is a macroscopic one-step network observer that estimates static path flows from link volume measurements based on a stochastic user equilibrium (SUE) modeling assumption in a congested network (Bell, 1995; Bell et al., 1997). The estimation problem is transformed into one of smooth optimization, which is iteratively solved. The model was enhanced by multiple user classes and a simple analytical queuing model to represent traffic flow dynamics (Bell et al., 1996) and was successfully implemented in various research and development projects (Bell and Grosso, 1999). The PFE's non-stochastic user equilibrium counterpart had been proposed in (Sherali et al., 1994; Sherali et al., 2003) and was further advanced in (Nie and Lee, 2002; Nie et al., 2005). PFEs also serve as OD matrix estimators since an OD flow is the sum of the path flows between its OD pair.

All PFEs and OD matrix estimators are confined to their underlying modeling assumptions. PFEs only consider static demand per time slice. Time-dependent OD matrix estimators represent demand correlations across subsequent time slices in a simplified and aggregate way, e.g., by autoregressive processes or polynomial trends (Ashok, 1996; Zhou, 2004). These approaches disregard many aspects of real travel behavior, which results from highly individualized activity patterns and likewise complex constraints (Bowman and Ben-Akiva, 1998; Kitamura, 1988; Kitamura, 1996; Vovsha et al., 2004). That is, even if a PFE or an OD matrix estimator is applied to a fully microscopic DTA simulator, the aggregate estimator is unable to account for those facets that amount to the microscopic modeling approach.

Random utility models (RUMs) capture travel behavior at the individual level, and sophisticated calibration procedures for this class of models are available (Ben-Akiva and Lerman, 1985; Bierlaire, 2003; Train, 2003). However, in order to maintain tractability, their calibration procedures require a mathematically well-behaved link between observations and model parameters. Here, this link is given through a DTA microsimulator. We are

not aware of any work that calibrates a RUM in suchlike conditions.

A calibration of the UrbanSim microsimulator in a Bayesian setting was recently reported (Sevcikova et al., 2007), where a sampling importance resampling (SIR) type algorithm is applied to the estimation of almost 300 model parameters. However, concerns regarding the computation times for larger problems are mentioned.

The remainder of this article is organized as follows. Section 2 introduces the terminology of our calibration framework and specifies the requirements for its application. Section 3 derives the Bayesian estimator at a conceptual level, and Section 4 describes its operationalization. Three exemplary advancements towards more specific applications are given in Section 5. Case studies of practically relevant size are presented in Section 6. Finally, Section 7 concludes the article.

2 Framework requirements

This section is organized in two parts. First, the considered type of DTA simulator is described. Second, the additional requirements for an application of our calibration methodology are specified.

Throughout this article, we denote probability density functions by a lowercase p and probability mass functions by an uppercase P. Instead of noting the probability that random variable X takes value x by some expression of the form $\Pr(X = x)$, we briefly write $\Pr(x)$ and avoid ambiguities by self-explanatory variables.

2.1 Considered DTA simulator

Agents and plans

We assume a microsimulation-based approach where every traveler is modeled as an individual **agent** $n = 1 \ldots N$. At every point in simulated time, every agent n disposes of a **plan** $\mathcal{U}_n$ that describes the intended travel behavior of that agent. A typical plan comprises a sequence of trips that connect intermediate stops during which activities are conducted, including all associated timing information. We

subsequently write $\{\mathcal{U}\}$ as a shortcut for the whole population's plan set $\{\mathcal{U}_1, \ldots, \mathcal{U}_N\}$.

Supply simulator

The **supply simulator** executes the plans of all agents simultaneously on the network. It models the physical interactions of the agents, including congestion. The result of such a **network loading** are the **network conditions** $\mathcal{X}$, which comprise all time-dependent, aggregate network characteristics (such as flows, densities, velocities) that are relevant to the decision making of the agents.

Formally, the supply simulator draws from a distribution $p(\mathcal{X}|\{\mathcal{U}\})$ of the network conditions given a particular plan set $\{\mathcal{U}\}$ in the population. In its most widespread form, this distribution is implicitly defined through a stochastic supply microsimulator. However, a deterministic, macroscopic supply simulator where $p(\mathcal{X}|\{\mathcal{U}\})$ collapses into a singleton is just as feasible.

Demand simulator

The **demand simulator** models the decision making of travelers. It maps, for every agent $n = 1 \ldots N$ individually, the prevailing network conditions $\mathcal{X}$ on a plan $\mathcal{U}_n$ the agent chooses in these conditions. We denote by $P_n(\mathcal{U}_n|\mathcal{X})$ the probability that plan $\mathcal{U}_n$ is chosen by agent n in network conditions $\mathcal{X}$, and we denote by C_n agent n's **choice set** of available plan alternatives.

It is assumed that the agents' plan choice distributions are independent once the network conditions are given. That is,

$$P(\{\mathcal{U}\}|\mathcal{X}) = \prod_{n=1}^{N} P_n(\mathcal{U}_n|\mathcal{X}), \tag{1}$$

which implies that the agents do not interact directly but only through the aggregate network conditions. This is a reasonable assumption for large-scale and/or time-critical simulations where traffic flow dynamics are typically represented by aggregate laws of motion ("mesoscopic simulators") instead of vehicle-by-vehicle interactions ("car-following models") (Astarita et al., 2001; Ben-Akiva et al., 2001; De Palma and Marchal, 2002; Mahmassani, 2001; Nökel and Schmidt, 2002).

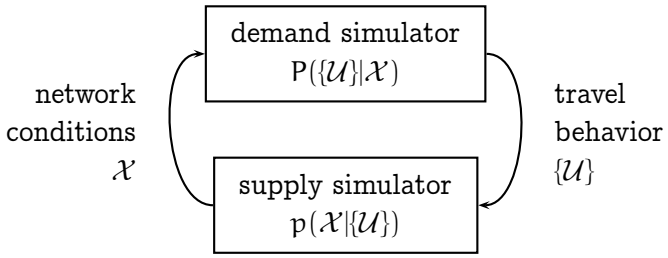

Figure 1: Interactions between demand and supply

The choice distributions $P_n(\mathcal{U}_n|\mathcal{X})$ and the choice sets C_n are arbitrary and entirely transparent to the proposed calibration approach. The demand simulator is only required to generate realizations of these distributions.

Iterative simulation logic

So far, the DTA simulator is defined in terms of a supply simulator and a demand simulator. The supply simulator specifies the conditional probability $p(\mathcal{X}|\{\mathcal{U}\})$ of the network conditions given a set of plans, and the demand simulator specifies the conditional probability $P(\{\mathcal{U}\}|\mathcal{X})$ of the population's plan choice given the network conditions.

A solution to the DTA problem can be expressed in terms of a joint distribution $p(\{\mathcal{U}\}, \mathcal{X})$ that describes a situation in which demand and supply are consistent with each other, cf. Figure 1. It typically is impossible to draw directly from this distribution, but it is possible to alternately draw from the two conditionals $p(\mathcal{X}|\{\mathcal{U}\})$ and $P(\{\mathcal{U}\}|\mathcal{X})$. After a burn-in period, these draws can be tested for convergence towards a stationary distribution, and their continuation in stationary conditions allows to extract the relevant characteristics of $p(\{\mathcal{U}\}, \mathcal{X})$ (Cascetta, 1989; Nagel et al., 1998; Ross, 2006; Watling and Hazelton, 2003). This simulation logic is drafted in Algorithm 1 where the iteration **cycle counter** is denoted by c.[1]

Speaking in terms of plans, Algorithm 1 constitutes a Markov Chain Monte Carlo procedure that utilizes a transition distribution

$$P(\{\mathcal{U}\}^{(c)}|\{\mathcal{U}\}^{(c-1)}) = \int P(\{\mathcal{U}\}^{(c)}|\mathcal{X})p(\mathcal{X}|\{\mathcal{U}\}^{(c-1)})d\mathcal{X} \qquad (2)$$

[1]This logic clearly resembles Gibbs sampling. We proceed pragmatically at this point in that we assume a converging simulation system of the above type to be available, but we do not make further assumptions about its technicalities.

Algorithm 1 Iterative dynamic traffic assignment

1. Initialize cycle counter $c = 0$.
2. Choose initial network conditions $\mathcal{X}^{(0)}$ (e.g., free-flow conditions).
3. Repeat for as many iterations as necessary to extract relevant characteristics in stationary conditions:
   (a) Increase c by one.
   (b) Replanning. For $n = 1 \dots N$, draw $\mathcal{U}_n^{(c)}$ from $P_n(\mathcal{U}_n^{(c)}|\mathcal{X}^{(c-1)})$.
   (c) Network loading. Draw $\mathcal{X}^{(c)}$ from $p(\mathcal{X}^{(c)}|\{\mathcal{U}\}^{(c)})$.

to eventually draw from a stationary distribution of plan occurrences that satisfies

$$P(\{\mathcal{U}\}) = \sum_{\{\mathcal{V}\}} P(\{\mathcal{U}\}|\{\mathcal{V}\})P(\{\mathcal{V}\}) \tag{3}$$

where $\sum_{\{\mathcal{V}\}}$ sums over all possible plan sets $\{\mathcal{V}\}$.

This logic is equally applicable to simulate a SUE-based planning model and a telematics model where drivers are spontaneous and imperfectly informed. From a simulation point of view, the only difference between these two models is that a SUE demand simulator typically utilizes all information from the most recent network loading, whereas a telematics demand simulator generates every elementary decision of a plan only based on such information that could have actually been gathered up to the according point in simulated time (Bottom, 2000).

2.2 Additional calibration requirements

Likelihood function

Denote by $\mathcal{Y}$ the sensor data that is available for the duration of the considered simulation/estimation period. It comprises trajectories of traffic flows, densities or velocities that are obtained at a limited set of network locations. Since these are aggregate traffic characteristics, they are assumed to follow a distribution $p(\mathcal{Y}|\mathcal{X})$ that is only dependent on the aggregate network conditions $\mathcal{X}$, which in turn result from the population's plan choice $\{\mathcal{U}\}$. The resulting distribution

of the sensor data $\mathcal{Y}$ given the plans $\{\mathcal{U}\}$ constitutes the likelihood function

$$p(\mathcal{Y}|\{\mathcal{U}\}) = \int p(\mathcal{Y}|\mathcal{X})p(\mathcal{X}|\{\mathcal{U}\})d\mathcal{X}. \quad (4)$$

This is a complicated function, which comprises the distribution $p(\mathcal{X}|\{\mathcal{U}\})$ that is only implicitly defined through the supply simulator. For a stochastic supply simulator, it also requires to solve a high-dimensional integral.

In order to deal with these difficulties, we rely on an efficiently evaluable approximation

$$p(\mathcal{Y}|\{\mathcal{U}\}) \approx \text{const} \cdot \prod_{n=1}^{N} p(\mathcal{Y}|\mathcal{U}_n). \quad (5)$$

For now, we constrain ourselves to the observation that this approximation can likewise be written as

$$\ln p(\mathcal{Y}|\{\mathcal{U}\}) \approx \text{const} + \sum_{n=1}^{N} \ln p(\mathcal{Y}|\mathcal{U}_n). \quad (6)$$

That is, we require the log-likelihood to be approximated by a linear superposition of individual log-likelihoods for all agents, which will turn out to be a manageable problem under moderate assumptions. This approximation will be utilized to decompose the calibration problem in a computationally tractable way.

Interface with simulation software

So far, the presentation has been at the conceptual level. Now, we account for the fact that the calibration logic needs to be inserted into a concrete simulation system. This can be realized in the following way.

Whenever the simulation sequence controller, i.e., the software that is responsible for calling the sub-models of the simulator in a meaningful order, requires that an agent selects a plan, it executes the demand simulator for that agent. This function call needs to be diverted to the calibration logic, which in turn calls the demand simulator one or more times and returns the result of exactly one of these calls. Basically, this allows the estimator to evaluate a whole set of behavioral alternatives in consideration of the measurements before selecting the most realistic one for actual execution in the simulation system.

3 Bayesian calibration

Aggregate measurements alone are insufficient to calibrate the plan choice distributions of all agents because usually there are many plan combinations that generate the same observations. Here, this problem is resolved by the incorporation of additional behavioral information in a Bayesian setting.

3.1 General approach

Assume that an iterative DTA simulator as specified in Section 2 is available and that this simulator repeatedly draws from the **prior transition distribution** $\mathsf{P}(\{\mathcal{U}\}^{(c)}|\{\mathcal{U}\}^{(c-1)})$ defined in (2) in order to ultimately draw from the **stationary prior distribution** $\mathsf{P}(\{\mathcal{U}\})$ defined in (3).

The according **stationary posterior distribution** $\mathsf{P}(\{\mathcal{U}\}|\mathcal{Y})$ of the plan choice given the measurements $\mathcal{Y}$ is, according to Bayes' law,

$$\mathsf{P}(\{\mathcal{U}\}|\mathcal{Y}) = \frac{\mathsf{p}(\mathcal{Y}|\{\mathcal{U}\})}{\mathsf{p}(\mathcal{Y})}\mathsf{P}(\{\mathcal{U}\}). \tag{7}$$

Substituting (3) in this expression yields, after some manipulations,

$$\mathsf{P}(\{\mathcal{U}\}|\mathcal{Y}) = \sum_{\{\mathcal{V}\}} \mathsf{P}(\{\mathcal{U}\}|\{\mathcal{V}\},\mathcal{Y})\mathsf{P}(\{\mathcal{V}\}|\mathcal{Y}) \tag{8}$$

where

$$\mathsf{P}(\{\mathcal{U}\}|\{\mathcal{V}\},\mathcal{Y}) = \frac{\mathsf{p}(\mathcal{Y}|\{\mathcal{U}\})\mathsf{P}(\{\mathcal{U}\}|\{\mathcal{V}\})}{\sum_{\{\mathcal{W}\}}\mathsf{p}(\mathcal{Y}|\{\mathcal{W}\})\mathsf{P}(\{\mathcal{W}\}|\{\mathcal{V}\})}. \tag{9}$$

$\{\mathcal{W}\}$ is yet another plan set variable. A derivation of this result can be found in Appendix A.

The calibration objective is to draw from the posterior (8) instead of the prior (3). The structural coincidence of both equations indicates that, just as a DTA simulator generates a sequence of draws from the prior transition distribution $\mathsf{P}(\{\mathcal{U}\}^{(c)}|\{\mathcal{U}\}^{(c-1)})$ in order to eventually represent the stationary prior distribution, a sequence of draws from the **posterior transition distribution** $\mathsf{P}(\{\mathcal{U}\}^{(c)}|\{\mathcal{U}\}^{(c-1)},\mathcal{Y})$ eventually represents the stationary posterior distribution. Relying on this assumption, we need to generate draws from the posterior transition distribution (9) in order to solve the calibration problem.

The DTA simulator draws from the prior transition distribution defined in (2) by the following two steps:

1. Draw $\mathcal{X}^{(c-1)}$ from $p(\mathcal{X}^{(c-1)}|\{\mathcal{U}\}^{(c-1)})$ by a run of the supply simulator.
2. Draw $\{\mathcal{U}\}^{(c)}$ from the **prior choice distribution** $P(\{\mathcal{U}\}^{(c)}|\mathcal{X}^{(c-1)})$, cf. (1), by running the demand simulator for all agents.

We would like to maintain this logic in the calibration. Since we are only interested in a calibration of the behavioral model, we assume the supply simulator to be correctly modeled. Then, drawing from the posterior transition distribution requires the following two steps:

1. Draw $\mathcal{X}^{(c-1)}$ from $p(\mathcal{X}^{(c-1)}|\{\mathcal{U}\}^{(c-1)})$. There is no need to account for the measurements here because the supply simulator is accurately modeled.
2. Draw $\{\mathcal{U}\}^{(c)}$ from the **posterior choice distribution**

$$P(\{\mathcal{U}\}^{(c)}|\mathcal{X}^{(c-1)}, \mathcal{Y}) = \frac{p(\mathcal{Y}|\{\mathcal{U}\}^{(c)})P(\{\mathcal{U}\}^{(c)}|\mathcal{X}^{(c-1)})}{\sum_{\{\mathcal{W}\}} p(\mathcal{Y}|\{\mathcal{W}\})P(\{\mathcal{W}\}|\mathcal{X}^{(c-1)})}. \tag{10}$$

   in consideration of the measurements. That is, incorporate the sensor data into the demand simulation.

The ambition to sample from distribution (10) is faced with two major challenges:

- The likelihood $p(\mathcal{Y}|\{\mathcal{U}\}^{(c)})$ is a complicated function. According to (4), it comprises the network loading, and, therefore, it is expensive to evaluate and not available in closed form.
- The prior choice distribution $P(\{\mathcal{U}\}^{(c)}|\mathcal{X}^{(c-1)})$ is specified in a likewise simulation-based manner. Only draws from this distribution are available through the demand simulator.

We proceed by presenting a tractable approach to these problems, where we assume that a factorized likelihood (5) is available. Hereafter, we describe how this factorization is obtained.

3.2 Individual-level approach

We begin by expanding the denominator of the posterior choice distribution (10):

$$P(\{\mathcal{U}\}^{(c)}|\mathcal{X}^{(c-1)},\mathcal{Y}) = \frac{p(\mathcal{Y}|\{\mathcal{U}\}^{(c)})P(\{\mathcal{U}\}^{(c)}|\mathcal{X}^{(c-1)})}{\sum_{\mathcal{W}_1\in C_1}\cdots\sum_{\mathcal{W}_N\in C_N} p(\mathcal{Y}|\{\mathcal{W}\})P(\{\mathcal{W}\}|\mathcal{X}^{(c-1)})}. \quad (11)$$

A substitution of the prior choice distribution (1) and the factorized likelihood (5) yields, by an application of the distributive law,

$$P(\{\mathcal{U}\}^{(c)}|\mathcal{X}^{(c-1)},\mathcal{Y}) \approx \prod_{n=1}^{N} P_n(\mathcal{U}_n^{(c)}|\mathcal{X}^{(c-1)},\mathcal{Y}) \quad (12)$$

where we define the **individual-level posterior choice distribution**

$$P_n(\mathcal{U}_n^{(c)}|\mathcal{X}^{(c-1)},\mathcal{Y}) = \frac{p(\mathcal{Y}|\mathcal{U}_n^{(c)})P_n(\mathcal{U}_n^{(c)}|\mathcal{X}^{(c-1)})}{\sum_{\mathcal{W}_n\in C_n} p(\mathcal{Y}|\mathcal{W}_n)P_n(\mathcal{W}_n|\mathcal{X}^{(c-1)})}. \quad (13)$$

The pivotal feature of this approximation is that the population's joint posterior choice distribution (10) is decomposed into a product of individual-level posterior choice distributions (13) that can be evaluated agent by agent. Operationally, this is convenient since it allows to maintain the demand simulator's individual-level choice logic during the calibration. Logically, we have decomposed the joint calibration problem for N agents into N much smaller problems.

Given that the individual-level likelihood $p(\mathcal{Y}|\mathcal{U}_n)$ is available, a universally applicable method to draw from the individual-level posterior (13) can be given. Denote by $P_{\text{accept},n}(\mathcal{U}_n|\mathcal{Y})$ the **acceptance probability** for plan $\mathcal{U}_n$ from agent n's choice set C_n. It is defined by

$$P_{\text{accept},n}(\mathcal{U}_n|\mathcal{Y}) = p(\mathcal{Y}|\mathcal{U}_n)/D_n \quad (14)$$

where D_n must be such that

$$D_n \geq \max_{\mathcal{W}_n\in C_n} p(\mathcal{Y}|\mathcal{W}_n) \quad (15)$$

for (14) to be a proper probability. If repeated draws taken from the individual-level prior $P_n(\mathcal{U}_n|\mathcal{X}^{(c-1)})$ are accepted with probability $P_{\text{accept},n}(\mathcal{U}_n|\mathcal{Y})$ and are rejected otherwise, then the first accepted draw constitutes a draw

Algorithm 2 Generic accept/reject estimator

1. Initialize cycle counter $c = 0$.
2. Choose initial network conditions $\mathcal{X}^{(0)}$ (e.g., free-flow conditions).
3. Repeat for as many iterations as necessary to extract relevant characteristics in stationary conditions:
   (a) Increase c by one.
   (b) Replanning. For $n = 1 \ldots N$, do:
      i. Draw $\mathcal{U}_n^{(c)}$ from $P_n(\mathcal{U}_n^{(c)}|\mathcal{X}^{(c-1)})$.
      ii. With probability $1 - P_{\text{accept},n}(\mathcal{U}_n^{(c)}|\mathcal{Y})$, goto step 3(b)i.
      iii. Retain the first accepted draw $\mathcal{U}_n^{(c)}$.
   (c) Network loading. Draw $\mathcal{X}^{(c)}$ from $p(\mathcal{X}^{(c)}|\{\mathcal{U}\}^{(c)})$.

from the individual-level posterior $P_n(\mathcal{U}_n|\mathcal{X}^{(c-1)}, \mathcal{Y})$. Algorithm 2 outlines this procedure. Its correctness is verified in Appendix B.

The technical requirements for an application of this method coincide with the specification given in Section 2.2: The simulation sequence controller eventually asks every agent for its selected plan, which, by definition, is a draw from that agent's prior choice distribution. If this call is redirected to an implementation of the above procedure, the suchlike generated plan is a sample from the agent's posterior choice distribution. That is, only step 3b of Algorithm 1 needs to be modified.

The only assumption made here is that the likelihood can be factorized with sufficient precision and in a computationally efficient way. The type of behavior to be estimated and the prior choice distribution implemented by the demand simulator are arbitrary. If a choice set enumeration is possible, the acceptance probabilities' denominator should be chosen such that strict equality holds in (15) in order to minimize the number of rejected draws. If a choice set enumeration is infeasible, e.g., because the choice set is implicitly specified through a constructive behavioral algorithm, this denominator can be 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 that are based on acceptance "probabilities" greater than one. The frequency at which such values occur allows to appraise the resulting loss in estimation precision. This provides a practically attractive balancing mechanism between estimation precision and computational efficiency, and it goes without a choice set enumeration.

In principle, the accept/reject procedure could also be deployed to draw from the exact posterior choice distribution (10) by generating joint proposal draws $\{\mathcal{U}\}$ for the whole population and using an acceptance probability that is proportional to the full likelihood $p(\mathcal{Y}|\{\mathcal{U}\})$. However, this approach is computationally infeasible in all but trivial cases since each evaluation of the suchlike re-defined acceptance probability requires an evaluation of the exact likelihood, including a network loading, while the number of draws that is needed until a single accept occurs may become very large.

4 Making the framework operational

The previous section relies on the availability of a factorized likelihood (5). In this section, we describe how this factorization can be obtained, given an arbitrary supply simulator. Instead of factorizing the likelihood directly, we equivalently linearize the log-likelihood.

4.1 Preparatory modeling assumptions

Let a plan $\mathcal{U}_n$ be formally specified as a large vector of indicator variables which only take values of zero or one. Every such plan element indicates if the considered agent n desires to execute a particular **elementary action**. There are three types of elementary actions: network entries, network exits, and turning moves within the network. Every elementary action comes with an associated time stamp. Examples are "enter the network on link 3043 at 7:30", "turn from link 3043 to link 9425 at 7:34", and "leave the network on link 9425 at 7:39".

We introduce a fictitious "null" link that represents any location outside of the road network. It allows for a unified representation of all elementary actions as (from-link, to-link, time step) triples where a network entry is identified by a null from-link and a network exit is identified by a null to-link. The aforementioned indicator variables that comprise plan $\mathcal{U}_n$ of agent n are subsequently denoted by $u_{ij,n}(k) \in \{0, 1\}$ where i is the from-link, j is the to-link, and k is the time step. The corresponding **demand turning flow** $d_{ij}(k)$ reflects the desired turning flow in the entire population. It is specified by

$$d_{ij}(k) = \frac{1}{T} \sum_{n=1}^{N} u_{ij,n}(k) \tag{16}$$

where T is the simulation time step duration. For mathematical convenience, we subsequently treat this flow as a real-valued quantity although it is clear that a finite agent population can only generate discrete flow levels.

Being unable to predict the exact travel times in the network, an agent is likely to observe an inconsistency between the intended timing of its elementary actions and what it experiences in the supply simulator. In an equilibrium-based planning simulation, this conflict is iteratively resolved until the observations of all agents become consistent with their expectations. In an iterated telematics simulation, however, the agents' predictive abilities are constrained by the very telematics modeling assumption such that this type of consistency cannot be expected, cf. the last paragraph of Section 2.1.

Our subsequent discussion builds on the availability of plan sets $\{\mathcal{U}\}$ that are unbiased with respect to the resulting network conditions $\mathcal{X}$ in that the timings of their elementary actions have been adjusted during or after the network loading such that the plans do not systematically deviate from what the agents have actually experienced during the last iteration. This (technically straightforward) procedure is necessary to evaluate the physical implications of the population's choices without being misled by the simulated lack of information in telematics conditions. Note that this post-processing can be conducted outside of the simulation loop and thus takes no effect on the model specification. In particular, it implies that (16) exhibits no systematic bias in comparison to the simulated network flows. However, this does not eliminate the random inconsistencies between plans

and network conditions that result from the continuous variability of the travel times even in stationary conditions.

4.2 Linear network loading

A **linear network loading** describes a situation in which the time-dependent travel times on all links in the network are known and fixed. This implies that there are no interactions between the flows, which move through an exogenously specified network environment. The resulting flow on any link becomes a linear superposition of all path flows across that link. For a microsimulator, this implies that the agents linearly superpose on each link. In order to obtain a mathematically tractable relation between demand and resulting network conditions, we formally approximate the true dynamics of the supply simulator by a linear network loading based on the most recent iteration's travel times.

These assumptions imply that the demand flows $d_{ij}(k)$ are not only unbiased but even perfect predictors of the physically occurring network flows, i.e,

$$q_j(k) = \sum_i d_{ij}(k) \tag{17}$$

where $q_j(k)$ denotes the simulated entry flow of link j in time step k. This goes beyond the previously assumed ex-post consistency of demand and resulting travel times because here it is assumed that travel times do not change with demand levels at all. Possible imprecisions in this approximation result from two causes. First, there is the continuous variability of the real travel times even in stationary conditions. Second, the assumption of constant travel times implies that the inflow of links at the capacity limit increases beyond this limit if the demand continues to grow. That is, (17) becomes an increasingly imprecise representation of the supply simulator as congestion increases.[2] An alternative approximation that properly captures congestion is discussed in Section 5.1.

Denote the measured counterpart of $q_j(k)$ by $y_j(k)$ and maintain the symbol $\mathcal{Y}$ for the set of all available measurements. For simplicity and without loss

[2]Note that a "proportional assignment", which is widely and successfully assumed in the field of time-dependent OD matrix estimation, implies the same assumption of constant travel times. That is, although (17) is consistent only in uncongested conditions, the state of practice suggests its applicability even in the case of congestion.

of generality, we only consider univariate normal measurement distributions of link entry flows[3] such that the log-likelihood takes the form

$$\ln p(\mathcal{Y}|\{\mathcal{U}\}) = \text{const} - \sum_j \sum_k \frac{(q_j(k) - y_j(k))^2}{2\sigma_j^2} \tag{18}$$

where σ_j^2 is the variance of the sensor data on link j. Linearizing this function with respect to the (presumably real-valued) flows yields

$$\ln p(\mathcal{Y}|\{\mathcal{U}\}) \approx \text{const} + \sum_j \sum_k \frac{y_j(k) - q_j^0(k)}{\sigma_j^2} q_j(k) \tag{19}$$

where $q_j^0(k)$ are the flows around which the linearization takes place and the constant addend comprises all terms independent of the flows $q_j(k)$. A substitution of (17) and (16) in this equation results in

$$\ln p(\mathcal{Y}|\{\mathcal{U}\}) \approx \text{const} + \sum_{n=1}^{N} \sum_{ijk} \lambda_{ij}(k) u_{ij,n}(k) \tag{20}$$

where

$$\lambda_{ij}(k) = \frac{y_j(k) - q_j^0(k)}{T\sigma_j^2}. \tag{21}$$

Through this, the log-likelihood is approximated by a linear superposition of agent-specific terms. Every (real-valued) $\lambda_{ij}(k)$ coefficient represents the effect of a single elementary action on the log-likelihood. Apart from the measurements and their characteristics, only the simulated flows $q_j^0(k)$ on all sensor-equipped links j need to be known.

Collecting all $\lambda_{ij}(k)$ elements in a single vector $\Lambda = (\lambda_{ij}(k))_{ijk}$, we utilize the inner product

$$\langle \Lambda, \mathcal{U}_n \rangle = \sum_{ijk} \lambda_{ij}(k) u_{ij,n}(k) \tag{22}$$

to write

$$\ln p(\mathcal{Y}|\{\mathcal{U}\}) \approx \text{const} + \sum_{n=1}^{N} \langle \Lambda, \mathcal{U}_n \rangle. \tag{23}$$

[3]The incorporation of flow sensors that are not located at the immediate upstream end of a link only requires to account for the (constant) lag until entering vehicles reach the sensor. Since the density on a link results from an integration of its in- and outflow, and since the velocity on a link can be related to its flow and density, measurements of these quantities can also be accounted for with some additional modeling effort.

The desired likelihood factorization is now obtained by applying the exponential function to both sides of (23). This results in

$$p(\mathcal{Y}|\{\mathcal{U}\}) \approx \text{const} \cdot \prod_{n=1}^{N} \exp\langle \Lambda, \mathcal{U}_n \rangle, \tag{24}$$

which implies a definition

$$p(\mathcal{Y}|\mathcal{U}_n) = \exp\langle \Lambda, \mathcal{U}_n \rangle \tag{25}$$

of the individual-level likelihood used in (5) to approximate the full likelihood. $p(\mathcal{Y}|\mathcal{U}_n)$ itself is not a proper probability density function because it does not integrate up to one over all $\mathcal{Y}$ values, which, however, is irrelevant for the purposes of the approximation.

4.3 Algorithm

The previous section decomposes the full likelihood $p(\mathcal{Y}|\{\mathcal{U}\})$ into the factors (25). This approximation relies on a reasonably precise linearization of the log-likelihood. However, given the plan set of a particular DTA iteration, a local linearization of the log-likelihood around the resulting network conditions under the assumption of constant travel times may not be an appropriate model of the log-likelihood given a different plan set and different network conditions. We therefore propose to identify the *expected* values of the Λ coefficients. More specifically, we are interested in the expectations in *posterior* conditions since these conditions prevail upon convergence of the calibration, where we want the likelihood factorization to hold with greatest precision.

When identifying the expectations of Λ, the difference between prior and posterior network conditions needs to be accounted for. We propose a fixed-point approach for this purpose: 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 on stable $\bar{\Lambda}$ coefficients that in turn are most appropriate given this very posterior. For illustrative purposes, the method of successive averages (MSA) is applied to this problem in Algorithm 3. Note that the acceptance probabilities $P_{\text{accept},n}(\mathcal{U}_n|\mathcal{Y}) \propto p(\mathcal{Y}|\mathcal{U}_n)$, cf. (14), are now replaced by $P_{\text{accept},n}(\mathcal{U}_n|\bar{\Lambda}) \propto \exp\langle \bar{\Lambda}, \mathcal{U}_n \rangle$ according to (25).

Algorithm 3 Linearization-based accept/reject estimator

1. Initialize cycle counter $c = 0$.
2. Choose initial network conditions $\mathcal{X}^{(0)}$ (e.g., free-flow conditions).
3. Repeat for as many iterations as necessary to extract relevant characteristics in stationary conditions:
   (a) Increase c by one.
   (b) Linearization. $\Lambda^{(c)} = \left(\frac{y_j(k) - q_j^{(c)}(k)}{T\sigma_j^2} \right)_{ijk}$.
   (c) MSA Update. $\bar{\Lambda}^{(c)} = \frac{c-1}{c}\bar{\Lambda}^{(c)} + \frac{1}{c}\Lambda^{(c)}$.
   (d) Replanning. For $n = 1 \ldots N$, do:
      i. Draw $\mathcal{U}_n^{(c)}$ from $P_n(\mathcal{U}_n^{(c)}|\mathcal{X}^{(c-1)})$.
      ii. With probability $1 - P_{\text{accept},n}(\mathcal{U}_n^{(c)}|\bar{\Lambda}^{(c)})$, goto step 3(d)i.
      iii. Retain the first accepted draw $\mathcal{U}_n^{(c)}$.
   (e) Network loading. Draw $\mathcal{X}^{(c)}$ from $p(\mathcal{X}^{(c)}|\{\mathcal{U}\}^{(c)})$.

Intuitively, this algorithm works like a controller that steers the agents towards a reasonable fulfillment of the measurements: The Λ coefficients calculated in step 3b are positive if the measured flow on a particular link is higher than the simulated flow such that the resulting acceptance probabilities prefer plans that cross this link. Vice versa, if the measured flow is lower than the simulated flow, the according Λ coefficients are negative and plans that cross this link are penalized.

This procedure calibrates whatever choice dimensions are represented by the demand simulator, is compatible with an arbitrary supply simulator, and is fully consistent with the execution logic of a typical DTA microsimulator. While the power of this approach certainly depends on the feasibility of a linear log-likelihood approximation, the utilization of expected Λ coefficients makes it robust even in nonlinear conditions.

5 Advancements

This section presents three independent enhancements of the hitherto described calibration methodology, the first two of which are also deployed in the subsequent case studies: an improved linearization of the log-likelihood that properly accounts for congestion effects but requires a deterministic supply simulator, a computationally efficient alternative to the accept/reject procedure that relies on a utility-driven demand simulator, and a straightforward extension of the methodology to the calibration of structural model parameters.

5.1 Deterministic supply simulator

An analytical linearization of the log-likelihood function is possible if a deterministic and macroscopic supply simulator is available. Without loss of generality, we assume this simulator to be represented by a state space model

$$\mathbf{x}(0) = \mathbf{x}_0 \tag{26}$$

$$\mathbf{x}(k+1) = \mathbf{f}[\mathbf{x}(k), \boldsymbol{\beta}(k), k] \tag{27}$$

where vector $\mathbf{x}(k)$ denotes the model's state in time step k and single-commodity flow splits $\boldsymbol{\beta}(k) = (\beta_{ij}(k))$ from every upstream link i to every downstream link j at every intersection are exogenously provided. To start with, we postpone the explicit formalization of entry and exit flows in this model. The vector-valued transition function $\mathbf{f}$ defines the dynamics of the supply simulator. We assume that the Jacobians $\partial\mathbf{f}[\ldots]/\partial\mathbf{x}(k)$ and $\partial\mathbf{f}[\ldots]/\partial\boldsymbol{\beta}(k)$ are available. Operational implementations of this model class are described, e.g., in (Flötteröd, 2008) for inner-urban traffic and (Kotsialos et al., 2002) for freeways.

In order to execute a microscopic plan set $\{\mathcal{U}\}$ in the macroscopic model, the demand turning flows (16) are normalized into flow splits according to

$$\beta_{ij}(k) = \frac{d_{ij}(k)}{\sum_{j'} d_{ij'}(k)} \tag{28}$$

such that the network flows are split at intersections consistently with the

frequency of turning move occurrences in the plans.[4] This specification requires the same consistency between desired travel times in the plans and those observed in the supply simulator as discussed in Section 4.1. Operational implementations of this requirement are described, e.g., in (Chabini, 2001; Flötteröd, 2008).

A linearization of (28) with respect to the (presumably real-valued) demand turning flows yields

$$\beta_{ij}(k) \approx \sum_{l} \frac{\mathbf{1}(j = l) - \beta^0_{ij}(k)}{\sum_{j'} d^0_{ij'}(k)} d_{il}(k) + \text{const} \tag{29}$$

where $d^0_{ij'}(k)$ are the demand flows around which the linearization takes place, $\beta^0_{ij}(k)$ are the according flow splits, $\mathbf{1}(\cdot)$ is the indicator function, and the constant addend comprises all terms that are independent of the demand flows $d_{il}(k)$. Substituting (16) in this equation results in

$$\beta_{ij}(k) \approx \sum_{n=1}^{N} \sum_{l} \frac{\mathbf{1}(j = l) - \beta^0_{ij}(k)}{T \sum_{j'} d^0_{ij'}(k)} u_{il,n}(k) + \text{const.} \tag{30}$$

This approximates every flow split as a sum of agent-specific terms. We now exploit the differentiability of the supply simulator's dynamics: Appendix C shows how the log-likelihood $\ln p(\mathcal{Y}|\{\mathcal{U}\})$ is differentiated with respect to the flow splits such that a linear approximation

$$\ln p(\mathcal{Y}|\{\mathcal{U}\}) \approx \sum_{ijk} \frac{d \ln p(\mathcal{Y}|\{\mathcal{U}\})}{d\beta_{ij}(k)} \beta_{ij}(k) + \text{const} \tag{31}$$

can be obtained. A substitution of (30) in this equation yields

$$\ln p(\mathcal{Y}|\{\mathcal{U}\}) \approx \sum_{n=1}^{N} \sum_{ilk} \lambda_{il}(k) u_{il,n}(k) + \text{const} \tag{32}$$

where

$$\lambda_{il}(k) = \sum_{j} \frac{d \ln p(\mathcal{Y}|\{\mathcal{U}\})}{d\beta_{ij}(k)} \cdot \frac{\mathbf{1}(j = l) - \beta^0_{ij}(k)}{T \sum_{j'} d^0_{ij'}(k)}. \tag{33}$$

In order to extend this result to the previously disregarded network entry and exit flows, it is sufficient to incorporate these flows in the linearization (31) before the substitution of (30). This Λ specification requires a

[4] To avoid divisions by zero, all demand turning flows should be temporally smoothed before (28) is evaluated.

modification of step 3b in Algorithm 3 but otherwise leaves the calibration procedure unchanged.

The mathematical approximation of the supply simulator that underlies this development is superior to a linear assignment, which does not account for the backward effects of congested road segments: If the demand changes such that the desired inflow into a link at the capacity limit increases, the model given here accounts for the fact that the supply simulator only increases the flow split towards this link, not the inflow itself. If the supply simulator captures spillback, the consequence of a suchlike increased demand is not an unrealistically increased downstream flow but a physically correct upstream propagation of the congestion, which is properly reflected in the linearization (31) from which (33) is obtained.

5.2 Utility-driven demand simulator

While the accept/reject estimator of Section 3.2 is arguably the most general technique to influence agent behavior, it is by no means the only one. This section describes an alternative method that is tailored towards utility-driven demand simulators.

The individual-level posterior choice distribution (13) constitutes the starting point of this development. It is restated here for ease of reference:

$$P_n(\mathcal{U}_n^{(c)}|\mathcal{X}^{(c-1)}, \mathcal{Y}) = \frac{p(\mathcal{Y}|\mathcal{U}_n^{(c)})P_n(\mathcal{U}_n^{(c)}|\mathcal{X}^{(c-1)})}{\sum_{\mathcal{W}_n \in C_n} p(\mathcal{Y}|\mathcal{W}_n)P_n(\mathcal{W}_n|\mathcal{X}^{(c-1)})}. \tag{34}$$

We assume that the demand simulator implements a model of the following structure:

$$P_n(\mathcal{U}_n^{(c)}|\mathcal{X}^{(c-1)}) = \frac{PS_n(\mathcal{U}_n^{(c)}|\mathcal{X}^{(c-1)}) \exp[V_n(\mathcal{U}_n^{(c)}|\mathcal{X}^{(c-1)})]}{\sum_{\mathcal{W}_n \in C_n} PS_n(\mathcal{W}_n|\mathcal{X}^{(c-1)}) \exp[V_n(\mathcal{W}_n|\mathcal{X}^{(c-1)})]} \tag{35}$$

where $V_n(\mathcal{U}_n^{(c)}|\mathcal{X}^{(c-1)})$ denotes the systematic utility of plan $\mathcal{U}_n^{(c)}$ as perceived by individual n given the network conditions $\mathcal{X}^{(c-1)}$. In a route choice context, the real-valued PS coefficients are denoted as "path sizes" that account for route overlap (Ben-Akiva and Bierlaire, 1999). However, their specification is arbitrary in the present context, and (35) basically constitutes a plain multinomial logit model the choice probabilities of which

are scaled by the PS coefficients. A substitution of (35) in (34) yields

$$P_n(\mathcal{U}_n^{(c)}|\mathcal{X}^{(c-1)},\mathcal{Y}) = \frac{PS_n(\mathcal{U}_n^{(c)}|\mathcal{X}^{(c-1)})\exp[V_n(\mathcal{U}_n^{(c)}|\mathcal{X}^{(c-1)}) + \ln p(\mathcal{Y}|\mathcal{U}_n^{(c)})]}{\sum_{\mathcal{W}_n\in C_n} PS_n(\mathcal{W}_n|\mathcal{X}^{(c-1)})\exp[V_n(\mathcal{W}_n|\mathcal{X}^{(c-1)}) + \ln p(\mathcal{Y}|\mathcal{W}_n)]}. \tag{36}$$

This posterior is structurally identical to its prior. The only difference is that $\ln p(\mathcal{Y}|\mathcal{U}_n^{(c)})$ is added to the systematic utility of every considered plan $\mathcal{U}_n^{(c)}$. This utility modification allows to force a demand simulator that implements the prior (35) to immediately draw from the posterior (36), and it avoids the computational overhead of a possibly large number of rejections in the accept/reject procedure.

In order to apply the utility-driven estimator, only step 3d in Algorithm 3 needs to be adjusted. It is noteworthy that a heuristic application of this technique is possible even if the demand simulator does not implement the prior (35). Such an approach is based on a weaker theoretical foundation, but it may still produce practically useful results. The case studies of Section 6 take this line.

5.3 Identification of structural model parameters

Let the demand simulator be parameterized with an individual-level parameter vector θ_n for every agent $n = 1\ldots N$, denote the individually parameterized prior plan choice distributions by $P_n(\mathcal{U}_n|\mathcal{X},\theta_n)$, and assume that a prior parameter distribution $p(\theta_n)$ is available. In complete analogy to the derivation given in Section 3.2, an individual-level posterior

$$p_n(\mathcal{U}_n,\theta_n|\mathcal{X},\mathcal{Y}) = \frac{p(\mathcal{Y}|\mathcal{U}_n)P_n(\mathcal{U}_n|\mathcal{X},\theta_n)p(\theta_n)}{\sum_{\mathcal{W}_n\in C_n} p(\mathcal{Y}|\mathcal{W}_n)\int P_n(\mathcal{W}_n|\mathcal{X},\theta')p(\theta')d\theta'} \tag{37}$$

of agent n's joint choice and parameter distribution given the measurements can be formulated. (Note that the assumption of real-valued parameters is made only for notational convenience. This result hold equally for discrete-valued and even non-ordinal parameters.)

The following version of the accept/reject estimator draws from this posterior:

1. Draw θ_n from $p(\theta_n)$.

2. Draw $\mathcal{U}_n$ from $P_n(\mathcal{U}_n|\mathcal{X}, \Theta_n)$.

3. Accept $(\Theta_n, \mathcal{U}_n)$ with the acceptance probability $P_{\text{accept},n}(\mathcal{U}_n)$ defined in (14) and approximated in (25). Otherwise, goto 1.

If this logic is embedded within the estimator, characteristics of the posterior parameter distribution can be extracted from the accepted parameters in posterior conditions. That is, our methodology is, without conceptual modifications, applicable to the calibration of the demand simulator's parameters – the arguably most apparent example of which are the coefficients of a random utility model.

6 Case studies

The capabilities of our calibration framework are demonstrated by a set of selected experimental results. More results, including a more comprehensive description of the experimental setting, can be found in (Flötteröd, 2008). Some deviations from that reference are due to a slightly more realistic experimental setting considered here.

6.1 Setup

The test case comprises the metropolitan region of Greater Berlin, including a network of 2459 links and 1083 nodes. A synthetic population of 206 353 travelers with complete activity plans is available for this scenario (Rieser et al., 2007). All experiments are constrained to the time span from 6 to 9 am, which exhibits the most variable traffic conditions because of the morning rush hour.

In order to generate systematically different scenarios, we assume that a time-independent toll of 0.24 EUR/km is charged on all link in the city center shown in Figure 2 and that no toll is charged outside of this area. The unitless utility $V(\mathcal{U})$ of a route $\mathcal{U}$ is specified as

$$V(\mathcal{U}) = (-\text{tt}(\mathcal{U}) - \text{toll}(\mathcal{U})/\text{VOT})/1\text{s} \tag{38}$$

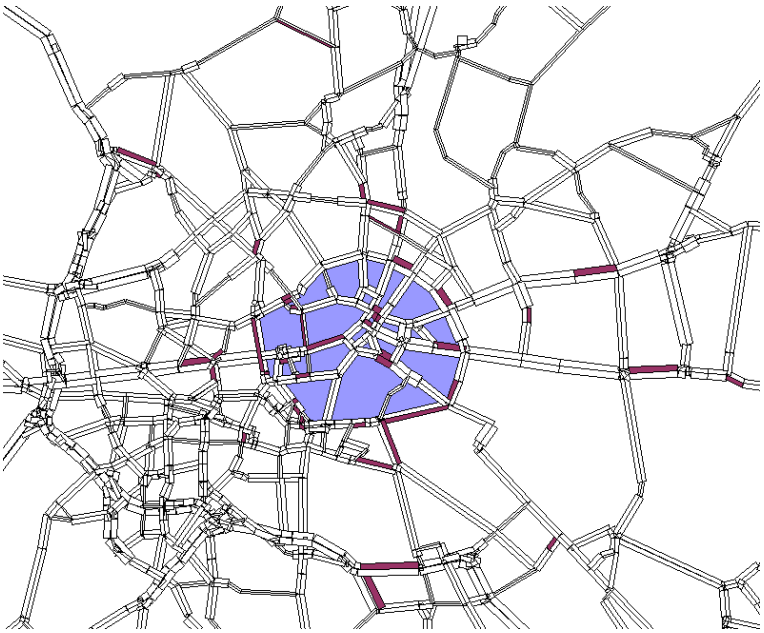

Figure 2: Berlin network

A time-independent toll of 0.24 EUR/km is charged in the highlighted area. One flow sensor is located in the center of each shaded link.

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 is a monetary **value of time** that is determinant for a particular scenario.

We deploy a simple DTA simulator where route choice is the only behavioral degree of freedom. In every iteration, the demand simulator recalculates the routes for 10% of all agents in the following way: A "proposed route" is generated by calculating a time-dependent best path based on a *randomly* chosen VOT and the preceding iteration's network conditions. The proposed route is adopted by the considered agent if it has a higher utility than the hitherto applied route given the *actual* VOT of the simulated scenario and the preceding iteration's network conditions. Otherwise, the agent maintains its previous route. The deterministic and macroscopic supply simulator complies with the specification of Section 5.1.

The utility-driven estimator of Section 5.2 is adopted. It does not interfere with the generation of a proposed route but only affects the subsequent choice between the proposed route and the hitherto applied route. As explained before, this constitutes a heuristic application of this estimator

because the demand simulator does not implement (35). Consequently, it also puts the estimator's robustness in suchlike conditions to test.

6.2 Equilibrium conditions

This section investigates the estimator's ability to calibrate an equilibrium-based planning model.

We generate, by simulation, a synthetic reality where the toll has no perceptible effect on the drivers, and we collect time dependent flows at 50 widespread sensor locations; see Figure 2 for an example. Synthetic measurement data is generated by averaging these (still time-dependent) flows over many iterations of the DTA simulator in stationary conditions. For simplicity, the measurements are assumed to follow independent and identical normal distributions, which yields a log-likelihood of the form (18).

During estimation, the prior assumption is that drivers react according to a 12 EUR/h VOT to the toll. The question thus becomes in how far the estimator, given 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 synthetic reality. It is emphasized that the estimator is not aware of the VOT parameter but adjusts the simulated driver behavior only based on the synthetic sensor data.

The estimator's ability to reproduce the flow measurements is evaluated by the root mean square error

$$\mathrm{RMSq} = \sqrt{\frac{1}{K|A'|}\sum_{k}\sum_{a\in A'}(q_a^{\mathrm{real}}(k) - q_a^{\mathrm{estim}}(k))^2} \tag{39}$$

where K is the number of considered time steps, A' is the set of 50 sensor-equipped links, $q_a^{\mathrm{real}}(k)$ is the measured flow on link a in time step k, and $q_a^{\mathrm{estim}}(k)$ is its estimated counterpart. The estimator's ability to extrapolate the measurements on the entire network is evaluated by

$$\mathrm{RMSx} = \sqrt{\frac{1}{K|A|}\sum_{k}\sum_{a\in A}(x_a^{\mathrm{real}}(k) - x_a^{\mathrm{estim}}(k))^2} \tag{40}$$

where A is the set of *all* links in the network, $x_a^{\mathrm{real}}(k)$ is the real occupancy (in vehicles) on link a in time step k, and $x_a^{\mathrm{estim}}(k)$ is its estimated counter-

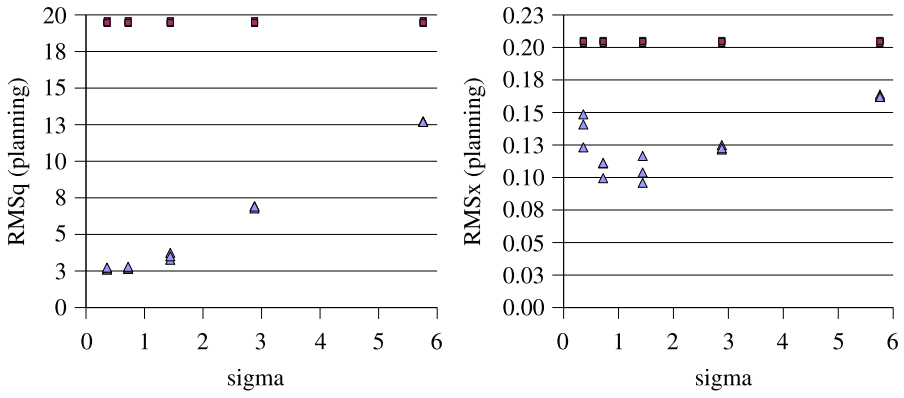

Figure 3: Result overview for planning experiments

part. The subsequently described experiments evaluate RMSq and RMSx in slightly different ways:

- The experiments in equilibrium conditions, i.e., those described in the remainder of this very section, evaluate (39) and (40) based on "real" and "estimated" values that are averaged over many iterations of their respective simulation/estimation run. This reduces the stochasticity in the evaluation to a minimum and resembles a planning setting in which the measurements are averaged over many days. These results are taken from (Flötteröd, 2008) without modification.

- Section 6.3's telematics results in offline conditions are still based on averaged "estimated" values, but utilize "real" data that is truly sampled from a single iteration of the simulation that generates the synthetic reality. This accounts for the fact that in a telematics application the estimator reconstructs the traffic state of a single day only. Finally, Section 6.4's telematics experiments in online conditions utilize the same sampled "real" data as in offline conditions but also sample the "estimated" values for computational reasons. These results go beyond what is described in (Flötteröd, 2008).

Figure 3 shows RMSq and RMSx values that result from different assumptions about the standard deviation σ of the sensor data's normal distributions. The light triangles over each σ value represent three independent

estimation results. The dark squares result from four plain simulations of the prior scenario, which is equivalent to running the estimator without sensor input. All results are quite stable in that there is limited variability among repeated runs such that, often enough, the dots lie on top of each other and cannot be distinguished.

RMSq increases monotonously with σ. This is plausible: the less certain the measurements the less weight is put on their reproduction. The greatest estimation improvement over a plain simulation of the prior scenario is 86%.

A non-monotonous relation between σ and RMSx can be observed. As σ grows, the measurement influence vanishes and the estimation quality gracefully deteriorates towards that of plain simulation. However, as σ decreases, a minimum value of RMSx is encountered after which a further decrease of σ results in an increased validation error. The most likely explanation of this effect is overfitting. The greatest RMSx reduction of 48% reflects the estimator's ability to spatiotemporally extrapolate the sensor data.

The quality of these results needs to be emphasized in light of the difficulty of the considered problem. The estimator adjusts a purely simulation-based route choice model at the individual level to measurements that are obtained at only 50 sensor locations out of 2459 links. The link between the observations and the calibrated model is given through a fully dynamical, highly nonlinear traffic flow model. Furthermore, the estimation problem is solved subject to likewise dynamical SUE constraints, which are implicitly represented through the iterative logic of the DTA simulator.

6.3 Non-equilibrium conditions, offline

For the experiments in non-equilibrium conditions, only the first day after the implementation of the toll is considered. That is, the drivers are aware of typical travel times without the toll and of the toll itself. However, they have not yet learned the altered traffic conditions that result from other travelers' changed behavior in response to the toll. In order to apply the estimator in these conditions, it is necessary to run iterations while the drivers remain on their initial level of knowledge. This knowledge is generated beforehand by running many iterations of a planning simulation

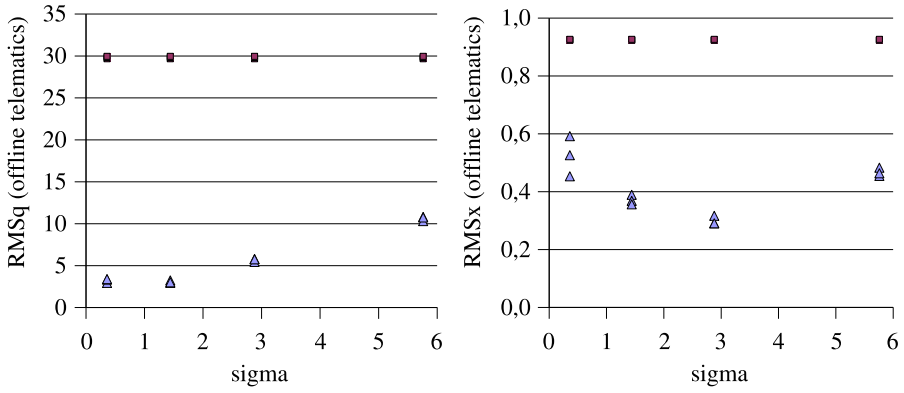

Figure 4: Result overview for offline telematics experiments

without toll and saving the travel times of every iteration. These travel times are then used by replanning travelers in the iterated estimation.

We investigate to what degree the estimator can push a prior SUE scenario towards a distinct out-of-equilibrium situation. For this purpose, we assume a prior equilibrium scenario without toll, and we generate the measurements from a synthetic reality where the drivers react to the toll with a 12 EUR/h VOT. All data that is extracted from the synthetical reality is now truly sampled. That is, the averaging over many iterations (days) assumed in the planning experiments is now omitted, and the flow and occupancy data representing the synthetic reality is generated from a single iteration of the according simulation run. However, the estimated flows and occupancies that enter RMSq and RMSx through (39) and (40) are still averaged over many iterations of their respective estimation run.

Figure 4 shows the resulting error measures over different σ values. Again, the light triangles represent estimation results and the dark squares represent plain simulations of the prior scenario. Distinct and stable improvements are generated. At $\sigma = 2.88$, the estimator reduces RMSq by 81% and achieves a maximum RMSx reduction of 68% when compared to a plain simulation of the prior scenario.

The non-equilibrium assumption in the synthetic reality implies that at the first day of the toll's installation the presumably most advantageous route choice for most drivers who so far have traversed the toll zone is now to

avoid this area but to bypass it as sharply as possible in order to minimize the increase in travel time. This causes a strong, unforeseen congestion on the roads that immediately encircle the toll zone. This congestion, which is not at all captured in the prior scenario that represents an equilibrium situation without toll, is successfully enforced by the estimator.

6.4 Non-equilibrium conditions, online

A rolling horizon logic is deployed that runs the estimator in simulated online conditions. 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 an identical logic as in offline conditions. 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, 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 most recent iteration of the previous estimation period. This logic is continued until 9:00.

A measurement standard deviation of $\sigma = 2.88$ is maintained in all runs because it achieved the best results in the previous offline experiments. Figure 5 provides separate results for every 30-minute estimation period ending at 7 through 9 am. The light bars in the front represent (from left to right) RMSx values obtained at the end of 5, 10, 20, 30, 40, and 50 iterations per estimation period. They are drawn on top of dark error bars that result from plain rolling horizon simulations of the prior scenario with respective iteration numbers. These simulations follows an identical logic as the estimator, only that the measurements are not accounted for.

Since the online context considered here does not allow for an averaging of estimation results over many iterations, the RMSx value of each estimation period is calculated from the occupancies of the last iteration of that estimation period only. That is, both the true and estimated occupancies that enter RMSx in (40) are now truly sampled. All RMSx values are averaged over three independent experiments.

The estimation and simulation errors rise over time as the traffic volumes increase in the morning rush hour. Estimation reduces RMSx on average by

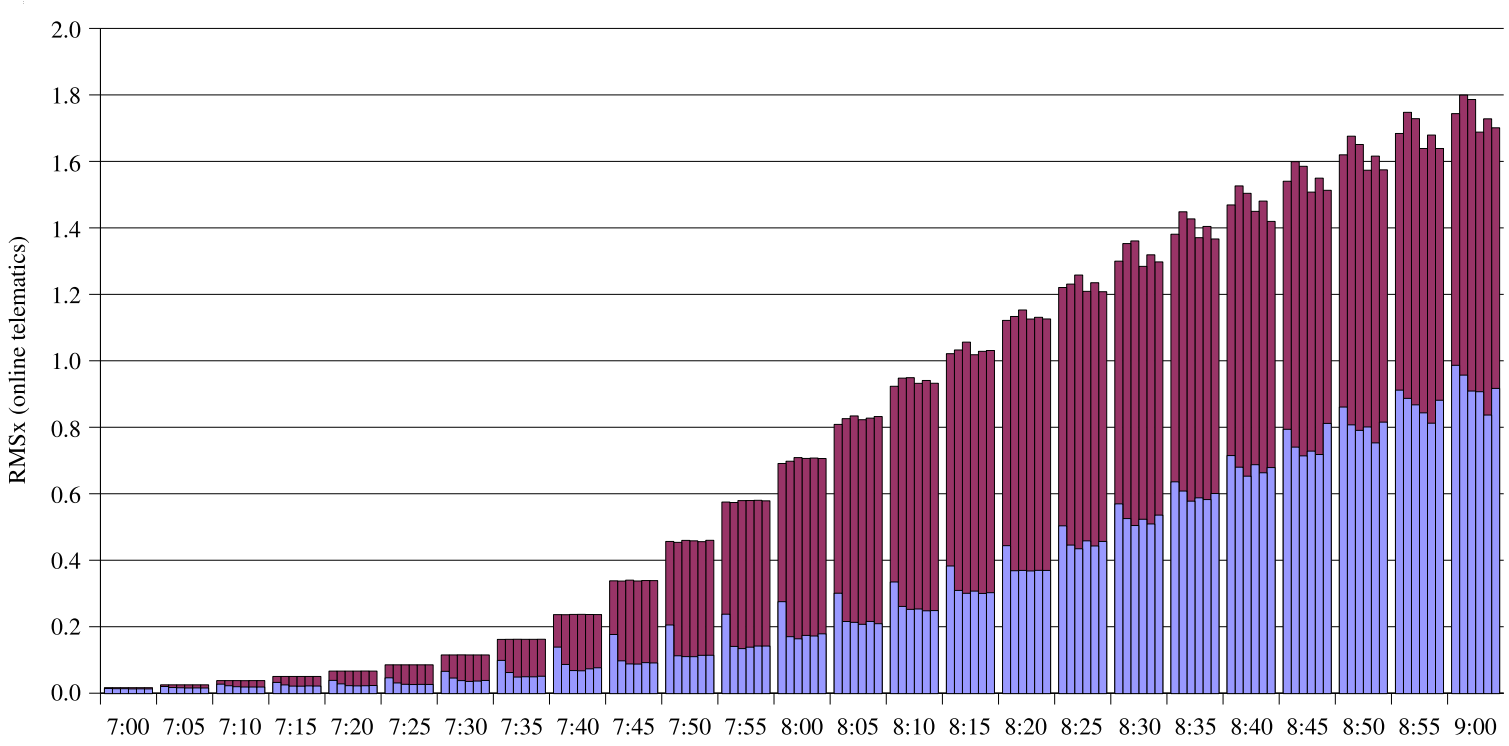

Figure 5: RMSx for rolling horizon estimation

The light bars in front represent (from left to right per time label) validation errors RMSx obtained after 5, 10, 20, 30, 40, and 50 iterations per estimation period. They are drawn on top of dark RMSx error bars that result from plain rolling horizon simulations of the prior scenario with respective iteration numbers. All RMSx values are averaged over three independent experiments.

half when compared to plain simulation. Conducting only 5 iterations per estimation period results in slightly lower improvements when compared to 10 iterations and more. Running beyond 10 iterations yields no further improvements.

An evaluation of the average prediction errors over a 0 to 30 minute time interval (omitted due to space restrictions; see (Flötteröd, 2008) for similar results) affirms these observations: The estimator reduces the RMSx prediction error by approximately 40% in the later time periods, and the computational effort of executing more than 10 iterations per estimation period does not further improve the predictions. The current implementation of the estimator accomplishes 6 iterations per 5-minute interval on a 3.2 GHz Pentium 4 machine with 2 GB RAM such that optimal results require a speedup of less than 2.

In summary, these case studies show that the estimator

- is compatible with a fully dynamical DTA simulator in both equilibrium and non-equilibrium conditions;
- reproduces the sensor data well and significantly improves the correctness of the global traffic situation;
- allows for the real-time calibration of practically relevant scenarios with standard computer equipment.

7 Summary

We have presented a new calibration framework that overcomes many of the simplifying modeling assumptions typically adopted in the calibration of dynamic traffic simulators. Our approach allows for the estimation of arbitrary behavioral patterns at the individual level in a Bayesian setting where aggregate measurements such as traffic counts are combined with a simulation-based representation of the analyst's prior knowledge. The approach is compatible with both an equilibrium-based modeling assumption and a telematics model where drivers are spontaneous and imperfectly informed. The experimental results show that the method generates significant model improvements and that it is capable of estimating practically

relevant scenarios in real-time. We also present various methodological advancements within the scope of our framework, which accentuates its structural adequacy to the problem under consideration.

8 Acknowledgments

This research was funded in part by the German research foundation DFG under the grant "State estimation for traffic simulations as coarse grained systems". Most experiments were run on the computing cluster of TU Berlin's mathematical faculty. GF thanks École Polytechnique Fédérale de Lausanne for funding and hospitality during the compilation of this article.

A Derivation of stationary posterior distribution

This appendix derives (8). First, (3) is substituted in (7) and the result is rearranged:

$$\begin{aligned}
\mathsf{P}(\{\mathcal{U}\}|\mathcal{Y}) &= \frac{\mathfrak{p}(\mathcal{Y}|\{\mathcal{U}\})}{\mathfrak{p}(\mathcal{Y})}\mathsf{P}(\{\mathcal{U}\}) \\
&= \frac{\mathfrak{p}(\mathcal{Y}|\{\mathcal{U}\})}{\mathfrak{p}(\mathcal{Y})}\sum_{\{\mathcal{V}\}}\mathsf{P}(\{\mathcal{U}\}|\{\mathcal{V}\})\mathsf{P}(\{\mathcal{V}\}) \\
&= \frac{\mathfrak{p}(\mathcal{Y}|\{\mathcal{U}\})}{\mathfrak{p}(\mathcal{Y})}\sum_{\{\mathcal{V}\}}\frac{\mathsf{P}(\{\mathcal{U}\}|\{\mathcal{V}\})\mathsf{P}(\{\mathcal{V}\})}{\mathsf{P}(\{\mathcal{V}\}|\mathcal{Y})}\mathsf{P}(\{\mathcal{V}\}|\mathcal{Y}) \\
&= \frac{\mathfrak{p}(\mathcal{Y}|\{\mathcal{U}\})}{\mathfrak{p}(\mathcal{Y})}\sum_{\{\mathcal{V}\}}\frac{\mathsf{P}(\{\mathcal{U}\}|\{\mathcal{V}\})\mathsf{P}(\{\mathcal{V}\})}{\mathfrak{p}(\mathcal{Y}|\{\mathcal{V}\})\mathsf{P}(\{\mathcal{V}\})/\mathfrak{p}(\mathcal{Y})}\mathsf{P}(\{\mathcal{V}\}|\mathcal{Y}) \\
&= \sum_{\{\mathcal{V}\}}\left(\frac{\mathfrak{p}(\mathcal{Y}|\{\mathcal{U}\})\mathsf{P}(\{\mathcal{U}\}|\{\mathcal{V}\})}{\mathfrak{p}(\mathcal{Y}|\{\mathcal{V}\})}\right)\mathsf{P}(\{\mathcal{V}\}|\mathcal{Y}).
\end{aligned} \tag{41}$$

This defines the stationary posterior distribution $\mathsf{P}(\{\mathcal{U}\}|\mathcal{Y})$ with the posterior transition distribution

$$\mathsf{P}(\{\mathcal{U}\}|\{\mathcal{V}\},\mathcal{Y}) = \frac{\mathfrak{p}(\mathcal{Y}|\{\mathcal{U}\})\mathsf{P}(\{\mathcal{U}\}|\{\mathcal{V}\})}{\mathfrak{p}(\mathcal{Y}|\{\mathcal{V}\})}. \tag{42}$$

Since

$$\sum_{\{\mathcal{U}\}} \mathrm{P}(\{\mathcal{U}\}|\{\mathcal{V}\},\mathcal{Y}) = 1 \tag{43}$$

must hold for the latter, it can equivalently be written as

$$\mathrm{P}(\{\mathcal{U}\}|\{\mathcal{V}\},\mathcal{Y}) = \frac{\mathrm{p}(\mathcal{Y}|\{\mathcal{U}\})\mathrm{P}(\{\mathcal{U}\}|\{\mathcal{V}\})}{\sum_{\{\mathcal{W}\}} \mathrm{p}(\mathcal{Y}|\{\mathcal{W}\})\mathrm{P}(\{\mathcal{W}\}|\{\mathcal{V}\})}. \tag{44}$$

B Derivation of accept/reject estimator

Given the acceptance probabilities $\mathrm{P}_{\mathrm{accept},n}(\mathcal{U}_n|\mathcal{Y})$ defined in (14), the overall probability of a rejection for agent n is

$$\mathrm{P}_{\mathrm{reject},n}(\mathcal{X},\mathcal{Y}) = 1 - \sum_{\mathcal{V}_n \in \mathrm{C}_n} \mathrm{P}_{\mathrm{accept},n}(\mathcal{V}_n|\mathcal{Y})\mathrm{P}_n(\mathcal{V}_n|\mathcal{X}). \tag{45}$$

Consequently, the probability that $\mathcal{U}_n$ is the first accepted draw can be expressed as

$$\begin{aligned} & \sum_{d=0}^{\infty} \mathrm{P}^d_{\mathrm{reject},n}(\mathcal{X},\mathcal{Y})\mathrm{P}_{\mathrm{accept},n}(\mathcal{U}_n|\mathcal{Y})\mathrm{P}_n(\mathcal{U}_n|\mathcal{X}) \\ = & \frac{\mathrm{P}_{\mathrm{accept},n}(\mathcal{U}_n|\mathcal{Y})\mathrm{P}_n(\mathcal{U}_n|\mathcal{X})}{1 - \mathrm{P}_{\mathrm{reject},n}(\mathcal{X},\mathcal{Y})} \\ = & \frac{\mathrm{P}_{\mathrm{accept},n}(\mathcal{U}_n|\mathcal{Y})\mathrm{P}_n(\mathcal{U}_n|\mathcal{X})}{\sum_{\mathcal{V}_n \in \mathrm{C}_n} \mathrm{P}_{\mathrm{accept},n}(\mathcal{V}_n|\mathcal{Y})\mathrm{P}_n(\mathcal{V}_n|\mathcal{X})}, \end{aligned} \tag{46}$$

which coincides with the definition of $\mathrm{P}_n(\mathcal{U}_n|\mathcal{X},\mathcal{Y})$ in (13).

C Linearization of log-likelihood function

The dynamics of the deterministic supply simulator are represented by

$$\mathbf{x}(k+1) = \mathbf{f}[\mathbf{x}(k), \boldsymbol{\beta}(k), k], \tag{47}$$

cf. (27). For the sake of generality, a functional

$$\Phi(\mathcal{X}) = \sum_{k=1}^{K} \varphi[\mathbf{x}(k), k] \tag{48}$$

of the macroscopic system states $\mathcal{X} = \{\mathbf{x}(k)\}_k$ is linearized with respect the turning fractions $\boldsymbol{\beta}(k) = (\beta_{ij}(k))$. (Specifically, one would choose $\Phi(\mathcal{X}) = \ln p(\mathcal{Y}|\{\mathcal{U}\})$ and observe that $\mathcal{X} = \mathcal{X}(\{\mathcal{U}\})$ holds for a deterministic supply simulator.) A self-contained exposition of a well-known procedure for linearizing Φ with respect to the $\boldsymbol{\beta}(k)$ is given in the following (Kotsialos et al., 2002; Pearson and Sridhar, 1966).

Denote

$$\Phi(k) = \sum_{\kappa=k}^{K} \varphi[\mathbf{x}(\kappa), \kappa] \tag{49}$$

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

$$\Phi(k) = \begin{cases} \varphi[\mathbf{x}(k), k] + \Phi(k+1) & k = 1 \dots K-1 \\ \varphi[\mathbf{x}(K), K] & k = K. \end{cases} \tag{50}$$

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

$$\frac{d\Phi(k)}{d\mathbf{x}(k)} = \begin{cases} \dfrac{\partial \varphi[\mathbf{x}(k), k]}{\partial \mathbf{x}(k)} + \dfrac{d\Phi(k+1)}{d\mathbf{x}(k)} & k = 1 \dots K-1 \\ \dfrac{\partial \varphi[\mathbf{x}(K), K]}{\partial \mathbf{x}(K)} & k = K. \end{cases} \tag{51}$$

Since the interplay between variables at different time steps is fully defined by state equation (47),

$$\frac{d\Phi(k+1)}{d\mathbf{x}(k)} = \left(\frac{\partial \mathbf{f}[\mathbf{x}(k), \boldsymbol{\beta}(k), k]}{\partial \mathbf{x}(k)} \right)^T \frac{d\Phi(k+1)}{d\mathbf{x}(k+1)} \tag{52}$$

holds for $k < K$, where $\mathbf{x}(k+1) = \mathbf{f}[\dots]$ is used and superscript ^{T} denotes the transpose.

Now, sensitivities with respect to $\boldsymbol{\beta}(k)$ result from

$$\frac{d\Phi(\mathcal{X})}{d\boldsymbol{\beta}(k)} = \left(\frac{\partial \mathbf{f}[\mathbf{x}(k), \boldsymbol{\beta}(k), k]}{\partial \boldsymbol{\beta}(k)} \right)^T \frac{d\Phi(k+1)}{d\mathbf{x}(k+1)}, \tag{53}$$

where $\partial \varphi[\mathbf{x}(k), k] / \partial \boldsymbol{\beta}(k)$ disappears since $\boldsymbol{\beta}(k)$ influences no state earlier than $\mathbf{x}(k+1)$. In summary, $d\Phi(\mathcal{X})/d\boldsymbol{\beta}(k)$ is obtained in a two-pass-procedure.

1. Using (52), solve (51) recursively for $k = K \ldots 1$. Moving backwards through time introduces a "far sightedness" into the calculation that is necessary to predict the influence of present state variations on future system states.

2. Determine the influence of the flow splits by (53) for $k = 0 \ldots K - 1$.

References

Ashok, K., 1996, *Estimation and Prediction of Time-Dependent Origin-Destination Flows*, PhD thesis, Massachusetts Institute of Technology.

Astarita, V., Er-Rafia, K., Florian, M., Mahut, M. and Velan, S., 2001, A comparison of three methods for dynamic network loading, *Transportation Research Record* **1771**, 179–190.

Bell, M., 1991, The estimation of origin-destination matrices by constrained generalised least squares, *Transportation Research Part B* **25**(1), 13–22.

Bell, M., 1995, Stochastic user equilibrium assignment in networks with queues, *Transportation Research Part B* **29**(2), 125–137.

Bell, M. and Grosso, S., 1999, Estimating path flows from traffic counts, *in* W. Brilon, F. Huber, M. Schreckenberg and H. Wallentowitz (eds), *Traffic and Mobility: Simulation–Economics–Environment*, Springer, pp. 85–102.

Bell, M., Lam, W. and Iida, Y., 1996, A time-dependent multi-class path flow estimator, *in* J.-B. Lesort (ed.), *Proceedings of the 13th International Symposium on Transportation and Traffic Theory*, Pergamon, Lyon, France, pp. 173–193.

Bell, M., Shield, C., Busch, F. and Kruse, G., 1997, A stochastic user equilibrium path flow estimator, *Transportation Research Part C* **5**(3/4), 197–210.

Ben-Akiva, M. and Bierlaire, M., 1999, Discrete choice methods and their applications to short-term travel decisions, *in* R. Hall (ed.), *Handbook of Transportation Science*, Kluwer, pp. 5–34.

Ben-Akiva, M., Bierlaire, M., Burton, D., Koutsopoulos, H. and Mishalani, R., 2001, Network state estimation and prediction for real–time transportation management applications, *Networks and Spatial Economics* **1**, 293–318.

Ben-Akiva, M. and Lerman, S., 1985, *Discrete Choice Analysis*, MIT Press series in transportation studies, The MIT Press.

Bierlaire, M., 2002, The total demand scale: a new measure of quality for static and dynamic origin-destination trip tables, *Transportation Research Part B* **36**(9), 837–850.

Bierlaire, M., 2003, BIOGEME: a free package for the estimation of discrete choice models, *Proceedings of the 3rd Swiss Transport Research Conference*, Monte Verita/Ascona.

Bierlaire, M. and Crittin, F., 2006, Solving noisy large scale fixed point problems and systems of nonlinear equations, *Transportation Science* **40**(1), 44–63.

Bierlaire, M. and Toint, P., 1995, MEUSE: an origin-destination estimator that exploits structure, *Transportation Research Part B* **29**(1), 47–60.

Bottom, J., 2000, *Consistent Anticipatory Route Guidance*, PhD thesis, Massachusetts Institute of Technology.

Bowman, J. and Ben-Akiva, M., 1998, Activity based travel demand model systems, *in* P. Marcotte and S. Nguyen (eds), *Equilibrium and advanced transportation modelling*, Kluwer, pp. 27–46.

Cascetta, E., 1984, Estimation of trip matrices from traffic counts and survey data: a generalised least squares estimator, *Transportation Research Part B* **18**(4/5), 289–299.

Cascetta, E., 1989, A stochastic process approach to the analysis of temporal dynamics in transportation networks, *Transportation Research Part B* **23**(1), 1–17.

Cascetta, E., Inaudi, D. and Marquis, G., 1993, Dynamic estimators of origin-destination matrices using traffic counts, *Transportation Science* **27**, 363–373.

Cascetta, E. and Posterino, N., 2001, Fixed point approaches to the estimation of o/d matrices using traffic counts on congested networks, *Transportation Science* **35**(2), 134–147.

Chabini, I., 2001, Analytical dynamic network loading problem, *Transportation Research Record* **1771**, 191–200.

De Palma, A. and Marchal, F., 2002, 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.

Flötteröd, G., 2008, *Traffic State Estimation with Multi-Agent Simulations*, PhD thesis, Berlin Institute of Technology, Berlin, Germany.

Kitamura, R., 1988, An evaluation of activity-based travel analysis, *Transportation* **15**, 9–34.

Kitamura, R., 1996, Applications of models of activity behavior for activity based demand forecasting, *Proceedings of the Activity-Based Travel Forecasting Conference*, New Orleans, LA, USA, pp. 119–150.

Kotsialos, A., Papageorgiou, M., Diakaki, C., Pavlis, Y. and Middelham, F., 2002, Traffic flow modeling of large-scale motorway networks using the macroscopic modeling tool METANET, *IEEE Transactions on Intelligent Transportation Systems* **3**(4), 282–292.

Maher, M., 1983, Inferences on trip matrices from observations on link volumes: a Bayesian statistical approach, *Transportation Research Part B* **17**(6), 435–447.

Maher, M., Zhang, X. and Van Vliet, D., 2001, 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.

Mahmassani, H. S., 2001, Dynamic network traffic assignment and simulation methodology for advanced system management applications, *Networks and Spatial Economics* **1**(3/4), 267–292.

Nagel, K., Rickert, M., Simon, P. and Pieck, M., 1998, The dynamics of iterated transportation simulations, *Proceedings of the 3rd Triennial Symposium on Transportation Analysis*, San Juan, Puerto Rico.

Nie, Y. and Lee, D.-H., 2002, An uncoupled method for the equilibrium-based linear path flow estimator for origin-destination trip matrices, *Transportation Research Record* **1783**, 72–79.

Nie, Y., Zhang, H. and Recker, W., 2005, Inferring origin-destination trip matrices with a decoupled GLS path flow estimator, *Transportation Research Part B* **39**(6), 497–518.

Nökel, K. and Schmidt, M., 2002, Parallel DYNEMO: meso-scopic traffic flow simulation on large networks, *Networks and Spatial Economics* **2**(4), 387–403.

Pearson, J. and Sridhar, R., 1966, A discrete optimal control problem, *IEEE Transactions on Automatic Control* **11**(2), 171–174.

Raney, B. and Nagel, K., 2006, An improved framework for large-scale multi-agent simulations of travel behavior, *in* P. Rietveld, B. Jourquin and K. Westin (eds), *Towards better performing European Transportation Systems*, Routledge, pp. 305–347.

Rieser, M., Nagel, K., Beuck, U., Balmer, M. and Rümenapp, J., 2007, Truly agent-oriented coupling of an activity-based demand generation with a multi-agent traffic simulation, *Proceedings of the 86. Annual Meeting of the Transportation Research Board*, Washington, DC, USA.

Ross, S., 2006, *Simulation*, fourth edn, Elsevier.

Sevcikova, H., Raftery, A. and Waddell, P., 2007, Assessing uncertainty in urban simulations using Bayesian melding, *Transportation Research Part B* **41**(6), 652–669.

Sherali, H., A.Narayan and Sivanandan, R., 2003, Estimation of origin-destination trip-tables based on a partial set of traffic link volumes, *Transportation Research Part B* **37**(9), 815–836.

Sherali, H. and Park, T., 2001, Estimation of dynamic origin-destination trip tables for a general network, *Transportation Research Part B* **35**(3), 217–235.

Sherali, H., Sivanandan, R. and Hobeika, A., 1994, A linear programming approach for synthesizing origin-destination trip tables from link traffic volumes, *Transportation Research Part B* **28**(3), 213–233.

Spiess, H., 1987, A maximum likelihood model for estimating origin-destination models, *Transportation Research Part B* **21**(5), 395–412.

Train, K., 2003, *Discrete Choice Methods with Simulation*, Cambridge University Press.

van Zuylen, H. and Willumsen, L. G., 1980, The most likely trip matrix estimated from traffic counts, *Transportation Research Part B* **14**(3), 281–293.

Vovsha, P., Bradley, M. and Bowman, J., 2004, Activity-based travel forecasting models in the United States: progress since 1995 and prospects for the future, *Proceedings of the EIRASS Conference on Progress in Activity-Based Analysis*, Maastricht, The Netherlands.

Waddell, P., Ulfarsson, G., Franklin, J. and Lobb, J., 2007, Incorporating land use in metropolitan transportation planning, *Transportation Research Part A* **41**(5), 382–410.

Watling, D. and Hazelton, M., 2003, The dynamics and equilibria of day-to-day assignment models, *Networks and Spatial Economics* **3**(3), 349–370.

Yang, H., 1995, Heuristic algorithms for the bilevel origin/destination matrix estimation problem, *Transportation Research Part B* **29**(4), 231–242.

Yang, H., Sasaki, T. and Iida, Y., 1992, Estimation of origin-destination matrices from link traffic counts on congested networks, *Transportation Research Part B* **26**(6), 417–434.

Zhou, X., 2004, *Dynamic Origin-Destination Demand Estimation and Prediction for Off–Line and On–Line Dynamic Traffic Assignment Operation*, PhD thesis, University of Maryland, College Park.