Title
Real-time estimation of distributed parameters systems: Applicaton to traffic monitoring

Author
Work, Daniel Benjamin

Publication Date
2010

Real–time estimation of distributed parameters systems: Applicaton to traffic monitoring

by

Daniel Benjamin Work

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering – Civil and Environmental Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Alexandre M. Bayen, Chair
Professor Carlos Daganzo
Professor Pieter Abbeel

Fall 2010

Real–time estimation of distributed parameters systems: Applicaton to traffic monitoring

Abstract

Real–time estimation of distributed parameters systems: Applicaton to traffic monitoring

by

Daniel Benjamin Work

Doctor of Philosophy in Engineering – Civil and Environmental Engineering

University of California, Berkeley

Professor Alexandre M. Bayen, Chair

This dissertation is motivated by the practical problem of highway traffic estimation using velocity measurements from GPS enabled mobile devices such as cell phones. In order to simplify the estimation procedure, a velocity model for highway traffic is constructed, which results in a dynamical system in which the observation operator is linear. It presents a new scalar hyperbolic *partial differential equation* (PDE) model for traffic velocity evolution on highways, based on the seminal *Lighthill-Whitham-Richards* (LWR) PDE for density. Equivalence of the solution of the new velocity PDE and the solution of the LWR PDE is shown for quadratic flux functions. Because this equivalence does not hold for general flux functions, a discretized model of velocity evolution based on the Godunov scheme applied to the LWR PDE is proposed. Using an explicit instantiation of the weak boundary conditions of the PDE, the discrete velocity evolution model is generalized to a network, thus making the model applicable to arbitrary highway networks. The resulting velocity model is a nonlinear and nondifferentiable discrete time dynamical system with a linear observation operator, for which a Monte Carlo based ensemble Kalman filtering data assimilation algorithm is applied.

The model and estimation technique is evaluated with experimental data obtained from a large-scale field experiment known as *Mobile Century*. The velocity estimates using GPS data from cellphones is compared to velocity estimates using inductive loop detector data from the PeMS system. More than 900 estimation simulations are performed using various volumes of GPS data and inductive loop detector data collected during the experiment, which show travel times can be reconstructed to less than 10% error with sufficient GPS data, loop data, or a combination of both. All data collected during the field experiment and used in the simulations are available for download at `http://traffic.berkeley.edu`.

To my parents,

Bill Work and Kathy Work

Contents

List of Figures

List of Tables

Acknowledgments

I am extremely grateful for the opportunity to work with Professor Alexandre Bayen, who has served as my PhD thesis advisor, mentor, and friend at Berkeley. I am indebted to him for his endless support for all aspects of my professional and personal development.

I would also like to thank Professors Carlos Daganzo, Pieter Abbeel, Samer Madanat, and James Demmel who provided a number of valuable suggestions during my qualifying examination. Additionally, my work has been advanced by several discussions with Professor Daganzo, who pointed me towards a discrete velocity model to circumvent some challenges found in the continuous domain. Professor Madanat has served as a mentor to me since my first semester at Berkeley. I am inspired by the work of Professor Abbeel on machine learning approaches to traffic estimation, and Professor Demmel's work on parallel computing, both of which have given me new perspectives on large-scale estimation problems.

I would also like to thank Professors Steven Glaser and Raja Sengupta for sharing their perspectives on systems approaches for civil engineering problems, and for providing research and creative guidance on a number of occasions. In addition to serving on my preliminary examination, they have opened their labs to me and treated me as one of their own students.

It has been a privilege for me to work with Professor Benedetto Piccoli at Rutgers University, who introduced me to the problem of the LWR PDE on networks, and who has provided mathematical guidance throughout my PhD.

This thesis would not have been possible without the support of Dr. Quinn Jacobson, Dr. Ken Tracton, and the Mobile Internet Services Systems team at Nokia Research Center in Palo Alto. In addition to their support for the *Mobile Century* and *Mobile Millennium* projects, they introduced me to the privacy challenges around mobile sensing with cell phones during my time as a visiting researcher and intern.

I would like to show my gratitude to Dr. Olli-Pekka Tossavainen for many fruitful discussions on inverse problems and data assimilation, and for suggesting the use of ensemble Kalman filtering for traffic estimation problems. I am especially grateful for several productive collaborations with Sebastien Blandin on topics related to conservation laws, including velocity inversion on first-order models and phase transition second-order models. Working with each of them has been a wonderful experience.

I would also like to thank my fellow students in the Bayen lab. You have each given me support in so many ways, such as brainstorming ideas and collaborating on articles, donating your time to help with field experiments, and providing me encouragement on the numerous late nights and long weekends in the lab. It has been a true pleasure to work with each of you.

I would like to acknowledge Tom West, Joe Butler, and the staff of *California Center for innovative Transportation* for transforming *Mobile Century* and *Mobile Millennium* from research ideas into large-scale experiments and software systems. Thank you for making CCIT an exciting and fun place to work.

Special thanks goes to my friends in Berkeley. You never let me spend too much time in the lab when the weather was perfect or the surf was up. Thanks for introducing me to California!

Lastly, I would like to thank my parents Bill and Kathy Work, my sister Amy Work, and Alexandra Kolla for all of their love and support.

Chapter 1

Introduction

1.1 Motivation

Distributed parameters systems (DPS) are a class of systems which arise naturally in large-scale civil and environmental engineering problems. A distributed parameter system one whose spatial variation of its infinite dimensional state (distributed parameters) plays an important role in the evolution of the system in time. Thus, a distributed parameter system can be used to model contaminants propagating in rivers and estuaries, air quality and pollution dispersion in urban areas, the behavior of structures under wind or seismic loads, and traffic congestion on roadways, to name a few examples. A common mathematical representation of a distributed parameter system is in the form of a *partial differential equation* (PDE).

1.1.1 The forward problem

The first formulation to study distributed parameters systems, known as the *direct* or *forward problem*, is illustrated in Figure 1.1. First, an abstraction of the physical world must be constructed, typically in the form of a mathematical equation which describes the system's evolution. Second, the system's initial condition, boundary conditions, and model parameters must be defined. Finally, the model is solved or simulated forward in time, so that the behavior of the system can be studied and analyzed.

In Figure 1.1, arrows show how information flows between the physical world, the distributed parameter system abstraction, and the computational model simulations. In particular, information from the physical world enters the model simulations only through the mathematical model.

This process has two fundamental limitations which prevent a direct matching between events occurring in the simulated environment and that which occurs in reality.

- The mathematical model is only an abstraction of reality in which the physical world

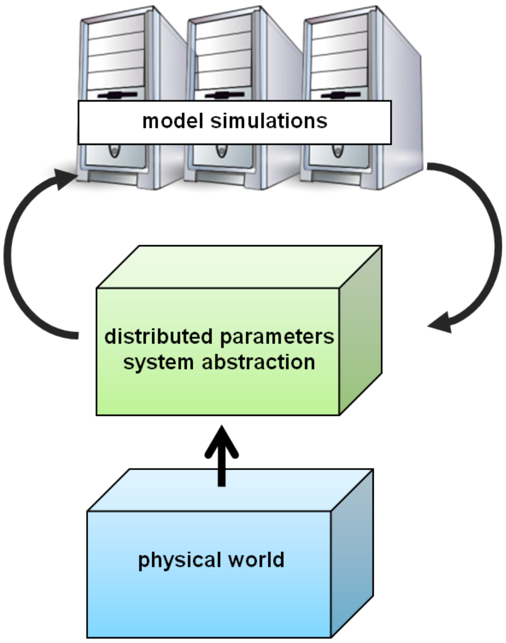


Figure 1.1: The forward problem.

is approximated and simplified. Thus the model contains some error inherent to the abstraction. This modeling error can be difficult to quantify, and therefore the accuracy of the model may be hard to determine.

- Often, the initial condition, boundary conditions, and model parameters are only known approximately, if at all, which adds to the uncertainty of the model.

1.1.2 The estimation problem

To address the limitations of the forward problem formulation described above, a related *estimation problem* can also be studied. In the estimation problem, the mathematical model is augmented with additional information from the physical world in the form of data from sensors. This process is illustrated in Figure 1.2. Like the forward problem, the physical world is again used to build the mathematical abstraction in the form of a partial differential equation. Moreover, it is also used to generate observations from the physical world through sensor data. The process of combining the model and the data is known as estimation. This process is also referred to as parameter estimation or *inverse modeling* when the goal is to estimate parameters in the system. When the objective is to estimate the state of the system, it is called state estimation or *data assimilation* [54].

Estimation algorithms can be described according to how they incorporate new sensor data into the estimate of the system. When the estimation problem is solved *online*, the algorithm uses sensor data piece by piece as it becomes available, without the need for all of the data at once. On the other hand, an *off-line* or *batch* algorithm requires all measurements at the same time. In practice, online algorithms may achieve estimates more quickly since only a portion of the data is needed at any time, but perhaps at the cost of improved accuracy achieved by batch algorithms.

Estimation algorithms can also be characterized according to the time constraints under which they operate. A *real–time* algorithm has strict deadlines on the timing of the competition, while an algorithm which is not real–time does not. The timing deadline is often a function of the rate at which the physical system evolves, so that information produced by a real–time algorithm can be used to control the physical world before it becomes outdated or obsolete. When this is achieved, the computation infrastructure and the physical world become tightly coupled, creating a *cyber–physical system*.

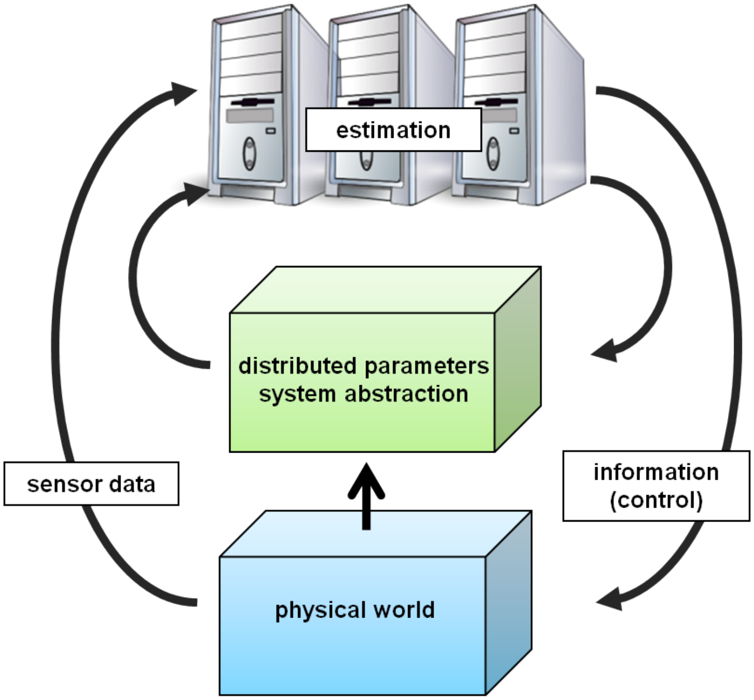


Figure 1.2: The estimation problem, with a feedback loop to the physical world.

One of the fundamental challenges for estimation problems for distributed parameters systems is the acquisition of sensor data, precisely because the system is distributed in space. In the context of traffic monitoring for highways, data acquisition is typically achieved through the placement of dedicated sensing infrastructure deployed in the pavement, such as an inductive loop detector, or sensors adjacent to the infrastructure such as video, radar or RFID. Due to the expense of installation and maintenance, it is difficult to achieve sensor coverage at a global scale.

1.1.3 Impact of the mobile Internet on distributed parameters systems

The recent explosion of Internet connected mobile devices such as smartphones is having a dramatic impact on data acquisition for distributed parameters systems. Their low cost, portability, and computational capabilities make smartphones useful for numerous applications in which they act as sensors moving with humans, embedded in the built infrastructure. Large scale applications include traffic flow estimation [77, 78], which is a rapidly expanding field at the heart of mobile internet services.

These devices also act as a content gateway for real–time *location based services*, which has significantly increased the need for accurate and timely information. In the case of traffic, information which is accurate but not real–time, or real–time but inaccurate, can instantly be detected by users receiving traffic information on Internet connected devices in their vehicles. In contrast, earlier paradigms of traffic monitoring, where congestion estimates were delivered to traffic websites or news stations and consumed by users before the trip begins, makes determination of the estimate's accuracy significantly harder.

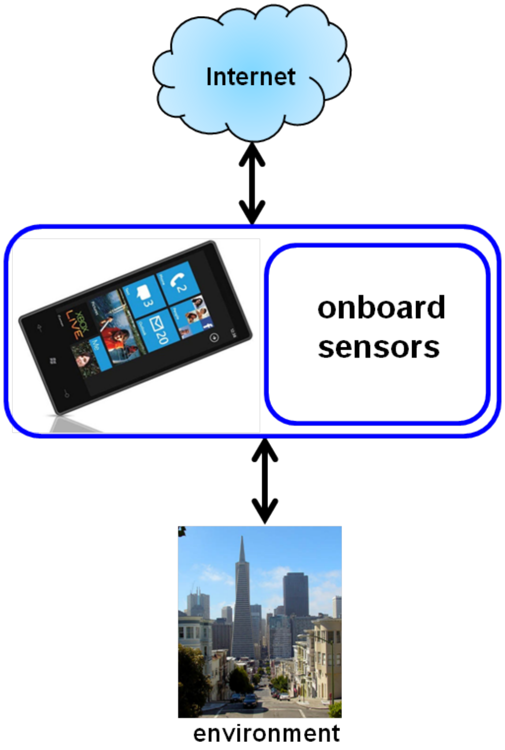


Figure 1.3: Mobile phones act as a sensing and communication platform, bridging the physical world with the Internet.

1.2 Problem: Velocity estimation from GPS-equipped mobile devices

Motivated both by the availability of GPS data from mobile devices, and by the need for increased accuracy demanded by mobile Internet services, the goal of this dissertation is to develop a real-time estimation algorithm for monitoring traffic using velocity data from mobile devices. This work constructs a model for the evolution of a velocity field $v(x,t)$ on a highway segment $x \in [0,L]$, which is a distributed parameter system. Vehicles labeled by $i \in \mathbb{N}$ travel along the highway with trajectories $x_i(t)$, and measure the velocity $v(x_i(t),t)$ along their trajectories.

These discrete measurements are then combined with the velocity evolution model, and together they are used to reconstruct or estimate the function $v(x,t)$, in real–time, using an online Ensemble Kalman filtering framework. Fig. 1.4 illustrates the process: the evolution of the velocity field $v(x,t)$ can be depicted as a surface, which is to be reconstructed. A subset of the vehicles is sampled along their trajectories. For illustration purposes in the figure, four vehicles are sampled at time $t = t_m$, which produces four points on the $v(x,t)$ surface which can be used by the algorithm to reconstruct the surface.

When the surface $v(x,t_m)$ is estimated using measurements up to time $t_1 = t_m$, $v(x,t_m)$ is known as a filtered estimate, which is the focus of this dissertation. Two related problems include prediction and smoothing. If the surface $v(x,t_m)$ is estimated using measurements up to time $t_2 < t_m$, the resulting estimate is known as a prediction. Finally, when the surface $v(x,t_m)$ is estimated using measurements up to time $t_3 > t_m$, the result is known as a smoothed estimate.

To address one important issue in sensing with mobile devices, we consider velocity measurements which are obtained through a privacy-aware architecture introduced by Nokia, called *virtual trip lines* (VTLs) [38]. VTLs are virtual geographic markers which act as triggers for mobile sensing, and therefore can be viewed as a spatial sampling strategy. The main constraint VTLs place on traffic estimation is that it is not possible to track vehicles (full vehicle trajectories are never disclosed), or identify measurements as belonging to the same vehicle. [38].

1.3 Related work

1.3.1 Traffic flow theory

The origins of traffic flow theory dates back to the 1950's with the pioneering works of Lighthill and Whitham [55], and independently Richards [64], who proposed a macroscopic model of traffic based on conservation of vehicles. The model, now known as the *Lighthill–Whitham–Richards* (LWR) PDE, is a nonlinear hyperbolic conservation law. The main mathematical challenge to the LWR PDE, and more generally systems of conservation laws, is

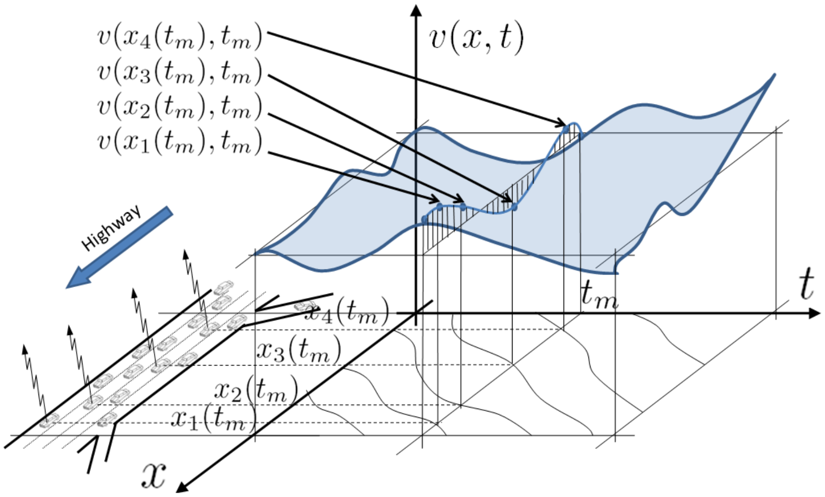


Figure 1.4: Illustration of the distributed velocity field $v(x,t)$ to be reconstructed from GPS samples. Four samples $v_i(x_i(t),t)$ are shown at $t = t_m$, from vehicles i transmitting their data (indicated by up-arrows above the vehicles).

the development of discontinuities (shocks), which can occur in finite time even from smooth initial conditions. The existence and uniqueness of the *Cauchy problem* (i.e. an initial value problem) for conservation laws on infinite domain are achieved through a suitable entropy condition introduced in the seminal works of Oleinik [61] and Kruzkov [49] (See also the existence result of Glimm [31]).

The introduction of a boundary condition for systems of conservation laws was first treated by Bardos, Leroux, and Nedelec [7]. The well-posedness of a scalar *initial boundary value problem* for a conservation law with a convex flux function was addressed by Le Floch [27] and recently by Frankowska [28]. The recent work of Strub and Bayen [70] instantiates the well-posedness of the initial boundary value problem for the LWR PDE explicitly.

Soon after the introduction of the scalar LWR PDE, higher–order traffic models were introduced in an attempt to reconcile some of the deficiencies of first-order models. These models augment the mass conservation equation with a momentum equation, the most notable being the model of Payne [63]. However this model had several deficiencies, including characteristics moving faster than the average velocity of traffic, and vehicles moving backward, as pointed out by del Castillo [23], and in particular Daganzo's “Requiem for second-order fluid approximations of traffic flow” [22]. These problems were addressed and the models were “resurrected” independently by Aw and Rascle [5] and Zhang [82], leading to the class of *Aw–Rascle–Zhang* (ARZ) models.

A class of phase transition models was introduced by Colombo [15], which combines a scalar conservation law in free flow with a 2×2 system of conservation laws in congestion. The global well posedness of phase transition models such as [15] was given by [16]. This model was later extended by Blandin et. al. [9] to simplify the phase transition analysis.

In the discrete domain, the *Cell Transmission Model* was introduced by Daganzo [20, 21] as a mass conserving traffic model consistent with the LWR PDE using a Godunov discretization [32, 51]. Papageorgiou [62] introduced a discrete model which is a modified version of the discretized Payne model.

1.3.2 Traffic estimation

The process of recursively estimating traffic conditions using a traffic flow model and experimental data begins with the 1970's with the early work of Szeto and Gazis [72]. Using a mass conservation equation with flow measurements at the segment edges, they applied an *extended Kalman filter* (EKF) to estimate the traffic density in the Lincoln Tunnel in New York City. The extended Kalman filter is a widely used extension of the recursive minimal variance estimator known as the Kalman filter [46]. For state estimation on nonlinear systems, the model equation and observation equation are linearized to fit the framework of Kalman filtering, resulting in a suboptimal filter.

In the early 1980s, a modified version of Payne model was used for a variety of estimation and control problems, in particular through the work of Papageorgiou and his collaborators[17, 62, 75, 76]. In [17], Cremer and Papageorgiou introduced the parameter estimation problem with experimental data on this model, and in [62], extended Kalman filtering is applied to this model for state estimation. The work of Wang et al [75] details the simultaneous solution of the state and parameter estimation problem, again with extended Kalman filtering. The article [75] also provides a concise review of related estimation problems appearing in traffic. The interested reader should also see [76], which provides some results of EKF on the modified Payne model as implemented an experimental testbed known as the Renaissance system.

A key ingredient of these works [17, 62, 75, 76] is the differentiability of the numerical scheme employed for the second order model of traffic used, which is a feature the first-order CTM does not possess. The early work of Szeto and Gazis [72], and later Gazis and Liu [30] circumvent this issue for first-order models by directly observing the flows at ends of the road segments, which enables the application of extended Kalman filtering.

Sun, Munoz, and Horowitz [71] treat the nonlinearity of the CTM by recognizing it can be transformed into a switching state space model, which enables the use of a set of linear equations to describe the state evolution for the distinct flow regimes on the highway (e.g. highway is in free-flow or congestion). The density state estimation problem is then solved with a *mixture Kalman filter* for the purpose of ramp metering. In [34, 35], specific modes of the dynamics presented in [71] are used to incorporate Lagrangian velocity trajectories into an extension of the CTM, called the *Switched Mode Model* (SMM), using mixture Kalman filtering.

Recently, the cell transmission model has been used in state estimation problems through increasingly advanced nonlinear filters, including *unscented Kalman filtering* (UKF) [43] in the work of Mihaylova, Boel, Hegyi [58], particle filtering by Mihaylova and Boel [57], Mihaylova, Boel, Hegyi [58], and Sau et al [65]. In [58], the particle filter is shown to perform better than UKF, but has a higher computational cost. Implementation of particle filtering techniques on high dimensional systems (several thousand states or more), remains an open challenge due to inherent scalability challenges for particle filters [69]. Other treatments of traffic estimation include adjoint–based control and data assimilation in [41, 42].

1.4 Contributions and organization of the dissertation

1.4.1 Contributions

This dissertation contains several new contributions to the problem of estimating traffic conditions from GPS velocity data.

- **A velocity evolution equation consistent with hydrodynamic theory**. This dissertation presents two models for velocity evolution on roadways.
    - The first model, known as the *Lighthill-Whitham-Richards for velocity* (LWR–v), is a velocity–based partial differential equation with weak solutions consistent with the classical LWR PDE for the Greenshields flux function. For general flux functions, we prove that this equivalence does not hold, which is a negative result.
    - The second model, known as the *Cell Transmission Model for velocity* (CTM–v), is a discrete evolution equation derived from a Godunov discretization scheme applied to an integral form of the LWR PDE. Its consistency with the Godunov discretization scheme for the LWR PDE, also known as the *Cell Transmission Model*, is ensured by equivalence of the Riemann solvers used in the numerical scheme.
    - We extend the CTM-v model to networks, using a generalized Riemann solver at vertices in the network which is consistent with the density Riemann solvers of Coclite, Garavello, and Piccoli [14] and Daganzo [21] by construction.
- **Solution of the velocity estimation problem using ensemble Kalman filtering**
    - Using the CTM–v, we pose the state estimation problem as a non-linear nondifferentiable dynamical system with a linear observation operator. By using the velocity as the state instead of density (as would be the case for the CTM), we avoid the need to linearize a nonlinear observation operator. The recursive velocity state estimation problem is then solved using *ensemble Kalman filtering* (EnKF).
    - We prove the non-differentiability of LWR PDE with a Godunov discretization (also the CTM), around model states which generate a standing shock wave.

This fact prevents direct application of the widely popular nonlinear extension of Kalman filtering, known as extended Kalman filtering, to these models.

- **Experimental implementation and assessment**
  - The velocity estimation results of the ensemble Kalman filtering estimation approach applied to the CTM–v velocity evolution model are presented using data collected from the *Mobile Century* field experiment, which ran a prototype version of the algorithm (online and in real–time).
  - We compare travel times computed from the CTM–v EnKF velocity estimation algorithm using different volumes of data collected from probe vehicles only, inductive loop detectors only, and probe vehicles and inductive loop detectors with travel times collected from high resolution video data during the *Mobile Century* experiment. At low volumes of probe data, the addition of data from inductive loops significantly reduces the estimation error. On the *Mobile Century* site, travel time estimates using only data from probe vehicles are as good as or better than estimates using only data from inductive loop detectors, provided a sufficient number of probe measurements are collected.

1.4.2 Organization

This dissertation is organized as follows. We begin with a derivation and review of the LWR PDE and its important mathematical properties in Chapter 2. The LWR PDE serves as the basis for the velocity evolution equations derived in Chapter 3.

In particular, in Chapter 3 we establish the equivalence of the proposed LWR–v model in the velocity domain and the LWR model in the density domain for a quadratic flux function (called the Greenshields model), and prove that this equivalence does not hold for general flux functions, which is a negative result. For general flux functions, we derive a discrete velocity evolution equation, the CTM–v, which is consistent with an integral form of the LWR PDE, and extend this model to networks through selection of a suitable Riemann solver at network vertices.

In Chapter 4, we pose the estimation problem in state space form, yielding a discrete space discrete time nonlinear nondifferentiable velocity evolution equation with a linear observation operator, and solve it with ensemble Kalman filtering. The *Mobile Century* experiment and privacy aware sampling of GPS enabled smart phones, and present experimental results of the CTM–v EnKF velocity estimation algorithm on this data.

In Chapter 5, we address experimentally the accuracy trade-offs between estimates using data from inductive loops compared to estimates using data from probe vehicles, by analyzing travel times computed from the velocity field estimated using the ensemble Kalman filtering algorithm. We present conclusions and discuss future in Chapter 6.

Chapter 2

LWR PDE

2.1 Introduction

The main objective of this chapter is to describe a mathematical model of traffic evolution which expresses conservation of vehicles, known as the *Lighthill-Whitham-Richards* (LWR) *partial differential equation* (PDE) [55, 64], and review the important mathematical attributes of this model. The new contributions in this chapter are as follows.

- **Strong boundary conditions**. By transforming the classical statement of weak boundary conditions for scalar conservation laws applicable to the LWR PDE into mutually exclusive conditions, we derive an explicit statement for boundary conditions for the LWR PDE on a finite domain to be applied in the strong sense.
- **Non-differentiability of the discretized LWR PDE around arbitrary model states**. We prove that the LWR PDE is not differentiable around model states resulting in a standing shock wave. This result is the main motivation for developing filtering techniques which do not suffer from the same shortcomings as the extended Kalman filter, in particular the ensemble Kalman filter.

The chapter is organized as follows. In Section 2.2 we recall the derivation of the LWR PDE as an integral equation expressing conservation of vehicles on a stretch of roadway, and note that when the density is smooth, it yields the well-known LWR PDE. We also show non-smooth solutions satisfy the integral form of the equation when the *Rankine-Hugoniot* jump condition is satisfied. In Section 2.3, we treat non-smooth solutions to the PDE by considering a weak formulation of the partial differential equation, and a suitable entropy condition to guarantee uniqueness of solutions. Proper formulation of the weak boundary conditions and the derivation of the strong boundary conditions are given. In Section 2.4, we introduce the Riemann problem and its solutions to provide further clarity on the results in Section 2.3. Numerical discretization of the LWR PDE using a Godunov scheme and its relation to the Riemann problem is presented in Section 2.5.1. We also show that this discretization yields a discrete time discrete space evolution equation for density, which cannot be linearized around states resulting in a standing shock wave.

2.2 Derivation of a mass conservation law for traffic

In this section, we derive the well-known Lighthill–Whitham–Richards partial differential equation [55, 64]. Let $\rho(x,t)$ be the vehicle density (the number of vehicles per unit length) at the point x in space and t in time, and let $Q(\cdot)$ be the flux (number of vehicles per unit time) as a function of the density. The flux function $Q(\cdot)$ is defined in an interval $[0, \rho_{\max}]$, where $\rho_{\max}$ is the maximal density, sometimes referred to as "jam density". The total number of vehicles on a segment between two points x_1 and x_2 is given by $\int_{x_1}^{x_2} \rho(x,t)dx$. Assuming vehicles do not appear or disappear within the segment, we have:

$$\begin{aligned} \frac{d}{dt}\int_{x_1}^{x_2} \rho(x,t)dx &= Q(\rho(x_1,t)) - Q(\rho(x_2,t)) && (2.1)\\ &= -Q(\rho(x,t))|_{x_1}^{x_2} \\ &= -\int_{x_1}^{x_2} \frac{\partial}{\partial x} Q(\rho(x,t))dx && (2.2) \end{aligned}$$

Equation (2.1) can be understood in the following way. Consider a segment of roadway shown in Figure 2.1, with vehicles entering from the left and exiting to the right. The change in the number of vehicles in the segment over time is just the difference between the number vehicles which entered at x_1, given by $Q(\rho(x_1,t))$ and the number that leave at x_2, given by $Q(\rho(x_2,t))$.

When $\rho(x,t)$ is smooth[1], (2.2) can be rewritten as

$$\int_{x_1}^{x_2} \left(\frac{\partial \rho(x,t)}{\partial t} + \frac{\partial Q(\rho(x,t))}{\partial x} \right) dx = 0 \tag{2.3}$$

Since (2.3) holds for any x_1 and x_2, we obtain the seminal LWR PDE model [55, 64]:

$$\begin{aligned} \frac{\partial \rho(x,t)}{\partial t} + \frac{\partial Q(\rho(x,t))}{\partial x} &= 0 \quad (x,t) \in (-\infty, +\infty) \times (0,T) && (2.4)\\ \rho(x,0) &= \rho_0(x) \quad x \in (-\infty, +\infty) && (2.5) \end{aligned}$$

which is the macroscopic traffic flow model expressing conservation of vehicles along and infinite stretch of roadway from time $t = 0$ through $t = T$, augmented with the initial condition ρ_0.

---

[1] This turns out to be a critical assumption, since often the density profile contains discontinuities such as shocks.

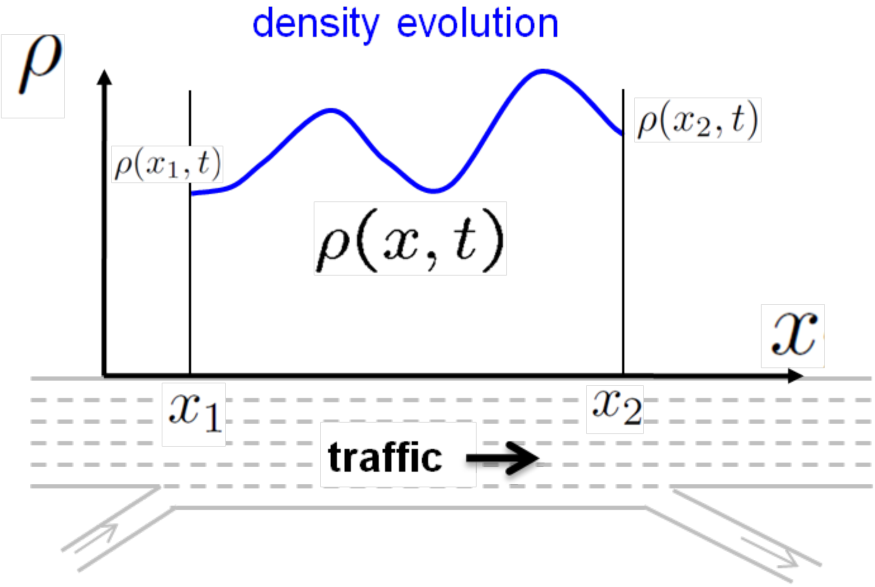


Figure 2.1: The LWR PDE describes the evolution of density on the roadway.

For traffic applications, the flux function $Q(\cdot)$ is generally assumed to be concave and piecewise C^1. This function may be approximated by strictly concave C^2 flux functions with superlinear growth to fit the framework of [8] and [27], which is used to define existence and uniqueness properties of scalar conservation laws (such as the LWR PDE) on a finite domain. In the transportation engineering community, the flux function $Q(\cdot)$ is also known as the *fundamental diagram.*

The fundamental assumption of the LWR PDE is that the average vehicle velocity can be defined in terms of the density alone. With this assumption, we introduce the velocity function $V(\cdot)$ of the density in $[0, \rho_{\max}]$. Then the flux function reads:

$$Q\left(\rho\right) = \rho\, V(\rho) \tag{2.6}$$

Solutions to the LWR PDE can be constructed through the method of characteristics. For this, one needs to transform the partial differential equation into a system of ordinary differential equations along curves $(x(z), t(z))$. We seek solutions on the curves of the form

$$\frac{d\rho\left(x(z), t(z)\right)}{dz} = F\left(x(z), t(z), \rho\left(x(z), t(z)\right)\right) \tag{2.7}$$

where $F(\cdot,\cdot,\cdot)$ is a function to be determined. Applying the chain rule to the left side of equation (2.7) yields

$$\frac{d\rho\left(x(z), t(z)\right)}{dz} = \frac{\partial\rho}{\partial x}\frac{dx}{dz} + \frac{\partial\rho}{\partial t}\frac{dt}{dz} \tag{2.8}$$

Note that for smooth functions $Q(\cdot)$, the LWR PDE can be written in quasi-linear form:

$$\frac{\partial \rho(x,t)}{\partial t} + Q'(\rho(x,t))\frac{\partial \rho(x,t)}{\partial x} = 0 \tag{2.9}$$

If we let $\frac{dx}{dz} = Q'(\rho\left(x(z),t(z)\right))$ and $\frac{dt}{dz} = 1$, and substitute into equation (2.8) we have

$$\frac{d\rho\left(x(z),t(z)\right)}{dz} = \frac{\partial \rho\left(x(z),t(z)\right)}{\partial x} Q'(\rho\left(x(z),t(z)\right)) + \frac{\partial \rho\left(x(z),t(z)\right)}{\partial t} = 0 \tag{2.10}$$

where the second equality is given by equation (2.9). This means the solution $\rho\left(x(z),t(z)\right)$ is constant on the characteristics. Moreover, solving $\frac{dt}{dz} = 1$, with the initial condition $t(0) = 0$ yields $t = s$. Therefore $\frac{dx}{dz} = \frac{dx}{dt} = Q'\left(\rho\left(x(z),t(z)\right)\right)$. Thus, we obtain three well-known and important properties of the LWR PDE:

- The density is constant along characteristic curves.
- The speed of the characteristics is given by the slope of the flux function $Q(\cdot)$.
- Because the density is constant along the characteristic curves, the speed of each characteristic curve is a constant.

More details about the method of characteristics can be found in [24]. With these three properties, we can now point out the crux of all mathematical difficulties which arise when solving the LWR PDE. Even from smooth initial conditions, shocks may develop in finite time, and classical (smooth) solutions to the PDE may not exist.

For the purpose of illustration, we consider the flux function given by $Q(\rho) = \rho - \rho^2$ where $\rho \in [0,1]$. The speed of the characteristic curves is positive for $\rho \in \left[0, \frac{1}{2}\right)$, and negative for $\rho \in \left(\frac{1}{2}, 1\right]$. If the initial condition is specified such that upstream characteristic curves have positive velocity, and downstream characteristic curves have negative velocity, the characteristic curves may intersect, yielding a point where the solution is discontinuous.

Moreover, discontinuous solutions satisfy the integral form of the conservation law [53], as we will now show. First, apply the integral form of the conservation law (2.1) over a small interval $x \in [x_1, x_1 + \Delta x]$ in space and $t \in [t_1, t_1 + \Delta t]$ in time:

$$\begin{aligned} &\int_{x_1}^{x_1+\Delta x} \rho(x, t_1 + \Delta t)dx - \int_{x_1}^{x_1+\Delta x} \rho(x, t_1)dx \\ &= \int_{t_1}^{t_1+\Delta t} Q(\rho(x_1, t))dt - \int_{t_1}^{t_1+\Delta t} Q(\rho(x_1 + \Delta x, t))dt \end{aligned}$$

Then let s denote the speed of a shock which exists over the full interval, which connects the two states ρ^- and ρ^+ on the left and right sides of the shock, respectively (Figure 2.2)

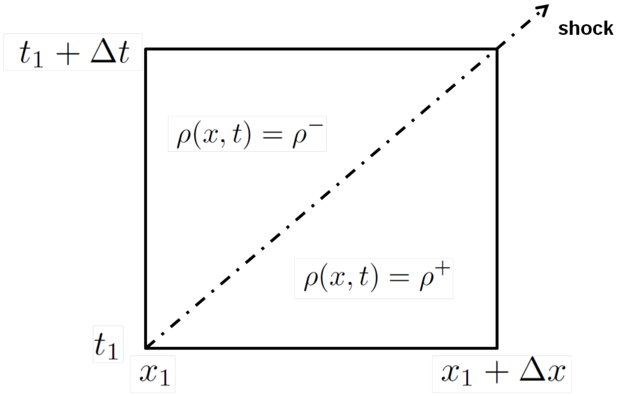


Figure 2.2: A shock traveling at speed s connects the states ρ^- and ρ^+ over a small interval $x \in [x_1, x_1 + \Delta x]$ in space and $t \in [t_1, t_1 + \Delta t]$ in time.

and choose $(\Delta x, \Delta t)$ sufficiently small so that ρ^-, ρ^+, and s can be viewed as a constant. Then we have

$$\Delta x \rho^- - \Delta x \rho^+ = \Delta t Q(\rho^-) - \Delta t Q(\rho^+) + O(\Delta t^2) \tag{2.11}$$

where $O(\Delta t^2)$ accounts for the small variation in the fluxes at the boundaries. Since the shock speed over this small interval is given as $s = \frac{\Delta x}{\Delta t}$, we can substitute $\Delta x = s\Delta t$ in (2.11), divide by Δt, and take the limit as $\Delta t \to 0$, yielding

$$s(\rho^- - \rho^+) = Q(\rho^-) - Q(\rho^+)$$

After rearranging terms, this leads to the *Rankine–Hugoniot* jump condition for the speed of the shock:

$$s = \frac{Q(\rho^+) - Q(\rho^-)}{\rho^+ - \rho^-} \tag{2.12}$$

The important result is that if (2.12) is satisfied, then the discontinuity satisfies the integral form of the LWR PDE (2.1).

To reconcile the fact that discontinuous solutions to the LWR PDE (2.4) can arise in finite time, even from smooth initial data, and that discontinuous solutions satisfying the Rankine–Hugoniot jump condition (2.12) also satisfy the integral form of the conservation law (2.1), we must consider a more general class of solutions to the LWR PDE known as *weak* solutions. We describe this next.

2.3 Weak solutions and the entropy condition

We begin by assuming temporarily that $\rho(x,t)$ is smooth, and satisfies the LWR PDE (2.4) and (2.5). Then we introduce a smooth test function with compact support $\varphi(x,t) \in C_c^1((-\infty, \infty) \times [0,T))$. Since the LWR PDE is satisfied, we have

$$\int_{-\infty}^{\infty}\int_{0}^{T}\left(\frac{\partial\rho(x,t)}{\partial t}+\frac{\partial Q(\rho(x,t))}{\partial x}\right)\varphi(x,t)dtdx=0 \tag{2.13}$$

Now by applying integration of parts, we can rewrite (2.13) so that the derivatives appear only on the smooth test function:

$$\int_{-\infty}^{\infty}\int_{0}^{T}\left(\frac{\partial\varphi(x,t)}{\partial t}\rho(x,t)+\frac{\partial\varphi(x,t)}{\partial x}Q(\rho(x,t))\right)dtdx=-\int_{-\infty}^{\infty}\rho_0(x)\varphi(x,0)dx \tag{2.14}$$

Note that only the initial condition appears, due to the compact support of φ, and that (2.14) no longer requires a smooth density profile. A solution satisfying (2.14) is known as a *weak solution* to the LWR PDE, and can be shown to be equivalent to the integral form (2.1).

An unfortunate result of the weak formulation is that, due to the possibility of discontinuities, the solution is no longer unique (as will be discussed further in Example 4 in Section 2.4).

One approach to isolate a unique solution is to consider an entropy function of the density, with the property that the entropy is conserved when the density profile is smooth, and that the entropy either increases or decreases due to discontinuities in the density profile. Thus the entropy acts as an indicator of discontinuities, and can be used to isolate a unique solution. This approach leads to the following definition.

Definition 1 (Weak entropy solution [10, 49])**.** A weak entropy solution $\rho(\cdot,\cdot)$ of (2.4) and (2.5) is defined as follows:

$$\int_{-\infty}^{\infty}\int_{0}^{T}\left(|\rho(x,t)-k|\;\frac{\partial}{\partial t}\varphi(x,t)+\operatorname{sgn}(\rho(x,t)-k)\;(Q(\rho(x,t))-Q(k))\;\frac{\partial}{\partial x}\varphi(x,t)\right)dxdt$$
$$+\int_{-\infty}^{\infty}|\rho_0(x)-k|\varphi(x,0)dx\geq 0 \qquad \forall\varphi\in C_c^2\left([-\infty,\infty]\times[0,T);\mathbb{R}_+\right),\forall k\in\mathbb{R}$$

where $\operatorname{sgn}(x)=1$ for $x>0$, $\operatorname{sgn}(x)=-1$ for $x<0$, and $\operatorname{sgn}(x)=0$ for $x=0$.

In practice, it is often easier to use an equivalent entropy condition which explicitly defines admissible shocks. For a smooth concave flux function $Q(\cdot)$ a discontinuous solution connecting two states ρ^- and ρ^+ propagating at speed s satisfies the Lax entropy condition if:

$$Q'(\rho^+)\leq s\leq Q'(\rho^-) \tag{2.15}$$

When the LWR PDE is specified on a finite domain:

$$\frac{\partial\rho(x,t)}{\partial t}+\frac{\partial Q(\rho(x,t))}{\partial x}=0 \quad (x,t)\in(0,L)\times(0,T) \tag{2.16}$$

special consideration must be made at the boundaries to ensure well posed problem, as given by the next two definitions.

Definition 2 (Left weak boundary condition - concave flux function [8, 70])**.** For a general flux function $Q(\cdot)$, the proper weak description of the left boundary condition for (2.16) in terms of the trace of the solution $\rho(0,t)$ and the left boundary data $\rho_l(t)$ is as follows:

$$\sup_{k\in D(\rho(0,t),\rho_l(t))} \left(\operatorname{sgn}\left(\rho(0,t)-\rho_l(t)\right)\left(Q(\rho(0,t))-Q(k)\right)\right)=0 \quad \text{for } a.e.\ t>0 \tag{2.17}$$

where $D(x,y)=[\inf(x,y),\sup(x,y)]$.

Definition 3 (Right weak boundary condition - concave flux function [8, 70])**.** For a general flux function $Q(\cdot)$, the proper weak description of the right boundary condition for (2.16) in terms of the trace of the solution $\rho(L,t)$ and the right boundary condition $\rho_r(t)$ is as follows:

$$\inf_{k\in D(\rho(L,t),\rho_r(t))} \left(\operatorname{sgn}\left(\rho(L,t)-\rho_r(t)\right)\left(Q(\rho(L,t))-Q(k)\right)\right)=0 \quad \text{for } a.e.\ t>0 \tag{2.18}$$

where $D(x,y)=[\inf(x,y),\sup(x,y)]$.

It was proposed in [27] to write boundary conditions in such a way that the entropy solution to equation (2.16) exists and is unique, for in domain bounded on the left snd unbounded on the right. For a strictly convex continuously differentiable flux function under sufficient regularity of the boundary data $\rho_l(\cdot)$ and $\rho_r(\cdot)$, an equivalent formulation of (2.17) and (2.18) can be obtained. In [28], it is shown that continuity of the boundary data is sufficient for an equivalent formulation. In our case, this formulation reads for the left boundary:

$$\begin{aligned}&\text{for } a.e.\ t>0,\\ &\begin{cases} \qquad \rho(0,t)=\rho_l(t)\\ \text{xor}\quad Q'(\rho(0,t))\le 0 \text{ and } Q'(\rho_l(t))\le 0 \text{ and } \rho(0,t)\ne\rho_l(t)\\ \text{xor}\quad Q'(\rho(0,t))\le 0 \text{ and } Q'(\rho_l(t))> 0 \text{ and } Q(\rho(0,t))\le Q(\rho_l(t))\end{cases}\end{aligned} \tag{2.19}$$

and for the right boundary:

$$\begin{aligned}&\text{for } a.e.\ t>0,\\ &\begin{cases} \qquad \rho(L,t)=\rho_r(t)\\ \text{xor}\quad Q'(\rho(L,t))\ge 0 \text{ and } Q'(\rho_r(t))\ge 0 \text{ and } \rho(L,t)\ne\rho_r(t)\\ \text{xor}\quad Q'(\rho(L,t))\ge 0 \text{ and } Q'(\rho_r(t))< 0 \text{ and } Q(\rho(L,t))\le Q(\rho_r(t))\end{cases}\end{aligned} \tag{2.20}$$

where $\rho_l(\cdot)$ and $\rho_r(\cdot)$ are functions of $C^0(0,T)$.

The preceding equations (2.19) and (2.20) is a description of cases for which (2.17) and (2.17) is satisfied. Note the description is slightly different from [70] in that the sets defined on each line of (2.19) and (2.20) are mutually exclusive. For example, first line of

(2.19) corresponds to the case when the trace of the solution $\rho(0,t)$ takes the value of the boundary data $\rho_l(t)$, which is analogous to a prescription of the boundary condition in the strong sense. The second line and third lines correspond to cases which satisfy (2.17), but where the value of the trace does not take the value prescribed at the boundary.

We now expand on the first line of equations (2.19)– (2.20) in order to state explicitly the set of the boundary data, trace pairs for which the boundary data is prescribed in the strong sense.

Lemma 1 (Strong boundary conditions - concave flux)**.** For a strictly concave flux function $Q(\cdot)$, the cases for strong boundary conditions read as follows: for *a.e.* $t > 0$,

$$\begin{aligned}&\rho(0,t) = \rho_l(t) \text{ iff}\\ &\begin{cases} & Q'(\rho(0,t)) \geq 0 \text{ and } Q'(\rho_l(t)) \geq 0\\ \text{xor} & Q'(\rho(0,t)) \leq 0 \text{ and } Q'(\rho_l(t)) \leq 0 \text{ and } u\left(0,t\right) = \rho_l\left(t\right)\\ \text{xor} & Q'(\rho(0,t)) \leq 0 \text{ and } Q'(\rho_l(t)) > 0 \text{ and } Q\left(u\left(0,t\right)\right) > Q(\rho_l\left(t\right))\end{cases}\end{aligned} \tag{2.21}$$

and for *a.e.* $t \geq 0$,

$$\begin{aligned}&u\left(L,t\right) = \rho_r\left(t\right) \text{ iff}\\ &\begin{cases} & Q'(\rho(L,t)) \leq 0 \text{ and } Q'(\rho_r(t)) \leq 0\\ \text{xor} & Q'(\rho(L,t)) \geq 0 \text{ and } Q'(\rho_r(t)) \geq 0 \text{ and } u\left(L,t\right) = \rho_r\left(t\right)\\ \text{xor} & Q'(\rho(L,t)) \geq 0 \text{ and } Q'(\rho_r) < 0 \text{ and } Q\left(u\left(L,t\right)\right) > Q(\rho_r\left(t\right))\end{cases}\end{aligned} \tag{2.22}$$

Proof. We prove the case of the left boundary condition for a concave flux and note a similar argument holds for the right boundary and in the case of convex flux functions. Beginning with the statement of weak boundary conditions, (2.19) we can write: for *a.e.* $t > 0$,

$$\begin{aligned}&\rho(0,t) \neq \rho_l(t) \text{ iff}\\ &\begin{cases} & Q'(\rho(0,t)) \leq 0 \text{ and } Q'(\rho_l(t)) \leq 0 \text{ and } \rho(0,t) \neq \rho_l(t)\\ \text{xor} & Q'(\rho(0,t)) \leq 0 \text{ and } Q'(\rho_l(t)) > 0 \text{ and } Q(\rho(0,t)) \leq Q(\rho_l(t))\end{cases}\end{aligned}$$

If we are not in one of these two cases, then by taking their complement, we must have either

$$\begin{cases} & Q'(\rho(0,t)) \geq 0 \text{ and } Q'(\rho_l(t)) \geq 0\\ \text{xor} & Q'(\rho(0,t)) \leq 0 \text{ and } Q'(\rho_l(t)) \leq 0 \text{ and } \rho\left(0,t\right) = \rho_l\left(t\right)\\ \text{xor} & Q'(\rho(0,t)) \leq 0 \text{ and } Q'(\rho_l(t)) > 0 \text{ and } Q\left(\rho\left(0,t\right)\right) > Q(\rho_l\left(t\right))\\ \text{xor} & Q'(\rho(0,t)) > 0 \text{ and } Q'(\rho_l(t)) < 0\end{cases} \tag{2.23}$$

For the fourth line in (2.23), for *a.e.* $t > 0$ we will have $Q'(\rho(0,t)) = 0$, so it is removed and the conditions for strong left boundary conditions are obtained. □

2.4 The Riemann problem

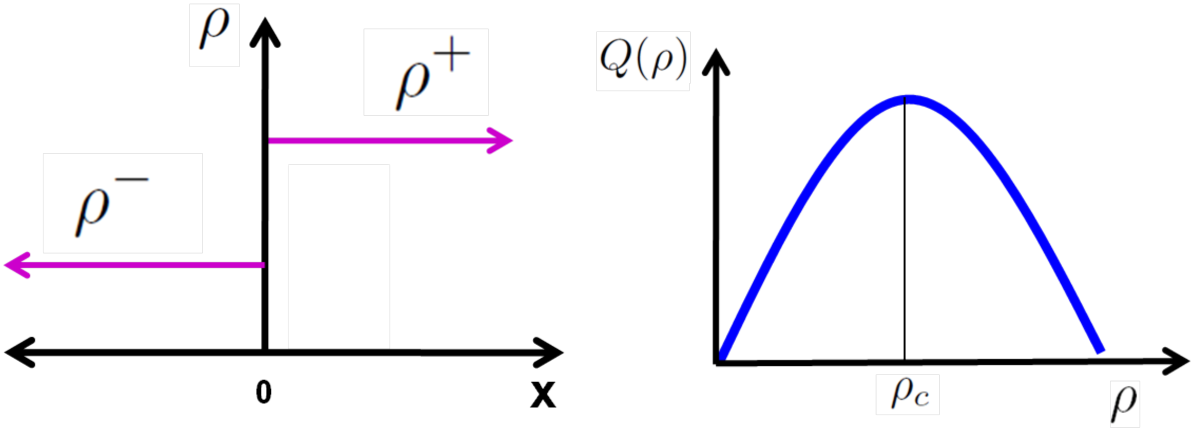
In order to provide clarity on the weak entropy solution to the LWR PDE on a bounded domain, will now introduce the Riemann problem, which is the Cauchy problem equation (2.4) with a Heaviside initial condition. By tracking the evolution of the characteristic curves on this problem, we can easily recover the weak boundary conditions (2.19) and (2.20). Later, it will be used again for numerical discretization of the PDE, and for its extension to networks.

The proof of a global solution to (2.4)–(2.5) by successive local solutions of Riemann problems is due to [18]. We now proceed to explain case-by-case all of the behaviors of the LWR PDE which can arise from the Riemann problem.

Let the initial data for equation (2.4) be given by (Figure 2.3a):

$$\rho_0(x) = \begin{cases} \rho^- & \text{if } x < 0 \\ \rho^+ & \text{if } x > 0 \end{cases} \tag{2.24}$$

and let us again consider a flux function of the form $Q(\rho) = \rho - \rho^2$ (Figure 2.3b). Note that the flux function is increasing for all $\rho \in [0, \rho_c)$, and decreasing for all $\rho \in (\rho_c, \rho_{\max}]$, where $\rho_c = \frac{1}{2}$ is the critical density and $\rho_{\max} = 1$ is the maximal density.



(a) (b)

Figure 2.3: (a) Initial data for the Riemann problem; (b) quadratic flux function.

2.4.1 Riemann solver

The weak entropy Riemann solver for the Riemann problem (2.4) and (2.24) is given by

- If $\rho^+ > \rho^-$,

$$\rho(x,t) = \begin{cases} \rho^- & \text{if } \frac{x}{t} < s \\ \rho^+ & \text{if } \frac{x}{t} > s \end{cases} \tag{2.25}$$

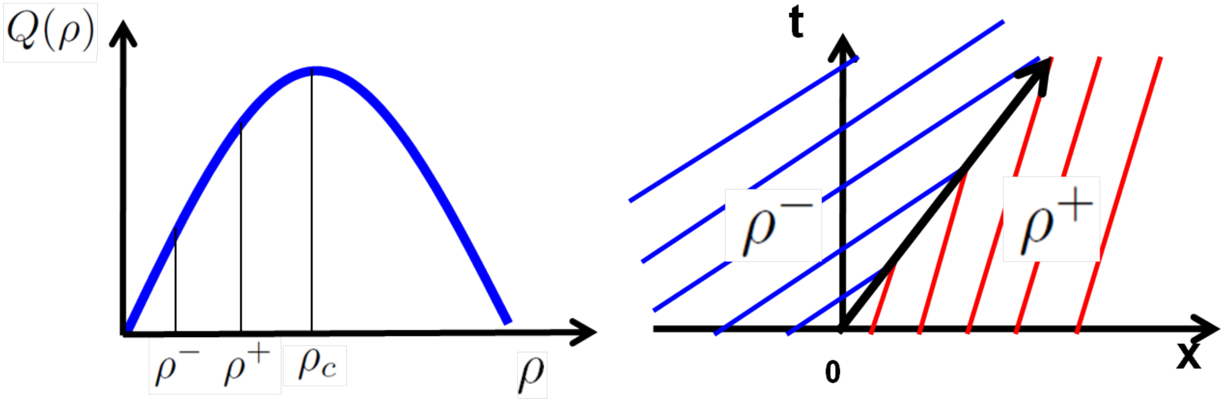
where $s = \frac{Q(\rho^+)-Q(\rho^-)}{\rho^+-\rho^-}$.

- If $\rho^+ < \rho^-$,

$$\rho(x,t) = \begin{cases} \rho^- & \text{if } \frac{x}{t} < Q'(\rho^-) \\ (Q')^{-1}(\frac{x}{t}) & \text{if } Q'(\rho^-) < \frac{x}{t} < Q'(\rho^+) \\ \rho^+ & \text{if } \frac{x}{t} > Q'(\rho^+) \end{cases} \tag{2.26}$$

We now proceed case-by-case through the solutions of the Riemann problem.

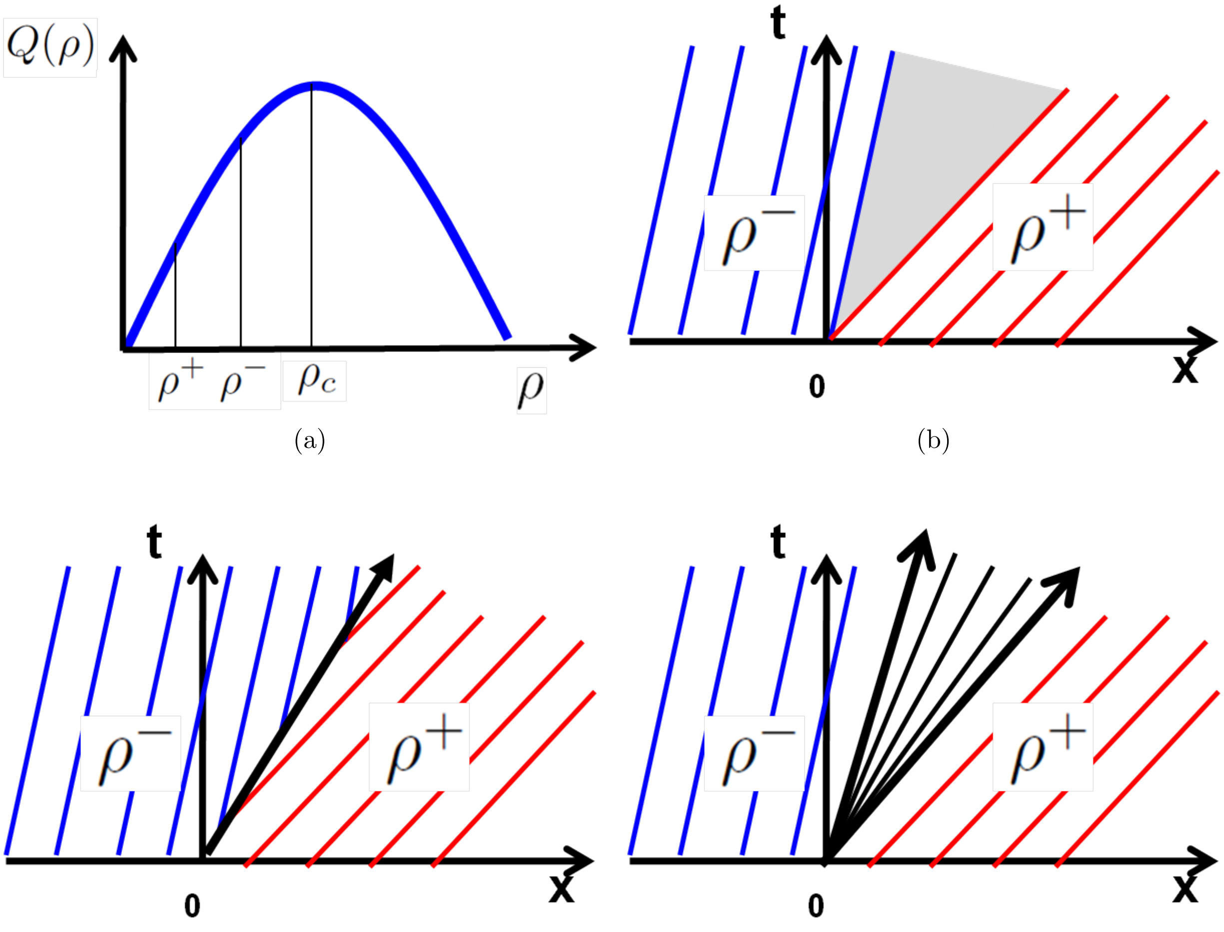
Example 1 (Small shock moving forward)**.** Let the initial data be defined such that $\rho^- \leq \rho^+ \leq \rho_c$ (Figure 2.4a). Then the speeds of the characteristics are ordered by $Q'(\rho^-) \geq Q'(\rho^+) \geq 0$. Since the characteristics on the left move faster than the characteristics on the right, they intersect and a small shock wave is formed (Figure 2.4b). The speed of the shock is given by the Rankine-Hugoniot relation $s = \frac{Q(\rho^+)-Q(\rho^-)}{\rho^+-\rho^-} \geq 0$, and so the shock travels forward. Moreover, since $Q'(\rho^-) \geq s \geq Q'(\rho^+)$, the shock is entropy admissible.



(a) (b)

Figure 2.4: Riemann problem solved by a small shock wave moving forward. (a) initial data; (b) evolution of the characteristic curves.

Example 2 (Rarefaction wave moving forward)**.** Let the initial data be defined such that $\rho^+ \leq \rho^- \leq \rho_c$ (Figure 2.5a). Then the speeds of the characteristics are ordered by $Q'(\rho^+) \geq Q'(\rho^-) \geq 0$. Now the characteristics on the right move faster than the characteristics on the left, and an envelope appears (solid gray area in Figure 2.5b) through which no characteristic curves emanating from the initial condition pass through. Because the weak form of the LWR PDE allows for discontinuities, a forward-moving shock is a weak admissible solution (Figure 2.5c). Moreover, a self-similar solution in the form of a rarefaction wave moving forward (2.26) is also a solution (Figure 2.5d), which does not require a discontinuity. Note, however, that only the rarefaction wave satisfies the entropy condition, since for the shock we have $Q'(\rho^-) \ngeq s \ngeq Q'(\rho^+)$.



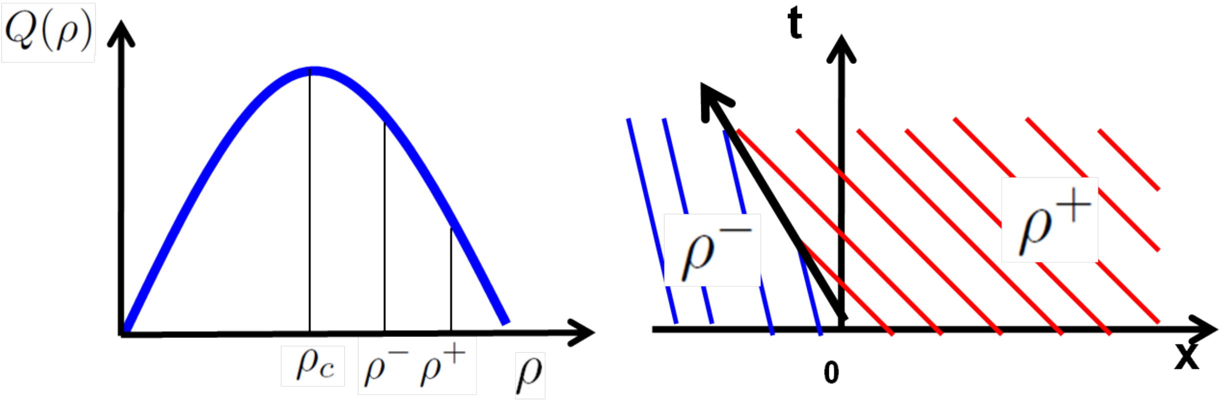
(a) (b)

(c) (d)

Figure 2.5: Riemann problem solved by a rarefaction wave moving forward. (a) initial data; (b) an envelope appears which is not determined by characteristic curves emanating from the initial data; (c) a weak (but not entropy admissible) evolution of the characteristic curves; (d) entropy admissible evolution of the characteristic curves.

Example 3 (Small shock moving back)**.** Let the initial data be defined such that $\rho_c \leq \rho^- \leq$

ρ^+ (Figure 2.6a). Then the speeds of the characteristics are ordered by $0 \geq Q'(\rho^-) \geq Q'(\rho^+)$. Since the characteristics on the left move slower than the characteristics on the right, they intersect and a small shock wave is formed (Figure 2.6b). The speed of the shock is given by the Rankine-Hugoniot relation $s = \frac{Q(\rho^+)-Q(\rho^-)}{\rho^+-\rho^-} \leq 0$, and so the shock travels back. Moreover, since $Q'(\rho^-) \geq s \geq Q'(\rho^+)$, the shock is entropy admissible.



(a)

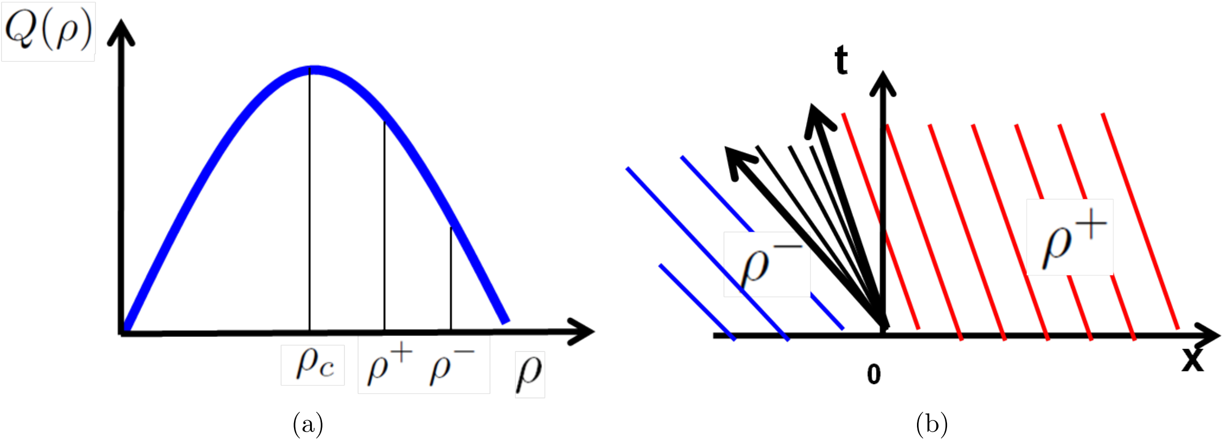
(b)

Figure 2.6: Riemann problem solved by a small shock wave moving back. (a) initial data; (b) evolution of the characteristic curves.

Example 4 (Rarefaction wave moving back)**.** Let the initial data be defined such that $\rho_c \leq \rho^+ \leq \rho^-$ (Figure 2.7a). Then the speeds of the characteristics are ordered by $0 \geq Q'(\rho^+) \geq Q'(\rho^-)$. Now the characteristics on the left move faster than the characteristics on the right, and again an envelope appears through which no characteristic curves emanating from the initial condition pass through (Similar to Example 2). Since the entropy condition prevents the formation of the shock, a rarefaction wave is formed (Figure 2.7b).

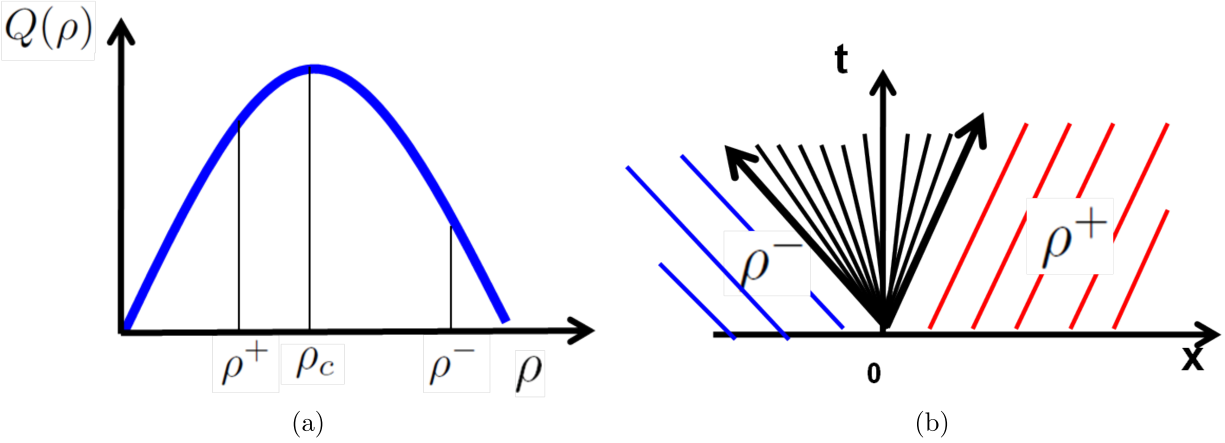
Example 5 (Rarefaction wave moving forward and back)**.** Let the initial data be defined such that $\rho^+ \leq \rho_c \leq \rho^-$ (Figure 2.8a). Then the speeds of the characteristics are ordered by $Q'(\rho^+) \geq 0 \geq Q'(\rho^-)$. Now the characteristics on the left move left, and the characteristics on the right side move right, so again and envelope appears through which no characteristic curves emanating from the initial condition pass through. Since the entropy condition prevents the formation of the shock, a rarefaction wave is formed (Figure 2.8b).

Example 6 (Big shock wave moving back)**.** Let the initial data be defined such that $\rho^- \leq \rho_c \leq \rho^+$, and additionally $Q(\rho^+) < Q(\rho^-)$ (Figure 2.9a). Then the speeds of the characteristics are ordered by $Q'(\rho^-) \geq Q'(\rho^+) \geq 0$. Since the characteristics on the left move right, and the characteristics on the right move left, they intersect and a big shock



(a) (b)

Figure 2.7: Riemann problem solved by a rarefaction wave moving back. (a) initial data; (b) evolution of the characteristic curves.

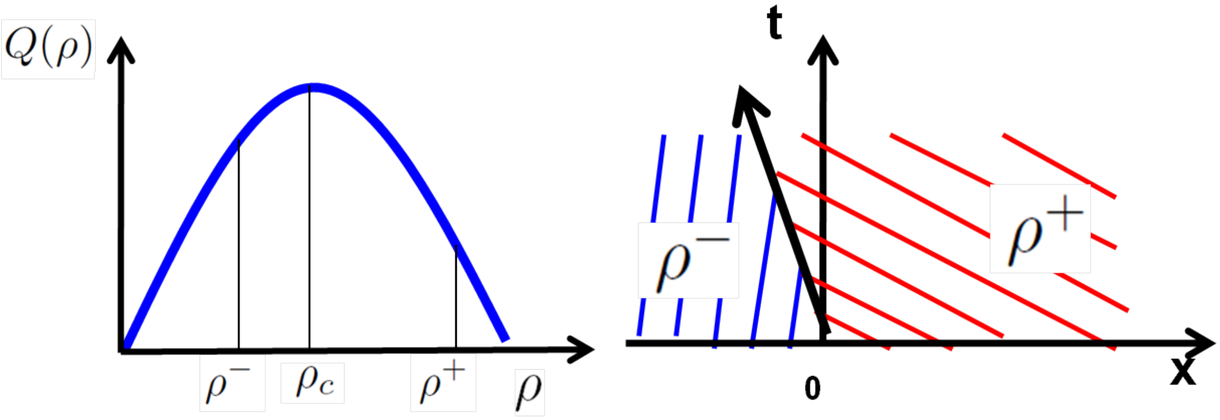


(a) (b)

Figure 2.8: Riemann problem solved by a rarefaction wave moving back and forward. (a) initial data; (b) evolution of the characteristic curves.

wave is formed (Figure 2.9b). The speed of the shock is given by the Rankine-Hugoniot relation $s = \frac{Q(\rho^+)-Q(\rho^-)}{\rho^+-\rho^-} \geq 0$, and since the numerator is negative and the denominator is positive, the shock travels back. Moreover, since $Q'(\rho^-) \geq s \geq Q'(\rho^+)$, the shock is entropy admissible.

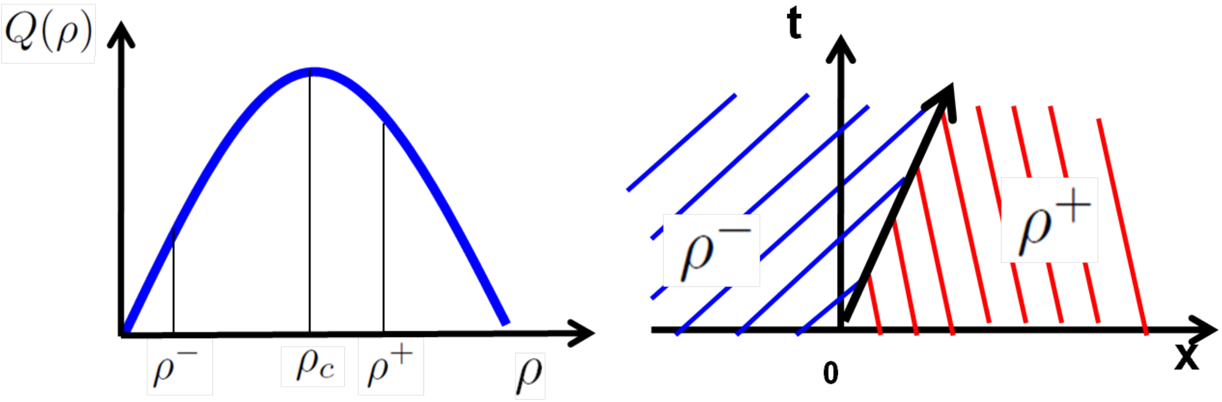
Example 7 (Big shock wave moving forward)**.** Let the initial data be defined such that $\rho^- \leq \rho_c \leq \rho^+$, and additionally $Q(\rho^+) > Q(\rho^-)$ (Figure 2.10a). Then the speeds of the characteristics are ordered by $Q'(\rho^-) \geq Q'(\rho^+) \geq 0$. Since the characteristics on the left move right, and the characteristics on the right move left, they intersect and a big shock wave is formed (Figure 2.10b. The speed of the shock is given by the Rankine-Hugoniot relation



(a) (b)

Figure 2.9: Riemann problem solved by a big shock moving back. (a) initial data; (b) evolution of the characteristic curves.

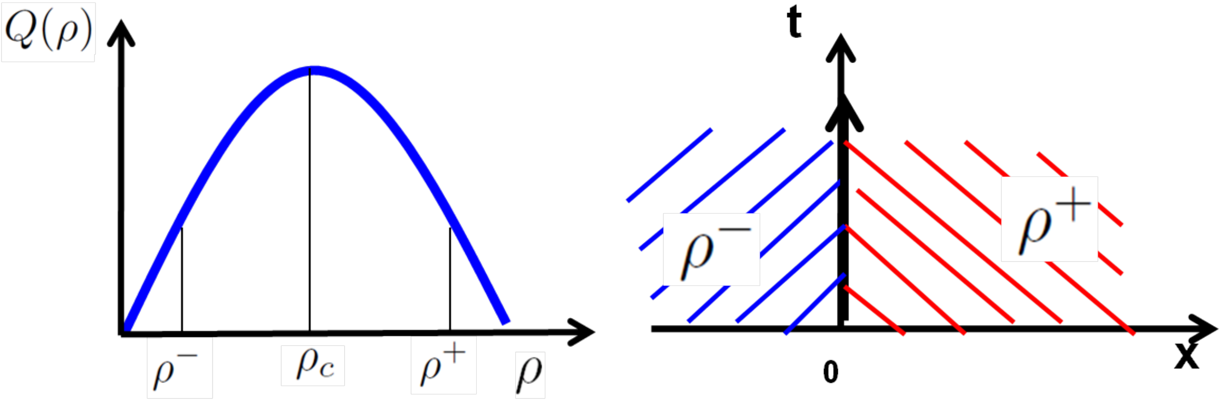
$s = \frac{Q(\rho^+)-Q(\rho^-)}{\rho^+-\rho^-} \geq 0$, and since now both the numerator and denominator are positive, the shock travels forward. Moreover, since $Q'(\rho^-) \geq s \geq Q'(\rho^+)$, the shock is entropy admissible.



(a) (b)

Figure 2.10: Riemann problem solved by a big shock moving forward. (a) initial data; (b) evolution of the characteristic curves.

Example 8 (Big stationary shock wave)**.** Let the initial data be defined such that $\rho^- \leq \rho_c \leq \rho^+$, and additionally $Q(\rho^+) = Q(\rho^-)$ (Figure 2.11a). Then the speeds of the characteristics are ordered by $Q'(\rho^-) \geq Q'(\rho^+) \geq 0$. Since the characteristics on the left move right, and the characteristics on the right move left, they intersect and a big shock wave is formed (Figure 2.11b). The speed of the shock is given by the Rankine-Hugoniot relation $s = \frac{Q(\rho^+)-Q(\rho^-)}{\rho^+-\rho^-} \geq 0$. Because the numerator is zero, the shock wave is stationary. Moreover, since $Q'(\rho^-) \geq s \geq Q'(\rho^+)$, the shock is entropy admissible.



(a) (b)

Figure 2.11: Riemann problem solved by a big stationary shock. (a) initial data; (b) evolution of the characteristic curves.

It is easy to verify that the previous examples are exhaustive on the set of initial conditions for the Riemann problem. The solutions are shown graphically in Figure 2.12. Note that the entropy condition only allows shocks above the line $Q'(\rho^+) = Q'(\rho^-)$, and so solutions below the line form rarefaction waves. The curve corresponding to the big stationary shock is defined by $Q(\rho^+) = Q(\rho^-)$, and its shape depends on the exact flux function used.

2.4.2 Weak boundary conditions revisited

With the solution to the Riemann problem now defined, it is possible to verify the statement of weak boundary conditions given by (2.19) and (2.20), through an interpretation of the Riemann solver. We now proceed to show this for the left boundary condition. Consider the Riemann problem (2.4) with the following initial data defined by (2.4):

$$\rho_0(x) = \begin{cases} \rho_l(0) & \text{if } x < 0 \\ \rho(0,0) & \text{if } x > 0 \end{cases} \tag{2.27}$$

where $\rho_l(0)$ is the left boundary condition we would like to apply (but which may not hold) and $\rho(0,0)$ is the trace of the solution approaching the left boundary. In order to determine if the boundary data applies on the domain, we only need to solve the Riemann

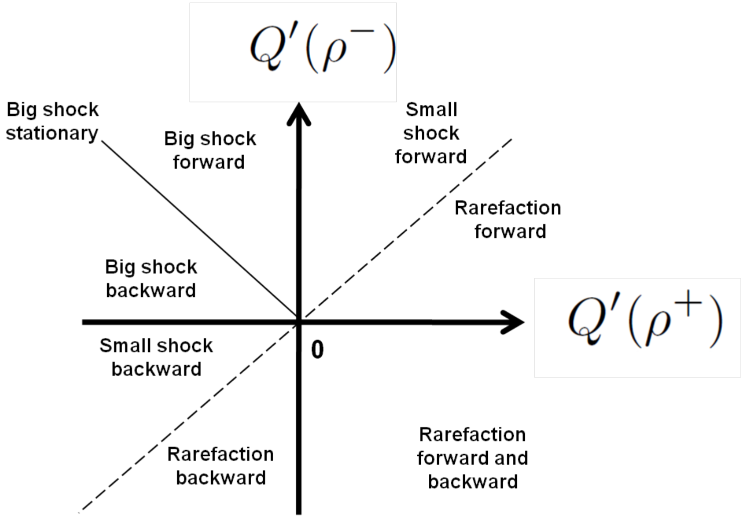


Figure 2.12: Summary of the various Riemann problem solutions as a function of the speed of the characteristics of the initial data.

problem and identify if the characteristic curves associated with $\rho_l(0)$ cross the boundary $x = 0$ for $t > 0$. If so, then the boundary data will carry into the domain, and the boundary condition will hold in the strong sense. Looking back at the previous Riemann problem, it is clear this is true when the solution to the Riemann problem is a forward moving rarefaction wave or a forward moving shock wave. This corresponds to the first line of the left weak boundary condition (2.19).

On the other hand, if the Riemann problem results in a rarefaction wave moving back or a shock wave moving back, then the boundary condition will not hold unless the boundary data and the trace have the same value (the initial condition for the Riemann problem is a single constant value). In the context of a boundary control problem, in this pathological case one must choose the boundary control to be the unique value dictated by the trace, and any perturbation in the boundary control value would cause the solution to be implemented in the weak sense again.

When the solution to the Riemann problem yields a big shock wave moving backward, it is covered by the third line of (2.19), while a small shock wave or a rarefaction wave moving backward corresponds to the second line of (2.19). Finally, the solution to the Riemann problem resulting in a rarefaction wave with characteristics pointing both forward and back corresponds to the second line of (2.19), since for $t > 0$, $Q'(\rho(0,t)) = 0$.

2.5 Numerical discretization

2.5.1 Godunov Scheme

We now describe the Godunov discretization scheme used to numerically approximate weak entropy solutions to the LWR PDE. We discretize the time and space domains by introducing a discrete time step ΔT, indexed by $n \in \{0, \cdots, n_{\max}\}$ and a discrete space step Δx, indexed by $i \in \{0, \cdots, i_{\max}\}$. Let us integrate equation (2.1) over a single timestep, yielding:

$$\begin{aligned} &\int_{x_{i-1/2}}^{x_{i+1/2}} \rho(x, t_{n+1})dx - \int_{x_{i-1/2}}^{x_{i+1/2}} \rho(x, t_n)dx \\ &= \int_{t_n}^{t_{n+1}} Q(\rho(x_{i-1/2}, t))dt - \int_{t_n}^{t_{n+1}} Q(\rho(x_{i+1/2}, t))dt \end{aligned} \tag{2.28}$$

We introduce the variables ρ_i^n and Q_i^n as follows:

$$\rho_i^n \approx \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \rho(x, t_n)dx \tag{2.29}$$

$$Q_i^n \approx \frac{1}{\Delta T} \int_{t_n}^{t_{n+1}} Q(\rho(x_i, t))dt \tag{2.30}$$

where ρ_i^n approximates the average density in the i^{th} cell at time t_n and Q_i^n approximates the average flux at x_i over the time interval $[t_n, t_{n+1}]$. Then after substituting into (2.28) and rearranging terms, we obtain

$$\rho_i^{n+1} = \rho_i^n - \frac{\Delta T}{\Delta x} \left(Q_{i+1/2}^n - Q_{i-1/2}^n \right) \tag{2.31}$$

which is the basis of the Godunov discretization scheme.

In particular, the Godunov discretization scheme is as follows.

1. Approximate the function $\rho(x, t)$ with a piecewise constant function $\bar{\rho}(x, t)$, where $\bar{\rho}(x, t)$ is constant in each cell. Then

$$\begin{aligned} \rho_i^n &= \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \bar{\rho}(x, t_n)dx \\ &= \bar{\rho}(x_i, t_n) \end{aligned}$$

2. Solve the Riemann problems at the cell boundaries $x_{i+1/2}$ and $x_{i-1/2}$. Let $G(\rho_i^n, \rho_{i+1}^n)$ denote the flux at $x_{i+1/2}$ when the Riemann problem is solved between the two states

ρ_i^n and ρ_{i+1}^n. Because $\bar{\rho}(x,t)$ is piecewise constant, the Riemann problem can be solved exactly, and the flux $G(\rho_1, \rho_2)$ is given as (Figure 2.13):

$$G(\rho_1, \rho_2) = \begin{cases} Q(\rho_2) & \text{if } \rho_c \leq \rho_2 \leq \rho_1 \\ Q(\rho_c) & \text{if } \rho_2 \leq \rho_c \leq \rho_1 \\ Q(\rho_1) & \text{if } \rho_2 \leq \rho_1 \leq \rho_c \\ \min(Q(\rho_1), Q(\rho_2)) & \text{if } \rho_1 \leq \rho_2 \end{cases} \tag{2.32}$$

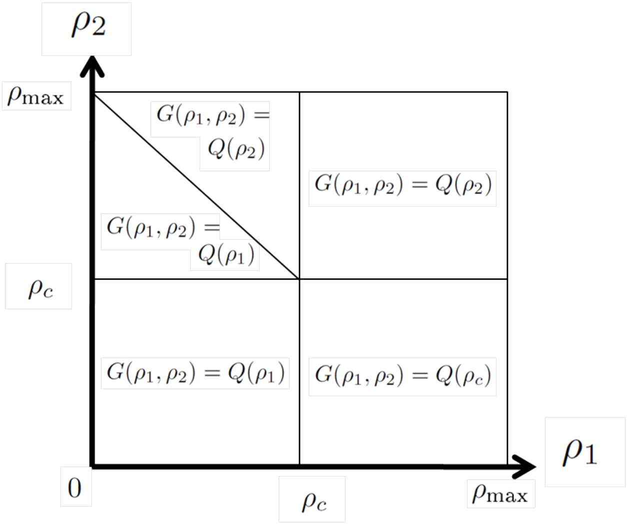


Figure 2.13: Graphical representation of the numerical flux function equation (2.32) as a function of ρ_1 and ρ_2. Note that the line connecting (ρ_c, ρ_c) and $(0, \rho_{\max})$ may have different shapes depending on the flux function $Q(\cdot)$.

This yields

$$\begin{aligned} Q_{i+1/2}^n &= \frac{1}{\Delta T} \int_{t_n}^{t_{n+1}} Q(\bar{\rho}(x_{i+1/2}, t)) dt \\ &= G(\rho_i^n, \rho_{i+1}^n) \end{aligned}$$

3. Compute the density at the next timestep according to (2.31).

In practice, the scheme is implemented as the nonlinear discrete evolution equation:

$$\rho_i^{n+1} = \rho_i^n - \frac{\Delta T}{\Delta x} \left(G\left(\rho_i^n, \rho_{i+1}^n\right) - G\left(\rho_{i-1}^n, \rho_i^n\right) \right) \tag{2.33}$$

In order to ensure numerical stability, the time and space steps are coupled by the CFL condition [53]: $|\alpha_{\max}|\Delta t \leq \Delta x$ where $\alpha_{\max}$ denotes the maximal characteristic speed. This restriction guarantees that the solution of the Riemann problem at each cell boundary is independent of the Riemann problems at adjacent boundaries.

This discrete model is derived independently by Daganzo [20, 21] as a macroscopic traffic model consistent with the LWR PDE, commonly referred to as the *Cell Transmission Model* in the transportation engineering community.

2.5.2 A note on linearization

Later in this dissertation we discuss the traffic velocity estimation problem using a velocity evolution equation consistent with the discretized LWR PDE. Because of the nonlinearity of the hyperbolic conservation law for density, standard linear estimation techniques such as the Kalman filter cannot be used. Moreover, we now show that the LWR PDE discretized with the Godunov discretization scheme cannot be linearized around an arbitrary model state. This unfortunate fact prevents the use of extended Kalman filtering for traffic estimation when using this model (the Cell Transmission Model). The set of states under which the model cannot be linearized corresponds to the case when the Riemann problem is solved with a stationary shock.

Theorem 1. Let $Q(\rho)$ be a smooth C^1 concave flux function with $\rho \in [0, \rho_{\max}]$, with a maximum obtained at ρ_c. Let $\rho^n = \left[\rho_0^n, \cdots, \rho_{i_{\max}}^n\right]$ denote the vector of states at time n. The LWR PDE is discretized according to the Godunov scheme (2.32) and (2.33). The discrete model can be linearized if and only if no standing shockwaves are formed.

Proof. It is easy to see from (2.33) that the model can be linearized if and only if the function $G(\cdot, \cdot)$ is differentiable. We determine the differentiability of (2.32) by an exhaustive computation of the partial derivatives $G(\cdot, \cdot)$ with respect to each of the inputs. Note that $G(\cdot, \cdot)$ is continuous for all $(\rho_1, \rho_2) \in [0, \rho_{\max}] \times [0, \rho_{\max}]$, so it remains to check if the partial derivatives exist and are continuous.

We begin by computing $\frac{\partial G(\rho_1, \rho_2)}{\partial \rho_1}$.

- For $\rho_1, \rho_2 > \rho_c$, we have $\frac{\partial G(\rho_1, \rho_2)}{\partial \rho_1} = 0$. This corresponds to either a small shock or a rarefaction wave moving backward, and $G(\rho_1, \rho_2)$ depends only on ρ_2.

- For $\rho_1 > \rho_c > \rho_2$, we have $\frac{\partial G(\rho_1, \rho_2)}{\partial \rho_1} = 0$. This corresponds to a rarefaction wave with characteristics moving forward and backward, and $G(\rho_1, \rho_2)$ is a constant.

- For $\rho_1 > \rho_2 = \rho_c$, G(ρ_1, ρ_2) is again independent of ρ_1, and so $\frac{\partial G(\rho_1, \rho_2)}{\partial \rho_1} = 0$.

- For $\rho_1, \rho_2 < \rho_c$, we have $\frac{\partial G(\rho_1, \rho_2)}{\partial \rho_1} = Q'(\rho_1)$, which corresponds to either a small shock or a rarefaction wave moving forward.

- For $\rho_2 < \rho_1 = \rho_c$, we have $\lim_{\rho_1 \to \rho_c^+} \frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1} = 0$, and $\lim_{\rho_1 \to \rho_c^-} \frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1} = \lim_{\rho_1 \to \rho_c^-} Q'(\rho_1)$ $= 0$, since $Q(\cdot)$ is maximized at ρ_c, so the partial derivative is continuous.
- The case when $\rho_1 < \rho_c < \rho_2$ corresponds to a big shock, so we must further specify the ordering of the flux. Let $Q(\rho_2) > Q(\rho_1)$, so the shock moves forward. In this case, $\frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1} = Q'(\rho_1)$.
- When $\rho_1 < \rho_2 = \rho_c$ and $Q(\rho_2) > Q(\rho_1)$, we have $\frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1} = Q'(\rho_1)$ and the solution is again a forward moving shock.
- When $\rho_1 < \rho_c < \rho_2$ and $Q(\rho_2) < Q(\rho_1)$, we have $\frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1} = 0$ since $G(\rho_1, \rho_2)$ depends only on ρ_2. The solution is a big shock moving back.
- When $\rho_1 = \rho_c < \rho_2$ and $Q(\rho_2) < Q(\rho_1)$, we have $\frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1} = 0$ in the solution is again a shock moving back.
- The last case is when $\rho_1 < \rho_c < \rho_2$ and $Q(\rho_2) = Q(\rho_1)$. For a given ρ_2, let ρ^* be defined such that $\rho^* < \rho_c < \rho_2$ and $Q(\rho^*) = Q(\rho_2)\}$. Then $\lim_{\rho_1 \to \rho^{*+}} \frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1} = 0$, and $\lim_{\rho_1 \to \rho^{*-}} \frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1} = Q'(\rho_1) > 0$ from the concavity of the flux function. Thus, solution is not differentiable around this point, which corresponds to a standing shock wave.

The proof for $\frac{\partial G(\rho_1,\rho_2)}{\partial \rho_2} = 0$ proceeds similarly, and is also differentiable everywhere except around states corresponding to a standing shock wave. The computation of the partial derivatives are summarized in Figure 2.14.

□

Direct application of extended Kalman filtering [74], or implicit switching between linearized regimes [73] will not be applicable around these points. Fortunately, in practical traffic estimation problems, the state around which the model is linearized will rarely result in a stationary shock; the numerical values of the flux almost always cause the shock to move slightly forward or back. Nevertheless, this still creates a theoretical challenge which has not yet been addressed in a traffic estimation literature.

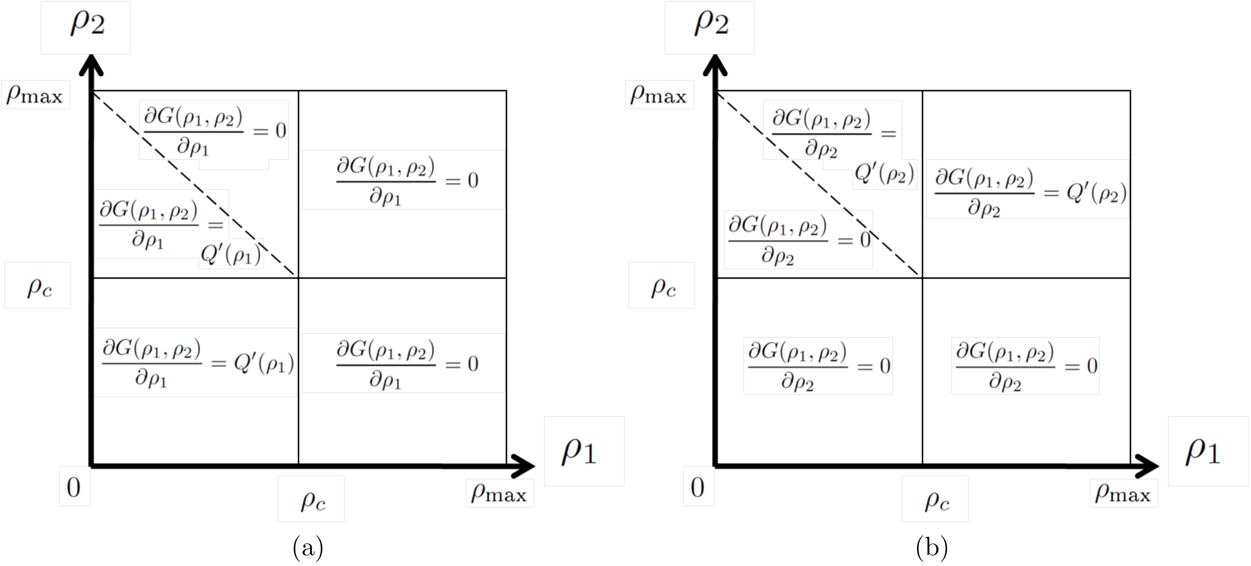


Figure 2.14: Summary of the computation of (a) $\frac{\partial G(\rho_1,\rho_2)}{\partial \rho_1}$; (b) $\frac{\partial G(\rho_1,\rho_2)}{\partial \rho_2}$. Note that the line connecting (ρ_c, ρ_c) and $(0, \rho_{\max})$ may have different shapes depending on the flux function $Q(\cdot)$, and represents the set for which $G(\cdot,\cdot)$ is not differentiable.

Chapter 3

Derivation of a velocity evolution equation

3.1 Introduction

Motivated by the problem of estimating traffic conditions using only velocity measurements from mobile phones, this chapter focuses on the development of a mathematical model for velocity evolution consistent with the LWR PDE. The contributions in this chapter are as follows.

- **Derivation of a velocity partial differential equation**. When the relationship between velocity and density is affine (as is the case for the Greenshields flux function), we derive a new velocity conservation law consistent with the weak form of the LWR PDE, called the *LWR–v* PDE.
- **Limitation of the velocity partial differential equation**. For general invertible velocity functions which are not affine, we show that there is no equivalent velocity conservation law. This is a negative result.
- **Derivation of a discrete velocity evolution equation**. For general, nonlinear invertible velocity functions, we derive a numerical approximation to the integral form of the LWR PDE (2.1), which describes velocity evolution on a discrete domain and overcomes the above limitation. We call this model the *Cell Transmission Model for velocity* (CTM–v), due to similarities with the CTM model.
- **Extension to networks**. We discuss how to extend the velocity evolution equation to networks of roads by using a generalized Riemann solver consistent with [14, 21].

The chapter begins with an introduction of several velocity functions which have been historically used in the LWR PDE, in Section 3.2. In Section 3.3, a conservation law for velocity consistent with the LWR PDE is derived for the Greenshields velocity function. We prove for general invertible velocity functions that this equivalence cannot be achieved. In Section 3.4 we circumvent this issue by developing a discrete time discrete space velocity evolution equation consistent with the discretized LWR PDE. The model is extended to networks in Section 3.5.

3.2 Velocity functions

In order to obtain a velocity of local evolution equation consistent with density, we require the velocity function used in the LWR PDE (2.6) to be invertible. The algebraic expression of the velocity function is a modeling choice, and it is typically constructed to fit experimental data.

Introduced in 1935, one of the earliest velocity functions considered is the Greenshields [33] affine velocity function:

$$v = V_G\left(\rho\right) = v_{\max}\left(1 - \rho/\rho_{\max}\right)$$

where $v_{\max}$ is the maximum (freeflow) velocity, and $\rho_{\max}$ is the maximum (jam) density. This model remains a useful mathematical model because of its algebraic simplicity, despite disagreements with observed traffic data. Since it expresses a linear relationship between speed and density, it is clearly invertible as:

$$\rho = V_G^{-1}\left(v\right) = \rho_{\max}\left(1 - v/v_{\max}\right) \tag{3.1}$$

The widely used Daganzo–Newell velocity function assumes a constant velocity in freeflow and a hyperbolic velocity in congestion:

$$v = V_{\mathrm{DN}}(\rho) = \begin{cases} v_{\max} & \text{if } \rho \le \rho_c \\ -w_f\left(1 - \frac{\rho_{\max}}{\rho}\right) & \text{otherwise} \end{cases}$$

where $v_{\max}$, $\rho_{\max}$, ρ_c and w_f are respectively the maximum velocity, maximum density, critical density at which the flow transitions from free-flow to congested, and the backwards propagating wave speed, respectively. Because the Daganzo-Newell velocity function is not strictly monotonic in freeflow, it cannot be inverted.

In order to use the Daganzo-Newell model in a velocity setting, we approximate it by the Smulders velocity function [68], with a linear expression in free-flow and a hyperbolic expression in congestion:

$$v = V_{\mathrm{S}}(\rho) = \begin{cases} v_{\max}\left(1 - \frac{\rho}{\rho_{\max}}\right) & \text{if } \rho \le \rho_c \\ -w_f\left(1 - \frac{\rho_{\max}}{\rho}\right) & \text{otherwise} \end{cases}$$

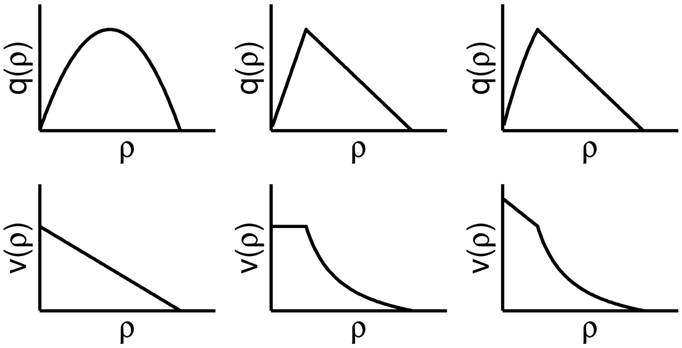


Figure 3.1: Fundamental diagrams (top row) and velocity functions (bottom row) for Greenshields (left), Daganzo-Newell (center), and Smulders (right).

For continuity of the flux at the critical density ρ_c, the additional relation $\frac{\rho_c}{\rho_{\max}} = \frac{w_f}{v_{\max}}$ must be satisfied.

The Smulders velocity function can be inverted to obtain the velocity as a function of density:

$$\rho = V_{\mathrm{S}}^{-1}(v) = \begin{cases} \rho_{\max}\left(1 - \frac{v}{v_{\max}}\right) & \text{if } v \geq v_c \\ \rho_{\max}\left(\frac{1}{1+\frac{v}{w_f}}\right) & \text{otherwise} \end{cases} \tag{3.2}$$

where v_c is the critical velocity: $v_c = V(\rho_c)$. This Smulders velocity function yields a quadratic-linear flux function as illustrated in Figure 3.1.

Unless noted otherwise, we assume that the velocity function is invertible throughout the remainder of this dissertation.

3.3 Derivation of a velocity PDE in conservative form for the Greenshields flux function

In this section, we derive a velocity PDE in conservative form for the Greenshields flux and we show that for other C^1 velocity functions, there is no velocity transport equation equivalent to the LWR equation. The important result shown here is that unless the velocity function is affine (i.e., the Greenshields case), there will not be equivalence between weak solutions to the derived velocity PDE and the weak solutions of the density PDE written in terms of the velocity.

First, we introduce the notion of a weak velocity solution to the LWR PDE. Assuming that the velocity function is invertible with inverse $V^{-1}(\cdot)$, the PDE (2.16) in weak form for

$\rho(\cdot,\cdot)$ is equivalent to the following formulation for $v(\cdot,\cdot)$:

$$\int_0^L \int_0^T \left(V^{-1}(v(x,t))\, \frac{\partial \varphi}{\partial t}(x,t) + Q(V^{-1}(v(x,t)))\frac{\partial \varphi}{\partial x}(x,t) \right) dxdt \\ + \int_0^L V^{-1}(v_0(x))\, \varphi(x,0) dx = 0 \quad \forall \varphi \in C_c^2\left([0,L] \times [0,T)\right) \tag{3.3}$$

In order to use existing numerical analysis schemes for the PDE we want to obtain, we would like to transform the weak formulation (3.3) into the following conservation law for velocity with initial condition $v_0(\cdot)$:

$$\begin{cases} \frac{\partial}{\partial t} v(x,t) + \frac{\partial}{\partial x} R\left(v(x,t)\right) = 0 \\ v(x,0) = v_0(x) \end{cases} \tag{3.4}$$

By analogy with the classical LWR equation, the velocity PDE (3.4) is called LWR-v PDE. Because the flux function $R(v)$ in the velocity conservation law (3.4) is convex, the weak boundary conditions are given as follows:

Definition 4 (Weak boundary conditions - convex flux function [8, 27])**.** For a convex flux function $R(\cdot)$, the weak formulation of boundary conditions reads:

$$\text{for } a.e.\ t > 0, \\ \begin{cases} & v(0,t) = v_l(t) \\ \text{xor} & R'(v(0,t)) \leq 0 \text{ and } R'(v_l(t)) \leq 0 \text{ and } v(0,t) \neq v_l(t) \\ \text{xor} & R'(v(0,t)) \leq 0 \text{ and } R'(v_l(t)) > 0 \text{ and } R(v(0,t)) \geq R(v_l(t)) \end{cases}$$

and

$$\text{for } a.e.\ t > 0, \\ \begin{cases} & v(L,t) = v_r(t) \\ \text{xor} & R'(v(L,t)) \geq 0 \text{ and } R'(v_r(t)) \geq 0 \text{ and } v(L,t) \neq v_r(t) \\ \text{xor} & R'(v(L,t)) \geq 0 \text{ and } R'(v_r(t)) < 0 \text{ and } R(v(L,t)) \geq R(v_r(t)) \end{cases}$$

where $v_l(\cdot)$, $v_r(\cdot)$ are functions of $C^0(0,T)$. The functions $v_l(\cdot)$ and $v_r(\cdot)$ are the strong boundary conditions one wants to apply at the left and the right boundaries.

We can now state the main result of this section, which defines the velocity functions for which a velocity evolution PDE in conservative form can be constructed.

Theorem 2. For a velocity function piecewise analytic in $[0,\rho_{\max}]$, the velocity PDE in weak form (3.3) is equivalent to system (3.4) if and only if the velocity function is affine (Greenshields case).

Proof. The proof proceeds in two steps. Beginning with equation (3.3) instantiated for the Greenshields velocity function $V_G(\cdot)$ defined by (3.1), we show that the conservation equation obtained is the one from system (3.4). Substitution of the explicit expression of V_G^{-1} in (3.3) yields:

$$\begin{aligned}\int_0^L\int_0^T \rho_{\max}\frac{\partial}{\partial t}\varphi(x,t)dxdt - \int_0^L\int_0^T \frac{\rho_{\max}}{v_{\max}}\,v(x,t)\,\frac{\partial}{\partial t}\varphi(x,t)dxdt \\ + \int_0^L\int_0^T Q_G\left(\rho_{\max}-\frac{\rho_{\max}}{v_{\max}}\,v(x,t)\right)\frac{\partial}{\partial x}\varphi(x,t)dxdt \\ - \int_0^L \frac{\rho_{\max}}{v_{\max}}\,v_0(x)\,\varphi(x,0)dx + \int_0^L \rho_{\max}\,\varphi(x,0)dx = 0\end{aligned}$$

where $Q_G(\rho)=\rho\,V_G(\rho)$. Since $\varphi\in C_c^2([0,L]\times[0,T))$ the first term equals $-\int_0^L \rho_{\max}\,\varphi(x,0)dx$ and cancels with the last term. Multiplication by $-\frac{v_{\max}}{\rho_{\max}}$ gives:

$$\begin{aligned}\int_0^L\int_0^T v(x,t)\,\frac{\partial}{\partial t}\varphi(x,t)dxdt + \int_0^L v_0(x)\,\varphi(x,0)dx \\ - \int_0^L\int_0^T \frac{v_{\max}}{\rho_{\max}}\,Q_G\left(\rho_{\max}-\frac{\rho_{\max}}{v_{\max}}\,v(x,t)\right)\frac{\partial}{\partial x}\varphi(x,t)dxdt = 0\end{aligned}$$

which means that v is a weak solution of the PDE:

$$\frac{\partial}{\partial t}v(x,t)+\frac{\partial}{\partial x}\left(R_G(v(x,t))\right)=0$$

with the initial condition $v(x,0)=v_0(x)$, and the velocity flux function

$$R_G(v)=-\frac{v_{\max}}{\rho_{\max}}\,Q_G(V_G^{-1}(v))=v^2-v_{\max}\,v$$

This completes the first part of the proof.

Now, we show that the Rankine-Hugoniot jump condition [24, 53] is not conserved in the transformation from (2.16) to (3.4) for the general case, which means that the equivalence is not obtained for general flux functions.

First, note that a necessary condition to have equivalence between the LWR PDE (2.16) and the LWR-v PDE (3.4) is to have the same characteristic speeds for a state ρ in (2.16) and for the state $V(\rho)$ in (3.4). This yields $Q'(V^{-1}(v))=R'(v)$. Integrating this relation between any states (ρ_1,v_1) and (ρ_2,v_2) we obtain:

$$\int_{v_1}^{v_2} Q'(V^{-1}(v))dv=\int_{v_1}^{v_2} R'(v)dv$$

Using the variable change $v = V(\rho)$, we obtain:

$$\int_{\rho_1}^{\rho_2} Q'(\rho)\, V'(\rho) d\rho = \int_{v_1}^{v_2} R'(v) dv \tag{3.5}$$

Next, at a discontinuity, the Rankine-Hugoniot jump condition [24, 53] reads:

$$\frac{Q(\rho_2) - Q(\rho_1)}{\rho_2 - \rho_1} = \frac{R(v_2) - R(v_1)}{v_2 - v_1} \tag{3.6}$$

which we can rewrite as:

$$\int_{v_1}^{v_2} R'(v) dv = \frac{v_2 - v_1}{\rho_2 - \rho_1} \int_{\rho_1}^{\rho_2} Q'(\rho) d\rho \tag{3.7}$$

If we substitute equality (3.5) into equation (3.7) we obtain:

$$\int_{\rho_1}^{\rho_2} Q'(\rho)\, V'(\rho) d\rho = \frac{V(\rho_2) - V(\rho_1)}{\rho_2 - \rho_1} \int_{\rho_1}^{\rho_2} Q'(\rho) d\rho$$

which translates to:

$$\int_{\rho_1}^{\rho_2} V'(\rho)\, (V(\rho) + \rho\, V'(\rho)) d\rho = \left(\frac{1}{\rho_2 - \rho_1} \int_{\rho_1}^{\rho_2} V'(\rho) d\rho \right) \left(\int_{\rho_1}^{\rho_2} (V(\rho) + \rho\, V'(\rho)) d\rho \right) \tag{3.8}$$

If we define the function G_{ρ_1} in $[\rho_1, \rho_i]$ by $G_{\rho_1}(\rho_2) = \frac{1}{\rho_2 - \rho_1} \int_{\rho_1}^{\rho_2} V'(\rho) d\rho$, on intervals on which V is smooth, we can write:

$$V'(\rho_2)\, (V(\rho_2) + \rho_2\, V'(\rho_2)) = G'_{\rho_1}(\rho_2)\, (\rho_2\, V(\rho_2) - \rho_1\, V(\rho_1)) + G_{\rho_1}(\rho_2)\, (V(\rho_2) + \rho_2\, V'(\rho_2)) \tag{3.9}$$

Given the expression of G_{ρ_1}, if we differentiate $(\rho_2 - \rho_1)\, G_{\rho_1}(\rho_2)$ w.r.t ρ_2 we obtain for all ρ_2 in $[\rho_1, \rho_i]$:

$$((\rho_2 - \rho_1)\, G_{\rho_1}(\rho_2))' = G_{\rho_1}(\rho_2) + (\rho_2 - \rho_1)\, G'_{\rho_1}(\rho_2) = V'(\rho_2)$$

Thus if we factor $V(\rho_2) + \rho_2\, V'(\rho_2)$ in the first and last term of (3.9) and if we replace $G_{\rho_1}(\rho_2) - V'(\rho_2)$ by $-(\rho_2 - \rho_1)\, G'_{\rho_1}(\rho_2)$ we obtain:

$$G'_{\rho_1}(\rho_2)\, ((\rho_2\, V(\rho_2) - \rho_1\, V(\rho_1)) - (\rho_2 - \rho_1)\, (V(\rho_2) + \rho_2\, V'(\rho_2))) = 0 \tag{3.10}$$

The second term in the product can be written as $Z(\rho_1, \rho_2) = Q(\rho_2) - Q(\rho_1) - (\rho_2 - \rho_1)\, Q'(\rho_2)$. So either $Q(\cdot)$ is affine and $Z(\rho_1, \rho_2)$ is zero, either Q is strictly concave or strictly convex and $Z(\rho_1, \rho_2)$ is different from zero, and the first term of (3.10) must be zero. If the first term in (3.10) is zero, it means that V is of the form $V(\rho) = a\, \rho + b$. If the second term is

zero, it means that V is of the form $V(\rho) = \frac{a}{\rho} + b$. So we obtain a necessary condition that V must be piecewise affine or hyperbolic.
If there exists a point $\rho_i \in [0, \rho_{\max}]$ s.t. V has a different algebraic expression for $\rho > \rho_i$ and $\rho < \rho_i$, simple algebra shows that the equality of the Rankine-Hugoniot speeds (3.6) does not hold in general. Therefore V is either of the form $a\,\rho + b$ in $[0, \rho_{\max}]$, or $\frac{a}{\rho} + b$ in $[0, \rho_{\max}]$. The second possibility is excluded by assumption on V (unbounded speed as ρ goes to zero). $\square$

Thus for more realistic traffic models with nonlinear velocity functions, it is not possible to derive a PDE model for velocity in conservation form (3.4).

3.4 Numerical approximation of the velocity evolution equation

Since a partial differential equation for velocity consistent with the LWR PDE does not exist for arbitrary invertible velocity functions, we instead return to the integral form of the LWR PDE (2.1) to perform the variable change. Then we will derive a Godunov scheme for velocity to approximate the solution.

Following the same procedure as in Section 2.5.1, we discretize the time and space domains by introducing a discrete time step ΔT, indexed by $n \in \{0, \cdots, n_{\max}\}$ and a discrete space step Δx, indexed by $i \in \{0, \cdots, i_{\max}\}$. Let us integrate equation (2.1) over a single timestep, and apply the variable change $\rho(x,t) = V^{-1}(v(x,t))$:

$$\begin{aligned} &\int_{x_{i-1/2}}^{x_{i+1/2}} V^{-1}(v(x,t_{n+1}))dx - \int_{x_{i-1/2}}^{x_{i+1/2}} V^{-1}(v(x,t_n))dx \\ &\qquad = \int_{t_n}^{t_{n+1}} Q(V^{-1}(v(x_{i-1/2},t)))dt - \int_{t_n}^{t_{n+1}} Q(V^{-1}(v(x_{i+1/2},t)))dt \end{aligned} \tag{3.11}$$

Define the piecewise constant function $\bar{v}(x,t) := V^{-1}(\bar{\rho}(x,t))$ where $\bar{\rho}(x,t)$, is a piecewise constant approximation of $\rho(x,t)$ and $\bar{\rho}(x,t)$ is a constant in each cell. We introduce the variable v_i^n and recall the definition of Q_i^n as follows:

$$v_i^n = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \bar{v}(x,t_n)dx \tag{3.12}$$

$$Q_i^n \approx \frac{1}{\Delta T} \int_{t_n}^{t_{n+1}} Q(V^{-1}(v(x_i,t)))dt \tag{3.13}$$

where v_i^n approximates the average velocity in the i^{th} cell at time t_n and Q_i^n approximates the average flux at x_i over the time interval $[t_n, t_{n+1}]$. Substituting into (3.11) and rearranging, we obtain

$$V^{-1}(v_i^{n+1}) = V^{-1}(v_i^n) - \frac{\Delta T}{\Delta x}\left(Q_{i+1/2}^n - Q_{i-1/2}^n\right) \tag{3.14}$$

which is the basis of the Godunov discretization scheme for velocity evolution. The full algorithm is as follows.

1. Approximate the function $v(x,t)$ with a piecewise constant function $\bar{v}(x,t)$, where $\bar{v}(x,t)$ is constant in each cell.

2. Solve the Riemann problems at the cell boundaries $x_{i+1/2}$ and $x_{i-1/2}$. Let $\tilde{G}(v_i^n, v_{i+1}^n)$ denote the flux at $x_{i+1/2}$ when the Riemann problem is solved between the two states v_i^n and v_{i+1}^n. For consistency with the density evolution, we require $\tilde{G}(v_i^n, v_{i+1}^n) = G(V^{-1}(v_i^n), V^{-1}(v_{i+1}^n))$ where $G(\rho_1, \rho_2))$ is given by (2.32), and solves the Riemann problem exactly. Moreover, let us define $\tilde{Q}(v) := Q(V^{-1}(v))$, so by substitution into (2.32), we obtain

$$\tilde{G}(v_1, v_2) = \begin{cases} \tilde{Q}(v_2) & \text{if } V^{-1}(\rho_c) \le V^{-1}(v_2) \le V^{-1}(v_1) \\ \tilde{Q}(v_c) & \text{if } V^{-1}(v_2) \le V^{-1}(v_c) \le V^{-1}(v_1) \\ \tilde{Q}(v_1) & \text{if } V^{-1}(v_2) \le V^{-1}(v_1) \le V^{-1}(v_c) \\ \min\left(\tilde{Q}(v_1),\ \tilde{Q}(v_2)\right) & \text{if } V^{-1}(v_1) \le V^{-1}(v_2) \end{cases} \tag{3.15}$$

Note that if $\rho_1 \le \rho_2$, with $v_1 = V(\rho_1)$ and $v_2 = V(\rho_2)$, then $v_1 \ge v_2$ when $V(\cdot)$ is monotonically decreasing (which is typically the case for traffic applications). Then $\tilde{G}(v_i^n, v_{i+1}^n)$ is given by:

$$\tilde{G}(v_1, v_2) = \begin{cases} \tilde{Q}(v_2) & \text{if } v_c \ge v_2 \ge v_1 \\ \tilde{Q}(v_c) & \text{if } v_2 \ge v_c \ge v_1 \\ \tilde{Q}(v_1) & \text{if } v_2 \ge v_1 \ge v_c \\ \min\left(\tilde{Q}(v_1),\ \tilde{Q}(v_2)\right) & \text{if } v_1 \ge v_2 \end{cases} \tag{3.16}$$

This yields

$$\begin{aligned} Q_{i+1/2}^n &= \frac{1}{\Delta T}\int_{t_n}^{t_{n+1}} Q(V^{-1}(\bar{v}(x_{i+1/2}, t)))dt \\ &= \tilde{G}(v_i^n, v_{i+1}^n) \end{aligned}$$

3. Compute the density at the next timestep according to (3.14).

In practice, the scheme is implemented as the nonlinear discrete evolution equation:

$$v_i^{n+1} = V\left(V^{-1}\left(v_i^n\right) - \frac{\Delta T}{\Delta x}\left(\tilde{G}\left(v_i^n, v_{i+1}^n\right) - \tilde{G}\left(v_{i-1}^n, v_i^n\right)\right)\right) \tag{3.17}$$

which we call the *Cell Transmission Model for velocity* CTM-v.

Example 9 (Smulders model)**.** After evaluation of the function (3.2), equation (3.16) reduces to:

$$\tilde{G}\left(v_1, v_2\right) = \begin{cases} v_2 \rho_{\max}\left(\frac{1}{1+\frac{v_2}{w_f}}\right) & \text{if } v_c \geq v_2 \geq v_1 \\ v_c \rho_{\max}\left(1 - \frac{v_c}{v_{\max}}\right) & \text{if } v_2 \geq v_c \geq v_1 \\ v_1 \rho_{\max}\left(1 - \frac{v_1}{v_{\max}}\right) & \text{if } v_2 \geq v_1 \geq v_c \\ \min\left(V_{\mathrm{S}}^{-1}\left(v_1\right) v_1,\ V_{\mathrm{S}}^{-1}\left(v_2\right) v_2\right) & \text{if } v_1 \geq v_2 \end{cases} \tag{3.18}$$

We choose not to simplify the last line in (3.18) due to the piecewise analytical expression of function $V_{\mathrm{S}}^{-1}(\cdot)$.

We note that the evolution of the velocity field at each discrete point on an edge except at the boundary points v_0^n and $v_{i_{\max}}^n$ is well defined by (3.17) and (3.18). At these boundaries, the equations

$$v_0^{n+1} = V\left(V^{-1}\left(v_0^n\right) - \frac{\Delta T}{\Delta x}\left(\tilde{G}\left(v_0^n, v_1^n\right) - \tilde{G}\left(v_{-1}^n, v_0^n\right)\right)\right) \tag{3.19}$$

$$v_{i_{\max}}^{n+1} = V\left(V^{-1}\left(v_{i_{\max}}^n\right) - \frac{\Delta T}{\Delta x}\left(\tilde{G}\left(v_{i_{\max}}^n, v_{i_{\max}+1}^n\right) - \tilde{G}\left(v_{i_{\max}-1}^n, v_{i_{\max}}^n\right)\right)\right) \tag{3.20}$$

contain references to the ghost sells v_{-1}^n and $v_{i_{\max}+1}^n$, which are points which do not lie in the physical domain. The values of v_{-1}^n and $v_{i_{\max}+1}^n$ are given by the prescribed boundary conditions to be imposed on the left and right side of the domain respectively. Note that these boundary values do not always affect the physical domain because of the nonlinear operator (3.18), which causes the boundary conditions to be implemented in the weak sense (See Section 2.4.2).

3.5 Extension of the model to networks

3.5.1 Network model and edge boundary conditions at junctions

We now describe the extension of the velocity evolution equation to networks. On each edge, the velocity field evolves according to (3.17), with an important modification in the computation of the points at the boundary. Instead of implementing ghost points, it is natural to require the left and right boundary conditions to be a function of upstream and downstream edges, so that the velocity field can be evolved across the network.

We model the highway transportation network as a directed graph consisting of vertices $\nu \in \mathcal{V}$ and edges $e \in \mathcal{E}$. Let L_e be the length of edge e. The spatial and temporal variables are $x \in [0,\ L_e]$, and $t \in [0, +\infty)$ respectively. In order to model traffic flow across the network, we define a junction $j \in \mathcal{J}$ as a tuple $(\nu_j,\ I_j,\ \mathcal{O}_j) \subseteq \mathcal{V} \times \mathcal{E} \times \mathcal{E}$, consisting of a single vertex $\nu_j \in \mathcal{V}$, a set of incoming edges indexed by $e_{\text{in}} \in \mathcal{I}_j$, and a set of outgoing edges indexed by $e_{\text{out}} \in \mathcal{O}_j$.

In general, extending the velocity model to handle these networks is challenging because one must prescribe a unique solution to the velocity Riemann problem on junctions, where multiple road segments merge or diverge. In general, even with mass conservation and a natural extension of the entropy condition, a unique solution to the Riemann problem in the density domain is not guaranteed, and therefore a unique solution in the velocity domain is not guaranteed either.

To illustrate the problem, let us consider the density Riemann problem for a junction with one incoming edge ($\mathcal{I} = \{1\}$) and to outgoing edges ($\mathcal{O} = \{2, 3\}$), with initial conditions given by $\rho_1(x, 0) = \rho_{\max}$ and $\rho_2(x, 0) = \rho_3(x, 0) = 0$, shown in Figure 3.2a. Let us also assume that the flux functions on each edge are the same.

An infinite number of solutions exist which satisfy the LWR PDE on each edge. For example, Figure 3.2b shows one such solution, where no vehicles pass through the vertex. The solution is given by $\rho_1(x, 0) = \rho_{\max}$ and $\rho_2(x, 0) = \rho_3(x, 0) = 0$. Because the data is piecewise constant on each edge, of the LWR PDE is trivially satisfied.

Other solutions can be constructed with vehicles passing through the vertex, shown in Figure 3.2c and 3.2d. In Figure 3.2c, the maximal number of vehicles that can be sent from the incoming edge are sent, but all vehicles are received by edge two, and none are received by edge three. In this case, the solution satisfying the LWR PDE on each edge is given by

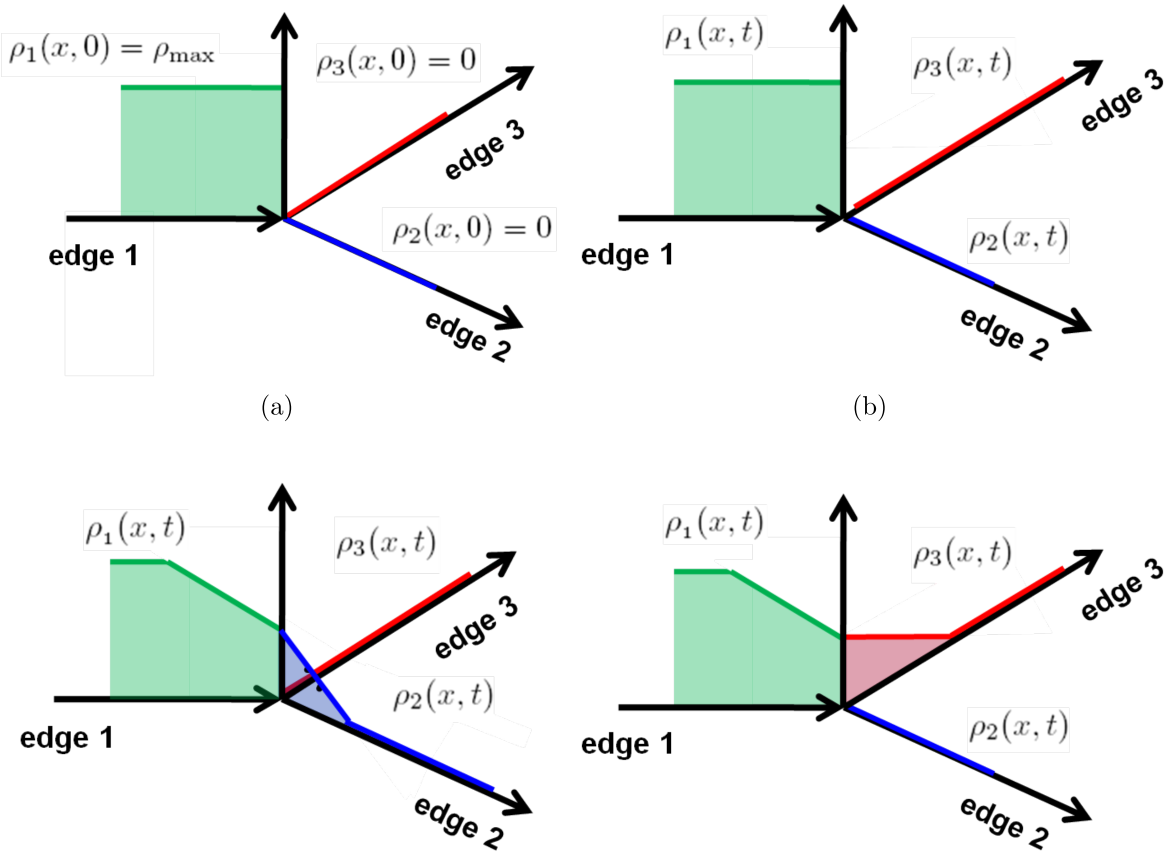
$$\rho_1(x,t) = \begin{cases} \rho_{\max} & \text{if } x < (Q')^{-1}(\rho_{\max})t, \\ (Q')^{-1}(\frac{x}{t}) & \text{otherwise} \end{cases}$$

$$\rho_2(x,t) = \begin{cases} (Q')^{-1}(\frac{x}{t}) & \text{if } x < (Q')^{-1}(0)t, \\ 0 & \text{otherwise} \end{cases}$$

$$\rho_3(x,t) = 0$$

Figure 3.2d shows the opposite scenario, where all vehicles are sent to edge three, and none are sent to edge two (the solutions of $\rho_2(x, t)$ and $\rho_3(x, t)$ are interchanged).

To resolve this nonuniqueness, several Riemann solvers in the density domain have been proposed in the literature. The first solver is due to Holden and Risebro [40], which maximizes a strictly convex function of the individual edge fluxes into and out of the junction, subject to mass conservation. Because the mass conservation constraints are linear, and the objective



(a) (b)

(c) (d)

Figure 3.2: Riemann problem for a diverge (a) initial condition; (b) a solution with no vehicles crossing the junction; (c) a solution with all vehicles received by edge 2; (d) a solution with all vehicles received by edge 3.

function is strictly convex, the solution to the Riemann problem is unique. In the context of the discretized LWR PDE (cell transmission model), Daganzo [21] provides a unique evolution of density where the total flux is maximized across the junction, subject to flow allocation parameters which encode the proportion of flow from an incoming edge associated to an outgoing edge. For diverge problems, a unique solution is constructed with the aid of additional priority parameters which specify preference for flows from upstream edges when a downstream link comes congested. Coclite, Garavello, and Piccoli [14] formalize this Riemann solver in the continuous domain and show existence of a global solution using wave front tracking [10]. A Riemann solver with internal state dynamics was also proposed by Labacque [52], and a multilane solver was introduced by Herty and Klar in [37] to better model traffic flow through intersections. The incremental transfer principle introduced by Daganzo [19] is a also Riemann solver which can be used to model exit and high occupancy vehicle lanes, yielding more realistic flow dynamics at intersections. The interested reader is referred to the recent book [29] for a detailed mathematical treatment of traffic flows on networks.

In what follows, we summarize the Riemann solver [14, 21], which will be the basis of our velocity model extension to networks.

We look for unique description of the evolution of the velocity dynamics at the junctions. Following the conditions for uniqueness of [29], we present three physically motivated restrictions on the dynamics, namely *(i)* conservation of vehicles across the junction, *(ii)* vehicles follow a set route across the junction, which define how the traffic flux from edges into the junction are routed to the outgoing edges *(iii)* traffic flow across the junction is maximized. Conditions *(i)* and *(ii)* imply that for the edge boundaries at the junction, boundary conditions must hold in the strong sense. This creates an upper bound on the flows on each edge into and out of the junction, which can be computed. By transforming these conditions into the velocity domain, the velocity evolution at the junctions can be determined by solving a linear programming problem.

Physical constraints

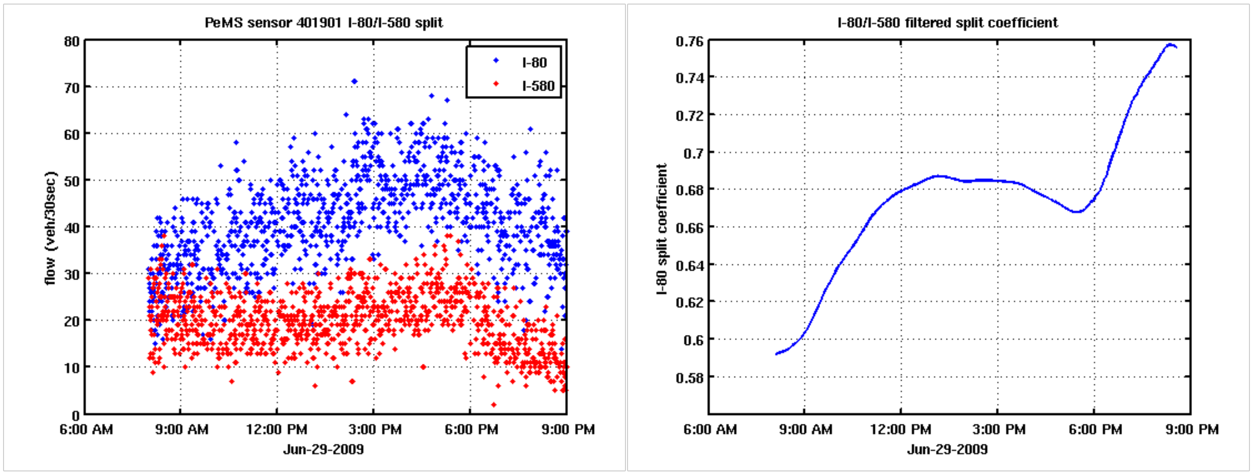
Consider a junction j with $|\mathcal{I}_j|$ incoming edges and $|\mathcal{O}_j|$ outgoing edges. First, we assume that the junction has no storage capacity, so all vehicles which enter the junction must also exit the junction. Conservation of the number of vehicles across the junction gives rise to the constraint that the total flux into the junction must equal the total flux out of the junction:

$$\sum_{e_{\text{in}} \in \mathcal{I}_j} \tilde{Q}_{e_{\text{in}}} \left(v_{e_{\text{in}}} \left(L_{e_{\text{in}}}, t \right) \right) = \sum_{e_{\text{out}} \in \mathcal{O}_j} \tilde{Q}_{e_{\text{out}}} \left(v_{e_{\text{out}}} \left(0, t \right) \right) \tag{3.21}$$

Next, we assume that the total volume of traffic entering from an incoming edge is distributed amongst the outgoing edges according to an allocation parameter $\alpha_{j,e_{\text{in}},e_{\text{out}}}(t) \geq 0$. The allocation matrix $A_j \in [0,1]^{|\mathcal{O}_j| \times |\mathcal{I}_j|}$, where $A_j(e_{\text{out}}, e_{\text{in}}) = \alpha_{j,e_{\text{out}},e_{\text{in}}}$, encodes the aggregate routing information of the traffic across the junction. That is, for all vehicles entering the junction j on edge e_{in} , $\alpha_{j,e_{\text{out}},e_{\text{in}}}$ denotes the proportion of vehicles which will exit the junction through edge e_{out}. This proportion can be determined empirically using historical origin-destination tables, or by analyzing the volumes of data collected near the junction (See Figure 3.3a). Because the vertex has no storage capacity, the sum of allocated flows from a fixed incoming link across all outgoing flows must be equal to one:

$$\sum_{e_{\text{out}} \in \mathcal{O}_j} \alpha_{e_{\text{out}},e_{\text{in}}} = 1 \tag{3.22}$$

Note that constraints *(i)* and *(ii)* combined imply $A_j \tilde{Q}_{e_{\text{in}}} = \tilde{Q}_{e_{\text{out}}}$. If we view the exiting flows from the incoming edges of the junction as a boundary condition for an out-



(a)

Figure 3.3: (a) Vehicle flows for the I80 – I580 diverge near Berkeley, California, obtained from the PeMS system; (b) filtered time-varying allocation parameter for flow to I80.

going edge, then the physical constraint $\sum_{e_{\text{in}}\in\mathcal{I}_j}\alpha_{e_{\text{out}},e_{\text{in}}}\tilde{Q}_{e_{\text{in}}}=\tilde{Q}_{e_{\text{out}}}$ for each e_{out} can be interpreted as a requirement that strong boundary conditions must be imposed on e_{out}. But strong boundary conditions (*i.e.* equality) cannot always be imposed for an arbitrary pair $\left(\sum_{e_{\text{in}}\in\mathcal{I}_j}\alpha_{e_{\text{out}},e_{\text{in}}}\tilde{Q}_{e_{\text{in}}},\ \tilde{Q}_{e_{\text{out}}}\right)$, so the statement of strong boundary conditions ((2.21) and (2.22) for a concave flux) provides upper bounds on the admissible incoming and admissible outgoing fluxes over which the flow is maximized (constraint (*iii*)). The maximum incoming admissible flux into the junction from edge e_{in} given a desired velocity $v_{e_{\text{in}}}$ to be prescribed in the strong sense is denoted by $\gamma^{\max}_{e_{\text{in}}}\left(v_{e_{\text{in}}}\right)$ (resp. $\delta^{\max}_{e_{\text{in}}}\left(\rho_{e_{\text{in}}}\right)$ for a given density). Similarly, the maximum outgoing admissible flux out of the junction from edge e_{out} given a desired velocity $v_{e_{\text{out}}}$ to be prescribed in the strong sense is denoted by $\gamma^{\max}_{e_{\text{out}}}\left(v_{e_{\text{out}}}\right)$ (resp. $\delta^{\max}_{e_{\text{in}}}\left(\rho_{e_{\text{in}}}\right)$ for a given density).

Thus the three conditions give rise to the following linear program for the fluxes (denoted by the vector dummy variable $\xi\in\mathbb{R}^{|\mathcal{I}|}$) on the incoming edges e_{in} for junction j:

$$\begin{array}{ll}\textbf{maximize} & 1^T\xi\\ \textbf{subject to} & A_j\xi\leq\gamma^{\max}_{\mathcal{O}_j}\\ & \mathbf{0}\leq\xi\leq\gamma^{\max}_{\mathcal{I}_j}\end{array}\tag{3.23}$$

where $\gamma^{\max}_{\mathcal{I}_j}:=\left(\gamma^{\max}_{e_{\text{in},1}},\cdots,\gamma^{\max}_{e_{\text{in},\left|\mathcal{I}_j\right|}}\right)$, $\gamma^{\max}_{\mathcal{O}_j}:=\left(\gamma^{\max}_{e_{\text{out},1}},\cdots,\gamma^{\max}_{e_{\text{out},\left|\mathcal{O}_j\right|}}\right)$ are the upper bounds on the fluxes on the edges entering and exiting the junction, to be computed subsequently. With the optimal solution to (3.23), denoted by $\xi^*_{e_{\text{in}}}$, the terms $\tilde{G}_{e_{\text{in}}}\left(v^n_{i_{\max}},v^n_{i_{\max}+1}\right)$ and $\tilde{G}_{e_{\text{out}}}\left(v^n_{-1},v^n_0\right)$ in the CTM-v (3.19) and (3.20) are given by:

$$\tilde{G}_{e_{\text{in}}}\left(v^n_{i_{\max}}, v^n_{i_{\max}+1}\right) = \xi^*_{e_{\text{in}}} \tag{3.24}$$

$$\tilde{G}_{e_{\text{out}}}\left(v^n_{-1}, v^n_{0}\right) = \sum_{e_{\text{in}}\in\mathcal{I}_j} \alpha_{e_{\text{out}},e_{\text{in}}}\xi^*_{e_{\text{in}}} \tag{3.25}$$

We note that the solution to this linear program is not always unique. In fact, for some instantiations of A_j, the gradient of the objective function may be normal to a facet of the constraint set polytope, in which case all feasible points on the facet will obtain the same objective value. This can be resolved with a technical condition on the coefficients A_j [14, 29], to explicitly prevent this nonuniqueness. However, when multiple links merge into a single link, additional priority constraints (detailed in [14, 21]) must be added to resolve the nonuniqueness, in which case the optimization problem becomes an integer program. Regardless, these optimization problems are small (typically only a few variables and less than 10 contstraints) and can be solved quickly, even by brute force.

Computation of the maximum admissible flux

First we introduce a function $\tau(\cdot)$, used to describe the domain for which we obtain admissible fluxes $Q(\cdot)$. For a continuous strictly concave C^0 flux function with $Q(0) = Q(\rho_{\max})$, the mapping from flux $Q(\rho)$ to ρ is double valued, with one value above and one value below the critical value ρ_c. For a given ρ, $\tau(\rho)$ is the map which produces the alternate ρ for the same flux. The function is expressed as follows:

$$Q(\tau(\rho)) = Q(\rho) \quad \forall\ \rho \in [0, \rho_{\max}]$$

$$\tau(\rho) \neq \rho \quad \forall \rho \in [0, \rho_{\max}]\backslash\{\rho_c\}$$

Given that $Q(\cdot)$ is in $C^0\left([0, \rho_{\max}]\right)$, strictly increasing in $[0, \rho_c)$ and strictly decreasing in $(\rho_c, \rho_{\max}]$ the following holds:

$$0 \leq \rho \leq \rho_c \Leftrightarrow \rho_c \leq \tau(\rho) \leq \rho_{\max}$$

We now define the upper bounds on the flux entering the junction from each incoming edge, and the flux leaving the junction on each outgoing edge. More precisely, for each incoming and outgoing link, we seek to find the upper bound on the admissible flux entering (resp. leaving) the link such that strong boundary conditions are imposed on the boundaries for all edges at the vertex. First we derive these admissible fluxes $\delta_{e_{\text{out}}}(\cdot)$ (resp. $\delta_{e_{\text{in}}}(\cdot)$) in terms of the trace of the density $\rho_{e_{\text{out}}}(0, t)$ (resp. $\rho_{e_{\text{in}}}(L, t)$), then apply the velocity inversion to arrive at admissible fluxes $\gamma_{e_{\text{out}}}(\cdot)$ (resp. $\gamma_{e_{\text{in}}}(\cdot)$) in terms of the trace of the velocity $v_{e_{\text{out}}}(0, t)$ (resp. $v_{e_{\text{in}}}(L, t)$).

For a strictly concave flux $Q(\cdot)$ with a maximum obtained at the critical value ρ_c we categorize the values of $\rho(0, \cdot)$ and $\rho_l(\cdot)$ for which (2.21) holds:

$$\text{for } a.e.\ t>0,\ \rho(0,t)=\rho_l(t) \text{ iff}$$
$$\begin{cases} & \rho(0,t)\in[0,\rho_c] \text{ and } \rho_l(t)\in[0,\rho_c] \\ \text{xor} & \rho(0,t)\in(\rho_c,\rho_{\max}] \text{ and } \rho_l(t)\in[0,\tau(\rho(0,t)))\cap\{\rho(0,t)\} \end{cases} \tag{3.26}$$

Recalling that incoming admissible fluxes are the set of fluxes corresponding to boundary data for the outgoing links which can be imposed in the strong sense, we can define the set of incoming admissible fluxes on an outgoing edge as:

- For $\rho_{e_{\text{out}}}(0,t)\in[0,\rho_{c,e_{\text{out}}}]$:

  $$\delta_{e_{\text{out}}}\left(\rho_{e_{\text{out}}}(0,t)\right)\in\Pi_{e\ \text{out}}\ \left(\rho_{e_{\text{out}}}(0,t)\right):=\left\{\hat{Q}:\ \exists\hat{\rho}\in[0,\rho_{c,e_{\text{out}}}]\,;\ \hat{Q}=Q\left(\hat{\rho}\right)\right\} \tag{3.27}$$

  where $\rho_{c,e_{\text{out}}}$ is the critical density on the edge e_{out}.

- For $\rho_{e_{\text{out}}}(0,t)\in[\rho_{c,e_{\text{out}}},\rho_{\max,e_{\text{out}}}]$:

  $$\delta_{e_{\text{out}}}\left(\rho_{e_{\text{out}}}(0,t)\right)\in\Pi_{e\ \text{out}}\ \left(\rho_{e_{\text{out}}}(0,t)\right):=$$
  $$\left\{\hat{Q}:\ \exists\hat{\rho}\in\{\rho_{e_{\text{out}}}(0,t)\}\cup[0,\tau\left(\rho_{e_{\text{out}}}(0,t)\right))\,;\hat{Q}=Q\left(\hat{\rho}\right)\ \right\} \tag{3.28}$$

Similarly, (2.22) can be rewritten in terms of outgoing admissible fluxes for incoming edges as:

- For $\rho_{e_{\text{in}}}(L_{e_{\text{in}}},t)\in[0,\rho_{c,e_{\text{in}}}]$:

  $$\delta_{e_{\text{in}}}\left(\rho_{e_{\text{in}}}(L_{e_{\text{in}}},t)\right)\in\Pi_{e_{\text{in}}}\left(\rho_{e_{\text{in}}}(L_{e_{\text{in}}},t)\right):=$$
  $$\left\{\hat{Q}:\ \exists\hat{\rho}\in\{\rho_{e_{\text{in}}}(L_{e_{\text{in}}},t)\}\cup(\tau(\rho_{e_{\text{in}}}(L_{e_{\text{in}}},t),\rho_{\max,e_{\text{in}}}]\,;\hat{Q}=Q\left(\hat{\rho}\right)\right\} \tag{3.29}$$

  where $\rho_{\max,e_{\text{in}}}$ is the maximum density on the edge e_{in}.

- For $\rho_{e_{\text{in}}}(L_{e_{\text{in}}},t)\in[\rho_{c,e_{\text{in}}},\rho_{\max,e_{\text{in}}}]$:

  $$\delta_{e_{\text{in}}}\left(\rho_{e_{\text{in}}}(L_{e_{\text{in}}},t)\right)\in\Pi_{e_{\text{in}}}\left(\rho_{e_{\text{in}}}(L_{e_{\text{in}}},t)\right):=$$
  $$\left\{\hat{Q}:\ \exists\hat{\rho}\in[\rho_{c,e_{\text{in}}},\rho_{\max,e_{\text{in}}}]\,;\ \hat{Q}=Q(\hat{\rho})\right\} \tag{3.30}$$

If the admissible flux is maximized, and written in terms of velocity, we obtain:

$$\gamma_{e_{\text{out}}}^{\max}\left(v_{e_{\text{out}}}(0,t)\right)=\begin{cases}\tilde{Q}(v_{c,e_{\text{out}}}) & \text{if } v_{e_{\text{out}}}(0,t)\in[v_{c,e_{\text{out}}},v_{\max,e_{\text{out}}}]\\ \tilde{Q}\left(v_{e_{\text{out}}}(0,t)\right) & \text{if } v_{e_{\text{out}}}(0,t)\in[0,v_{c,e_{\text{out}}}]\end{cases}$$

and

$$\gamma_{e_{\text{in}}}^{\max}\left(v_{e_{\text{in}}}(L_{e_{\text{in}}},t)\right) = \begin{cases} \tilde{Q}\left(v_{e_{\text{in}}}(L_{e_{\text{in}}},t)\right) & \text{if } v_{e_{\text{in}}}(L_{e_{\text{in}}},t) \in [v_{c,e_{\text{in}}}, v_{\max,e_{\text{in}}}] \\ \tilde{Q}\left(v_{c,e_{\text{in}}}\right) & \text{if } v_{e_{\text{in}}}(L_{e_{\text{in}}},t) \in [0, v_{c,e_{\text{in}}}] \end{cases}$$

which are the upper bounds used in (3.23).

Example 10 (Maximum admissible flux - Smulders model)**.** The maximum outgoing admissible flux is given as:

$$\gamma_{e_{\text{out}}}^{\max}\left(v_{e_{\text{out}}}(0,t)\right) = \begin{cases} \rho_{\max}\left(1-\frac{v_{c,e_{\text{out}}}}{v_{\max}}\right) v_{c,e_{\text{out}}} \\ \qquad \text{if } v_{e_{\text{out}}}(0,t) \in [v_{c,e_{\text{out}}}, v_{\max,e_{\text{out}}}] \\ \rho_{\max}\left(\frac{1}{1+\frac{v_{e_{\text{out}}}(0,t)}{w_f}}\right) v_{e_{\text{out}}}(0,t) \\ \qquad \text{if } v_{e_{\text{out}}}(0,t) \in [0, v_{c,e_{\text{out}}}] \end{cases} \tag{3.31}$$

and the maximum incoming admissible flux is given as:

$$\gamma_{e_{\text{in}}}^{\max}\left(v_{e_{\text{in}}}(L_{e_{\text{in}}},t)\right) = \begin{cases} \rho_{\max}\left(1-\frac{v_{e_{\text{in}}}(L_{e_{\text{in}}},t)}{v_{\max}}\right) v_{e_{\text{in}}}(L_{e_{\text{in}}},t) \\ \qquad \text{if } v_{e_{\text{in}}}(L_{e_{\text{in}}},t) \in [v_{c,e_{\text{in}}}, v_{\max,e_{\text{in}}}] \\ \rho_{\max}\left(\frac{1}{1+\frac{v_{c,e_{\text{in}}}}{w_f}}\right) v_{c,e_{\text{in}}} \\ \qquad \text{if } v_{e_{\text{in}}}(L_{e_{\text{in}}},t) \in [0, v_{c,e_{\text{in}}}] \end{cases} \tag{3.32}$$

3.5.2 Discrete CTM-v network algorithm

The *CTM-v network algorithm* is obtained by sequentially applying the CTM-v scheme on each link of the network and solving the junction conditions as presented in the previous section, which includes solving the LP (3.23) posed earlier. The algorithm as illustrated in Figure 3.4.

The network is thus marched in time and consists in a large scale discrete dynamical system which can be used for data assimilation and inverse modeling. Given the velocity field at each discrete point $i \in \{0, \cdots, i_{\max}\}$ on all edges of the network

$$v^n := \left[v_{0,e_0}^n, \cdots, v_{i_{\max},e_0}^n, \cdots, v_{0,e_{|\mathcal{E}|}}^n, \cdots, v_{i_{\max},e_{|\mathcal{E}|}}^n\right]$$

the velocity at time $t_{n+1}\Delta T$ is given by:

$$v^{n+1} = \mathcal{M}(v^n, \theta^n) \tag{3.33}$$

where θ^n represents the parameters of the velocity function on each link $(v_{\max}, v_c, w_f)$, flow allocation and priority parameters at the junctions, and boundary data at the network boundaries, and $\mathcal{M}(\cdot,\cdot)$ denotes the following update algorithm:

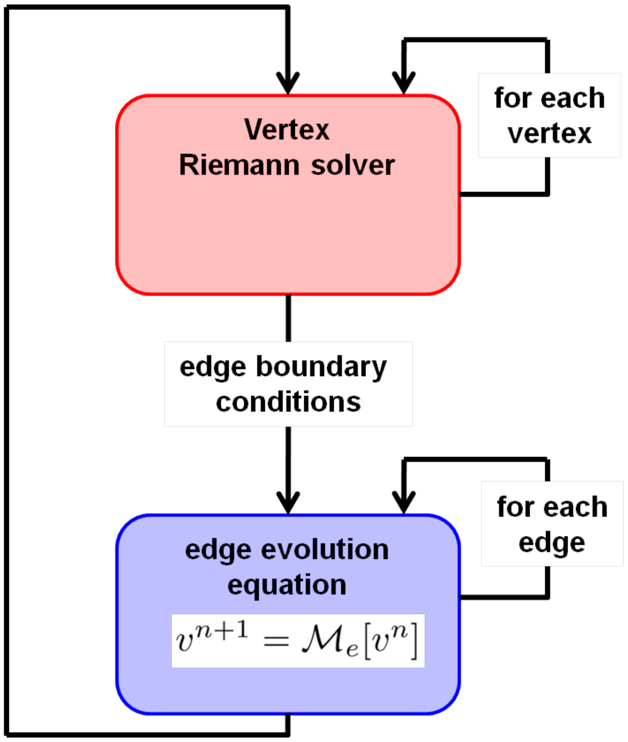


Figure 3.4: The discrete network velocity evolution equation proceeds into steps. First, the Riemann problem at each vertex is solved to determine the strong boundary conditions for each edge. Then, for each edge the velocity is evolved according to (3.17).

1. For all junctions $j \in \mathcal{J}$:
    - (a) Compute $\gamma^n_{i_{\max},e_{\text{in}}}\left(v^n_{i_{\max},e_{\text{in}}}\right)$ $\forall e_{\text{in}} \in \mathcal{I}_j$, and $\gamma^n_{0,e_{\text{out}}}\left(v^n_{0,e_{\text{out}}}\right)$ $\forall e_{\text{out}} \in \mathcal{O}_j$ using (3.31) and (3.32).
    - (b) Solve the LP (3.23) for ξ^*, and update $\tilde{G}_{e_{\text{in}}}\left(v^n_{i_{\max}}, v^n_{i_{\max}+1}\right)$ and $\tilde{G}_{e_{\text{out}}}\left(v^n_{-1}, v^n_0\right)$ through (3.24) and (3.25).
2. For all edges $e \in \mathcal{E}$: Compute $v^{n+1}_{i,e}$ $\forall i \in \{1, \cdots, i_{\max,e}\}$ according to the CTM-v (3.17), (3.19), and (3.20).

Chapter 4

Velocity estimation

4.1 Introduction

The goal of this chapter is to build an estimator to reconstruct the evolution of the velocity field on the highway, given the velocity measurements from GPS devices such as mobile phones, and test its performance on experimental data obtained from a large one-day field experiment known as *Mobile Century*. The new contributions in this chapter are as follows.

- **State space formulation**. We pose the velocity estimation problem in state space form. The resulting system has a nonlinear and nondifferentiable evolution equation, but contains a linear observation equation. This is an improvement over a nonlinear nondifferentiable evolution equation with a nonlinear observation equation when the discretized LWR PDE is used directly as an evolution equation.
- **Solution with ensemble Kalman filtering**. The resulting state estimation problem is solved using ensemble Kalman filtering, representing the first application of the technique for traffic monitoring.
- **Assessment on experimental data**. We assess the performance of the estimation algorithm on experimental data collected from the *Mobile Century* experiment. A prototype version of the velocity estimation algorithm ran live during the experiment, broadcasting results in real–time to monitors of the experiment.

The chapter is organized as follows. In Section 4.2 we pose the estimation problem in state space form, we describe the ensemble Kalman filtering technique and we compare it to extended Kalman filtering. We describe a mechanism for sampling GPS data from mobile phones in a privacy aware environment using virtual trip lines in Section 4.3, and describe a 100 vehicle field experiment known as *Mobile Century*. Finally, we conclude the chapter with experimental velocity estimation results from the *Mobile Century* experiment in Section 4.4.

4.2 Development of a recursive velocity estimation algorithm

4.2.1 State–space model

Before we begin with the estimation problem using the CTM–v model derived in Chapter 3, let us consider a traffic estimation problem in a more general form, in order to characterize the errors in the CTM–v model. We start by introducing a true state vector $\tilde{z}^n$, with dimension two times the number of vehicles in the transportation network at time n. In this vector, half of the elements correspond to the true positions of the individual vehicles, while the other half of the elements correspond to the velocities of the vehicles. This vector evolves according to

$$\tilde{z}^{n+1} = \tilde{\mathcal{M}}\left(\tilde{z}^n, \tilde{\theta}^n, \tilde{\eta}^n\right) \tag{4.1}$$

where $\tilde{\mathcal{M}}(\cdot,\cdot,\cdot)$ represents a function which maps the true previous position and velocity of all vehicles $\tilde{z}^n$ to their true position and velocity at the next timestep $\tilde{z}^{n+1}$, with the help of parameters such as driver behavior, vehicle performance characteristics, road geometry, etc. represented by $\tilde{\theta}^n$, and some stochastic input given by $\tilde{\eta}^n$. If the model of the true evolution $\tilde{\mathcal{M}}(\cdot,\cdot,\cdot)$, the parameters $\tilde{\theta}^n$, and the stochastic input $\tilde{\eta}^n$ were known, we could completely characterize the traffic evolution throughout the network for all time by evolving (4.1) forward. Unfortunately, each of the preceding components are unknown in practice. Thus, $\tilde{\mathcal{M}}(\cdot,\cdot,\cdot)$ is an abstraction which represents the true, error-free model of traffic evolution, but which we cannot instantiate due to its unknown form and inputs.

The true state vector representing all vehicle positions and velocities is related to the average traffic velocity vector in each discrete segment on the network v^n, which is the vector we are interested in estimating, by:

$$v^{n+1} = P(\tilde{z}^{n+1})$$

where $P(\cdot)$ is an averaging operator which computes the average traffic velocity in each discrete segment from the individual vehicles' velocities in each discrete segment. Note then that the average velocity can be computed according to

$$v^{n+1} = P\left(\tilde{\mathcal{M}}\left(\tilde{z}^n, \tilde{\theta}^n, \tilde{\eta}^n\right)\right)$$

Unfortunately, since $\tilde{\mathcal{M}}(\cdot,\cdot,\cdot)$ and its inputs are unknown, we need another evolution equation which is known to approximate this model. We will use the network velocity evolution algorithm $\mathcal{M}(\cdot,\cdot)$, given in Section 3.5.2. This algorithm consists of the following steps. For each vertex in the network, a linear program is solved such that strong boundary conditions are imposed on the incoming and outgoing edges of the junction. Next, the velocity field is updated according to the numerical scheme outlined earlier (which is nonlinear and non-differentiable). Then our approximate model is derived as follows:

$$\begin{aligned} v^{n+1} &= P\left(\tilde{\mathcal{M}}\left(\tilde{z}^n, \tilde{\theta}^n, \tilde{\eta}^n\right)\right) \\ &= \left[P\left(\tilde{\mathcal{M}}\left(\tilde{z}^n, \tilde{\theta}^n, \tilde{\eta}^n\right)\right) - \mathcal{M}\left(P(\tilde{z}^n), \theta^n\right)\right] + \mathcal{M}\left(P(\tilde{z}^n), \theta^n\right) \\ &= \mathcal{M}\left(P(\tilde{z}^n), \theta^n\right) + \hat{\eta}^n \\ &= \mathcal{M}\left(v^n, \theta^n\right) + \hat{\eta}^n \end{aligned} \tag{4.2}$$

The term θ^n represents the model parameters, and $\hat{\eta}^n$ represents the error due to the use of the approximate model in place of the unknown true model. Interestingly, $\mathcal{M}(\cdot, \cdot)$ is derived from a conservation law, and the principle of conservation of vehicles has no error. Instead, $\hat{\eta}^n$ appears through (i) the fundamental assumption of the LWR PDE, which is that velocity can be described as a function of density only, (ii) the nonuniqueness of solutions to conservation laws, and the choice of an entropy condition to isolate a unique solution, and (iii) the nonuniqueness of generalized Riemann solvers at junctions. It is not caused by uncertain boundary data or model parameters, which we treat next.

In practice, boundary data and model parameters θ^n are uncertain. If we choose to replace the true θ^n with an estimate $\bar{\theta}^n$, then (4.2) is modified by:

$$\begin{aligned} v^{n+1} &= \mathcal{M}\left(v^n, \theta^n\right) + \hat{\eta}^n \\ &= \left[\mathcal{M}\left(v^n, \theta^n\right) + \hat{\eta}^n - \mathcal{M}\left(v^n, \bar{\theta}^n\right)\right] + \mathcal{M}\left(v^n, \bar{\theta}^n\right) \\ &= \mathcal{M}\left(v^n, \bar{\theta}^n\right) + \eta^n \end{aligned} \tag{4.3}$$

Now the term η^n in (4.3) represents both the errors in the approximate model, and the errors caused by the incorrect model parameters $\bar{\theta}$. Unfortunately the preceding derivation of (4.3) shows that the true model $\mathcal{M}(\cdot, \cdot, \cdot)$ is needed to compute the statistics of these errors. This problem can be addressed in part through the aid of a more accurate but computationally more intensive model, see for example [44, 45], or by estimating the errors, for example by $\eta^n \sim (0, \mathbf{Q}^n)$, a zero–mean, white state noise with covariance $\mathbf{Q}^n$. The latter is the approach we will use for estimation of velocity fields from mobile phone data. The zero mean assumption and white noise assumption are introduced to simplify the presentation of various Kalman filtering algorithms, and can be relaxed with suitable adjustments made to the Kalman filtering algorithm and state space formulation [3, 26, 66].

We address the process by which velocity measurements are obtained from GPS equipped vehicles similarly. Let $\tilde{\mathbf{H}}^n$ be a linear operator which maps the velocity of vehicles which send measurements to the GPS velocity value measured by the vehicle at time n, and let y^n denote the GPS measurements which are received at time n. The network observation model is given by

$$y^n = \tilde{\mathbf{H}}^n \tilde{z}^n + \tilde{\chi}^n \tag{4.4}$$

where $\tilde{\chi}^n$ is the measurement error of the GPS device in each vehicle.

Several comments can be made about (4.4). First, note that $\tilde{\mathbf{H}}^n$ is in fact a linear operator, since it simply indicates the subset of vehicles from which measurements are obtained. Second, in order for $\tilde{\mathbf{H}}^n$ to be defined, vehicles must report a unique ID along with the velocity measurement, so that each measurement may be correctly mapped to the correct vehicle. If the identifiers are withheld from the measurements, for example for privacy reasons, then $\tilde{\mathbf{H}}^n$ is unknown and a data association problem must be solved to determine $\tilde{\mathbf{H}}^n$. The use of GPS position information may help solve the data association problem but does not replace a known $\tilde{\mathbf{H}}^n$. Finally, note that the GPS error $\tilde{\chi}^n$ may be correlated in time and across vehicles [59].

Because we are working with an aggregate state v^n, we derive an equivalent form of (4.4) as follows:

$$\begin{aligned} y^n &= \left[\tilde{\mathbf{H}}^n \tilde{z}^n + \tilde{\chi}^n - \mathbf{H}^n P(\tilde{z}^n)\right] + \mathbf{H}^n P(\tilde{z}^n) \\ &= \mathbf{H}^n P(\tilde{z}^n) + \chi^n \\ &= \mathbf{H} v^n + \chi^n \end{aligned} \tag{4.5}$$

The linear observation matrix $\mathbf{H}^n \in \{0,1\}^{p^n \times \kappa}$ encodes the p^n discrete cells on the highway for which the velocity is observed during discrete time step n and $\kappa = \sum_{e \in \mathcal{E}} (i_{\max,e} + 1)$ is the corresponding (total) number of cells in the network. The term χ^n now includes both the GPS error and the sampling error introduced when the sample vehicle's velocity is different from the average velocity of all vehicles in the discrete road segment from which the measurement is obtained. Note also that determination of the error statistics χ^n requires knowledge of the true state $\tilde{z}^n$, although approximate statistics could be computed through the use of microscopic simulation models and enhanced error modeling techniques [44, 45]. In this work, we approximate the statistics with a white, zero-mean observation noise $\chi^n \sim (0, \mathbf{R}^n)$, although again these assumptions can also be relaxed [3].

Interestingly, the term $\mathbf{H}^n$ itself may contain error, even in the macroscopic setting. This is because the GPS position of the vehicle is used to determine the location of the measurement, and therefore it determines the corresponding state vector associated to the measurement (which is stored in $\mathbf{H}^n$). The amount of error in $\mathbf{H}^n$ is determined by the amount of error in the GPS position, the size of the discretized road segments, and the proximity of the measurement to be discretized road segment boundaries. For example, if a measurement is received exactly at the boundary between to discrete road segments, then it is not clear to which element of the state vector the measurement should be mapped. This difficulty is circumvented in our application through the use of a spatial sampling technique known as a virtual trip line [38], which is a virtual marker in the form of a line segment encoded by two latitudes longitude coordinates, which triggers a measurement from a mobile phone when it is crossed. By careful placement of the virtual trip lines, and by employing virtual trip line specific filters, the amount of error in $\mathbf{H}^n$ can be made negligible, at the cost of perhaps fewer measurements. We describe virtual trip lines in more detail in Section 4.3.

We briefly describe an alternate formulation for estimating velocities, using the discretized LWR PDE directly, to compare against (4.3) and (4.5). Using a similar approach to the velocity derivation, the density ρ^n evolves according to

$$\rho^{n+1} = \mathcal{M}_\rho\left(\rho^n, \bar{\theta}^n_\rho\right) + \eta^n_\rho \tag{4.6}$$

$$y^n = \mathbf{H}^n V(\rho^n) + \chi^n_\rho \tag{4.7}$$

where $\mathcal{M}_\rho(\cdot,\cdot)$ is the discretized LWR PDE, $\bar{\theta}^n_\rho$ are the model parameters and boundary data, and η^n_ρ is the error in the conservation law associated with the approximation of velocity as a function of density $v = V(\rho)$ only, the nonuniqueness of entropy solutions to the LWR PDE, and the nonuniqueness of the Riemann problems at junctions. The term y^n is again the vector of GPS velocity measurements, $\mathbf{H}^n$ is the same linear operator which maps the measurements to the corresponding elements in the state vector, and χ^n_ρ is the error associated with the GPS error, the sampling error, and errors in the velocity function approximation $v = V(\rho)$.

When comparing the velocity formulation (4.3) and (4.5) to the density formulation (4.6) and (4.7), several observations can be made.

- Both the evolution equation for velocity (4.3) and the evolution equation for density (4.6) are in general nonlinear and nondifferentiable, due to the min operator in the Godunov discretization schemes (2.32) and (3.16), and in particular the existence of standing shockwaves as solutions to the LWR PDE (see Section 2.5.2 for the proof that the model is not differentiable the around this state). This comes in addition to the potential nondifferentiability of the flux function itself, for example like in the Newell–Daganzo flux function. Moreover, the generalized Riemann solver at junctions takes the form of an optimization problem (often a linear program), which also is not differentiable in general.

- Both models make the same fundamental assumption that the velocity can be represented as a function of density only. The velocity evolution equation places a further restriction on the velocity function, requiring that the velocity function the invertible. Note, however, when the velocity function is not invertible, the observation equation (4.7) makes estimating the density from velocity measurements more ill posed.

- By construction, both models use the same entropy solution and the same Riemann solver at junctions.

- While the observation model for the velocity state (4.5) is linear, the observation model for the density state (4.7) is linear only when the velocity function is linear (i.e. Greenshields). For nonlinear velocity functions, the observation equation would have to be linearized to fit a standard Kalman filtering framework.

- The observation model for the density state (4.7) relies on the velocity function $V(\cdot)$, as does the density evolution equation (4.6). Thus, in this formulation the model error η_ρ^n in the observation error χ_ρ^n are correlated. This is not the case in the velocity formulation, where the errors η^n and χ^n are independent.

In the remainder of this chapter, we elect to use the velocity evolution equation (4.3) and observation equation (4.5) due to the linearity of (4.5). As a possible extension of this work, it would be interesting to compare how the different formulations perform in practice.

4.2.2 Extended Kalman filtering for nonlinear systems

If equation (3.17) was differentiable in v^n, so would be the operator $\mathcal{M}(\cdot,\cdot)$ in (4.3), in which case the estimate for the state v^n could be obtained using the following traditional extended Kalman filtering equations:

- Forecast step (Time-update):

$$\begin{aligned} v_f^n &= \mathcal{M}\left(v_a^{n-1}, \bar{\theta}^{n-1}\right) \\ \mathbf{P}_f^n &= \mathcal{M}_L^{n-1}\mathbf{P}_a^{n-1}\left(\mathcal{M}_L^{n-1}\right)^T + \mathbf{Q}^{n-1} \end{aligned} \tag{4.8}$$

  where v_f^n (resp. v_a^n) is the forecast (analyzed) state estimate at time n, and $\mathcal{M}_L$ is the Jacobian matrix of mapping $\mathcal{M}$ (also known as the *tangent linear model*) defined as

$$\mathcal{M}_L^n = \frac{\partial \mathcal{M}(v_a^n, \bar{\theta}^n)}{\partial v_a^n} \tag{4.9}$$

- Analysis step (Measurement-update):

$$v_a^n = v_f^n + \mathbf{G}^n\left(y^n - \mathbf{H}^n v_f^n\right) \tag{4.10}$$

$$\mathbf{P}_a^n = \mathbf{P}_f^n - \mathbf{G}^n\mathbf{H}^n\mathbf{P}_f^n \tag{4.11}$$

$$\mathbf{G}^n = \mathbf{P}_f^n\left(\mathbf{H}^n\right)^T\left(\mathbf{H}^n\mathbf{P}_f^n\left(\mathbf{H}^n\right)^T + \mathbf{R}^n\right)^{-1} \tag{4.12}$$

  where $\mathbf{P}_f^n$ (resp. $\mathbf{P}_a^n$) is the error covariance of the forecast (analyzed) state at time n.

The initial conditions for the recursion are given by $v_a^0 = v^0$ and $\mathbf{P}_a^0 = \mathbf{P}^0$.

4.2.3 Ensemble Kalman filter

The ensemble Kalman filter was introduced by Evensen in [26] as an alternative to EKF to overcome specific difficulties with nonlinear state evolution models, including non-differentiability of the model and closure problems. Closure problems refer to the fact that

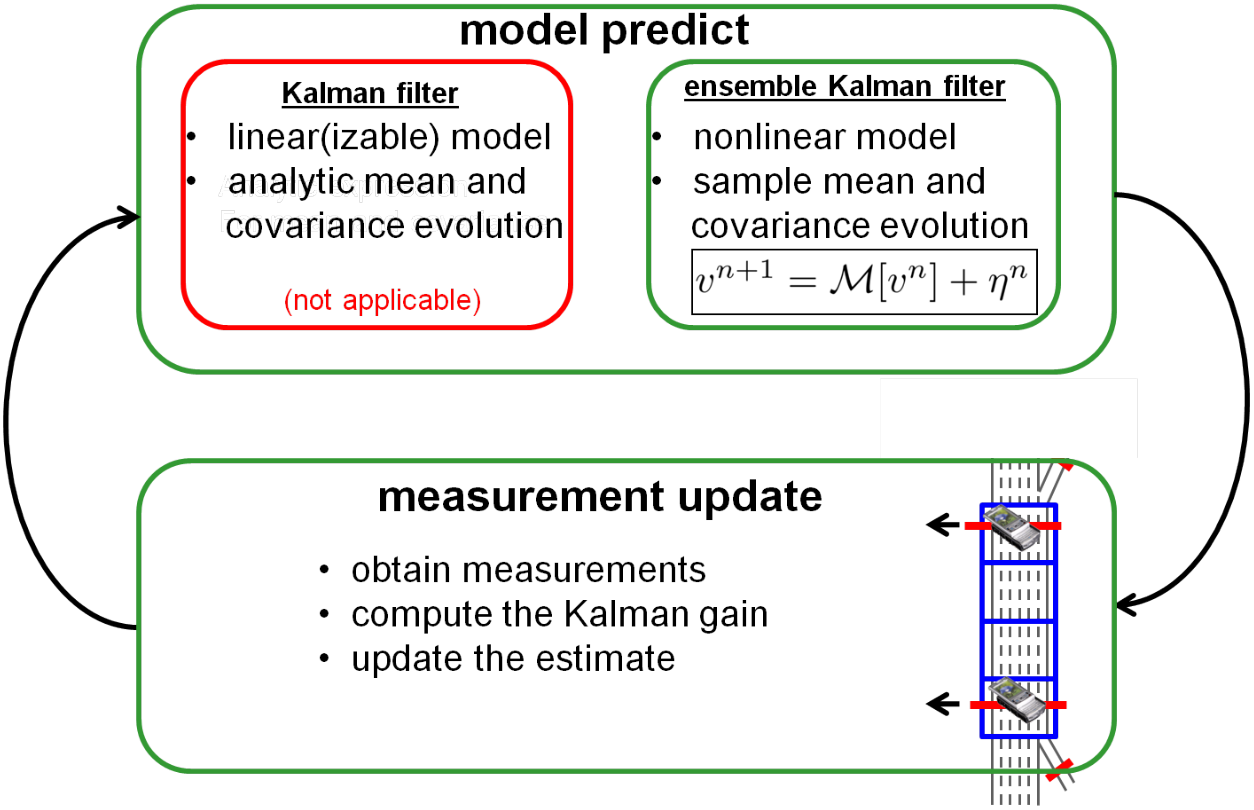


Figure 4.1: Illustration of the difference between extended Kalman filtering and ensemble Kalman filtering, and the iterative process predict–update.

in EKF, it is assumed that discarding the higher order moments from the evolution of the error covariance in (4.8) yields a good approximation. In cases in which this linearization approximation is invalid, it can cause an unbounded error variance growth [26]. To tackle this issue EnKF uses Monte Carlo (or ensemble integrations). By propagating the ensemble of model states forward in time, it is possible to calculate the mean and the covariances of the error needed at the analysis (measurement-update) step [11] and avoid the closure problem. Furthermore, a strength of EnKF is that it uses the standard update equations of EKF, except that the gain is computed from the error covariances provided by the ensemble of model states. Figure 4.1 illustrates the difference.

EnKF also comes with a relatively low computational cost. Namely, usually a rather limited number of ensemble members is needed to achieve a reasonable statistical convergence [11].

In traditional Kalman filtering, the error covariance matrices are defined in terms of the true state as $\mathbf{P}_f = E[(v_f - v_t)(v_f - v_t)^T]$ and $\mathbf{P}_a = E[(v_a - v_t)(v_a - v_t)^T]$ where $E[\cdot]$ denotes the average over the ensemble, v is the model state vector at particular time, and the subscripts f, a, and t represent the forecast, analyzed, and true state, respectively. Because the true state is not known, ensemble covariances for EnKF have to be considered. These covariance matrices are evaluated around the ensemble mean $\bar{v}$, yielding $\mathbf{P}_f \approx \mathbf{P}_{\text{ens},f} = E[(v_f - \bar{v}_f)(v_f - \bar{v}_f))^T]$ and $\mathbf{P}_a \approx \mathbf{P}_{\text{ens},a} = E[(v_a - \bar{v}_a)(v_a - \bar{v}_a)^T]$ where the subscript ens refers to the ensemble approximation. In [11], it is shown that if the ensemble mean is used as the best estimate, the ensemble covariance can consistently be interpreted as the error covariance of the best estimate. For complete details of derivation of the EnKF algorithm, the reader is referred to [26].

The ensemble Kalman filter algorithm can be summarized as follows [11, 26]:

1. *Initialization*: Draw K ensemble realizations $v_a^0(k)$ (with $k \in \{1, \cdots, K\}$) from a process with a mean speed $\bar{v}_a^0$ and covariance $\mathbf{P}_a^0$.

2. *Forecast*: Update each of the K ensemble members according to the CTM-v forward simulation algorithm in Section 3.5.2. Then update the ensemble mean and covariance according to:

$$v_f^n(k) = \mathcal{M}\left(v_a^{n-1}(k), \bar{\theta}^{n-1}\right) + \eta^n(k) \tag{4.13}$$

$$\bar{v}_f^n = \frac{1}{K}\sum_{k=1}^{K} v_f^n(k) \tag{4.14}$$

$$\mathbf{P}_{\text{ens},f}^n = \frac{1}{K-1}\sum_{k=1}^{K}\left(v_f^n(k) - \bar{v}_f^n\right)\left(v_f^n(k) - \bar{v}_f^n\right)^T \tag{4.15}$$

3. *Analysis*: Obtain measurements, compute the Kalman gain, and update the network forecast:

$$\mathbf{G}_{\text{ens}}^n = \mathbf{P}_{\text{ens},f}^n\left(\mathbf{H}^n\right)^T\left(\mathbf{H}^n\mathbf{P}_{\text{ens},f}^n\left(\mathbf{H}^n\right)^T + \mathbf{R}^n\right)^{-1} \tag{4.16}$$

$$v_a^n(k) = v_f^n(k) + \mathbf{G}_{\text{ens}}^n\left(y^n(k) - \mathbf{H}^n v_f^n(k) + \chi^n(k)\right) \tag{4.17}$$

4. Return to 2.

In (4.17), an important step is that at measurement times, the measurement vector y^n is represented by an ensemble indexed by k. This ensemble has the actual measurement as the mean and the variance of the ensemble is used to represent the measurement errors. This is done by adding perturbations $\chi^n(k)$ to the measurements drawn from a distribution with zero mean and covariance equal to the measurement error covariance matrix $\mathbf{R}^n$. This ensures that the updated ensemble has the correct analyzed covariance [11].

4.2.4 Large scale real–time implementation

The ensemble Kalman filter algorithm presented in the previous section is in a framework in which all of the unknown state variables on each edge in the network are updated simultaneously. This introduces the following problems. First, because the state covariance is represented through a limited number of ensemble members, non-physical correlations may arise. This means that the correlation matrix may incorrectly show correlation between distant parts of the highway network which do not correlate in practice. Secondly, the framework described previously requires the forecast error covariance in (4.15) to be computed for the entire highway network, for use in computing the Kalman gain in (4.16). When operating on large scale networks such as the San Francisco Bay Area, CA, the loading the covariance matrix into memory can easily require more than 2 GB of space, creating computational limitations for implementation.

To circumvent the above mentioned problems for practical implementations, we employ a *covariance localization method*. This approach limits the correlation between the velocity states on all edges in the network. For a given edge e, only nearby links (upstream and downstream in the network) can exhibit correlation, thereby removing correlation across distant parts of the network. These techniques have also been implemented for oceanography data assimilation problems (see e.g. [60]).

For the large scale traffic network estimation problem, localization also provides a computationally efficient way to update the state variables at the measurement update time in (4.16)–(4.17). Namely, due to the localization, the computation of the covariance matrix in (4.15) is transformed into a computation of numerous small localized covariance matrices for each edge in the network. These small scale covariance matrices are computed for each edge given its neighboring edges on which the correlation is assumed to be physically meaningful. Finally, this allows for the distributed solving of the update equations.

For the localization, we introduce a localization operator $\mathcal{L}_e$ for each edge e, which is constructed at the initialization stage. This operator indicates which velocity states on the other edges of the network are allowed to have correlation with the velocity state on the eth edge. The implementation of the EnKF algorithm described previously can be modified for localization by replacing the measurement update equations (4.15)-(4.17) with the following sub-algorithm:

For each edge $e \in \mathcal{E}$:

1. Using the localization operator $\mathcal{L}_e$, compute the localized forecast error covariance:

$$\mathbf{P}^n_{\text{ens},f,e} = \frac{1}{K-1} \sum_{k=1}^{K} \mathcal{L}_e \left(v^n_f(k) - \bar{v}^n_f \right) \times \left(\mathcal{L}_e (v^n_f(k) - \bar{v}^n_f) \right)^T \tag{4.18}$$

2. *Analysis*: Obtain measurements $y^n_{\text{meas},e}$ from edges that are indicated in $\mathcal{L}_e$, compute the Kalman gain, and update the the local forecast:

$$\mathbf{G}^n_{\text{ens},e} = \mathbf{P}^n_{\text{ens},f,e} \left(\mathbf{H}^n_e \right)^T \times \left(\mathbf{H}^n_e \mathbf{P}^n_{\text{ens},f,e} \left(\mathbf{H}^n_e \right)^T + \mathbf{R}^n_e \right)^{-1} \tag{4.19}$$

$$v^n_{a,e}(k) = \mathcal{L}_e \left(v^n_f(k) \right) + \mathbf{G}^n_{\text{ens},e} \left(y^n_{\text{meas},e} - \mathbf{H}^n_e v^n_f(k) + \chi^n_e(k) \right) \tag{4.20}$$

3. Return to 1.

It is worth noting that in practice, the operator $\mathcal{L}_e$ does not need to be constructed as a matrix in the computer memory and subsequently be used to do the relatively demanding matrix multiplications. In other words, the e^{th} edge has references to the forecasts and measurements of its neighboring edges needed to construct the localized forecast error covariance matrix.

A second performance optimization is achieved by avoiding construction of the covariance matrices directly. When the number of ensemble members is small with respect to the total state space, the covariance matrix $\mathbf{P}^n_{\text{ens},f}$ is low rank, and therefore significant computational savings can be achieved by working with a Cholesky decomposition of the covariance matrix. Algorithmic optimizations to the ensemble Kalman filter are explained in detail in the tutorial article [25], complete with pseudocode. For implementation in the *Mobile Millennium* system at UC Berkeley [83], we follow the implementation optimizations of [25].

4.3 Experimental setup

In this section, we describe how GPS velocity data can be obtained from mobile phones, using a technique known as virtual trip lines. This technique is implemented on a one-day field experiment, known as *Mobile Century*, which provides experimental data to assess the velocity estimation algorithm using ensemble Kalman filtering and the discrete velocity evolution equation.

4.3.1 The *Mobile Century* field experiment (February 8, 2008)

Nicknamed the *Mobile Century* experiment, a prototype privacy-aware data collection system was launched on February 8, 2008 and used to estimate traffic conditions for a day on I-880 near Hayward, CA (see Figure 4.2) With the help of 165 UC Berkeley students, 100 vehicles carrying Nokia N95 phones drove repeated loops of 6 to 10 miles in length continuously for 8 hours. This section of highway was selected specifically for its complex traffic properties, which include alternating periods of free-flowing, uncongested traffic, and slower moving traffic during periods of heavy congestion. These vehicles represented approximately 2% to 5% of the total volume of traffic on the main line of the highway during the experiment.

Figure 4.2: *Mobile Century* experiment site in the San Francisco Bay Area. Vehicles drove a subset of an 11.4 mile stretch of I880.

During the experiment, the 100 vehicles were divided into three groups, and each group covered a different subset of the stretch of freeway for experimental reasons. For example, one third of the vehicles drove north starting at Stevenson Blvd. to W. Tennyson Rd., before exiting, turning around, and driving south from W. Tennyson Rd. to Stevenson Blvd. A second group of vehicles drove in loops covering the freeway between Mowry Ave. and CA 92 / San Mateo Br., while the third group covered the stretch between Thorton Ave. and Winton Ave. In the afternoon, the three groups drove a shorter stretch of roadway to maintain a penetration rate between 2-5% as the traffic volume increased. When the experiment was concluded, it was identified that 77 of the cell phones running the experimental software were able to properly record the probe vehicles' positions and velocities. In addition to sending virtual trip line measurements, a local log on each device stored the position, time, and estimated speed at 3 second intervals for experiment analysis purposes. Thus, the GPS data recorded from these 77 vehicles is available for input to compute velocity estimates. The data obtained from these vehicles on the northbound stretch of roadway is shown in Figure 4.3a.

Because of privacy constraints, the full trajectories of the vehicles are never sent to the traffic estimation system. Instead, measurements are obtained from the mobile devices using *virtual trip lines* (VTLs) [38], which are virtual geographic line segments placed on the roadway. When a vehicle trajectory intersects a VTL, the phone reports its velocity to the system.

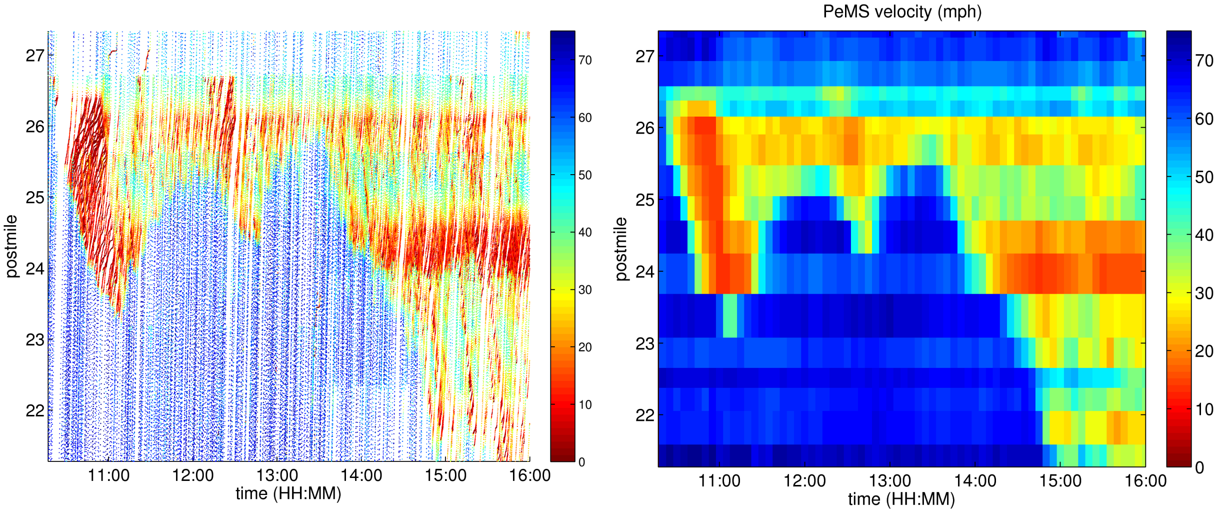
The section is also monitored with 17 inductive loop detectors, which are processed by the PeMS system to produce speed estimates every five minutes [1]. To construct a velocity contour (Figure 4.3b), the roadway is discretized into 17 links centered around the detectors. A complete description of the experiment and comparison of the VTL data and PeMS data can be found in [36]. The data collected during the experiment is downloadable from the project website [83].

During the experiment at approximately 10:30 am, a multiple car accident created significant unanticipated congestion for northbound traffic south of CA-92 (see Figure 4.3a). The California Highway Patrol reported an incident located at postmile 26.64 at 10:27 am, lasting 34 minutes [1], although GPS readings in Figure 4.3a show slowdowns in the area as early as 10:10 am. An earlier version of the EnKF CTM-v algorithm, running in real-time during the experiment, detected the accident's resulting bottleneck and corresponding shockwave [79], and broadcast the results to the web.

4.3.2 Sampling and data collection

A variety of sampling techniques can be used to collect data from GPS enabled mobile devices. In the case of the Nokia N95, the embedded GPS chip-set is capable of producing a geo-position (latitude, longitude, altitude) every three seconds. From this position data, an estimate of the velocity is produced in software at the same frequency. Over time, this vehicle trajectory and velocity information produces a rich history of the dynamics of the vehicle and the velocity field through which it evolves.

While this level of detail is particularly useful for traffic estimation, it can be extremely privacy invasive, since the device is ultimately carried by a single user. Even if personally identifiable information from the data is replaced with a randomly chosen ID through a



(a) (b)

Figure 4.3: I-880N experiment data. (a) Vehicle GPS logs stored locally on the phone. (b) PeMS velocity contour plot. Color denotes speed in mph. x-axis: time of day. y-axis: postmile.

process known as pseudo-anonymization, it is still possible to reidentify individuals from the trajectory data. For example, these pseudo-anonymous trajectories have been combined with free, publicly available data sets to determine the addresses of participants' homes [39].

The transmission of high frequency data without regard to location also wastes resources throughout the system, which can pose scalability problems. In addition disclosing sensitive information, the trajectory information on small roadways near users homes are of lower value to the general commuting public than major thoroughfares such as interstates. Thus, collection of low utility and highly sensitive data should be avoided when sampling using mobile devices.

A variety of methods can be used to address these problems. To manage privacy concerns, in addition to pseudo-anonomization of the trajectory data, the data can be further degraded until a sufficient level of privacy is attained. Common degradation approaches include (i) spatial obfuscation (ie blocking data collection from particular regions, such as home), (ii) increasing uncertainty in the data through noise addition, and (iii) location discretization approaches, which round the measurement to the nearest discrete grid point. The tradeoffs between the measurement utility and privacy under these degradation approaches have been analyzed with experimental data [48] and can be cast as a sampling strategy optimization problem [47].

The alternative sampling strategy which is implemented in this work is based on VTLs [38], which act as spatial triggers for phones to collect measurements and send updates. Each

VTL is composed of two GPS coordinates which make a virtual line drawn across a roadway of interest. Instead of periodic sampling in time, VTLs control disclosure of speed and location updates by sampling in space, creating updates at predefined geographic locations on roadways of interest.

In this sampling strategy, mobile devices monitor their speed and location using GPS and use the locally stored VTLs to determine when a VTL crossing occurs. When the phone intersects a VTL, the device can probabilistically send an update to a back end server with anonymized position, speed and direction information. The device may also probabilistically send the travel time observed between two consecutive trip lines.

A unique feature of this sampling strategy is that data points are only identified through the ID of the VTL, and not that of the mobile device which generated the update, so no privacy-invasive extended trajectories are collected. Through careful placement of trip lines, the system better suited to manage data quality and privacy than through a uniform temporal sampling interval.

4.3.3 System architecture

A prototype system architecture was implemented to test a VTL based sampling strategy (shown in Figure 4.4). The system consists of four layers: GPS-enabled smartphones in vehicles (driving public), a cellular network operator (network operator), cellular phone data aggregation and traffic estimation (Nokia/Berkeley), and information dissemination (Info Consumers). On each participating mobile device (or client), an application is executed which is responsible for the following functions: downloading and caching trip lines from the VTL server, detecting trip line traversal, and filtering measurements before transmissions to the service provider. To determine trip line traversals, the device checks if the line between the current GPS position and the previous GPS position intersects with any of the trip lines in its cache. Upon traversal, the mobile device creates an encrypted VTL update. The update comprises of a speed reading, timestamp, the trip line ID, and the of the direction trip line crossing. These VTL updates are transmitted to the ID proxy server over a secure channel.

Note that all data packets transmitted from the mobile device, regardless of the application (traffic, email, etc), must contain the mobile device identification information for billing by the network provider. Thus, in the *Mobile Century* system, an ID proxy server is used to first authenticate each client to prevent unauthorized updates, then remove the mobile device identification information from the data packets. It then forwards the anonymized updates to the VTL server. Since the VTL update is encrypted with the VTL server's public key (RSA encryption), the ID proxy server cannot access the VTL update content. It only has knowledge of which phone transmitted a VTL update, but no knowledge of the phone's position or speed information. Thus we prevent any single entity from observing both the identification data required by the network operator, and the sensing data. See [38] for a more detailed description of privacy protection in VTL based traffic monitoring.

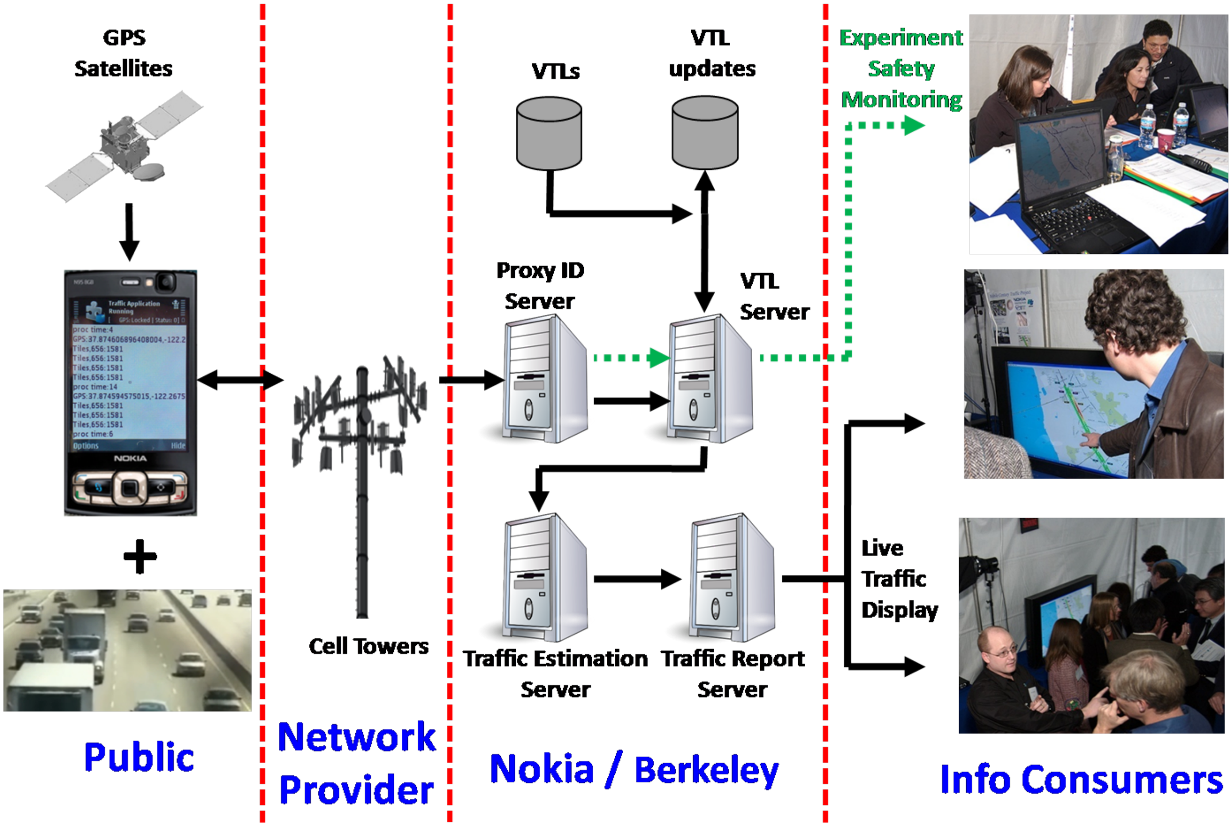


Figure 4.4: *Mobile Century* system architecture overview. The system consists of vehicles equipped with GPS-enabled smartphones, a cellular network provider, data collection infrastructure and traffic estimation, and information display. A live tracking infrastructure (shown in dashed green) was also required for the safety of the UC Berkeley student drivers during this experiment, but it is *not* part of the core system (shown in solid black).

The VTL server stores all trip lines in a VTL database and distributes trip lines within a given region to a mobile device upon receiving a VTL download request for that region. The VTL server also aggregates updates from a large number of probe vehicles in VTL update database and pushes the data to UC Berkeley algorithms for data assimilation (including the CTM–v EnKF algorithm), which run on a traffic estimation server. An estimate manager in the traffic estimation server monitors the performance of the various algorithms and transmits the resulting traffic estimates with highest confidence to the traffic report server.

The traffic report server then sends data to information consumers through a mapping interface on a web site. During the *Mobile Century* experiment, large displays were used on the experiment site to show the live traffic estimate. In a later version of the system developed for the *Mobile Millennium* project, the traffic information is also be accessible from the mobile devices running a traffic data collection client [83].

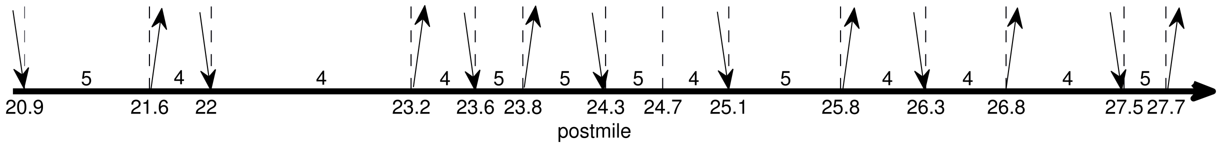


Figure 4.5: Road geometry of I-880N between Decoto Rd. (postmile 20.9) to the south and Winton Ave. (postmile 27.7) to the north. Arrows represent ramp entrance and exit locations, numbers represent the number of lanes on each of the 13 links.

The current VTL implementation generates approximately 1KB of update data for every two minutes per client while driving on a major road. Assuming an average two hours of driving per day on a major road, we expect the total data transfer is 60KB per day. The database servers can easily scale to large number of client updates since the bandwidth and the total data storage demands are rather small by current information industry standards.

4.4 Experimental Results

4.4.1 Numerical implementation

The network implemented for the results presented in this article is a 6.8 mile stretch of I-880N from the Decoto Rd. entrance ramp at postmile 20.9, to the Winton Ave. exit ramp at postmile 27.7. The network model consists of 13 edges and 14 junctions (six exit ramps, seven entrance ramps, and one lane drop), shown in Figure 4.5. The following link parameters are selected for this experiment: $\rho_{\max} = 200$ vehicles per lane per mile, $v_{\max} =$ 70 mph, and w_f= 13 mph. Each link is discretized into equal maximal length cells such that $\Delta x \leq 0.11$ miles and a time step $\Delta t = 5$ seconds is used to ensure numerical stability. The mainline boundary conditions are assumed to be free flowing at 67 mph with standard deviation of 2 mph, and the ramps are set at 30 mph with a standard deviation of 2 mph. The boundary conditions are implemented in the weak sense, and thus are not always imposed on the computational domain. The state noise covariance matrix $\mathbf{Q}^n$ is assumed to be diagonal with standard deviation 2 mph, and the measurement error covariance $\mathbf{R}^n$ is assumed to be diagonal with standard deviation 4 mph. Parameter estimation and characterization of the error covariance structures is the subject of ongoing work. An initial ensemble with 100 members with mean 67 mph is drawn from $\mathbf{P}_a^0$, which is assumed diagonal with standard deviation 4 mph. In one scenario, measurements are collected from ten evenly spaced VTLs, while a second scenario considers measurements collected from 40 evenly spaced VTLs. The estimation algorithm is implemented in Matlab and run on a dual core Intel i5 M540 2.53GHz machine with 4 GB RAM. The estimation algorithm on this experiment site runs just over 14 times faster than real time. For example, a six–hour simulation takes just under 25 minutes.

4.4.2 Comparison with inductive loop detectors

We present a comparison of the velocity estimate from the EnKF CTM-v algorithm using measurements from 10 and 40 VTLs (Figure 4.6a–4.6b) with the velocity estimate obtained from the PeMS system [1]. In order to compare the velocity contours, the EnKF CTM-v estimates are projected onto the coarse discretization induced by the location of the PeMS inductive loop detectors and their corresponding update frequency, then averaged. Because the inductive loops used in the PeMS system are also subject to errors, the resulting velocity contour should not be taken as the ground truth velocity contour.

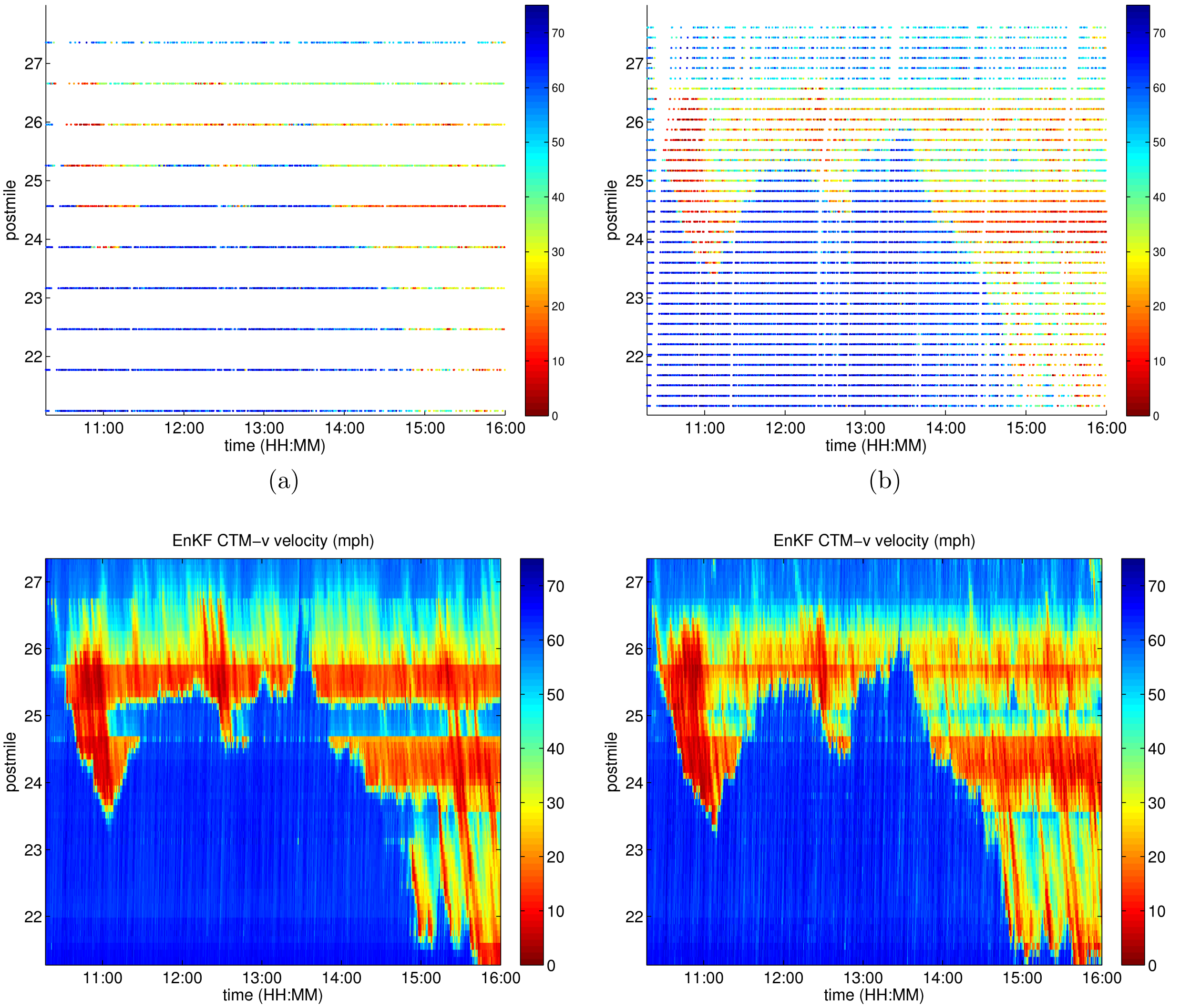
The results of the EnKF CTM-v with 10 VTLs (Figure 4.6c) and 40 VTLs (Figure 4.6d) show good agreement with the PeMS velocity estimate (Figure 4.3b). Both VTL and PeMS estimates capture important features of the congestion pattern, including the extent of the queue resulting from the accident, which propagates upstream to postmile 23.25 just after 11:00, before it begins to clear (see Figs. 4.6 and 4.3b). The effects of bottlenecks created by capacity decreases at postmiles 25.8 and 24.7 are also well described, and differ by less than 10 mph throughout most of the experiment when 40 VTLs are used (Figure 4.7b).

Features of the velocity model are also evident in Figure 4.6c-4.6d. In freeflow, information propagates downstream along characteristics, while in congestion information propagates upstream. Also, the discontinuities in the solution joining free flowing upstream sections with congested downstream sections are resolved with high granularity (see in particular the discontinuity caused by the morning accident, Figure 4.6c-4.6d). On the other hand, the PeMS estimates in the same region transition from freeflow speeds in excess of 65 mph to congested speeds around 20 mph over a period of 15 min.

One area where the model appears to underestimate the congestion appears between postmiles 24.7 and 25.1, in Figure 4.6c. Both the upstream and downstream sections are five lanes, while the intermediate section has only four lanes. The lane drop at postmile 24.7 acts as a bottleneck, and vehicle speeds increase after entering the four lane link. While speeds increase in both the raw GPS logs (Figure 4.3a) and the PeMS estimates (Figure 4.3b), the resulting velocity estimated from 10 VTLs is approximately 15 mph faster than the PeMS estimate (Figure 4.7a). The difference decreases with additional VTLs (Figure 4.7b).

The congestion resulting from the morning accident also highlights some of the differences between the EnKF CTM-v estimates created with 10 VTLs and 40 VTLs. Because the model does not predict accidents, measurements are needed to drive the ensemble states into congestion. Because the congestion is recorded on VTLs earlier and more frequently than with the coarser VTL spacing, the ensembles converge to the slower state more quickly. Additionally, because the congested state is slower, the difference in fluxes surrounding the discontinuity is increased, which in turn causes the shockwave speed to increase. Particularly around postmile 25, the decrease in velocity from the shockwave causes the difference between PeMS and EnKF CTM-v velocity measurements to increase with additional VTLs (Figure 4.7a–4.7b).

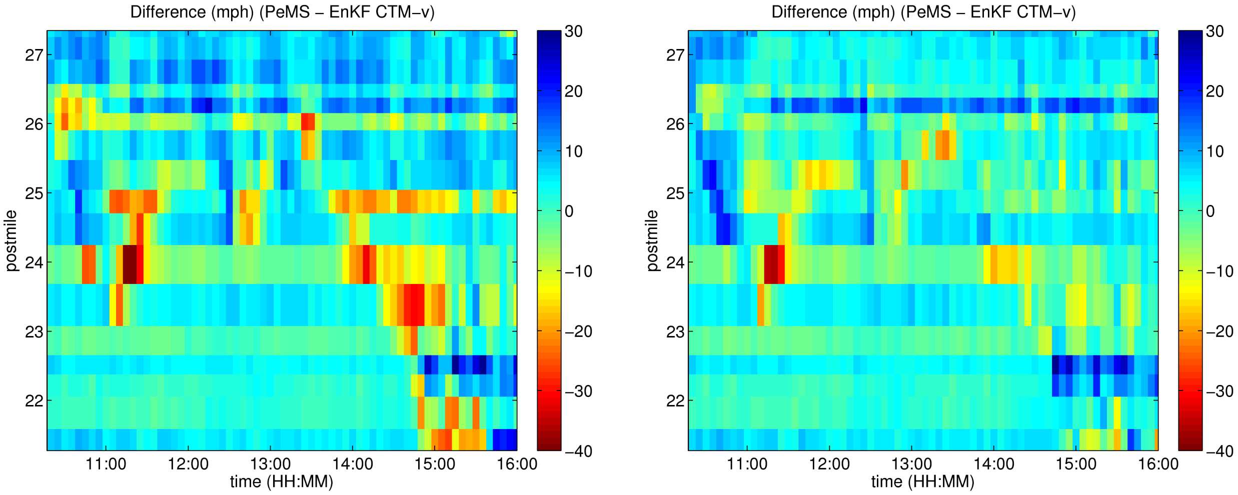
At postmile 26.3, the EnKF CTM-v and PeMS estimates differ by almost 20 mph through-



(a) (b)

(c) (d)

Figure 4.6: VTL measurements with (a) 10 VTLs and (b) 40 VTLs, and EnKF CTM-v velocity contour plots with (c) 10 VTLs and (d) with 40 VTLs. Color denotes speed in mph. x-axis: time of day. y-axis: postmile.



(a) (b)

Figure 4.7: PeMS and EnKF CTM-v comparison. Color denotes speed difference between PeMS and EnKF CTM-v with (a) 10 VTLs and (b) 40 VTLs, in mph. Color denotes speed in mph. x-axis: time of day. y-axis: postmile.

out the day (Figure 4.7a–4.7b). However, there is good agreement on the downstream cell centered at postmile 26.0 which is congested, and the upstream cell centered at postmile 26.5, which is freeflow, so disagreement comes from the transition between the two states. Another area of disagreement occurs in the afternoon rush hour between postmiles 20.9 and 23.6. The EnKF CTM-v estimates show several distinct shockwaves followed by faster traffic. These features are missed in the average speeds reported by PeMS in the region, which leads to high disagreement in this area.

Chapter 5

Travel time estimation

5.1 Introduction

The goal of this chapter is to assess to what degree GPS probe data act as a substitute for conventional traffic monitoring technologies such as inductive loop detectors. In particular, it addresses the trade-offs between between (*i*) velocity data collected from GPS smartphones in probe vehicles, and (*ii*), velocity data obtained from inductive loop detectors, for the purpose of computing travel times on a stretch of roadway. It is a case study using inductive loop detector data and probe data collected during the *Mobile Century* experiment. The new contributions in this section are as follows.

- **Assessment of travel time estimates using inductive loop detector data**. We show dynamic travel times can be estimated to within 6–13% with an average inductive loop detector spacing of 0.83 mi/sensor, with marginal improvement with increased sensor density.
- **Assessment of travel time estimates using virtual trip line-based probe data** We show dynamic travel times can be estimated to within 7% with 137.5 probe vehicles sending measurements on triplines spaced at 2.54 VTL/mi.

This chapter is organized as follows. In Section 5.2, we give an overview of the methodology used in this case study. The processing algorithm used for velocity estimation is given in Section 5.3, and the methods for computing travel times from the velocity field are described. In Section 5.4, the techniques for generating scenarios with various amounts of input data from probe vehicles and inductive loops are presented. In Section 5.5, the results of nearly 917 different scenarios using various amounts of inductive loop detector data and probe data for travel time estimation is presented and summarized.

5.2 Methodology overview

5.2.1 Description of the case study

We address the trade-offs between inductive loop detectors and probe data through a case study which uses experimental data collected during the *Mobile Century* experiment [36] described earlier. As introduced previously in Section 4.3, the data set collected during this experiment and used in this chapter is unique because of the large number of GPS equipped probe vehicles representing 2-5% of the traffic flow, the dense coverage of working inductive loop detectors on the experiment site, and the availability of travel time data obtained from video license plate re-identification. Thus, although the results presented in this chapter are still limited in geographic scope and in time, they are based on the most comprehensive publically available data set to date.

In order to assess the trade-offs between velocity data collected from GPS smartphones and velocity data obtained from inductive loop detectors, it is necessary to define the process by which the data is transformed into an estimate of travel time. In this chapter, we rely on the CTM–v EnKF velocity estimation algorithm developed in Chapter 4 and implemented as as part of the *Mobile Millennium* project [83] to produce a estimate of the velocity field, from which the travel time is computed. The resulting travel time computed from this process is then compared to the travel times recorded from the license plate re-identification video data.

With the data processing algorithm determined, we create a number of scenarios in which the volume of probe data and number of inductive loop detectors made available to the processing algorithm are adjusted. For example, this allows us to compare the accuracy of computing travel times when all of the probe data is made available, to travel times which are computed when only some of the probe data is available, to travel times when some probe data is available and some inductive loop detector data is available. In this way, we can quantify the trade-offs of various amounts of data from probes and inductive loop detector data in terms of increased or decreased accuracy of the computed travel times.

In order to describe and quantify the probe data made available to the travel time processing algorithm, we introduce two metrics of importance to probe data, namely the *penetration rate* and the *sampling strategy*. The penetration rate is defined as the percentage of cars on the roadway reporting probe data compared to the overall traffic flow, including the vehicles which do not send data. In addition to increasing the number of measurements, as the penetration rate increases, the sample of vehicles which generate measurements are more likely to be representative of the total traffic flow.

The sampling strategy refers to how data is collected from the probe vehicles, and can be used to increase or decrease the number of measurements made available for estimating travel times. The sampling strategy used in this chapter collects data from probe vehicles at fixed points in space using the spatial sampling technique of VTLs [38] (see Section 4.3.2). By decreasing spacing between the VTLs, the probe vehicles will send more measurements, with smaller spacing between measurements.

In order to modify the amount of data obtained from inductive loop detectors, the number of inductive loop detectors which are made available to the processing algorithm is adjusted. Because this chapter is a case study of a real highway, it is not possible to modify the location of the inductive loop detectors. Instead, given a fixed number of inductive loop detectors to include for a given scenario, we select the specific loop detectors such that they achieve as uniform of a spacing along the highway as is possible.

5.2.2 Related studies

Several studies have been conducted to assess the applicability of cell phone-based measurements for traffic monitoring [2, 4, 6, 12, 56, 67, 80, 81], including data generated from cell phone towers, which is less accurate than GPS. Bar-Gera [6] compared several months of network data from cellphones to inductive loop detector data on a 14 km freeway segment in Israel, and found them to be in good agreement. Liu et al. [56] evaluated a different network-based cell phone system in Minnesota, and compared travel times to license plate reidentification, and found the system generated results with varying accuracies. A summary of the major network-based cell phone experiments to date can be found in Liu et al. [56].

Several studies have also been conducted to assess the trade-offs between inductive loop detector data and data collected from GPS equipped probe vehicles. In Kwon et al. [50], it is shown that annual estimates of total delay, average duration of congestion, and average spatial extent of congestion can be made with less than 10% error by using either inductive loop detectors placed with half-mile spacing, or by using probe vehicle runs at a rate of about three vehicles an hour. Approximately four to six days of data is needed for reliable estimates from either data source.

The work of Herrera et al. [35] compares a nudging algorithm and a mixture Kalman filtering algorithm to examine how the addition of probe vehicle measurements sampled at a fixed time interval can decrease errors in estimating traffic velocity. On a 0.4 mile stretch of roadway, sampling 5% the traffic at 150 second intervals with inductive loops at both ends of the domain lead to a 16% improvement over the inductive loop detector data alone. The article also uses the *Mobile Century* experiment data to compare three scenarios of time-based sampling of probe vehicles, finding that probe data outperforms inductive loop detector data for estimating traffic velocity if a sufficient number of measurements can be obtained from probe vehicles. This chapter uses the same data set from *Mobile Century* as [35], but we now consider nearly one thousand scenarios to compare probe data to inductive loop detector data, beyond the three presented in [35].

5.3 Algorithm for estimating travel times

Given the velocity data obtained from inductive loop detectors and GPS equipped probe vehicles, a processing algorithm is needed to convert the velocity data into an estimate of travel time. The processing algorithm used in this chapter is the CTM–v EnKF algorithm developed in Chapters 3 and 4 and implemented in the *Mobile Millennium* system. The algorithm of takes velocity data from inductive loop detectors and probe vehicles as input, combines the data with a physical model of traffic evolution, and produces an improved estimate of the velocity along the full stretch of roadway. Using this improved estimate of velocity, an estimated travel time is computed using (i) an instantaneous method and (ii) a dynamic method, to compare against the travel times recorded from video data.

5.3.1 *Mobile Millennium* velocity estimation algorithm

We first present a few remarks on the performance of the CTM–v EnKF estimation algorithm.

First, it is noted that the algorithm was designed as part of the *Mobile Millennium* system, where it is not possible to track probe vehicles for privacy reasons. In other words, it assumed that the probe vehicles send measurements to the system only from pre-selected locations on the highway and, thus, no continuous GPS records from probes are available for the estimation algorithm. Hence, in this study, we also make the assumption that tracking of the vehicles is prohibited. In practice, it is expected that the performance of the estimation algorithm could be improved when tracking of individual probe vehicles is allowed.

Second, the *Mobile Millennium* algorithm does not directly estimate travel times. Instead, travel times are computed from the estimated velocity field, assuming a vehicle travels at the mean speed reported in each cell. Again, it is expected that the performance of the estimation algorithm could be further improved by directly estimating the travel times in addition to estimating the velocity field. Regardless of the potential for further improvement, preliminary studies of the processsing algorithm to compute travel times on the *Mobile Century* experimental data suggest the approach used in this report works well in practice.

Third, it should be noted that the flow model uses some historical flow information to help calibrate the model boundary conditions. In this chapter, historical inductive loop detector data from PeMS was used to estimate a constant (in time) average flow value for the Dumbarton (CA-84) and San Mateo bridge (CA-92) on-ramps feeding traffic to the experiment site. Similarly, loop detectors outside of the actual experiment site were used to estimate a constant flow value for the north end and south end of the experiment site. The use of coarsely approximated constant boundary values is by choice. Although time varying parameters could be used, leading to further improvements in the estimates, extremely precise calibration of a flow model is not generally possible in areas without existing sensing infrastructure. For this reason, we elect to use a sub optimal constant boundary value.

Next, the methods for computing the instantaneous and dynamic travel times from an estimated velocity field are described.

5.3.2 Methods for computing travel times

The *instantaneous method* of computing an estimate of the travel time along a stretch of roadway is as follows. At the time when the instantaneous travel time estimate is produced, the current estimate of the velocity field is recorded. The travel time of a vehicle is simulated, assuming the vehicle travels at the velocity estimated in each cell. The velocity field is assumed to remain constant in time, as the simulated vehicle travels through the velocity field. Thus, the method is an approximation of the true travel time a vehicle would experience, because in practice the velocity field would change as the simulated vehicle completes the trip. The main advantage of the instantaneous travel time is that it does not require a prediction of the evolution of the velocity field, and it should produce accurate travel times when the velocity does not change significantly during the computation.

The *dynamic method* of computing an estimate of the travel time is obtained similarly, with one important modification. Unlike the instantaneous method which assumes the velocity field does not change during the computation, in the dynamic method, the velocity field is updated during the computation. Thus, the traffic conditions are allowed to evolve while the vehicle travels along the roadway. In practice, the computation of a dynamic travel time has be done a posteriori, since the method requires knowledge of the speed evolution from the future time steps. Yet, under rapidly changing traffic conditions, the dynamic method will result in more accurate estimates for the travel times actually experienced by the drivers compared to instantaneous travel times.

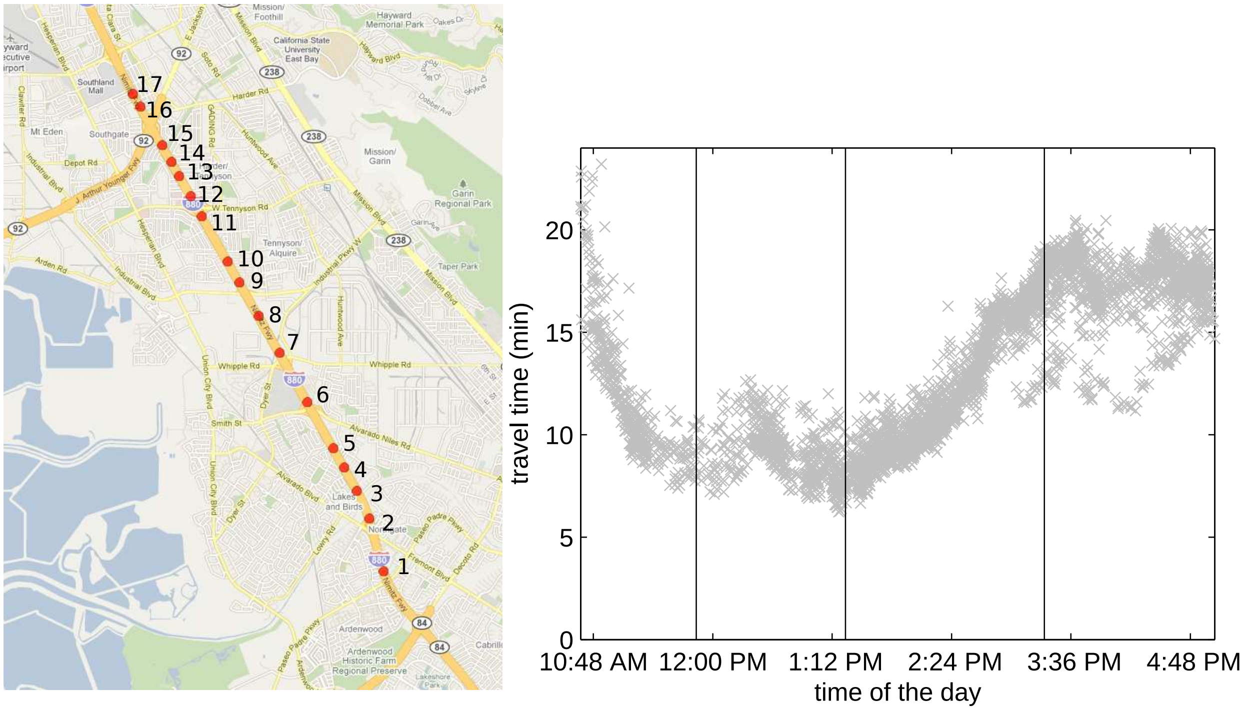
It is worth noting that the dynamic travel time for individual vehicles is measurable via the license plate recognition performed on the video data collected during the experiment. The differences in accuracy between the two travel time computation methods are further discussed in Section 5.5.

5.4 Data selection

The core topic of this study is to assess the trade offs between different amounts of probe data and inductive loop detector data for the purpose of estimating travel times. To achieve this, we algorithmically select different subsets of the inductive loop detector data and GPS probe data from the *Mobile Century* experiment, and use these subsets as inputs to the data processing algorithm described in the previous section. This section describes the various scenarios which modify the type and amount of the data which is made available for estimation, and the selection criteria which are used to generate the scenarios.

5.4.1 Application to the *Mobile Century* experiment data

Recall from Section 5.4.1 the *Mobile Century* site is covered with 17 *inductive loop detector* (ILD) stations which feed measurements into the PeMS system [1]. The inductive loop detectors record the sensor occupancy and vehicle counts every 30 seconds, which is processed by a *Mobile Millennium* filtering algorithm implemented as an extension of [13] in order to obtain the 30 second average velocity at the sensor. The locations of the inductive loop detector stations are shown in Figure 5.1a.



(a) (b)

Figure 5.1: (a) Location of the northbound inductive loop detector stations on the area were travel times are to be estimated; (b) Northbound travel times divided into four time bins from left to right: morning accident (10:00am-11:50am), free flow (11:50am-1:20pm), congestion building (1:20pm-3:20pm), and full congestion (3:20pm-). The travel times obtained from the license plate reidentification video recordings are marked with crosses.

Also, as part of the experiment, high definition video cameras were temporarily installed on bridges to record license plates of northbound traffic. The locations of the video cameras are shown in Figure 5.2. The travel times recorded from the re-identified vehicles traveling northbound between Decoto Rd. to the south and Winton Ave. to the north is shown in Figure 5.1b. During the morning, a 5 car accident caused significant delay, and some drivers experienced travel times in excess of 20 minutes around 10:48 Am. Between 11:50 AM and 1:20 PM, vehicles experience travel times between 8 and 10 minutes on the same stretch of roadway, which steadily increased from 1:20 PM to 3:20 PM. By 3:20 PM, most re-identified drivers experienced heavy evening congestion with travel times increasing to 15–20 minutes.

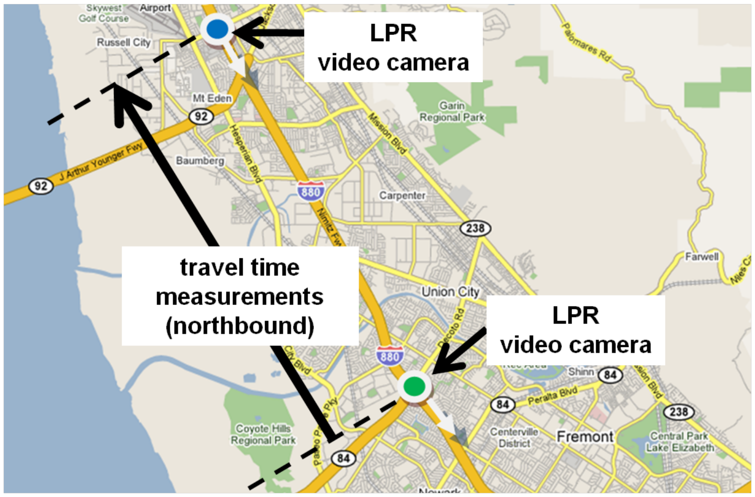


Figure 5.2: Location of video cameras used during the *Mobile Century* experiment for license plate reidentification to compute travel times on northbound I880.

5.4.2 Description of scenarios to be considered

The variables for modifying the amount of the input data for computing travel times considered in this report are as follows.

- **Number of inductive loop detectors.** We modify the number of inductive loop detectors which send data into the data processing algorithm.
- **Number of probe data measurements.** The amount of probe data can be modified in two ways.
    - **Penetration rate.** We modify the number of measurements by increasing or decreasing the penetration rate of the probe vehicles. This is achieved indirectly by changing the number of vehicles from which measurements are collected.
    - **Number of measurements per vehicle.** The second method of modifying the amount of probe data is to change the number of measurements made available from each vehicle. For space-based sampling, this is achieved by changing the number of locations where vehicles report measurements, which is encoded by the number of virtual trip lines.

By modifying the type and amount of inductive loop detector data and probe data through the techniques described above, various scenarios are created to test the impact of the data on computing travel times. In total, the number of scenarios run in this case study is 917. They are generated by creating combinations of the following input data:

- Nine different sets of inductive loop detector data, ranging from scenarios with 0 inductive loop detectors to 16 inductive loop detectors, increasing by increments of two detectors.
- 11 different penetration rates, ranging from scenarios when none of the 1,100 probe vehicle trajectories of *Mobile Century* are used, to scenarios when 100% of the probe vehicle trajectories are used, increasing by increments of 10%. This corresponds to an average rate of probe vehicles between 13.75 veh/hr and 137.5 veh/hr.
- 10 different sets of locations to collect space-based measurements, encoded by scenarios with nine evenly spaced virtual trip lines covering the experiment site (about 8.68 VTL/mi), to scenarios with 99 virtual trip lines (about 0.79 VTL/mi), increasing by increments of 10 virtual trip lines.

Thus the 917 scenarios are created by instantiating scenarios with all combinations of the 9 sets of inductive loop detector data sets, 11 probe penetration rates, and 10 sets of VTLs. The scenarios tested are summarized in Figure 5.1. In the remainder of this section, we describe the specific algorithms which select the data for each scenario.

5.4.3 Algorithms for data selection

Selection of inductive loop detector data

In order to modify the number of inductive loop detector stations which are made available for computing travel times, a simple selection selection criterion is developed for determining the sensors which are made available for estimation. Specifically, given a fixed number of inductive loop detectors to include, the inductive loop detector stations are selected in order to minimize the variance of the distance between consecutive sensors. This allows us to pick the sensors such that they are as uniformly distributed across the experiment site as is possible, given the fixed locations of the candidate inductive loop detector stations. We describe this criterion in detail in this section.

We consider a stretch of highway of length L, starting at $x = 0$ and ending at $x = L$, with n inductive loop detector stations located at $x_1, x_2, \cdots, x_n$, as shown in Figure 5.3.

Let S_i denote the spacing between sensor i and $i + 1$. In order to treat the boundaries without explicit knowledge of sensors outside the domain $x \in [0, L]$, it is assumed only half of the first inter-station spacing S_0 and the last inter-station spacing S_n is in the domain of interest. The weighted average spacing between the sensors is given by:

Table 5.1: A subset of runs used in the study.

Run	ILD stations	Probe type	Probe rate (veh/hr)	VTL/mi	Sampling interval (s)
1	1	No Probe	-	-	-
2	2	No Probe	-	-	-
3	3	No Probe	-	-	-
⋮	⋮	⋮	⋮	⋮	⋮
101	0	Space	13.75	0.79	-
102	0	Space	13.75	1.67	-
103	0	Space	13.75	2.54	-
104	0	Space	13.75	3.42	-
⋮	⋮	⋮	⋮	⋮	⋮
478	6	Space	110	6.93	-
479	6	Space	110	7.81	-
480	6	Space	110	8.68	-
481	6	Space	123.75	0.79	-
482	6	Space	123.75	1.67	-
⋮	⋮	⋮	⋮	⋮	⋮
915	16	Space	27.5	6.93	-
916	16	Space	27.5	7.81	-
917	16	Space	27.5	8.68	-

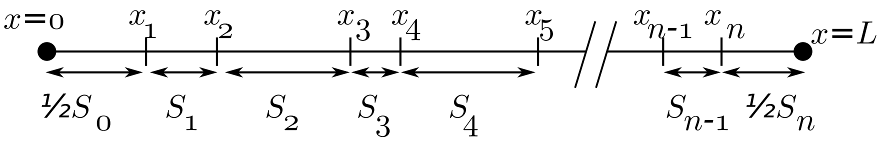


Figure 5.3: Highway segment of length L, with n inductive loop detector stations located at x_i.

$$\bar{S} = \frac{\frac{1}{2}S_0 + S_1 + S_2 \cdots + S_{n-1} + \frac{1}{2}S_n}{n} = \frac{L}{n} \tag{5.1}$$

where the first and last spacings have a weight $\frac{1}{2}$, since only half of these spacings actually lie within the $[0, L]$ domain. Note that the average spacing is independent of the specific locations of the sensors x_i, and consequently cannot be used as a selection criterion.

Instead, we use a selection criterion which explicitly takes the uniformity of the inter-sensor distances S_i into account. This is achieved by minimizing the variance σ^2 of the inter-station spacings $S_k, 0 \leq k \leq n$, given by:

$$\sigma^2 = \frac{1}{2n}(S_0 - \bar{S})^2 + \frac{1}{n}\sum_{1\leq i\leq n}(S_i - \bar{S})^2 + \frac{1}{2n}(S_n - \bar{S})^2 \tag{5.2}$$

Again, the first and last spacings have a weight $\frac{1}{2}$, since only half of these spacings actually lie within the $[0, L]$ domain.

In practice, rather than minimizing the variance σ^2, it is convenient to minimize an equivalent loop detector placement criterion denoted $\tilde{S}$:

$$\tilde{S}(x_1, x_2, \cdots, x_n) = 2\sqrt{\frac{x_1^2}{2n} + \frac{(L - x_n)^2}{2n} + \sum_{1\leq k<n}\frac{(\frac{x_{k+1}-x_k}{2})^2}{n}} \tag{5.3}$$

which is equal to σ^2 plus a constant offset. The best set of k inductive loop detector stations is then given by:

$$U^*(k) = \text{argmin}\{\tilde{S}(U) \quad | \quad U \subset \{x_1, x_2, ..., x_n\} \text{ and } |U| = k\} \tag{5.4}$$

where $|U|$ represents the number of elements in the set U. The resulting selections for the inductive loop detector stations are shown in Table 5.2.

For the case in which the chosen inductive loop detector stations are uniformly spaced within the section of interest, the criterion $\tilde{S}$ is equal to the average spacing spacing $\bar{S}$. Because $\bar{S}$ serves as a lower bound for $\tilde{S}$, the difference between $\tilde{S}$ and $\bar{S}$ indicates the degree of non uniformity of the sensor spacings caused by the fixed set from which the sensors are selected. Figure 5.4 shows the difference between the inductive loop detector placement criterion $\tilde{S}(U^*(k))$ and its lower bound, the average inductive loop detector spacing $\bar{S}(k)$, is small, indicating that the sensors are uniformly spaced.

Table 5.2: Inductive loop detector selection results. Given a number k, the selection algorithm returns the set $U^*(k)$ of k inductive loop detector stations which minimizes the inductive loop detector placement index $\tilde{S}(U^*(k))$. the labels in $U^*(k)$ correspond to the labels of the inductive loop detectors in Figure 5.1a.

k	$\tilde{S}(U^*(k))$ (mi)	$U^*(k)$
0	∞	$\emptyset$
1	6.51	{ 8 }
2	3.25	{ 4, 11 }
3	2.17	{ 3, 8, 14 }
4	1.63	{ 2, 6, 9, 15 }
5	1.33	{ 2, 6, 8, 11, 16 }
6	1.11	{ 1, 3, 6, 8, 11, 16 }
7	0.95	{ 1, 3, 6, 8, 10, 13, 16 }
8	0.83	{ 1, 3, 6, 7, 9, 11, 14, 16 }
9	0.73	{ 1, 2, 4, 6, 7, 9, 11, 14, 16 }
10	0.66	{ 1, 2, 4, 6, 7, 8, 10, 11, 14, 16 }
11	0.60	{ 1, 2, 3, 5, 6, 7, 8, 10, 11, 14, 16 }
12	0.55	{ 1, 2, 3, 5, 6, 7, 8, 10, 11, 13, 15, 16 }
13	0.51	{ 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 13, 15, 16 }
14	0.48	{ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 15, 16 }
15	0.46	{ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 15, 16, 17 }
16	0.43	{ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17 }
17	0.41	{ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 }

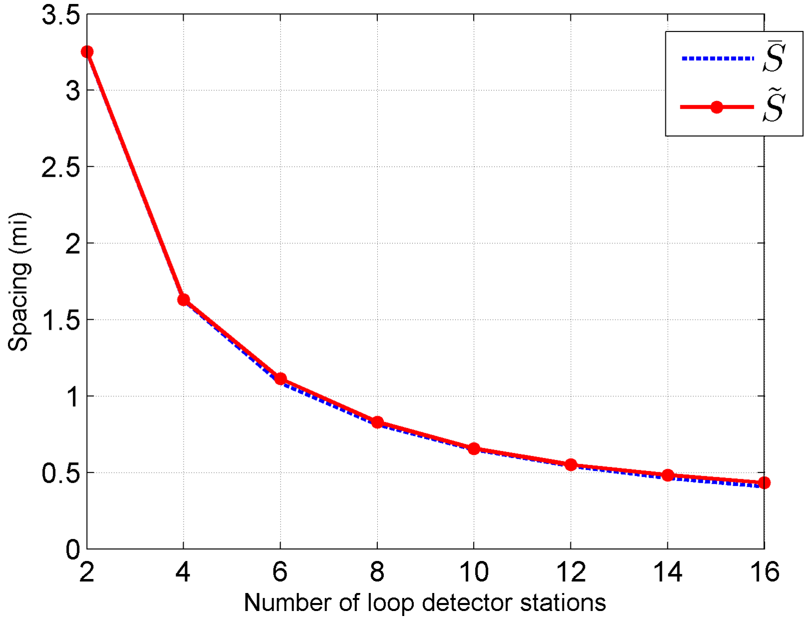


Figure 5.4: Change in the spacing between inductive loop detector stations, as a function of the number of loop detector stations used in the inductive loop detector selection algorithm.

Penetration rate for probe data

During the *Mobile Century* experiment, the GPS data obtained from 77 GPS equipped probe vehicles generated a total of 1,100 vehicle trajectories on I880 in the northbound direction. Each vehicle trajectory consists of the estimated vehicle position and velocity recorded at three second intervals. The trajectory data is filtered to guarantee physically meaningful acceleration and velocity data based on assumed vehicle dynamics. Less than 0.01% of the data were identified as outliers, which were replaced by an interpolated value.

Throughout the day, these 1,100 vehicle trajectories represent a 5 minute penetration rate which ranges between approximately 0% and 5%, depending on the time and location at which the penetration rate is estimated [36]. Figure 5.5 shows the 20 minute penetration rate estimated at the center of the *Mobile Century* experiment site (near inductive loop detector station 9 in Figure 5.1a), which varies between 1.5% and 3%.

Before criteria to modify the penetration rate are discussed, a few limitations of the penetration metric must be highlighted. In general, the penetration rate is difficult to determine for probe vehicles specifically because it depends on (*i*) the number of equipped probe vehicles, (*ii*) the total traffic flow, and (*iii*) the evolution of the traffic flow in space and time. Typically, only the total number of equipped probe vehicles is known to probe data providers. Similarly, the total traffic flow can only be estimated from counts recorded by inductive loop detectors at predefined locations. Finally, because the evolution of the traffic

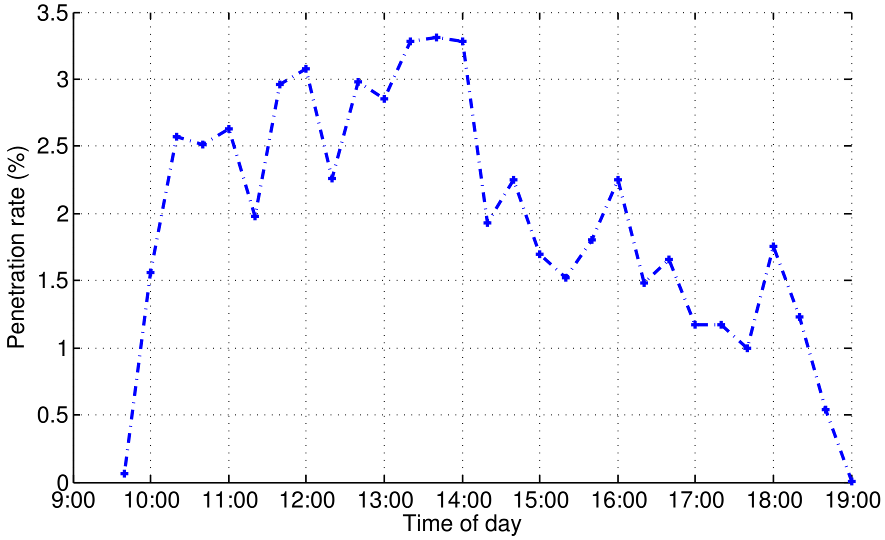


Figure 5.5: 20-minute average penetration rate in the center of the *Mobile Century* experiment site on I-880 NB.

flow is not under the control of the probe vehicles, it is nearly impossible to a priori specify a penetration rate which is both uniform in space, and uniform in time.

Because of the inherent difficulty in specifying the penetration rate a priori, we instead elect to directly modify the number of equipped probe vehicles as a proxy for modifying the penetration rate. The number of equipped probe vehicles in this study varies from 0% of the 1,100 vehicle trajectories, to 100% of the 1,100 vehicle trajectories, increasing by increments of 10%. Over the eight hour experiment, this corresponds to an average rate of probe vehicles between 0 veh/hr and 137.5 veh/hr. When a subset of the vehicle trajectories is required, the subset is determined by selecting the trajectories at random. For example, 50% of the collected probe data corresponds to exactly 550 vehicle trajectories (68.75 veh/hr), which are selected at random before the simulation. The corresponding 20 minute penetration rate at the center of the experiment site would then be half of the penetration rate shown in Figure 5.5, but only in the expected sense, since the trajectories are selected at random.

Space–based sampling

In order to modify the number of measurements used from each probe vehicle trajectory under spatial sampling, the number of locations where measurements are collected are modified. The locations where measurements are obtained are encoded through the placement of VTLs, which can be viewed as virtual geographic markers which trigger vehicles to send measurements when the vehicle trajectory intersects the VTL. A complete description of the VTL sampling strategy is described in detail in Hoh et al [38] and higlighted in Section 4.3.2.

Because the VTLs are not physical infrastructure, it is possible to place them anywhere on the experiment site. The determination of optimal VTL placement is complex, so instead we elect to place the VTLs uniformly across the experiment site. The number of VTLs, n_{VTL} tested in our scenarios varies from nine VTLs to 99 VTLs, increasing by increments of 10 VTLs. This corresponds to an average spacing between 0.72 to 7.1 VTL/mi. Note the number of VTLs used on the experiment site is significantly higher than the number of inductive loop detector stations. This is possible because unlike inductive loop detector stations, the marginal cost of virtual trip lines is small.

5.5 Results and discussion

In this section, we present the results of 917 runs with varying amounts of probe and inductive loop detector data. We also vary the type of travel time computed (instantaneous or dynamic). The quantification of error is described in Section 5.5.1, and the computational results are shown in Section 5.5.2.

5.5.1 Error quantification

Since validation data is available for dynamic travel times (see Section 5.3.2), an error metric is used to compare the velocity estimation algorithm output that has been converted to travel times with the travel time measured from video recordings. By using the travel time error as a performance metric, estimation algorithm results can be compared with the results obtained when using different types and quantities of the input data.

Since the license plate reidentification data provides a distribution of individual vehicle travel times (see Figure 5.1b), we define the true travel time as a one minute moving average of the recorded travel times. Figure 5.1b also shows the division of the experiment into four time periods that represent the different phases of the traffic during the experiment. These periods are (i) the *morning accident*, were travel times are decreasing as an incident clears, (ii) a *free flow* period during the middle of the day when travel times are low, (iii) a *congestion building* period before the evening rush hours, and (iv) and *full congestion* during the evening rush hours. Because of the different traffic conditions present in these time intervals, in addition to computing the error across the full day, the error is also computed for each time interval.

The error is quantified as follows. Let n be the number of estimates given in a period for which the error is to be computed, with each estimate indexed by i. The travel time error is computed as follows. Let $T_v(i)$ be the mean travel time from the video data at time i, $T_{\text{inst}}(i)$ be the estimated mean travel time computed with the instantaneous method at time i, and $T_{\text{dyn}}(i)$ be the estimated mean travel time computed with the dynamic method at time i, as in Sections 5.3.2 and 5.3.2, respectively. The *mean absolute percent error* (MAPE) for the travel time computed with the instantaneous method is:

$$\varepsilon_{\text{inst,MAPE}} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{T_v(i) - T_{\text{inst}}(i)}{T_v(i)}\right| \tag{5.5}$$

while the MAPE for the travel time computed with the dynamic method is:

$$\varepsilon_{\text{dyn,MAPE}} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{T_v(i) - T_{\text{dyn}}(i)}{T_v(i)}\right| \tag{5.6}$$

The MAPE is used in this study to aggregate the error in the model estimates over a given time period in order to produce a single value for the error with given input data and time period. Similarly a non-aggregated error would result in a representation of the error that is dependent on the time dimension of the results. Also, note that the spatial dimension of the error vanishes when the travel times are produced from the model estimated mean speed fields.

5.5.2 Computational results

Implementation

The estimates were computed using the existing *Mobile Millennium* highway model, implemented in Java. The model was run 917 times with various data inputs. Given the computationally intensive nature of the algorithms utilized, each run (an eight-hour simulation) took approximately 20 minutes to complete. Each run consisted in the computation of the mean speed field evolution and computation of both instantaneous and dynamic travel time every 30 seconds, from 10 am to 6 pm on the day of the experiment.

The runs took 315 CPU-hours, and were distributed on 8 servers equipped with 2.2 GHz dual core AMD Opteron CPUs and 8 GB RAM, which reduced the computation time to 39 hours. The travel time data was then extracted manually from the servers. Finally, this data was analyzed in Matlab.

To give an idea of the input data variability between the runs, a representative subset of the input data combinations is shown in Table 5.1. The table shows the travel time type, number of inductive loop detector stations, average rate of probe vehicles, and number of VTLs per mile. These parameters are presented as a function of the run number. The number of probe vehicle measurements used in each simulation is shown in Figure 5.6.

Using only inductive loop detector data in the model

The first analysis of the traffic estimates is based on the results obtained when using inductive loop detector loop detector data as the only input to the model. These results give us a baseline for the comparison between probe and loop detector data. A total number of 17 runs were conducted based only on the inductive loop detector data, by varying the

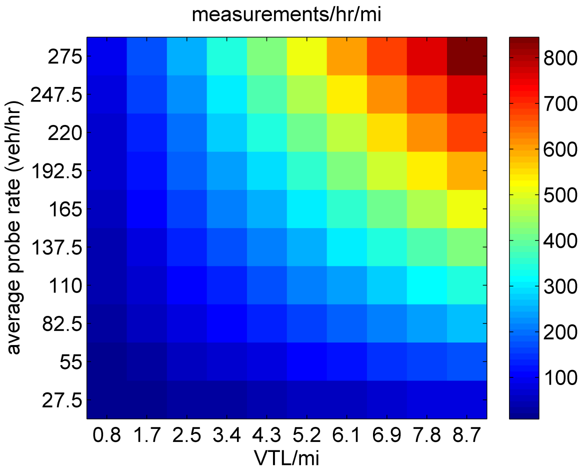
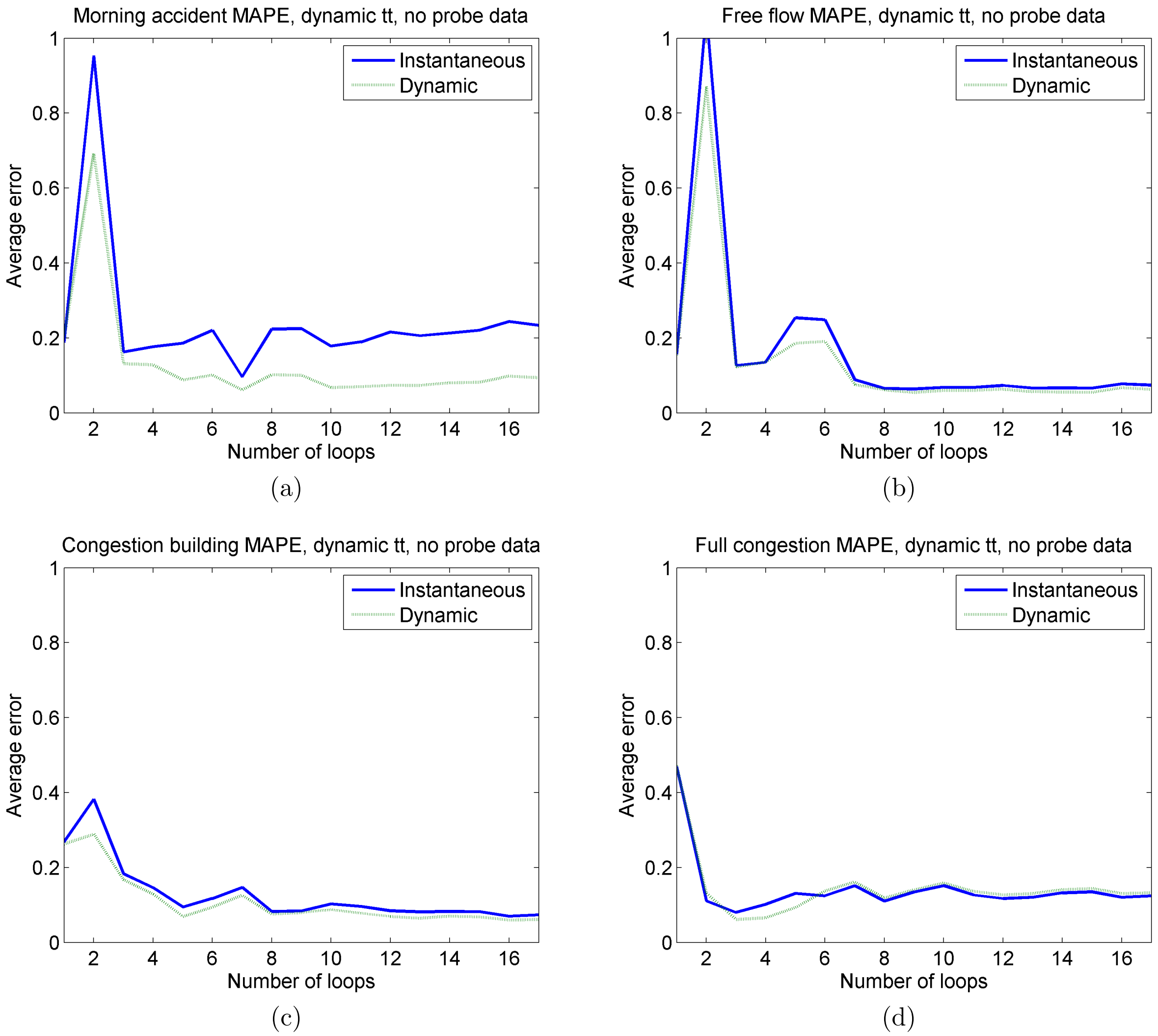


Figure 5.6: Number of probe vehicle measurements used in the simulations when using VTL data. See also Table A.1

number of sensors according to the selection algorithm in Section 5.4.3. Both instantaneous and dynamic travel times were computed. The labels of the inductive loop detector stations used in the estimation are presented in Table 5.2 as a function of the number of stations selected (see also Figure 5.1a).

The results of these runs are shown Figure 5.7. The subfigures show the estimation error broken down by time of the day, as defined in Figure 5.1b. During the morning accident (Figure 5.7a), the dynamic travel times converge to estimates with 7% error, while the instantaneous estimates remain above 20% error. The instantaneous and dynamic estimates have between 6% and 7% error during the free flow and congestion building periods (Figure 5.7b and 5.7c), and 13% error during the full congestion period (Figure 5.7d), with the instantaneous and dynamic estimates performing similarly.

The number of inductive loop detector stations used tends to have a positive impact on the quality of the estimate when less than eight inductive loop detector stations are used. Note that the curve is not monotonic decreasing. This is because when only a few sensors are deployed, the error becomes highly dependent on the placement of the sensors. It is expected that an optimal sensor placement algorithm would reduce the error. The threshold of eight inductive loop detector stations corresponds to the inductive loop detector placement index $\tilde{S}(U^*(8)) = 0.83$ mi (Table 5.2). However, using data from more than eight inductive loop detector stations does not improve the quality of the estimates. If fewer than three inductive loop detectors are used, the estimation error is unacceptably high, at some points reaching as high as 100% error.



(a) (b)

(c) (d)

Figure 5.7: MAPE computed using inductive loop detector data only, no probe data. Travel time is computed using the dynamic method (green dash) and instantaneous method (solid blue). x-axis: number of inductive loop detector sensors, y-axis: MAPE (a) morning incident; (b) free flow; (c) afternoon as congestion increases; (d) evening congestion.

Using only VTL data in the model

The second part of the analysis consists of the travel time estimates obtained when using VTL data only. The changing parameters of the input data are the number of VTLs deployed on the experiment site and the rate of the probe vehicles used to produce speed measurements at the locations of VTLs, see Table 5.1. The estimation errors of the travel times obtained with the dynamic method are shown in Figure 5.8.

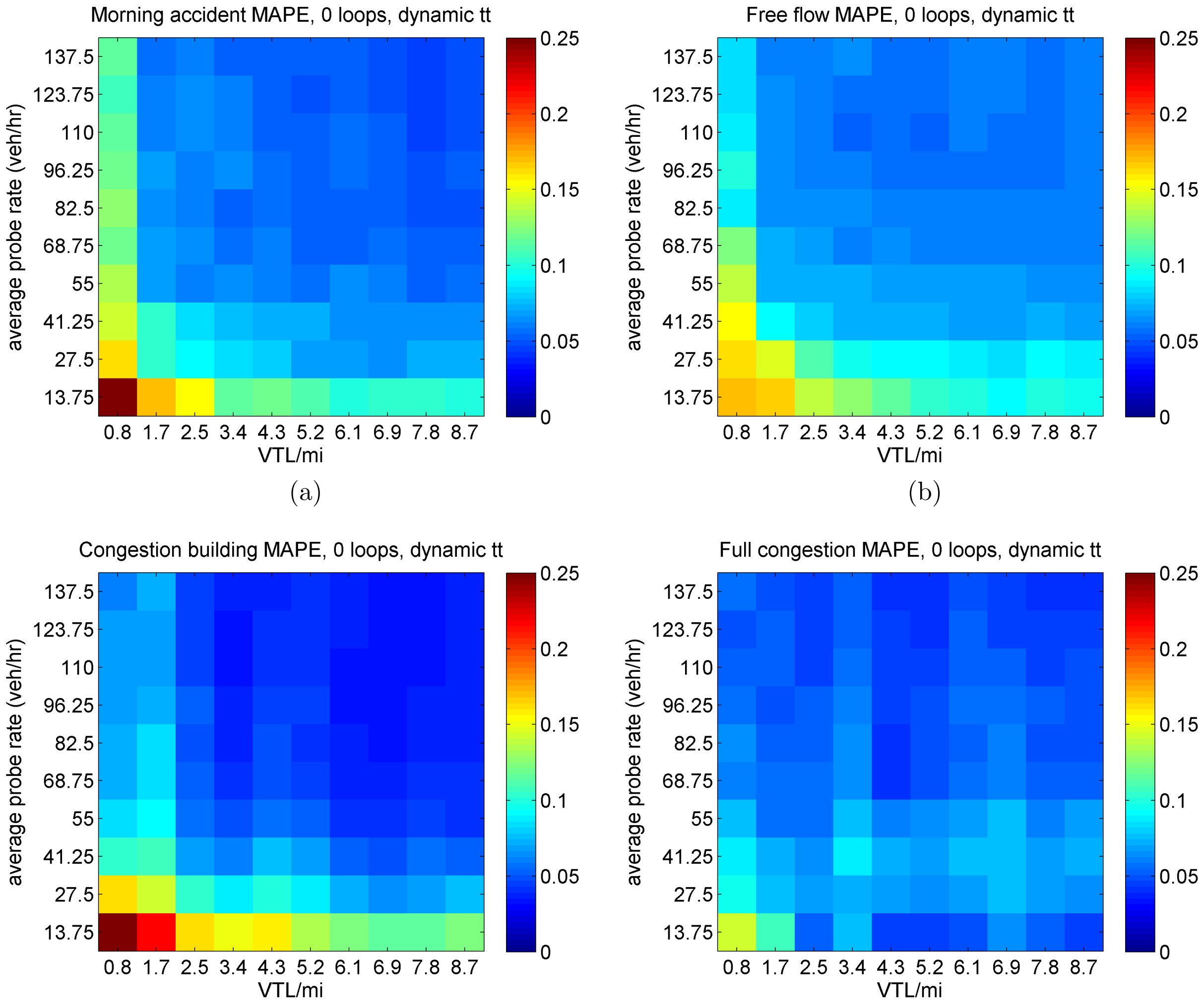
In each of the time periods, estimates of the travel time can be achieved with less than 6% MAPE, with sufficient probe vehicles and virtual trip lines. However, when more than 68.75 veh/hr are used with more than 2.54 VTL/mi, only small improvements in the accuracy of the estimates can be achieved. When compared with inductive loop detectors, using 68.75 veh/hr and 2.54 VTL/mi performs as well as the estimates using more than eight inductive loop detector stations during the morning accident, free flow, and congestion building periods, but has less than half the error of inductive loops during the full congestion period. When 68.75 veh/hr are used, the overall probe penetration rate as a percentage of the total number of vehicles is one half the values shown in Figure 5.5.

Mixing VTL and loop detector data

The dynamic travel time estimation errors using both VTL and loop detector data simultaneously is assessed in Figure 5.9, where the change in the dynamic travel time MAPE due to the addition of data from six inductive loop detectors is computed. The results shown are a representative subset of all the runs performed when mixing the two data types.

At low probe data rates during the morning accident, free flow, and congestion building periods, adding inductive loop detector data increases the accuracy of the dynamic travel time estimates. For example, during the morning accident (Figure 5.9a), with a probe rate of 13.75 veh/hr and a VTL spacing of 0.79 VTL/mi, adding inductive loop detector data reduced the error from 29% to 8%. During the full congestion period, the dynamic travel time estimate accuracy decreased when inductive loop detector data was added at low probe rates (13.75 veh/hr). This is likely due to the fact that the estimates based on virtual trip line data only were unusually accurate, even performing better than simulations with more probe vehicles.

At higher penetration rates (above 68.75 veh/hr) adding data from the six inductive loops has negligible effect, increasing or decreasing the accuracy only slightly. The exception is during the free flow period, when the MAPE increased (between 0.05 and 0.08) even at high probe rates, when 0.79 VTL/mi were used. The errors in the free flow period are magnified due to the small base travel time, which is under 10 minutes, and it is in fact not constant during the period (see Figure 5.1b). Moreover, it is clear from Figure 4.3 that there is an area of heavy congestion around postmile 26 even during the free flow period, which is difficult to capture correctly with sparse sampling.



Morning accident MAPE, 0 loops, dynamic tt

Free flow MAPE, 0 loops, dynamic tt

Congestion building MAPE, 0 loops, dynamic tt

Full congestion MAPE, 0 loops, dynamic tt

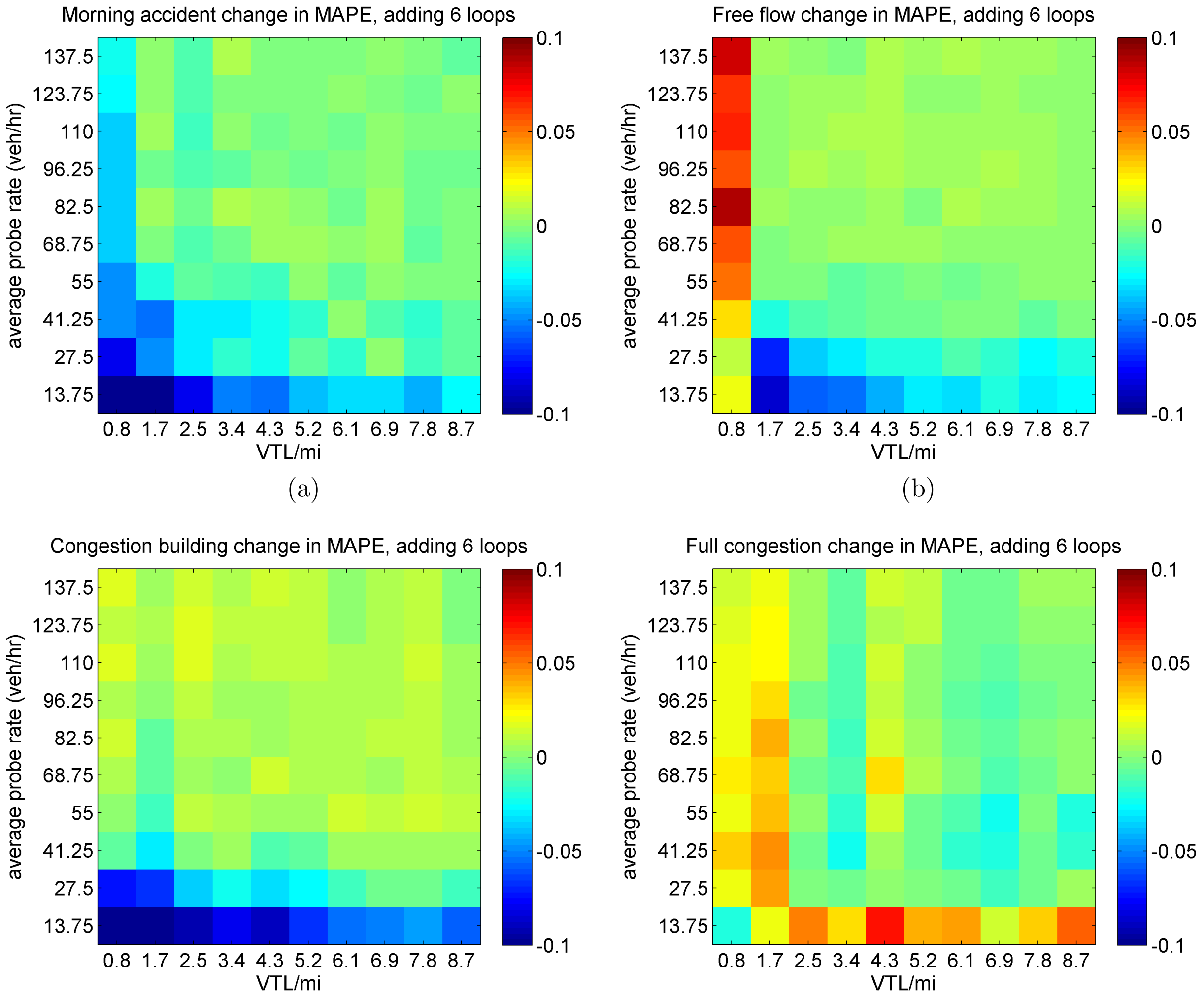
(a)

(b)

(c)

(d)

Figure 5.8: MAPE contours computed using VTL data only, no inductive loop detector sensors. Travel time is computed using the dynamic method. x-axis: number of VTLs, y-axis: average probe data rate (a) morning accident; (b) free flow; (c) congestion building; (d) full congestion. Color scale limited to 0.25. See also Table A.2–A.5.



(a) (b) (c) (d)

Figure 5.9: Change in MAPE contours when adding six inductive loop detectors to VTL data. x-axis: number of VTLs, y-axis: average probe data rate (a) morning accident; (b) free flow; (c) congestion building; (d) full congestion. Color scale limited to ±0.1. See also Table A.6–A.9.

Using instantaneous travel time as an estimate for dynamic travel time

Figure 5.10 shows a comparison of the estimation errors when using instantaneous and dynamic travel times for the morning accident period. Instantaneous travel times can be determined at any time on any route using the speed estimates, and used as a proxy for dynamic travel times. As was shown for the inductive loop detector data in Figure 5.7b and 5.7d, instantaneous and dynamic travel time estimates are very similar when traffic conditions change sufficiently slowly. The same holds when estimating travel times from probe data.

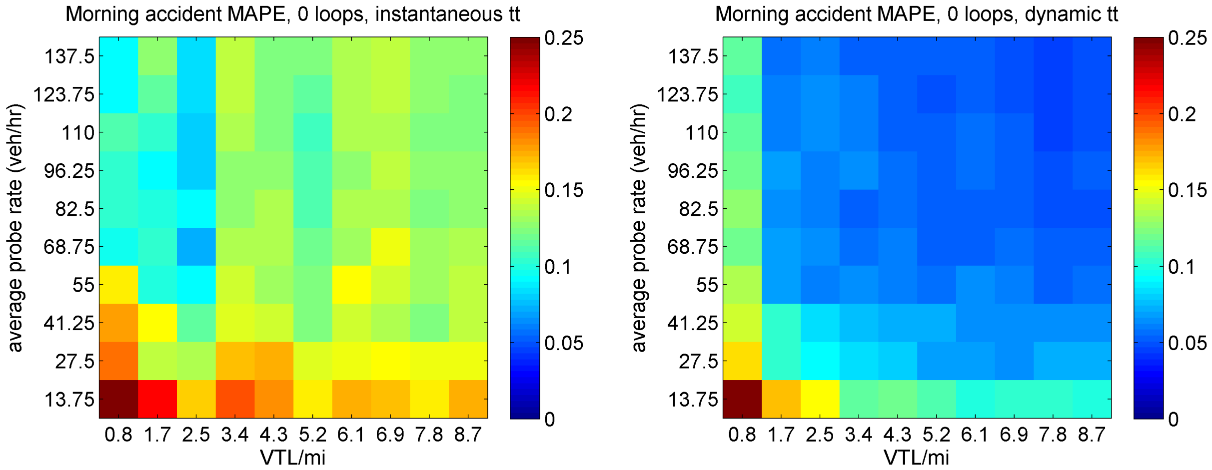
By looking at the instantaneous travel time errors in Figure 5.10a, an interesting result can be seen. The results suggest that adding more probe data results in an increased travel time estimation error. However, this result is expected, and can be explained by focusing on the scenarios (in Figure 5.10a) in which the penetration rate of the probe vehicles is low and no loop detectors are used. Here, the instantaneous travel time estimate performs well, and may seem like a valid estimate of the true travel time during the incident. However, this gives a misleading indication of the quality of these travel time estimates. The good performance of the instantaneous estimate is caused by the fact that the current state of the traffic (speed field) is very poorly captured in the underlying scenario and the speed of the traffic is heavily overestimated. This causes the instantaneous travel time estimate to perform as a good predictor of the future traffic conditions, namely, as a predictor of the clearing incident. When the number of probe measurements increases, the speed field estimate is captured more accurately, and the increased error in the travel time estimate is caused by the instantaneous approximation.

5.5.3 Summary of key results

The following is a summary of the key results found in this study:

1. **Achieving 10% error for dynamic travel times.** In this study, it was found that the dynamic travel time estimates can be achieved with less than 10% error when using a flow model with data assimilation, by using either inductive loop detector data, probe data, or a mixture of both inductive loop detector data in probe data. Moreover, the estimates from virtual trip line-based probe can achieve a higher degree of accuracy when all available probe data is used compared to estimates from inductive loop detectors when all inductive loops on the experiment site are used, although in general the performance is similar.

2. **Minimum loop detector spacing for travel time estimation.** In this study, using data from more than eight inductive loop detector stations (average spacing 0.83 miles) did not give extra benefit in the travel time estimation. The error remains constant between 6–13% depending on the time of day, regardless of the added loop detector stations.



(a) (b)

Figure 5.10: MAPE contours computed for the morning accident using VTL data and inductive loop detector sensors. x-axis: number of VTLs, y-axis: average probe data rate (a) 0 inductive loop detector sensors, instantaneous travel time; (b) 0 inductive loop detector sensors, dynamic travel time. Color scale limited to 0.25. See also Table A.10–A.11.

3. **Diminishing travel time accuracy improvement.** When sampling probe vehicles at a rate of 68.75 veh/hr with more than 2.54 VTL/mi, increasing the number of probe measurements by adding more probe vehicles or additional trip lines causes only small improvement on the travel time accuracy.

4. **A mixture of probe and loop detector data in travel time estimation.** It was found that when complementing loop detector data with probe vehicle data, better estimates for travel times are obtained, especially at low penetration rates. For example, if using loop detectors spaced more than 2.11 miles apart, probe data can give over 50% increase in the travel time accuracy. These results hold generally, independent of the sampling strategy of the probe vehicles.

Chapter 6

Conclusions and future work

The main contributions in this thesis are briefly outlined below

- **Review of weak entropy solutions to the LWR PDE**. In Chapter 2, we summarized the basic mathematical tools of the LWR PDE, which laid the groundwork for the development of a velocity evolution equation consistent with this model. We showed that significant mathematical challenges are introduced by the formation of shocks in the density profile, at which point classical solutions to the LWR PDE no longer exist. The introduction of more general weak solutions allows for shocks, but the uniqueness of solutions can only be guaranteed in presence of an additional entropy condition.

- **Development of two new velocity evolution equations consistent with hydrodynamic theory**. In Chapter 3, we introduced the continuous velocity evolution equation called the LWR–v, and a discrete velocity model, the CTM–v, as transformations of the LWR PDE and its discretization into the velocity domain. Again, due to the presence of shocks in the density domain, we showed that the LWR–v model only exists as a special case when the velocity function is affine. The CTM–v model was derived directly from the integral form of the LWR PDE to circumvent this issue for arbitrary invertible velocity functions. Using the Riemann solver in the density domain proposed by Daganzo [21] and Coclite, Garavello, and Piccoli [14], we developed a Riemann solver in the velocity domain to extend the CTM–v model to networks.

- **Velocity estimation with ensemble Kalman filtering**. In Chapter 4, we propose to solve the velocity estimation problem with the CTM–v model and velocity measurements obtained from GPS smartphones sampled with virtual trip lines using ensemble Kalman filtering, which is a highly scalable estimation algorithm. Moreover, ensemble Kalman filtering does not require the model to be linearizable, which is a feature not possessed by the CTM or the CTM–v due to standing shockwaves. The algorithm was run live during the *Mobile Century* field experiment, and showed that velocity field estimates from mobile phone data and the EnKF CTM–v algorithm are of comparable quality to estimates produced from inductive loops.

- **Travel time estimation comparison between probe data and inductive loop data**. In Chapter 5, travel time data collected during the mobile century experiment from license plate reidentification was used to assess the accuracy of travel times computed by the CTM–v EnKF algorithm using various amounts of probe data collected through VTLs, and inductive loop detector data. For this experiment, it was shown that estimates obtained with VTL probe data performed as well as or better than estimates using inductive loop detector data. While this result may vary at different locations, or with different estimation algorithms, is a strong indicator of the possibility for probe data to augment or replace inductive loop detector data for the purpose of estimating travel times.

Several other important findings have been uncovered through this work, which may be the focus of future research efforts.

- **Flow measurements from mobile devices**. One of the major challenges for estimating traffic conditions with probe data is in the quantity which is measured. Unlike inductive loop detectors, which count the number of vehicles crossing a point in time yielding the basic unit (vehicles) which are conserved, GPS smartphones do not directly measure vehicle flows. As the number of GPS equipped mobile devices continues to rapidly expand, it may be possible to model the proportion of equipped vehicles, and derive approximate flows to integrate into the estimation algorithm. This would be beneficial in areas which do not have dedicated sensing of the structure already deployed.

- **Heterogeneous sensor data fusion**. A likely future scenario faced by state departments of transportation is one in which low-cost probe data can be acquired in many areas. To improve the estimates of various transportation network performance metrics, this information may be combined with more expensive, but dedicated sensors such as radar, inductive loops, and video detection. Estimation of traffic conditions using all available data is challenging due to the various measured quantities (point velocities, flows, travel times, etc), and the spatial and temporal extent of the measurements, and will require the development of new models and estimation algorithms.

- **Traffic prediction algorithms** One of the important insights learned through the *Mobile Century* and *Mobile Millennium* projects is that traffic estimation will become an important subcomponent of other consumer focused location based services. A now often used example is one in which a traffic service is linked to a routing engine and a personal calendar, which could trigger warnings for events based on the user's current location and the expected time to reach the next appointment based on traffic conditions. Future services like this will rely not only on the ability to estimate current conditions accurately, but also accurate prediction. Even short-term travel time prediction could be enhanced by a velocity forecasting algorithm, as was shown in Chapter 5. Statistical inverse problems theory and machine learning theory offer tools to tackle this problem.

Appendix A

Supplementary tables

Table A.1: Number of probe vehicle measurements used in the simulations when using VTL data. See also Figure 5.5.2.

measurements/hr/mi										
probe rate	VTL/mi									
veh/hr	0.79	1.67	2.54	3.42	4.30	5.18	6.05	6.93	7.81	8.68
137.50	7977	15589	23363	31265	38559	46154	54615	62568	69802	77073
123.75	7147	13960	20923	27998	34529	41330	48902	56035	62496	69012
110.00	6383	12469	18685	25005	30836	36907	43671	50035	55809	61622
96.25	5588	10911	16356	21886	26996	32301	38218	43792	48847	53942
82.50	4811	9393	14076	18838	23239	27801	32892	37692	42032	46429
68.75	4012	7842	11749	15733	19410	23214	27465	31473	35102	38775
55.00	3237	6334	9489	12708	15681	18750	22185	25425	28358	31326
41.25	2435	4773	7149	9579	11821	14126	16712	19159	21367	23606
27.50	1631	3195	4784	6408	7908	9451	11187	12822	14297	15792
13.75	817	1587	2377	3185	3930	4698	5555	6373	7104	7848

Table A.2: Travel time MAPE (in %) using VTL probe data and 0 loop detector sensors. Travel time is computed using the dynamic method. See also Figure 5.8a.

Morning accident MAPE, 0 loops, dynamic tt										
probe rate	VTL/mi									
veh/hr	0.79	1.67	2.54	3.42	4.30	5.18	6.05	6.93	7.81	8.68
137.50	11.34	5.67	5.92	5.18	5.30	5.11	5.27	5.03	4.52	5.05
123.75	10.81	6.12	6.35	5.88	5.13	5.00	5.11	5.00	4.60	4.76
110.00	11.67	6.06	6.34	5.87	5.22	5.32	5.74	5.19	4.60	4.69
96.25	11.73	6.71	6.03	6.29	5.70	5.11	5.52	5.33	4.80	5.11
82.50	12.72	6.39	5.98	5.41	5.73	5.29	5.46	5.37	5.03	4.73
68.75	11.98	6.75	6.41	5.78	6.13	5.23	5.46	5.70	5.36	5.36
55.00	13.28	6.74	6.19	6.46	6.10	5.47	6.49	6.18	5.32	5.79
41.25	14.30	10.21	8.42	7.77	7.41	7.10	6.30	6.35	6.46	6.51
27.50	16.09	10.50	9.30	8.26	7.86	6.99	6.90	6.60	7.09	7.29
13.75	28.90	16.88	15.37	11.34	11.81	11.23	9.91	10.30	10.30	10.06

Table A.3: Travel time MAPE (in %) using VTL probe data and 0 loop detector sensors. Travel time is computed using the dynamic method. See also Figure 5.8b.

Free flow MAPE, 0 loops, dynamic tt										
probe rate	VTL/mi									
veh/hr	0.79	1.67	2.54	3.42	4.30	5.18	6.05	6.93	7.81	8.68
137.50	8.28	6.03	5.91	6.26	5.84	5.79	5.87	5.88	5.79	6.13
123.75	8.53	6.30	5.98	5.80	5.75	5.69	5.96	5.94	5.83	6.03
110.00	8.72	6.25	6.00	5.38	5.51	5.42	5.94	5.59	5.62	5.97
96.25	9.79	6.53	5.95	5.88	5.82	5.81	5.75	5.82	5.65	5.89
82.50	8.94	6.56	6.57	6.39	5.97	6.16	5.90	6.06	5.92	6.09
68.75	12.22	7.28	6.71	6.18	6.31	5.95	6.14	6.11	5.94	6.14
55.00	13.77	7.25	7.21	7.16	6.91	6.65	6.67	6.80	6.39	6.40
41.25	15.43	9.17	8.15	7.16	7.17	7.22	6.89	6.84	7.19	6.80
27.50	16.37	14.48	11.21	9.38	9.10	9.32	8.68	8.31	9.11	8.75
13.75	16.89	16.51	14.02	12.82	11.51	10.54	9.93	9.22	10.06	9.74

Table A.4: Travel time MAPE (in %) using VTL probe data and 0 loop detector sensors. Travel time is computed using the dynamic method. See also Figure 5.8c.

Congestion building MAPE, 0 loops, dynamic tt										
probe rate	VTL/mi									
veh/hr	0.79	1.67	2.54	3.42	4.30	5.18	6.05	6.93	7.81	8.68
137.50	6.10	7.42	4.56	3.56	3.74	3.94	3.67	3.42	3.43	3.76
123.75	6.69	6.93	4.66	3.50	4.06	4.00	3.76	3.39	3.45	3.85
110.00	6.66	6.94	4.59	3.49	4.28	4.02	3.33	3.34	3.37	3.70
96.25	6.67	7.41	5.25	3.85	4.52	4.38	3.51	3.37	3.74	3.86
82.50	7.04	8.25	4.84	3.65	4.71	4.26	3.58	3.40	3.75	3.89
68.75	7.36	8.55	5.40	4.26	4.85	4.68	3.81	3.67	4.05	4.08
55.00	8.35	9.01	5.60	4.86	5.70	5.42	4.14	4.05	4.37	4.18
41.25	10.16	10.57	6.80	5.89	7.57	6.66	5.15	5.01	5.59	5.18
27.50	16.38	14.07	10.34	8.79	9.99	8.70	7.40	6.40	6.95	7.80
13.75	30.18	21.70	16.30	15.05	15.98	13.41	12.34	11.38	11.47	12.47

Table A.5: Travel time MAPE (in %) using VTL probe data and 0 loop detector sensors. Travel time is computed using the dynamic method. See also Figure 5.8d.

Full congestion MAPE, 0 loops, dynamic tt										
probe rate	VTL/mi									
veh/hr	0.79	1.67	2.54	3.42	4.30	5.18	6.05	6.93	7.81	8.68
137.50	5.50	4.97	4.58	5.25	4.21	3.99	4.77	4.54	4.18	4.21
123.75	4.97	5.39	4.36	5.18	4.62	4.15	5.09	4.67	4.31	4.39
110.00	5.18	5.13	4.68	5.59	4.36	4.56	5.20	5.16	4.59	4.79
96.25	5.47	4.74	5.25	6.01	4.31	5.06	5.59	5.52	5.23	4.80
82.50	6.31	5.27	5.27	6.30	4.29	4.92	5.44	5.86	5.08	4.88
68.75	6.24	5.63	5.48	6.35	4.01	4.90	5.54	6.13	5.23	5.20
55.00	7.61	5.82	5.78	7.52	6.11	6.38	6.65	7.68	6.21	7.02
41.25	8.82	7.17	6.45	8.64	7.34	6.98	7.44	7.73	6.80	7.09
27.50	9.51	7.53	6.71	7.42	6.69	6.57	7.02	7.65	6.67	6.54
13.75	14.42	10.90	5.18	7.54	4.35	4.58	4.82	6.50	5.22	4.34

Table A.6: Travel time MAPE (in %) change when 6 loop detector sensors are used. Travel time is computed using the dynamic method. See also Figure 5.9a.

Morning accident change in MAPE, adding 6 loops, dynamic tt										
probe rate	VTL/mi									
veh/hr	0.79	1.67	2.54	3.42	4.30	5.18	6.05	6.93	7.81	8.68
137.50	-2.44	0.22	-1.08	0.72	-0.08	-0.18	-0.11	0.25	-0.22	-0.71
123.75	-2.63	0.13	-1.15	-0.07	-0.14	-0.21	0.14	-0.08	-0.34	0.01
110.00	-3.45	0.62	-1.38	0.12	-0.41	-0.20	-0.46	0.29	-0.09	-0.13
96.25	-3.59	-0.46	-1.13	-0.71	-0.29	-0.32	-0.30	0.27	-0.49	-0.41
82.50	-3.47	0.35	-0.51	0.66	0.46	0.10	-0.37	0.34	-0.45	-0.01
68.75	-3.65	-0.22	-1.22	-0.62	0.33	0.31	0.28	0.42	-0.64	-0.22
55.00	-4.88	-2.11	-0.71	-1.17	-1.28	0.26	-0.85	0.15	-0.29	-0.29
41.25	-4.87	-5.34	-3.08	-3.02	-2.32	-1.83	0.09	-0.97	-1.83	-0.70
27.50	-7.93	-4.75	-3.04	-1.81	-2.31	-0.69	-1.72	0.19	-1.54	-0.87
13.75	-20.84	-10.62	-7.86	-5.22	-5.47	-3.86	-3.28	-3.18	-4.35	-2.67

Table A.7: Travel time MAPE (in %) change when 6 loop detector sensors are used. Travel time is computed using the dynamic method. See also Figure 5.9b.

Free flow change in MAPE, adding 6 loops, dynamic tt										
probe rate	VTL/mi									
veh/hr	0.79	1.67	2.54	3.42	4.30	5.18	6.05	6.93	7.81	8.68
137.50	8.19	0.44	0.27	-0.24	0.63	0.32	0.70	0.59	0.38	0.15
123.75	6.32	0.04	0.53	0.40	0.71	0.20	0.23	0.54	0.15	0.27
110.00	6.79	0.17	0.39	0.92	0.79	0.61	0.34	0.61	0.47	0.26
96.25	5.90	0.19	0.75	0.45	0.75	0.36	0.55	0.89	0.55	0.24
82.50	8.82	0.52	0.14	0.04	0.53	-0.02	0.69	0.47	0.47	0.13
68.75	5.82	-0.20	0.13	0.43	0.47	0.60	0.24	0.29	0.10	0.29
55.00	5.16	-0.31	-0.15	-0.79	-0.45	-0.07	-0.53	-0.08	0.03	0.24
41.25	3.08	-1.99	-1.02	-0.79	-0.48	-0.62	-0.23	-0.25	-0.78	-0.08
27.50	1.05	-6.99	-3.64	-2.85	-1.89	-1.94	-1.24	-1.59	-2.71	-1.96
13.75	1.94	-8.70	-5.85	-5.60	-4.09	-3.10	-3.17	-2.10	-3.10	-2.78

Table A.8: Travel time MAPE (in %) change when 6 loop detector sensors are used. Travel time is computed using the dynamic method. See also Figure 5.9c.

Congestion building change in MAPE, adding 6 loops, dynamic tt										
probe rate	VTL/mi									
veh/hr	0.79	1.67	2.54	3.42	4.30	5.18	6.05	6.93	7.81	8.68
137.50	1.56	0.58	1.40	0.64	1.30	1.07	0.04	0.66	1.02	-0.09
123.75	1.02	0.73	1.71	0.96	1.05	1.09	0.15	0.70	1.16	-0.11
110.00	1.78	0.48	1.64	0.66	1.15	0.98	0.70	0.65	1.34	0.50
96.25	0.78	0.06	0.99	0.46	0.57	0.76	0.73	0.78	1.13	0.41
82.50	1.38	-0.83	0.89	0.93	0.46	0.68	0.63	0.97	1.01	0.45
68.75	0.88	-0.72	0.44	0.22	1.36	0.70	0.91	0.46	1.08	0.63
55.00	0.21	-1.46	1.00	0.75	0.47	0.36	1.32	1.17	1.53	1.22
41.25	-0.84	-2.88	-0.21	0.37	-1.04	-0.83	0.44	0.55	0.52	0.41
27.50	-7.37	-6.70	-3.63	-2.40	-3.19	-2.58	-1.46	-0.47	-0.61	-1.30
13.75	-21.00	-13.57	-9.24	-7.96	-9.02	-6.66	-5.58	-5.03	-4.64	-5.74

Table A.9: Travel time MAPE (in %) change when 6 loop detector sensors are used. Travel time is computed using the dynamic method. See also Figure 5.9d.

Full congestion change in MAPE, adding 6 loops, dynamic tt										
probe rate	VTL/mi									
veh/hr	0.79	1.67	2.54	3.42	4.30	5.18	6.05	6.93	7.81	8.68
137.50	1.33	2.02	0.53	-0.78	1.30	1.05	-0.34	-0.52	0.37	0.44
123.75	1.69	2.46	0.53	-0.70	0.82	1.10	-0.49	-0.54	0.20	0.28
110.00	1.96	2.39	0.34	-0.98	1.28	0.22	-0.47	-0.87	-0.30	-0.19
96.25	1.97	2.95	-0.36	-1.23	1.09	0.08	-0.68	-1.23	-0.49	-0.13
82.50	1.90	3.87	0.04	-1.34	1.44	0.47	-0.45	-0.89	-0.05	0.13
68.75	2.56	3.19	-0.43	-1.13	2.90	0.65	-0.20	-1.18	-0.39	0.02
55.00	1.95	3.49	0.09	-1.59	1.43	-0.60	-1.24	-2.34	-0.23	-2.02
41.25	3.37	4.51	-0.59	-2.31	0.37	-0.42	-1.60	-2.05	-0.53	-1.59
27.50	2.17	4.21	-0.27	-0.49	0.30	-0.10	-0.53	-1.48	-0.31	0.33
13.75	-1.96	2.02	4.93	3.03	7.04	3.77	4.20	1.33	3.35	5.42

Table A.10: Travel time MAPE (in %) using VTL probe data and 0 loop detector sensors. Travel time is computed using the instantaneous method. See also Figure 5.10a.

Morning accident MAPE, 0 loops, instantaneous tt										
probe rate	VTL/mi									
veh/hr	0.79	1.67	2.54	3.42	4.30	5.18	6.05	6.93	7.81	8.68
137.50	9.34	12.77	8.29	14.04	12.43	12.17	13.34	13.69	12.83	12.54
123.75	9.20	11.51	8.33	13.97	12.37	11.55	13.61	13.69	12.52	12.44
110.00	10.99	10.25	8.09	13.67	12.27	10.80	13.57	13.39	12.15	12.30
96.25	10.49	9.03	8.06	12.86	12.59	11.31	12.70	13.73	12.52	12.74
82.50	10.52	9.78	9.35	12.73	13.60	10.95	13.62	13.52	12.32	12.68
68.75	9.39	10.33	7.17	13.30	13.49	11.95	12.97	15.15	13.03	13.49
55.00	15.69	10.12	9.17	14.29	13.08	12.28	15.52	14.21	13.05	14.01
41.25	17.95	15.48	11.51	14.83	14.29	12.30	14.39	13.50	12.46	13.72
27.50	19.11	13.90	13.53	17.01	17.37	14.73	15.10	15.60	15.11	15.23
13.75	28.37	21.72	16.57	19.68	18.25	15.67	17.51	17.11	15.70	17.49

Table A.11: Travel time MAPE (in %) using VTL probe data and 0 loop detector sensors. Travel time is computed using the dynamic method. See also Figure 5.10b.

Morning accident MAPE, 0 loops, dynamic tt										
probe rate	VTL/mi									
veh/hr	0.79	1.67	2.54	3.42	4.30	5.18	6.05	6.93	7.81	8.68
137.50	11.34	5.67	5.92	5.18	5.30	5.11	5.27	5.03	4.52	5.05
123.75	10.81	6.12	6.35	5.88	5.13	5.00	5.11	5.00	4.60	4.76
110.00	11.67	6.06	6.34	5.87	5.22	5.32	5.74	5.19	4.60	4.69
96.25	11.73	6.71	6.03	6.29	5.70	5.11	5.52	5.33	4.80	5.11
82.50	12.72	6.39	5.98	5.41	5.73	5.29	5.46	5.37	5.03	4.73
68.75	11.98	6.75	6.41	5.78	6.13	5.23	5.46	5.70	5.36	5.36
55.00	13.28	6.74	6.19	6.46	6.10	5.47	6.49	6.18	5.32	5.79
41.25	14.30	10.21	8.42	7.77	7.41	7.10	6.30	6.35	6.46	6.51
27.50	16.09	10.50	9.30	8.26	7.86	6.99	6.90	6.60	7.09	7.29
13.75	28.90	16.88	15.37	11.34	11.81	11.23	9.91	10.30	10.30	10.06

Bibliography

[1] `http://pems.eecs.berkeley.edu/`.

[2] A. Alessandri, R. Bolla, and M. Repetto. Estimation of freeway traffic variables using information from mobile phones. In *Proc. American Control Conference the 2003*, volume 5, pages 4089–4094, Denver, CO, June 2003.

[3] B. Anderson and J. Moore. *Optimal filtering.* Prentice-Hall, inc, Englewood Cliffs, N.J., 1979.

[4] V. Astarita and M. Florianz. The use of mobile phones in traffic management and control. In *Proc. IEEE Intelligent Transportation Systems*, pages 10–15, Oakland, CA, August 2001.

[5] A. Aw and M. Rascle. Resurrection of 'second order' models of traffic flow. *SIAM Journal on Applied Mathematics*, 60(3):916–938, 2000.

[6] H. Bar-Gera. Evaluation of a cellular phone-based system for measurements of traffic speeds and travel times: A case study from Israel. *Transportation Research Part C*, 15(6):380–391, 2007.

[7] C. Bardos, A. Y. Leroux, and J. C. Nedelec. First order quasilinear equations with boundary conditions. *Communications in partial differential equations*, 4(9):1017–1034, 1979.

[8] C. Bardos, A. Y. Leroux, and J. C. Nedelec. First order quasilinear equations with boundary conditions. *Communications in partial differential equations*, 4(9):1017–1034, 1979.

[9] S. Blandin, D. Work, P. Goatin, B. Piccoli, and A. A. Bayen. A general phase transition model for vehicular traffic. In press, SIAM Journal on Applied Mathematics, March 2010.

[10] A. Bressan. *Hyperbolic Systems of Conservation Laws: The One-dimensional Cauchy Problem.* Oxford University Press, Oxford, UK, 2000.

[11] G. Burgers, P. Jan van Leeuwen, and G. Evensen. Analysis scheme in the ensemble Kalman filter. *Monthly Weather Review*, 126(6):1719–1724, 1998.

[12] P. Cheng, Z. Qiu, and B. Ran. Particle filter based traffic state estimation using cell phone network data. In *Proc. IEEE Intelligent Transportation Systems Conference ITSC '06*, pages 1047–1052, 2006.

[13] C. Claudel, M. Nahoum, and A. Bayen. Minimal error certificates for detection of faulty sensors using convex optimization. In *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*, pages 1177 –1186, 302009-oct.2 2009.

[14] G. M. Coclite, M. Garavello, and B. Piccoli. Traffic flow on a road network. *SIAM Journal on Mathematical Analysis*, 36(6):1862–1886, 2005.

[15] R. Colombo. Hyperbolic phase transitions in traffic flow. *SIAM Journal on Applied Mathematics*, 63(2):708–721, 2002.

[16] R. Colombo, P. Goatin, and F. Priuli. Global well posedness of traffic flow models with phase transitions. *Nonlinear Analysis: Theory, Methods & Applications*, 66(11):2413 – 2426, 2007.

[17] M. Cremer and M. Papageorgiou. Parameter identification for a traffic flow model. *Automatica*, 17(6):837–843, 1981.

[18] C. M. Dafermos. Polygonal approximation for solutions of the initial value problem for a conservation law. *Journal of mathematical analysis and applications*, 38:33 – 41, 1972.

[19] C. Daganzo, W. Lin, and J. Castillo. A simple physical principle for the simulation of freeways with special lanes and priority vehicles. *Transportation Research Part B*, 31(2):103 – 125, 1997.

[20] 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.

[21] C. F. Daganzo. The cell transmission model, part II: network traffic. *Transportation Research Part B*, 29(2):79–93, 1995.

[22] C. F. Daganzo. Requiem for second-order fluid approximations of traffic flow. *Transportation Research Part B*, 29(4):277 – 286, 1995.

[23] J. Del Castillo, P. Pintado, and F. Benitez. The reaction time of drivers and the stability of traffic flow. *Transportation Research Part B*, 28(1):35–60, 1994.

[24] L. C. Evans. *Partial Differential Equations.* American Mathematical Society, Providence, RI, 1998.

[25] G. Evensen. The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53(4):343–367, 2003.

[26] G. Evensen. *Data Assimilation: The Ensemble Kalman Filter.* Springer-Verlag, Berlin Heidelberg, 2007.

[27] P. Le Floch. Explicit formula for scalar non-linear conservation laws with boudary condition. *Math. Meth. Appl. Sci.*, 10:265–287, 1988.

[28] H. Frankowska. On Le Floch's solutions to the initial boundary value problem for scalar conservation laws. *Journal of hyperbolic differential equations*, 7(3):503 – 543, 2010.

[29] M. Garavello and B. Piccoli. *Traffic Flow on Networks.* American Institute of Mathematical Sciences on Applied Math. Springfield, MO, 2006.

[30] D. Gazis and C. Liu. Kalman filtering estimation of traffic counts for two network links in tandem. *Transportation Research Part B: Methodological*, 37(8):737 – 745, 2003.

[31] J. Glimm. Solutions in the large for nonlinear hyperbolic systems of equations. *Communications on Pure and Applied Mathematics*, 18(4):697–715, 1965.

[32] S. Godunov. A difference method for the numerical calculation of discontinuous solutions of hydrodynamic equations. *Mathematics Sbornik*, 47(3):271–306, 1959.

[33] B.D. Greenshields. A study of traffic capacity. *Highway Research Board*, 14:448–477, 1935.

[34] J.-C. Herrera and A. Bayen. Traffic flow reconstruction using mobile sensors and loop detector data. In 87^{th} *TRB Annual Meeting*, Washington D.C., Jan. 12-17 2008. Transportation Research Board.

[35] J.-C. Herrera and A. Bayen. Incorporation of lagrangian measurements in freeway traffic state estimation. *Transportation Research Part B*, 44(4):460 – 481, 2010.

[36] J.-C. Herrera, D. Work, R. Herring, J. Ban, Q. Jacobson, and A. Bayen. Evaluation of traffic data obtained via GPS-enabled mobile phones: the Mobile Century experiment. *Transportation Research Part C*, 18(4):568–583, 2010.

[37] M. Herty and A. Klar. Modeling, simulation, and optimization of traffic flow networks. *SIAM Journal on Scientific Computing*, 25(3):1066–1087, 2003.

[38] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. Bayen, M. Annavaram, and Q. Jacobson. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *6th International Conference on Mobile Systems, Applications, and Services*, pages 15–28, Breckenridge, CO, June 17-18 2008.

[39] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Enhancing security and privacy in traffic-monitoring systems. *IEEE Pervasive Computing*, 5(4):38–46, 2006.

[40] H. Holden and N.H. Risebro. a mathematical model of traffic flow on a network of unidirectional roads. *SIAM Journal on mathematical analysis*, 26(4):999 – 1017, July 1995.

[41] D. Jacquet, C. Canudas de Wit, and D. Koenig. Traffic control and monitoring with a macroscopic model in the presence of strong congestion waves. In *Proc. of the 44th IEEE Conference on Decision and Control, and European Control Conference*, pages 2164–2169, Sevilla, Spain, 2005.

[42] D. Jacquet, M. Krstic, and C. Canudas de Wit. Optimal control of scalar one-dimensional conservation laws. In *Proc. of the 25th American Control Conference*, pages 5213–5218, Minneapolis, MN, 2006.

[43] S. Julier and J. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[44] J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems.* Springer, New York, NY, 2005.

[45] J. Kaipio and E. Somersalo. Statistical inverse problems: Discretization, model reduction and inverse crimes. *Journal of Computational and Applied Mathematics*, 198(2):493–504, 2007.

[46] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME Journal of Basic Engineering*, 82:35–45, 1960.

[47] A. Krause, E. Horvitz, A. Kansal, and F. Zhao. Toward community sensing. In *IPSN 2008, International Conference on Information Processing in Sensor Networks*, St. Louis, MI, Apr. 2008.

[48] J. Krumm. Inference attacks on location tracks. In *Fifth International Conference on Pervasive Computing (Pervasive 2007)*, Toronto, Ontario, Canada., May 2007.

[49] S. Kruzkov. First order quasilinear equations in several independent variables. *Mathematics of the USSR-Sbornik*, 10(2):217, 1970.

[50] Jaimyoung Kwon, Karl Petty, and Pravin Varaiya. Probe vehicle runs or loop detectors? *Transportation and Research Record*, (2012):57 – 63, 2007.

[51] J.P. Lebacque. The godunov scheme and what it means for first order traffic flow models. In *13th International Symposium on Transportation and Traffic Theory*, pages 647–677, 1996.

[52] J.P. Lebacque. Intersection modeling, application to macroscopic network traffic flow models and traffic management. In *Traffic and Granular Flow 2003*, pages 261–278. Springer Berlin Heidelberg, 2005.

[53] R.J. LeVeque. *Numerical Methods for Conservation Laws.* Birkhäuser Verlag, Basel, Switzerland, 1992.

[54] J. M. Lewis, S. Lakshmivarahan, and S. Dhall. *Dynamic Data Assimilation: A Least Squares Approach.* Cambridge University Press, Cambridge, UK, 2006.

[55] M. Lighthill and G. Whitham. On kinematic waves. II. A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 229(1178):317–345, 1955.

[56] H. Liu, A. Danczyk, R. Brewer, and R. Starr. Evaluation of cell phone traffic data in minnesota. *Transportation Research Record*, (2086):1–7, 2008.

[57] L. Mihaylova and R. Boel. A particle filter for freeway traffic estimation. In *Proc. of the 43rd IEEE Conference on Decision and Control*, volume 2, pages 2106–2111, 2004.

[58] L. Mihaylova, R. Boel, and A. Hegyi. Freeway traffic estimation within recursive bayesian framework. *Automatica*, 43(2):290–300, 2007.

[59] P. Misra and P. Enge. *Global Positioning System: Signals, Measurements and Performance.* Lincoln, MA: Ganga-Jamuna Press, 2 edition, 2006.

[60] H.L. Mitchell, P.L. Houtekamer, and G. Pellerin. Ensemble size, balance, and model–error representation in an ensemble Kalman filter. *Montly Weather Review*, 130:2791–2808, 2002.

[61] O. Oleinik. Discontinuous solutions of non–linear differential equations. *Uspekhi Mat. Nauk*, 12(3):3–73, 1957.

[62] M. Papageorgiou. *Applications of Automatic Control Concepts to Traffic Flow Modeling and Control.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1983.

[63] H. Payne. Models of freeway traffic and control. *Simulation Council Proceedings*, 1:51–61, 1971.

[64] P. I. Richards. Shock waves on the highway. *Operations Research*, 4(1):42–51, 1956.

[65] J. Sau, N.E. El Faouzi, A. Ben Assa, and O. De Mouzon. Particle filter-based real-time estimation and prediction of traffic conditions. *Applied Stochastic Models and Data Analysis*, 12, 2007.

[66] D. Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches.* Wiley-Interscience, 2006.

[67] B. Smith, H. Zhang, M. Fontaine, and M. Green. *Cell phone probes as an ATMS tool.* Number UVACTS-15-5-79. Center for Transportation Studies, University of Virginia, June 2003.

[68] S. Smulders. Control of freeway traffic flow by variable speed signs. *Transportation Research Part B: Methodological*, 24(2):111 – 132, 1990.

[69] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136:4629–4640, 2008.

[70] I. Strub and A. Bayen. Weak formulation of boundary conditions for scalar conservation laws: An application to highway traffic modelling. *Int. J. Robust Nonlinear Control*, 16:733–748, 2006.

[71] X. Sun, L. Munoz, and R. Horowitz. Mixture Kalman filter based highway congestion mode and vehicle density estimator and its application. In *Proc. of the American Control Conference*, volume 3, pages 2098 – 2103, Boston, MA, 2004.

[72] M. Szeto and D. Gazis. Application of kalman filtering to the surveillance and control of traffic systems. *Transportation Science*, 6(4):419–439, 1972.

[73] C. Tampere and L. Immers. An extended Kalman filter application for traffic state estimation using CTM with implicit mode switching and dynamic parameters. In *2007 IEEE Intelligent Transportation Systems Conference*, pages 209 –216, Seattle, WA, Sept. 30–Oct. 3 2007.

[74] C. P. IJ. van Hinsbergen, T. Schreiter, F. S. Zuurbier, J. W. C. van Lint, and H. J. van Zuylen. Fast traffic state estimation with the localized extended kalman filter. In *13th International IEEE Annual Conference on Intelligent Transportation Systems*, pages 917 – 922, Madeira Island, Portugal, September 19–22 2010.

[75] 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.

[76] Y. Wang, M. Papageorgiou, A. Messmer, P. Coppola, A. Tzimitsi, and A. Nuzzolo. An adaptive freeway traffic state estimator. *Automatica*, 45(1):10–24, 2009.

[77] D. Work and A. Bayen. Impacts of the mobile internet on transportation cyber-physical systems: Traffic monitoring using smartphones. In *National Workshop for Research on High-Confidence Transportation Cyber-Physical Systems: Automotive, Aviation, & Rail*, Washington, DC, Nov. 18–20 2008.

[78] D. Work, A. Bayen, and Q. Jacobson. Automotive cyber-physical systems in the context of human mobility. In *National Workshop on High-Confidence Automotive Cyber-Physical Systems*, Troy, MI, April 2008.

[79] D. Work, O.-P. Tossavainen, S. Blandin, A. Bayen, T. Iwuchukwu, and K. Tracton. An ensemble Kalman filtering approach to highway traffic estimation using GPS enabled mobile devices. In *Proc. of the 47th IEEE Conference on Decision and Control*, pages 2141–2147, Cancun, Mexico, December 2008.

[80] J.L. Ygnace, C. Drane, Y.B. Yim, and R. de Lacvivier. *Travel time estimation on the San Francisco Bay area network using cellular phones as probes.* Number UCB-ITS-PWP-2000-18. California PATH Program, Institute of Transportation Studies, University of California, Berkeley, CA, 2000.

[81] Y. Yim and R. Cayford. *Investigation of vehicles as probes using global positioning system and cellular phone tracking: field operational test.* Number UCB-ITS-PWP-2001-9. California PATH Program, Institute of Transportation Studies, University of California, Berkeley, CA, 2001.

[82] H.M. Zhang. A non-equilibrium traffic model devoid of gas-like behavior. *Transportation Research Part B: Methodological*, 36(3):275–290, 2002.

[83] `http://traffic.berkeley.edu/`.