A Distributed Highway Velocity Model for Traffic State Reconstruction

Daniel B. Work, Sébastien Blandin, Olli-Pekka Tossavainen,
Benedetto Piccoli, and Alexandre M. Bayen

Abstract

This article 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 observation the operator is linear. This article 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, which enables the use of a Monte-Carlo based ensemble Kalman filtering data assimilation algorithm. Accuracy of the model and estimation technique is validated on experimental data obtained from a large-scale field experiment.

I. Introduction

A. Motivation

The convergence of communication, sensing, and multimedia platforms such as smartphones provides the engineering community with unprecedented monitoring capabilities. Standard smartphones include numerous sensors (accelerometers, light sensors, GPS), wireless connectivity ports (GSM, GPRS, Wi-Fi, bluetooth, infrared), and ever increasing computational power and memory. The rapid penetration of GPS in phones has enabled the explosion of new *Location Based Services*, heavily relying on spatial and context awareness. 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 [1], [2], which is a rapidly expanding field at the heart of mobile internet services.

With the cellular phone communication infrastructure in place and privacy aware smartphone sensing technology in full expansion [3], a large volume of data from mobile devices is now available [4]. Unlike traditional traffic sensors which typically measure vehicle flows from which vehicle densities can be computed, mobile devices report vehicle speeds or travel times along stretches of roadway. Numerous traffic estimation techniques developed in the literature rely on density based traffic models such as the *Lighthill-Whitham-Richards (LWR) partial differential equation (PDE) [5], [6]* and its discretization using the Godunov scheme [7], [8], [9] (also known as the *Cell Transmission Model (CTM)* [10], [11]

D. Work is a Ph.D. student in Systems Engineering, Department of Civil and Environmental Engineering, University of California, Berkeley, CA (e-mail: dbwork@berkeley.edu). Corresponding author.

S. Blandin is a Ph.D. student in Systems Engineering, Department of Civil and Environmental Engineering, University of California, Berkeley, CA.

O.-P. Tossavainen is a Post Doctoral researcher in Systems Engineering, Department of Civil and Environmental Engineering, University of California, Berkeley, CA, and with the Department of Physics, University of Kuopio, Kuopio, Finland.

B. Piccoli is the Research Director at Istituto per le Applicazioni del Calcolo 'M. Picone', Roma, Italy.

A. Bayen is an Assistant Professor in Systems Engineering, Department of Civil and Environmental Engineering, University of California, Berkeley,CA.

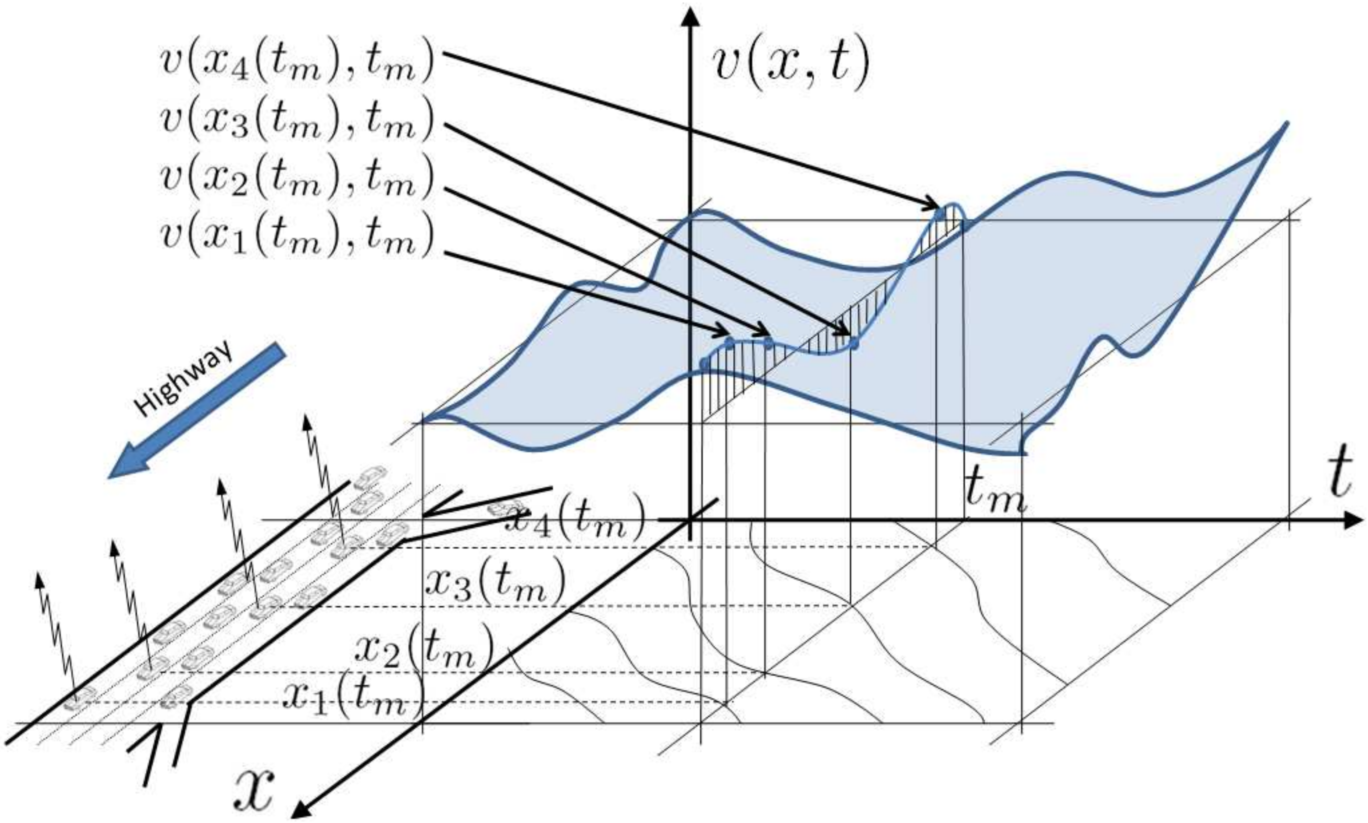

Figure 1. Illustration of the distributed velocity field $v(x, t)$ to be reconstructed from Lagrangian samples. Four samples $v_i(x_i(t), t)$ are shown at $t = t_m$, from vehicles i transmitting their data (indicated by up-arrows above the vehicles).

in the transportation literature). Thus, a key missing piece in creating a real–time system capable of monitoring traffic using mobile phones is a traffic flow model with velocity as the state. This article provides a mathematical approach to address this challenge: it presents a PDE model of traffic, applicable to smartphone collected data. The proposed model is new, and it simplifies the estimation problem when viewed in a state space framework because the state velocity variables are directly observed from the smartphone data.

B. Problem statement: Lagrangian data assimilation for distributed velocity fields

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 (Lagrangian measurements). These discrete measurements are used to reconstruct or estimate the function $v(x, t)$, in a process referred to as *data assimilation* or *inverse modeling* [12]. Fig. 1 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.

Data from mobile devices can be obtained through a variety of sampling strategies, including a new paradigm patented by Nokia, called *Virtual Trip Lines* (VTLs), which act as virtual triggers for mobile sensing [3].

C. Related work

Earlier studies have specifically addressed the traffic flow estimation problem using density evolution models and *Kalman Filtering* (KF) in its various forms. In [13], *Mixture Kalman Filtering* (MKF) was applied to the CTM [10] to estimate traffic densities for ramp metering. The nonlinear CTM was transformed into a switching state space model, which enabled 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). In [14], specific modes of the dynamics presented in [13] are used to incorporate Lagrangian velocity trajectories into an extension of the CTM, called the *Switched Mode Model* (SMM), using Kalman Filtering. A real–time algorithm for traffic estimation based on the *Extended Kalman Filter* (EKF) using a model resulting from the discretization of a PDE system for speed and density was used in [15]. A key ingredient of this work is the differentiability of the numerical scheme employed for the second order model of traffic used by the authors, a feature the model proposed in this work does not possess. Other treatments of traffic estimation include adjoint based control and data assimilation in [16], [17], *Unscented Kalman Filtering* (UKF) in [18] and *Particle Filtering* (PF) in [18], [19], [20].

A common feature for CTM based methods [14] described above is that the evolution of traffic state (typically density, not velocity) relies on a set of linearized equations which are needed in order to use the KF or EKF techniques. On the other hand, the PF technique is a nonlinear scheme for solving the Bayesian update problem, but has a higher computational cost.

Other studies have investigated the highway traffic estimation problem using cell phone tower information. In [21], an EKF was applied to a second order model of vehicle density and velocity, and validated in simulation. In practice, the modeling assumption that network providers can accurately provide both density and flow of the cellular phones currently on the highway of interest is limited, especially in dense and complex roadway networks. The work [22] uses a fully nonlinear Particle Filter to assimilate the mean velocity of a vehicle traveling between cell tower hand-off points, but also suffers from the same practical limitations in dense road networks. On the contrary, the velocity model and estimation procedure proposed in this work are motivated by practical requirements and technical limitations, and were validated in real–time and online with data obtained during a large–scale field experiment.

D. Outline and contribution of the article

This work is organized as follows. We propose a new model for evolution of velocity in the form of a PDE derived from the seminal LWR PDE in Section II-A. We establish the equivalence of the proposed model in the velocity and the density domain for a quadratic flux function (called the Greenshields model) in Section II-B. We prove that this equivalence does not hold for general flux functions, which is a negative result. For general flux functions, we use a transformation of the Godunov scheme which enables us to create a nonlinear discrete dynamical system for velocity evolution, which approximates the *entropy solution* of the LWR PDE in a compact domain (Section II-C). We then instantiate weak boundary conditions explicitly and derive the domain of boundary data for which strong boundary conditions can be prescribed (Section III). We extend the model to a network with the proper use of the strong boundary conditions, using linear programming to compute their values (Section III). The technique used to perform data assimilation with velocity measurements is described in Section IV, which uses an algorithm based on *Ensemble Kalman Filtering* (EnKF). The results of the estimation approach applied to the velocity evolution model are presented using data collected from the *Mobile Century* field experiment in Section V, which ran an earlier version of the algorithm (online and in real–time).

II. Mathematical Model of Traffic Velocity Evolution

A. Preliminaries

This section presents a review of the seminal first order hyperbolic conservation law for density, which serves as a basis for the creation of a class of velocity evolution models. Known as the *Lighthill-Whitham-Richards* (LWR) *partial differential equation* (PDE) *[5], [6]*, the macroscopic traffic flow model which describes the evolution of vehicle density ρ for a stretch of highway of length L over a time T is given as:

$$\frac{\partial \rho(x,t)}{\partial t} + \frac{\partial Q\left(\rho(x,t)\right)}{\partial x} = 0 \quad (x,t) \in (0,L) \times (0,T) \tag{1}$$

where $Q(\cdot)$ is a continuous and piecewise C^1 flux function defined in an interval $[0, \rho_{\max}]$, and $\rho_{\max}$ is the maximal density. The flux function $Q(\cdot)$ expresses the flow of vehicles as a function of the density, and is known as the *fundamental diagram* in the transportation engineering community [10], [11]. For traffic applications, the density flux function is usually concave.

Since transport equations such as (1) involve discontinuities which can appear in finite time, even from smooth initial conditions (see [23]), weak solutions to the density evolution model must be considered.

Definition 1: Weak solution. A weak solution $\rho(\cdot,\cdot)$ of equation (1) with initial condition $\rho_0(\cdot)$ is defined as follows:

$$\int_0^T \int_0^L \left(\rho(x,t) \frac{\partial}{\partial t}\varphi(x,t) + Q(\rho(x,t)) \frac{\partial}{\partial x}\varphi(x,t) \right) dxdt \\ + \int_0^L \rho_0(x)\, \varphi(x,0) dx = 0 \quad \forall \varphi \in C_c^1\left((0,L) \times [0,T)\right) \tag{2}$$

Uniqueness of a weak solution of equation (1) in a compact domain results from the proper formulation of a Cauchy problem, with appropriate initial and (weak) boundary conditions.

Theorem 1: Initial-boundary value problem for conservation laws. Let T, L be positive real numbers, $F(\cdot)$ be a continuous and piecewise C^1 function (the flux function). Then for any initial condition $u_0(\cdot)$ in $BV((0,L))$ [24], the initial-boundary value problem:

$$\begin{cases} \frac{\partial}{\partial t}u(x,t) + \frac{\partial}{\partial x}F\left(u(x,t)\right) = 0 \\ u(x,0) = u_0(x) \end{cases} \tag{3}$$

with an appropriate formulation of the weak boundary conditions has a unique entropy admissible weak solution $u(\cdot,\cdot)$ in $BV((0,L) \times [0,T))$.

Proof: We refer to [25] for a detailed proof. For more insights on weak solutions of scalar conservation laws on bounded domains, see [25], [26]. The interested reader could also refer to [9] for a synthesis on initial boundary value problem for transport equations and a traffic application. ■

1) Boundary conditions: The proper prescription of the boundary conditions for the initial-boundary value problem (3)is described next.

Definition 2: Left weak boundary condition - concave flux function. For a general concave flux function $F(\cdot)$, the proper weak description of the left boundary condition for the LWR PDE (3) was formulated in [26] in terms of the trace of the solution $u(0,t)$ and the left boundary data $u_l(t)$ one wants to apply as:

$$\sup_{k \in D(u(0,t), u_l(t))} \left(sgn\left(u(0,t) - u_l(t)\right) \left(F(u(0,t)) - F(k)\right)\right) = 0 \quad a.e.\ t > 0 \tag{4}$$

where $D(x,y) = [\inf(x,y), \sup(x,y)]$, and $u_l(\cdot)$ is a function of $BV(0,T)$.

More explicitly, the set of boundary data, trace pairs which satisfy (4) can be described similarly as in [9] by:

$$a.e.\ t > 0, \\ \begin{cases} & u(0,t) = u_l(t) \\ \text{xor} & F'(u(0,t)) \le 0 \text{ and } F'(u_l(t)) \le 0 \text{ and } u(0,t) \ne u_l(t) \\ \text{xor} & F'(u(0,t)) \le 0 \text{ and } F'(u_l(t)) > 0 \text{ and } F(u(0,t)) \le F(u_l(t)) \end{cases} \tag{5}$$

Remark 1: The preceding equation (5) is a description of cases for which (4) is satisfied, which is shown graphically in Fig. 2. Note the description is slightly different from [9] in that the sets defined on each line above are mutually exclusive. The first line of (5) corresponds to the case when the trace of

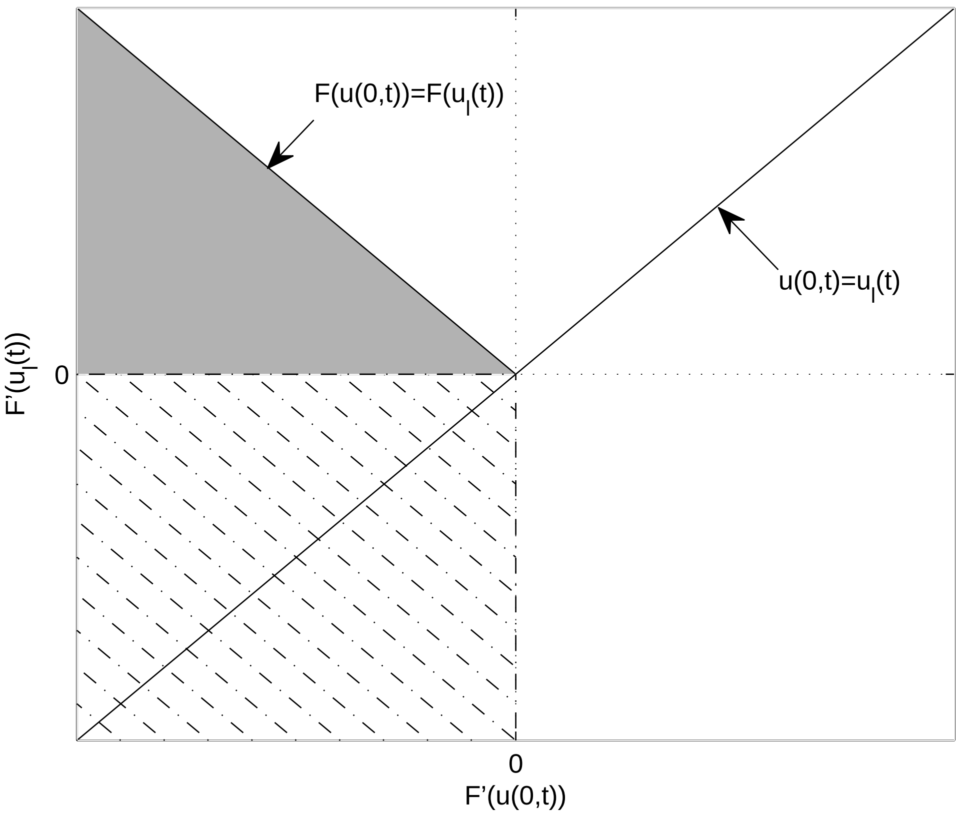

Figure 2. Graphical representation of the left boundary data, trace pairs for a concave flux which satisfy (5). x-axis: Characteristic speed of the trace of the solution $u(0,t)$. y-axis: Characteristic speed of the boundary data $u_l(t)$. The solid line labeled $u(0,t) = u_l(t)$ corresponds to the first line of (19), the dash-dot region corresponds to the second line of (19), and the solid gray region corresponds to the third line of (19). The curve $F(u(0,t)) = F(u_l(t))$ bounding the gray region depends on the choice of $F(\cdot)$, and is drawn as a straight line for illustration purposes. The region in solid white occurs for a set of times t with measure zero.

the solution $u(0,t)$ takes the value of the boundary data $u_l(t)$, 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 (4), but where the value of the trace does not take the value prescribed at the boundary. Finally, the white areas shown in Fig. 2 correspond to a zero measure set of time values for a boundary data, trace pair.

Definition 3: Right weak boundary condition - concave flux function. For a general concave flux function $F(\cdot)$, the the description of the right boundary condition for the LWR PDE (3) can be expressed in terms of the trace of the solution $u(0,t)$ and the right boundary data $u_r(t)$ one wants to apply as:

$$
\begin{aligned}
&a.e.\ t > 0, \\
&\begin{cases}
\quad\ u(L,t) = u_r(t) \\
\text{xor} \quad F'(u(L,t)) \geq 0 \text{ and } F'(u_r(t)) \geq 0 \text{ and } u(L,t) \neq u_r(t) \\
\text{xor} \quad F'(u(L,t)) \geq 0 \text{ and } F'(u_r(t)) < 0 \text{ and } F(u(L,t)) \leq F(u_r(t))
\end{cases}
\end{aligned} \tag{6}
$$

where $u_r(\cdot)$ is a function of $BV(0,T)$.

We now expand on the first line of equation (5) 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 $F(\cdot)$, the cases for strong boundary conditions reads as follows: $a.e.\ t > 0$,

$$
\begin{array}{l}
u(0,t) = u_l(t) \text{ iff} \\
\begin{cases}
\quad\;\; F'(u(0,t)) \geq 0 \text{ and } F'(u_l(t)) \geq 0 \\
\text{xor} \quad F'(u(0,t)) \leq 0 \text{ and } F'(u_l(t)) \leq 0 \text{ and } u\left(0,t\right) = u_l\left(t\right) \\
\text{xor} \quad F'(u(0,t)) \leq 0 \text{ and } F'(u_l(t)) > 0 \text{ and } F\left(u\left(0,t\right)\right) > F(u_l\left(t\right))
\end{cases}
\end{array}
\tag{7}
$$

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

$$
\begin{array}{l}
u\left(L,t\right) = u_r\left(t\right) \text{ iff} \\
\begin{cases}
\quad\;\; F'(u(L,t)) \leq 0 \text{ and } F'(u_r(t)) \leq 0 \\
\text{xor} \quad F'(u(L,t)) \geq 0 \text{ and } F'(u_r(t)) \geq 0 \text{ and } u\left(L,t\right) = u_r\left(t\right) \\
\text{xor} \quad F'(u(L,t)) \geq 0 \text{ and } F'(u_r) < 0 \text{ and } F\left(u\left(L,t\right)\right) > F(u_r\left(t\right))
\end{cases}
\end{array}
\tag{8}
$$

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 for convex flux functions. Beginning with the statement of weak boundary conditions, (5) we can write $a.e.\ t > 0$,

$$
\begin{array}{l}
u(0,t) \neq u_l(t) \text{ iff} \\
\begin{cases}
F'(u(0,t)) \leq 0 \text{ and } F'(u_l(t)) \leq 0 \text{ and } u(0,t) \neq u_l(t) \\
F'(u(0,t)) \leq 0 \text{ and } F'(u_l(t)) > 0 \text{ and } F(u(0,t)) \leq F(u_l(t))
\end{cases}
\end{array}
$$

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

$$
\begin{cases}
\quad\;\; F'(u(0,t)) \geq 0 \text{ and } F'(u_l(t)) \geq 0 \\
\text{xor} \quad F'(u(0,t)) \leq 0 \text{ and } F'(u_l(t)) \leq 0 \text{ and } u\left(0,t\right) = u_l\left(t\right) \\
\text{xor} \quad F'(u(0,t)) \leq 0 \text{ and } F'(u_l(t)) > 0 \text{ and } F\left(u\left(0,t\right)\right) > F(u_l\left(t\right)) \\
\text{xor} \quad F'(u(0,t)) > 0 \text{ and } F'(u_l(t) < 0
\end{cases}
\tag{9}
$$

For the fourth line in (9), $a.e.\ t > 0$ we will have $F'(u(0,t)) = 0$, so it is removed and the conditions for strong left boundary conditions are obtained. ■

2) Velocity inversion: According to the physical definition, flux is a product of density ρ and velocity v: $q = \rho\, v$. In practice, a flux function $Q(\cdot)$ such as the one appearing in (1) is expressed as a function of density only, by assuming that the velocity can be modeled as a function $V(\cdot)$ of density ρ in $[0, \rho_{\max}]$:

$$v = V\left(\rho\right) \tag{10}$$

Under this assumption, the velocity function (10) and the density flux function are formally linked by the following relation:

$$Q\left(\rho\right) = \rho\, V(\rho) \tag{11}$$

The algebraic expression of the form of the velocity function is a modeling choice, and it is typically constructed to fit experimental data. An example of a classical velocity function is given below.

Example 1: Greenshields velocity function [27]. Introduced in 1935, one of the earliest velocity functions considered is the Greenshields affine velocity function:

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

which remains a useful mathematical model because of its simplicity, despite disagreements with observed traffic data, where $v_{\max}$ is the maximum (freeflow) velocity, and $\rho_{\max}$ is the maximum (jam) density.
In order to develop a velocity evolution model from the density evolution equation, we require the velocity function to be invertible. If the velocity function $V\left(\cdot\right)$ is a function of $C^1\left([0, \rho_{\max}]\right)$, and we define I to be the image of $[0, \rho_{\max}]$ through $V\left(\cdot\right)$ (that is, $I = \{y | \exists x \in [0, \rho_{\max}] \text{ s.t. } y = V\left(x\right)\}$), then $V\left(\cdot\right)$ is invertible in I if and only if $\frac{dV(\rho)}{d\rho} \neq 0$ for all $\rho \in [0, \rho_{\max}]$. This is a direct application of the inverse function theorem on a compact one-dimensional domain. Clearly, the Greenshields velocity function is invertible.

In general, the velocity function is more commonly assumed to be continuous but only piecewise C^1, a feature used to model different properties of traffic in free-flow and congestion, such as the capacity drop originally described in [28]. If one considers a velocity function of $C^0([0,\rho_{\max}])$, which is piecewise C^1 on $[0,\rho_{\max}]$, then $V(\cdot)$ is invertible if and only if $V(\cdot)$ is strictly monotonic with the same monotonicity on $[0,\rho_{\max}]$. The proof of the invertibility of these piecewise C^1 velocity functions is a simple result in real analysis.

An example of a piecewise differentiable velocity function is the Daganzo - Newell velocity function.

Example 2: Daganzo - Newell velocity function. The widely used Daganzo-Newell velocity function assumes a constant velocity in free-flow and a hyperbolic velocity function in congestion:

$$v = V_{\text{DN}}(\rho) = \begin{cases} v_{\max} & \text{if } \rho \leq \rho_c \\ -w_f\left(1-\frac{\rho_{\max}}{\rho}\right) & \text{otherwise} \end{cases} \tag{12}$$

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 defined in (12) is not strictly monotonic in freeflow, it cannot be inverted.

When it physically makes sense, (*i.e.* it is possible to associate a density to each velocity), we define the density function $P(\cdot)$ in $[0,v_{\max}]$ as:

$$\rho = P(v) \tag{13}$$

Note that if the velocity function is strictly monotonic with the same monotonicity on $[0,\rho_{\max}]$, the following relation holds on $[0,v_{\max}]$: $P(\cdot) = V^{-1}(\cdot)$, and density flux (11) can be expressed on $[0,v_{\max}]$ as a function of v:

$$Q(\rho) = Q(V^{-1}(v)) = V^{-1}(v)v := \tilde{Q}(v)$$

For example, since the Greenshields velocity function is invertible, its density function is well defined.

Example 3: Greenshields density function. The Greenshields density function is defined by the affine function:

$$\rho = P_G(v) = V_G^{-1}(v) = \rho_{\max}\left(1 - v/v_{\max}\right) \tag{14}$$

Hyperbolic-Linear velocity function. In order to invert the Daganzo-Newell velocity function, we approximate it by replacing the constant free flow velocity profile with a linearly decreasing profile:

$$v = V_{\text{HL}}(\rho) = \begin{cases} v_{\max}\left(1-\frac{\rho}{\rho_{\max}}\right) & \text{if } \rho \leq \rho_c \\ -w_f\left(1-\frac{\rho_{\max}}{\rho}\right) & \text{otherwise} \end{cases} \tag{15}$$

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 density as a function of velocity can now be computed by:

$$\rho = V_{\text{HL}}^{-1}(v) = \begin{cases} \rho_{\max}\left(1-\frac{v}{v_{\max}}\right) & \text{if } v(x,t) \geq v_c \\ \rho_{\max}\left(\frac{1}{1+\frac{v}{w_f}}\right) & \text{otherwise} \end{cases} \tag{16}$$

where v_c is the critical velocity: $v_c = V(\rho_c)$. This hyperbolic-linear velocity function yields a quadratic-linear flux function as illustrated in figure 3.

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

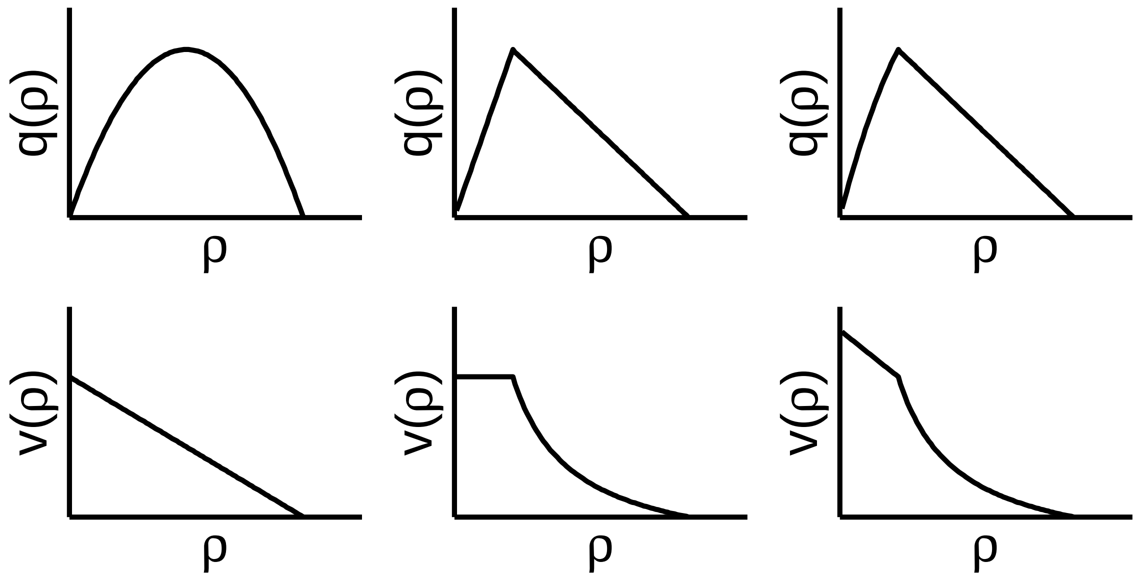

Figure 3. Fundamental diagrams (top row) and velocity functions (bottom row) for Greenshields (left), Daganzo-Newell (center), and quadratic-linear (left).

B. 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, the PDE (2) in weak form for $\rho(\cdot,\cdot)$ is equivalent to the following formulation for $v(\cdot,\cdot)$:

$$\int_0^T \int_0^L \left(P(v(x,t))\,\frac{\partial \varphi}{\partial t}(x,t) + Q(P(v(x,t)))\frac{\partial \varphi}{\partial x}(x,t) \right) dx dt \\ + \int_0^L P(v_0(x))\,\varphi(x,0) dx = 0 \quad \forall \varphi \in C_c^1\left((0,L)\times[0,T)\right) \tag{17}$$

This results from the substitution of expression (13) in equation (2).

In order to use existing numerical analysis schemes for the PDE we want to obtain, we would like to transform the weak formulation (17) into the following conservation law for velocity with initial condition $v_0(\cdot)$:

$$\begin{cases} \frac{\partial}{\partial t} v(x,t) + \frac{\partial}{\partial x} R\left(v(x,t)\right) = 0 \\ v(x,0) = v_0(x) \end{cases} \tag{18}$$

By analogy with the classical LWR equation, the velocity PDE (18) is called LWR-v PDE. Because the flux function $R(v)$ in the velocity conservation law (18) is convex, the weak boundary conditions are given as follows:

Definition 4: Weak boundary conditions- Convex flux function [26]. For a convex flux function $F(\cdot)$, the weak formulation of boundary conditions reads:

$$
\begin{aligned}
&a.e.\ t > 0,\\
&\begin{cases}
\quad u(0,t) = u_l(t) \\
\text{xor} \quad F'(u(0,t)) \leq 0 \text{ and } F'(u_l(t)) \leq 0 \text{ and } u(0,t) \neq u_l(t) \\
\text{xor} \quad F'(u(0,t)) \leq 0 \text{ and } F'(u_l(t)) > 0 \text{ and } F(u(0,t)) \geq F(u_l(t))
\end{cases}
\end{aligned}
\tag{19}
$$

and

$$
\begin{aligned}
&a.e.\ t > 0,\\
&\begin{cases}
\quad u(L,t) = u_r(t) \\
\text{xor} \quad F'(u(L,t)) \geq 0 \text{ and } F'(u_r(t)) \geq 0 \text{ and } u(L,t) \neq u_r(t) \\
\text{xor} \quad F'(u(L,t)) \geq 0 \text{ and } F'(u_r(t)) < 0 \text{ and } F(u(L,t)) \geq F(u_r(t))
\end{cases}
\end{aligned}
\tag{20}
$$

where $u_l(\cdot)$, $u_r(\cdot)$ are functions of $BV(0,T)$. The functions $u_l(\cdot)$ and $u_r(\cdot)$ 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 (17) is equivalent to system (18) if and only if the velocity function is affine (Greenshields case).

Proof: The proof proceeds in two steps. Beginning with equation (17) instantiated for the Greenshields density function (14), we show that the conservative equation obtained is the one from system (18). Substitution of the explicit expression of P_G in (17) yields:

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

where $Q_G(\rho) = \rho\, V_G(\rho)$. Since $\varphi \in C_c^1((0,L) \times [0,T))$ 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:

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

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

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

This completes the first part of the proof.

Now, we show that the Rankine-Hugoniot jump condition [8], [29] is not conserved in the transformation from (1) to (18) for the general case, which means that the equivalence is not obtained for general flux functions. A necessary condition to have equivalence between the LWR PDE (1) and the LWR-v PDE (18) is to have the same characteristics speeds for a state ρ in (1) and for the state $V(\rho)$ in (18). This yields $Q'(P(v)) = R'(v)$. Integrating this relation between any states (ρ_1, v_1) and (ρ_2, v_2) we obtain:

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

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

$$\int_{\rho_1}^{\rho_2} Q'(\rho)\, V'(\rho)d\rho = \int_{v_1}^{v_2} R'(v)dv \tag{21}$$

The Rankine-Hugoniot jump condition [8], [29] reads:

$$\frac{Q(\rho_2) - Q(\rho_1)}{\rho_2 - \rho_1} = \frac{R(v_2) - R(v_1)}{v_2 - v_1} \tag{22}$$

which we can rewrite as:

$$\int_{v_1}^{v_2} R'(v)dv = \frac{v_2 - v_1}{\rho_2 - \rho_1} \int_{\rho_1}^{\rho_2} Q'(\rho)d\rho \tag{23}$$

If we substitute equality (21) equation 23 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:

$$\int_{\rho_1}^{\rho_2} V'(\rho)\,(V(\rho) \;+\; \rho\, V'(\rho))d\rho \quad = \quad \left(\frac{1}{\rho_2 - \rho_1} \int_{\rho_1}^{\rho_2} V'(\rho)d\rho\right) \left(\int_{\rho_1}^{\rho_2} (V(\rho) + \rho\, V'(\rho))d\rho\right) \tag{24}$$

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:

$$V'(\rho_2)\,(V(\rho_2) + \rho_2\, V'(\rho_2)) = G'_{\rho_1}(\rho_2)\,(\rho_2\, V(\rho_2) - \rho_1\, V(\rho_1)) + G_{\rho_1}(\rho_2)\,(V(\rho_2) + \rho_2\, V'(\rho_2)) \tag{25}$$

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 (25) and if we replace $G_{\rho_1}(\rho_2) - V'(\rho_2)$ by $-(\rho_2 - \rho_1)\, G'_{\rho_1}(\rho_2)$ we obtain:

$$G'_{\rho_1}(\rho_2)\ ((\rho_2\, V(\rho_2) - \rho_1\, V(\rho_1)) - (\rho_2 - \rho_1)\,(V(\rho_2) + \rho_2\, V'(\rho_2))) = 0 \tag{26}$$

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 (26) must be zero. If the first term in (26) 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 is 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 (22) 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). ∎

C. Numerical approximation of the solution

The LWR-v PDE (18) can be discretized using the Godunov discretization scheme [30] to construct a nonlinear discrete time dynamical system [31]. The Godunov scheme computes an approximation of the weak solution to the PDE in conservative form in discrete time and space. Because of the equivalence of the solution of (17) and (18), the Godunov discretization and the velocity inversion commute, which is not the case for general flux functions.

Remark 2: For the case when the velocity function is not affine, the discrete velocity model must be constructed by applying the Godunov scheme directly to the LWR PDE, then applying the velocity inversion. Note that the order in which the operations are done is important, and that inversion before discretization for non-affine velocity functions would not lead to the solution of (17) [23].

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 $i \in \{0, i_{\max}\}$. Given the LWR PDE (1), application of the Godunov discretization scheme yields the following relation for the time evolution of the discretized solution of (1):

$$\rho_i^{n+1} = \rho_i^n - \frac{\Delta T}{\Delta x}\left(G\left(\rho_i^n, \rho_{i+1}^n\right) - G\left(\rho_{i-1}^n, \rho_i^n\right)\right) \tag{27}$$

In the above equation, ρ_i^n denotes the value of the computed solution at time step n and space step i. The Godunov flux $G\left(\rho_1, \rho_2\right)$ is defined as:

$$G\left(\rho_1, \rho_2\right) = \begin{cases} Q(\rho_2) & \text{if } \rho_c \leq \rho_2 \leq \rho_1 \\ Q(\rho_c) & \text{if } \rho_2 \leq \rho_c \leq \rho_1 \\ Q(\rho_1) & \text{if } \rho_2 \leq \rho_1 \leq \rho_c \\ \min\left(Q(\rho_1), Q(\rho_2)\right) & \text{if } \rho_1 \leq \rho_2 \end{cases} \tag{28}$$

In order to ensure numerical stability, the time and space steps are coupled by the CFL condition [8]: $\alpha_{\max}\frac{\Delta T}{\Delta x} \leq 1$ where $\alpha_{\max}$ denotes the maximal characteristic speed. This discrete model is commonly referred to as the Cell Transmission Model in the transportation engineering community [10], [11].

Note that if $\rho_1 \leq \rho_2$, with $v_1 = V(\rho_1)$ and $v_2 = V(\rho_2)$, then $v_1 \geq v_2$ when $V(\cdot)$ is monotonically decreasing (which is typically the case for traffic applications). Furthermore, since $V(\cdot)$ is invertible, from (11), we obtain the following relationship: $Q(\rho) = V^{-1}(v)v$. Finally, application of the inversion to (27) and (28) yields the *Cell Transmission Model for velocity* (CTM-v):

$$v_i^{n+1} = V\left(V^{-1}\left(v_i^n\right) - \frac{\Delta T}{\Delta x}\left(\tilde{G}\left(v_i^n, v_{i+1}^n\right) - \tilde{G}\left(v_{i-1}^n, v_i^n\right)\right)\right) \tag{29}$$

where the transformed Godunov velocity flux $\tilde{G}\left(v_1, v_2\right)$ is given by:

$$\tilde{G}\left(v_1, v_2\right) = \begin{cases} \tilde{Q}\left(v_2\right) & \text{if } v_c \geq v_2 \geq v_1 \\ \tilde{Q}\left(v_c\right) & \text{if } v_2 \geq v_c \geq v_1 \\ \tilde{Q}\left(v_1\right) & \text{if } v_2 \geq v_1 \geq v_c \\ \min\left(\tilde{Q}\left(v_1\right),\ \tilde{Q}\left(v_2\right)\right) & \text{if } v_1 \geq v_2 \end{cases} \tag{30}$$

Example 4: Hyperbolic-linear model. After evaluation of the function (16), equation (30) reduces to:

$$\tilde{G}\left(v_1, v_2\right) = \begin{cases} v_2\rho_{\max}\left(\frac{1}{1+\frac{v_2}{w_f}}\right) & \text{if } v_c \geq v_2 \geq v_1 \\ v_c\rho_{\max}\left(1 - \frac{v_c}{v_{\max}}\right) & \text{if } v_2 \geq v_c \geq v_1 \\ v_1\rho_{\max}\left(1 - \frac{v_1}{v_{\max}}\right) & \text{if } v_2 \geq v_1 \geq v_c \\ \min\left(V_{\text{HL}}^{-1}\left(v_1\right)v_1,\ V_{\text{HL}}^{-1}\left(v_2\right)v_2\right) & \text{if } v_1 \geq v_2 \end{cases} \tag{31}$$

We choose not to simplify the last equation in (31) due to the piecewise analytical expression of function $V_{\text{HL}}^{-1}(\cdot)$.
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 $v_{i_{\max}}^n$ are well defined by (29) and (31). At these boundaries the equations:

$$\begin{aligned} v_0^{n+1} &= V\left(V^{-1}\left(v_0^n\right) + \frac{\Delta T}{\Delta x}\left(\tilde{G}\left(v_0^n, v_1^n\right) - \tilde{G}\left(v_{-1}^n, v_0^n\right)\right)\right) \\ v_{i_{\max}}^{n+1} &= V\left(V^{-1}\left(v_{i_{\max}}^n\right) + \frac{\Delta T}{\Delta x}\left(\tilde{G}\left(v_{i_{\max}}^n, v_{i_{\max}+1}^n\right) - \tilde{G}\left(v_{i_{\max}-1}^n, v_{i_{\max}}^n\right)\right)\right) \end{aligned} \tag{32}$$

contain references to the ghost points 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 (31), which causes the boundary conditions to be implemented in the weak sense.

III. Extension of the Model to Networks

A. *Network model and edge boundary conditions at junctions*

We now show how to extend the velocity model to road networks in the presence of shocks and weak boundary conditions. This extension is addressed in the literature for density traffic models in the transportation engineering community using physical principles in [11], and also in a mathematical context in [32].

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 $\mathcal{J}_j := (\nu_j,\ I_j,\ \mathcal{O}_j) \subseteq \mathcal{V} \times \mathcal{E} \times \mathcal{E}$, 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 $e_{\text{out}} \in \mathcal{O}_j$. On each edge, the velocity field evolves according to (29), 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 links, so that the velocity field can be evolved across the network.

We look for unique description of the evolution of the velocity dynamics at the junctions. Following the conditions for uniqueness of [32], 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.

1) 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:

$$\sum_{e_{\text{in}} \in \mathcal{I}_j} \tilde{Q}_{e_{\text{in}}}\left(v_{e_{\text{in}}}\left(L_{e_{\text{in}}}, t\right)\right) = \sum_{e_{\text{out}} \in \mathcal{O}_j} \tilde{Q}_{e_{\text{out}}}\left(v_{e_{\text{out}}}\left(0, t\right)\right) \tag{33}$$

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. 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:

$$\sum_{e_{\text{out}} \in \mathcal{O}_j} \alpha_{e_{\text{out}}, e_{\text{in}}} = 1 \tag{34}$$

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 outgoing edge, then the physical constraint $\sum_{e_{\text{in}} \in \mathcal{I}} \alpha_{e_{\text{out}}, e_{\text{in}}} \tilde{Q}_{e_{\text{in}}} = \tilde{Q}_{e_{\text{out}}}$ 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 $\left(\sum_{e_{\text{in}} \in \mathcal{I}} \alpha_{e_{\text{out}}, e_{\text{in}}} \tilde{Q}_{e_{\text{in}}},\ \tilde{Q}_{e_{\text{out}}}\right)$, so the statement of strong boundary conditions ((7) and (8) 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 exiting fluxes (denoted by the vector dummy variable $\xi \in \mathbb{R}^{|\mathcal{I}|}$) on the incoming edges e_{in} for junction j:

$$\begin{array}{ll} \textbf{max:} & 1^T \xi \\ \textbf{s.t.}: & A_j \xi \leq \gamma^{\max}_{\mathcal{O}_j} \\ & \mathbf{0} \leq \xi \leq \gamma^{\max}_{\mathcal{I}_j} \end{array} \tag{35}$$

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 (35), 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 (32) are given by:

$$\tilde{G}_{e_{\text{in}}}\left(v^n_{i_{\max}}, v^n_{i_{\max}+1}\right) = \xi^*_{e_{\text{in}}}, \quad \tilde{G}_{e_{\text{out}}}\left(v^n_{-1}, v^n_0\right) = \sum_{e_{\text{in}} \in \mathcal{I}_j} \alpha_{e_{\text{out}}, e_{\text{in}}} \xi^*_{e_{\text{in}}} \tag{36}$$

Remark 3: We note that the solution to this linear program is not always unique. In fact, for some instantiations of A_j, the gradiant 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 in many cases by adding some noise to the coefficients of A_j. A second problem can occur when the maximum flow on an outgoing edge is an active constraint in the solution. When this occurs, the linear program must be augmented with additional priority constraints which describe how the flux from the incoming edges share the limited outgoing capacity. For more information on resolving the nonuniqueness of solutions to (35), the reader is referred to [32].

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

$$F(\tau(u)) = F(u) \quad \forall\ u \in [0, u_{\max}]$$

$$\tau(u) \neq u \quad \forall u \in [0, u_{\max}] \backslash \{u_c\}$$

Given that $F(\cdot)$ is in $C^0([0,u_{\max}])$, strictly increasing in $[0,u_c)$ and strictly decreasing in $(u_c,u_{\max}]$ the following holds:

$$0 \le u \le u_c \Leftrightarrow u_c \le \tau(u) \le u_{\text{max}}$$

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 $F(\cdot)$ with a maximum obtained at the critical value u_c we categorize the values of $u(0,\cdot)$ and $u_l(\cdot)$ for which which (7) holds:

$$\begin{aligned} &a.e.\ t>0,\ u(0,t)=u_l(t) \text{ iff} \\ &\begin{cases} & u(0,t)\in[0,u_c] \text{ and } u_l(t)\in[0,u_c] \\ \text{xor} & u(0,t)\in(u_c,u_{\text{max}}] \text{ and } u_l(t)\in[0,\tau(u(0,t)))\cap\{u(0,t)\} \end{cases} \end{aligned} \tag{37}$$

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

$$\delta_{e_{\text{out}}}\left(\rho_{e_{\text{out}}}(0,t)\right)\in\Pi_{e_{\text{out}}}\ \left(\rho_{e_{\text{out}}}(0,t)\right) := \left\{\hat{Q}:\ \exists\hat{\rho}\in[0,\rho_{c,e_{\text{out}}}];\ \hat{Q}=Q\left(\hat{\rho}\right)\right\} \tag{38}$$

  where $\rho_{c,e_{\text{out}}}$ is the critical density on the edge e_{out}.
- For $\rho_{e_{\text{out}}}(0,t)\in[\rho_{c,e_{\text{out}}},\rho_{\text{max},e_{\text{out}}}]$:

$$\begin{aligned} &\delta_{e_{\text{out}}}\left(\rho_{e_{\text{out}}}(0,t)\right)\in\Pi_{e_{\text{out}}}\ \left(\rho_{e_{\text{out}}}(0,t)\right) := \\ &\quad\left\{\hat{Q}:\ \exists\hat{\rho}\in\{\rho_{e_{\text{out}}}(0,t)\}\cup[0,\tau\left(\rho_{e_{\text{out}}}(0,t)\right));\hat{Q}=Q\left(\hat{\rho}\right)\ \right\} \end{aligned} \tag{39}$$

Similarly, (8) can be rewritten in terms of outgoing admissible fluxes for incoming edges as:

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

$$\begin{aligned} &\delta_{e_{\text{in}}}\left(\rho_{e_{\text{in}}}(L_{e_{\text{in}}},t)\right)\in\Pi_{e_{\text{in}}}\left(\rho_{e_{\text{in}}}(L_{e_{\text{in}}},t)\right) := \\ &\quad\left\{\hat{Q}:\ \exists\hat{\rho}\in\{\rho_{e_{\text{in}}}(L_{e_{\text{in}}},t)\}\cup(\tau(\rho_{e_{\text{in}}}(L_{e_{\text{in}}},t),\rho_{\text{max},e_{\text{in}}}];\hat{Q}=Q\left(\hat{\rho}\right)\right\} \end{aligned} \tag{40}$$

  where $\rho_{\text{max},e_{\text{in}}}$ is the maximum density on the edge e_{in}.
- For $\rho_{e_{\text{in}}}(L_{e_{\text{in}}},t)\in[\rho_{c,e_{\text{in}}},\rho_{\text{max},e_{\text{in}}}]$:

$$\delta_{e_{\text{in}}}\left(\rho_{e_{\text{in}}}(L_{e_{\text{in}}},t)\right)\in\Pi_{e_{\text{in}}}\left(\rho_{e_{\text{in}}}(L_{e_{\text{in}}},t)\right) := \left\{\hat{Q}:\ \exists\hat{\rho}\in[\rho_{c,e_{\text{in}}},\rho_{\text{max},e_{\text{in}}}];\ \hat{Q}=Q(\hat{\rho})\right\} \tag{41}$$

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

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

and

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

which are the upper bounds used in (35).

Example 5: Maximum admissible flux - Hyperbolic-linear model. The maximum outgoing admissible flux is given as:

$$\gamma_{e_{\text{out}}}^{\max}\left(v_{e_{\text{out}}}(0,t)\right) = \begin{cases} \rho_{\max}\left(1 - \frac{v_{c,e_{\text{out}}}}{v_{\max}}\right) v_{c,e_{\text{out}}} & \text{if } v_{e_{\text{out}}}(0,t) \in [v_{c,e_{\text{out}}}, v_{\max,e_{\text{out}}}] \\ \rho_{\max}\left(\frac{1}{1+\frac{v_{e_{\text{out}}}(0,t)}{w_f}}\right) v_{e_{\text{out}}}(0,t) & \text{if } v_{e_{\text{out}}}(0,t) \in [0, v_{c,e_{\text{out}}}] \end{cases} \tag{42}$$

and the maximum incoming admissible flux is given as:

$$\gamma_{e_{\text{in}}}^{\max}\left(v_{e_{\text{in}}}(L_{e_{\text{in}}},t)\right) = \begin{cases} \rho_{\max}\left(1 - \frac{v_{e_{\text{in}}}(L_{e_{\text{in}}},t)}{v_{\max}}\right) v_{e_{\text{in}}}(L_{e_{\text{in}}},t) & \text{if } v_{e_{\text{in}}}(L_{e_{\text{in}}},t) \in [v_{c,e_{\text{in}}}, v_{\max,e_{\text{in}}}] \\ \rho_{\max}\left(\frac{1}{1+\frac{v_{c,e_{\text{in}}}}{w_f}}\right) v_{c,e_{\text{in}}} & \text{if } v_{e_{\text{in}}}(L_{e_{\text{in}}},t) \in [0, v_{c,e_{\text{in}}}] \end{cases} \tag{43}$$

B. 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 (35) posed earlier. 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 $i = 0$ to $i = i_{\max}$ 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:

$$v^{n+1} = \mathcal{M}[v^n] \tag{44}$$

where $\mathcal{M}[\cdot]$ denotes the following update algorithm:

1) For all junctions $j \in \mathcal{J}$:
   a) Compute $\gamma_{i_{\max},e_{\text{in}}}^n\left(v_{i_{\max},e_{\text{in}}}^n\right)$ $\forall e_{\text{in}} \in \mathcal{I}_j$, and $\gamma_{0,e_{\text{out}}}^n\left(v_{0,e_{\text{out}}}^n\right)$ $\forall e_{\text{out}} \in \mathcal{O}_j$ using (42) and (43).
   b) Solve the LP (35) for ξ^*, and update $\tilde{G}_{e_{\text{in}}}\left(v_{i_{\max}}^n, v_{i_{\max}+1}^n\right)$ and $\tilde{G}_{e_{\text{out}}}\left(v_{-1}^n, v_0^n\right)$ through (36).
2) For all edges $e \in \mathcal{E}$: Compute $v_{i,e}^{n+1}$ $\forall i \in \{1, i_{\max}\}$ according to the CTM-v (29) and (32).

IV. Velocity Estimation

The goal of this section is to build an estimator to reconstruct the evolution of the velocity field on the highway. That is, we wish to estimate the velocity field v^n on the network at each time step n using velocity data obtained from the mobile devices.

A. State–space model

Given the velocity field at all points on the network at time $n\Delta t$, the velocity at time $(n+1)\Delta T$ is constructed using the CTM-v algorithm $v^{n+1} = \mathcal{M}[v^n]$, which is given by the CTM-v network algorithm in section III-B. 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). If we operate on the CTM-v model, rather than the CTM model, the observations of the state (*i.e.* the velocity measurements from mobile devices) can be modeled with a linear observation operator, which simplifies the estimation problem. For estimation purposes, we extend the model to

$$v^n = \mathcal{M}[v^{n-1}] + \eta^n \tag{45}$$

where $\eta^n \sim (0, \mathbf{Q}^n)$ is the Gaussian zero-mean, white state noise with covariance $\mathbf{Q}^n$.

A network observation model is given by:

$$y^n = \mathbf{H}^n v^n + \chi^n \tag{46}$$

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 $\kappa = \sum_{e\in\mathcal{E}}(i_{\text{max},e}+1)$ is the corresponding (total) number of cells in the network. The last term in expression (46) is the white, zero mean observation noise $\chi^n \sim (0, \mathbf{R}^n)$ with covariance matrix $\mathbf{R}^n$.

B. Extended Kalman Filtering for nonlinear systems

If equation (29) was differentiable in v^n, so would be the operator $\mathcal{M}[\cdot]$ in (45), in which case the optimal estimate for the state v^n could be obtained using the following traditional equations known as the Extended Kalman Filter:

- Forecast step (Time-update):

$$\begin{aligned} v_f^n &= \mathcal{M}[v_a^{n-1}] \\ \mathbf{P}_f^n &= \mathcal{M}_L^{n-1}\mathbf{P}_a^{n-1}\left(\mathcal{M}_L^{n-1}\right)^T + \mathbf{Q}^{n-1} \end{aligned} \tag{47}$$

  where $\mathcal{M}_L$ is the Jacobian matrix of mapping $\mathcal{M}$ (also known as the *tangent linear model*) defined as

$$\mathcal{M}_L^{n-1}(i,j) = \frac{\partial \mathcal{M}_i[v_a^{n-1}]}{\partial v_j^n} \tag{48}$$

- Analysis step (Measurement-update):

$$v_a^n = v_f^n + \mathbf{G}^n\left(y^n - \mathbf{H}^n v_f^n\right) \tag{49}$$

$$\mathbf{P}_a^n = \mathbf{P}_f^n - \mathbf{G}^n\mathbf{H}^n\mathbf{P}_f^n \tag{50}$$

$$\mathbf{G}^n = \mathbf{P}_f^n\left(\mathbf{H}^n\right)^T\left(\mathbf{H}^n\mathbf{P}_f^n\left(\mathbf{H}^n\right)^T + \mathbf{R}^n\right)^{-1} \tag{51}$$

  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 $\mathbf{P}_a^0 = \mathbf{P}^0$.

C. Ensemble Kalman Filter

The Ensemble Kalman Filter was introduced by Evensen in [33] 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 in EKF, it is assumed that discarding the higher order moments from the evolution of the error covariance in (47) yields a good approximation. In cases in which this linearization approximation is invalid, it can cause an unbounded error variance growth [33]. 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 [34] 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.

EnKF also comes with a relatively low numerical cost. Namely, usually a rather limited number of ensemble members is needed to achieve a reasonable statistical convergence [34].

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 [34], 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 [33].

The ensemble Kalman Filter algorithm can be summarized as follows [33], [34]:

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 (45) forward simulation algorithm. Then update the ensemble mean and covariance according to:

$$v_f^n(k) = \mathcal{M}[v_a^{n-1}(k)] + \eta^n(k) \tag{52}$$

$$\bar{v}_f^n = \frac{1}{K}\sum_{k=1}^{K} v_f^n(k) \tag{53}$$

$$\mathbf{P}_{\text{ens},f}^n = \frac{1}{K-1}\sum_{k=1}^{K} \left(v_f^n(k) - \bar{v}_f^n\right)\left(v_f^n(k) - \bar{v}_f^n\right)^T \tag{54}$$

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

$$\mathbf{G}_{\text{ens}}^n = \mathbf{P}_{\text{ens},f}^n \left(\mathbf{H}^n\right)^T \left(\mathbf{H}^n \mathbf{P}_{\text{ens},f}^n \left(\mathbf{H}^n\right)^T + \mathbf{R}^n\right)^{-1} \tag{55}$$

$$v_a^n(k) = v_f^n(k) + \mathbf{G}_{\text{ens}}^n \left(y_{\text{meas}}^n - \mathbf{H}^n v_f^n(k) + \chi^n(k)\right) \tag{56}$$

4) Return to 2.

In (56), an important step is that at measurement times, each measurement is represented by an ensemble. 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 a variance that is not too low [34].

1) 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 (54) to be computed for the entire highway network, then used for computing the Kalman gain in (55). When operating on large scale networks such as the San Francisco Bay Area, CA, the covariance matrix can easily require more than 2 GB of memory to load, 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. [35]).

For this large scale traffic network estimation problem, localization also provides a computationally efficient way to update the state variables at the measurement update time in (55)–(56). Namely, due to the localization, the computation of the covariance matrix in (54) 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 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 (54)-(56) 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:

$$\mathbf{P}^n_{\text{ens},f,e} = \frac{1}{K-1}\sum_{k=1}^{K} \mathcal{L}_e\left(v^n_f(k) - \bar{v}^n_f\right) \times \left(\mathcal{L}_e(v^n_f(k) - \bar{v}^n_f)\right)^T \quad (57)$$

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:

$$\mathbf{G}^n_{\text{ens},e} = \mathbf{P}^n_{\text{ens},f,e}\left(\mathbf{H}^n_e\right)^T \times \left(\mathbf{H}^n_e \mathbf{P}^n_{\text{ens},f,e}\left(\mathbf{H}^n_e\right)^T + \mathbf{R}^n_e\right)^{-1} \quad (58)$$

$$v^n_{a,e}(k) = \mathcal{L}_e\left(v^n_f(k)\right) + \mathbf{G}^n_{\text{ens},e}\left(y^n_{\text{meas},e} - \mathbf{H}^n_e v^n_f(k) + \chi^n_e(k)\right) \quad (59)$$

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.

V. Experimental Results

A. *Mobile Century case study (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 San Francisco, CA. With the help of 165 UC Berkeley students, 100 vehicles carrying Nokia N95 phones drove repeated loops of six to ten miles in length continuously for eight hours. These vehicles represented approximately 2% to 5% of the total volume of traffic on the main line of the highway during the experiment.

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. The section is also covered with existing loop detectors feeding into the PeMS system [36], which are used to assess the quality of the EnKF estimates.

The network implemented for the results presented in this article is a 7 mile stretch of I-880 northbound from the Decoto Rd. entrance ramp (south end), to the Winton Ave. exit ramp (north end). The network model consists of 13 edges and 14 junctions (6 exit ramps, 7 entrance ramps, and one lane drop). A total of 40 VTLs were placed on this highway segment with an average spacing of 0.17 miles.

At approximately 10:30 am, a multiple car accident created significant unanticipated congestion for northbound traffic south of CA-92 (see Fig. 4). An earlier version of the EnKF algorithm, running in real-time during the experiment, detected the accident's resulting bottleneck and corresponding shock wave [31]. It broadcast the speed contour of the highway and the resulting congestion in real time [37]. In Figs. 5-8, we present a comparison of the velocity estimate from the EnKF CTM-v algorithm using VTL data only with the velocity estimate obtained from the PeMS system [38], which provides loop detector data for the deployment area and serves as benchmark for this method.

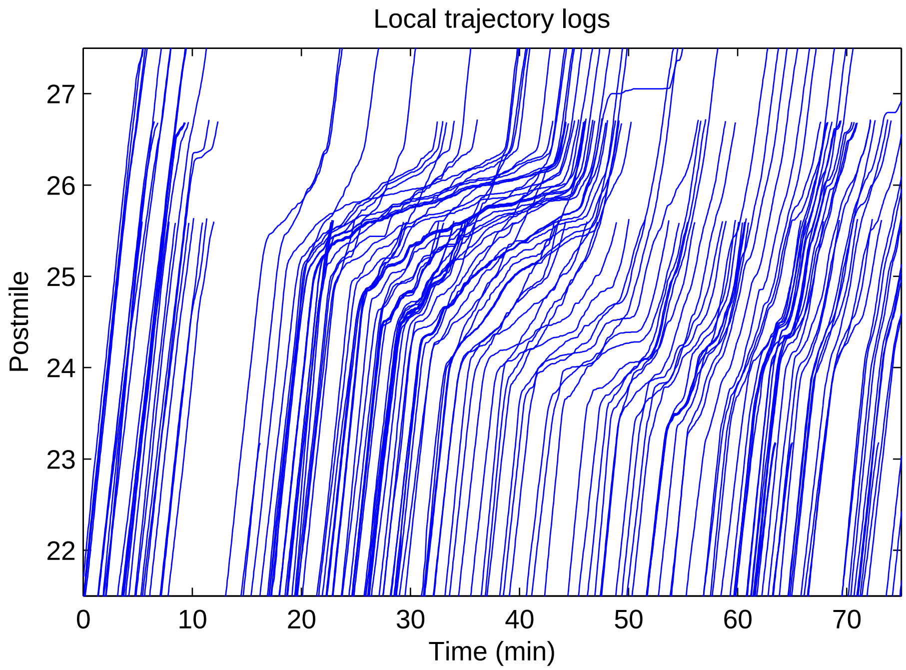

Figure 4. Local phone logs, I-880N, Feb. 8, 2008. In addition to the VTL updates, the raw trajectory of each device was recorded locally to the device as a backup for the data collection infrastructure for the purposes of this experiment only. The sharp decrease in the slope (velocity) of the trajectory corresponds to the vehicle encountering the shockwave and entering congestion. x-axis: time in minutes past 10:12am. y-axis: postmile between Decoto Rd. to the south (bottom) and Winton Ave. to the north (top). Trajectories are in the direction of increasing y.

In general, the results of the EnKF estimation show good agreement with the PeMS velocity estimate. In particular, the VTL-based sensing coupled with the EnKF algorithm captures the main features of the congestion pattern, including the length of the resulting queue, which extends just over two miles at 10:52 am (see Figs. 5 and 6). This proof of concept is an important step forward in mobile device-based sensing because of the sparsity of data used for the EnKF estimate. Unlike the loop detectors which sense every vehicle in each lane on the highway, but at fixed points in space, the mobile device-based sensing collects data from a very small fraction of vehicles. Furthermore, because of privacy considerations, the vehicles are not tracked in space; only a subset of the data logged by each device is used for estimation, sampling only anonymous location and speed updates triggered by VTLs. No extended vehicle-trajectory travel times are collected or used for estimation.

Note that there are some differences in the speed estimation shown in Figs. 5 and 6, as illustrated in in Fig. 7, which shows the relative difference between the EnKF and PeMS contour. In the free flowing regions, the relative difference is quite small. The absolute speed difference in this regime is shown with a dashed green line in Fig. 8 for a sample postmile of 22.8. As expected, the spikes in high relative difference in Fig. 7 occur in the queue resulting from the accident. The postmile with the greatest magnitude relative difference (PM 24.6, with absolute speed difference plotted as a dash dot red line in Fig. 7) occurs because of two factors. First, the EnKF estimates the velocity contour at a temporal resolution on the order of seconds, while the PeMS estimate shown is aggregated over a five minute window. Second, because the absolute speed in the congested regime is small, any difference in speed is amplified. Ultimately, the difference between PeMS and the EnKF on average is less than 10% across the network, which highlights the potential utility of mobile devices as a source of traffic data in the future.

VI. Conclusion and Future Work

This article presents a new scalar hyperbolic partial differential equation (PDE) model for the evolution of traffic velocity on highways, based on the seminal Lighthill-Whitham-Richards (LWR) PDE. It proves

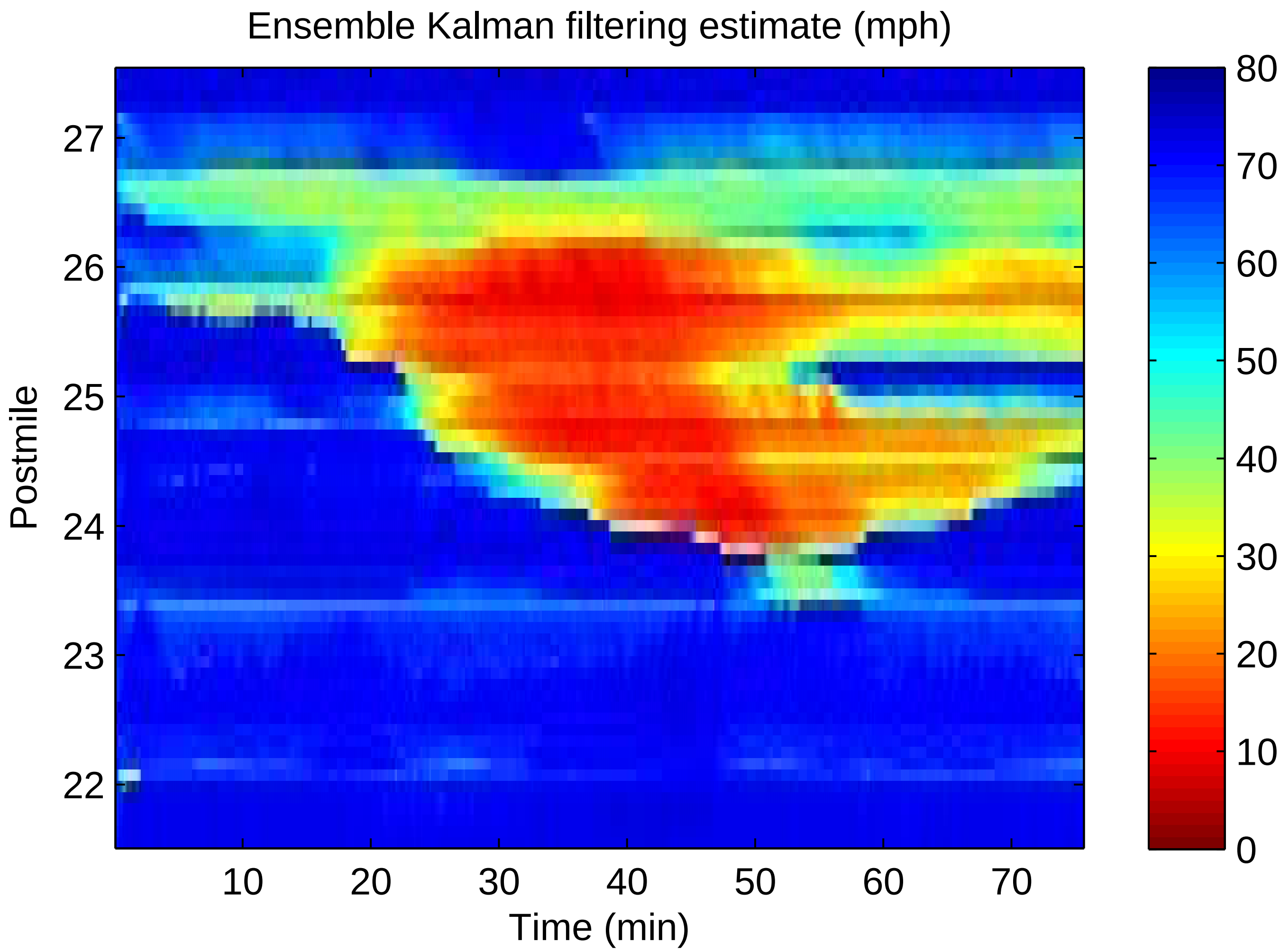

Figure 5. EnKF velocity contour plot, I-880N, Feb. 8, 2008. Color denotes speed in mph, with red denoting slow moving traffic, and blue denoting faster traffic. Vehicles travel from down to up. x-axis: time in minutes past 10:12am. y-axis: postmile between Decoto Rd. to the south (bottom) and Winton Ave. to the north (top).

the equivalence of the solution of the new PDE and the LWR PDE for quadratic flux functions, and proves that the equivalence does not hold for general flux functions. To circumvent this negative result, the article proposes a discretized model for the evolution of velocity, obtained using a transformation of the Godunov scheme. With an explicit instantiation of weak boundary conditions, the nonlinear discretized scheme is generalized to a network, thus making the model applicable to arbitrary highway systems. The resulting nonlinear time invariant dynamical system forms the basis of the Ensemble Kalman Filtering algorithm, which is introduced because of the nonlinearity and non-differentiability of the model. The algorithm was validated using velocity data obtained from GPS-equipped mobile phones in vehicles during the *Mobile Century* field experiment, and shows good agreement with velocity estimates from PeMS using loop detector data, even at penetration rates below five percent. This algorithm will be implemented next for a live system in which both fixed loop detector data and cell phone data is fused to produce traffic estimates in Northern California as part of a follow-up field operational test known as *Mobile Millennium* [37].

Acknowledgments

References

[1] 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.

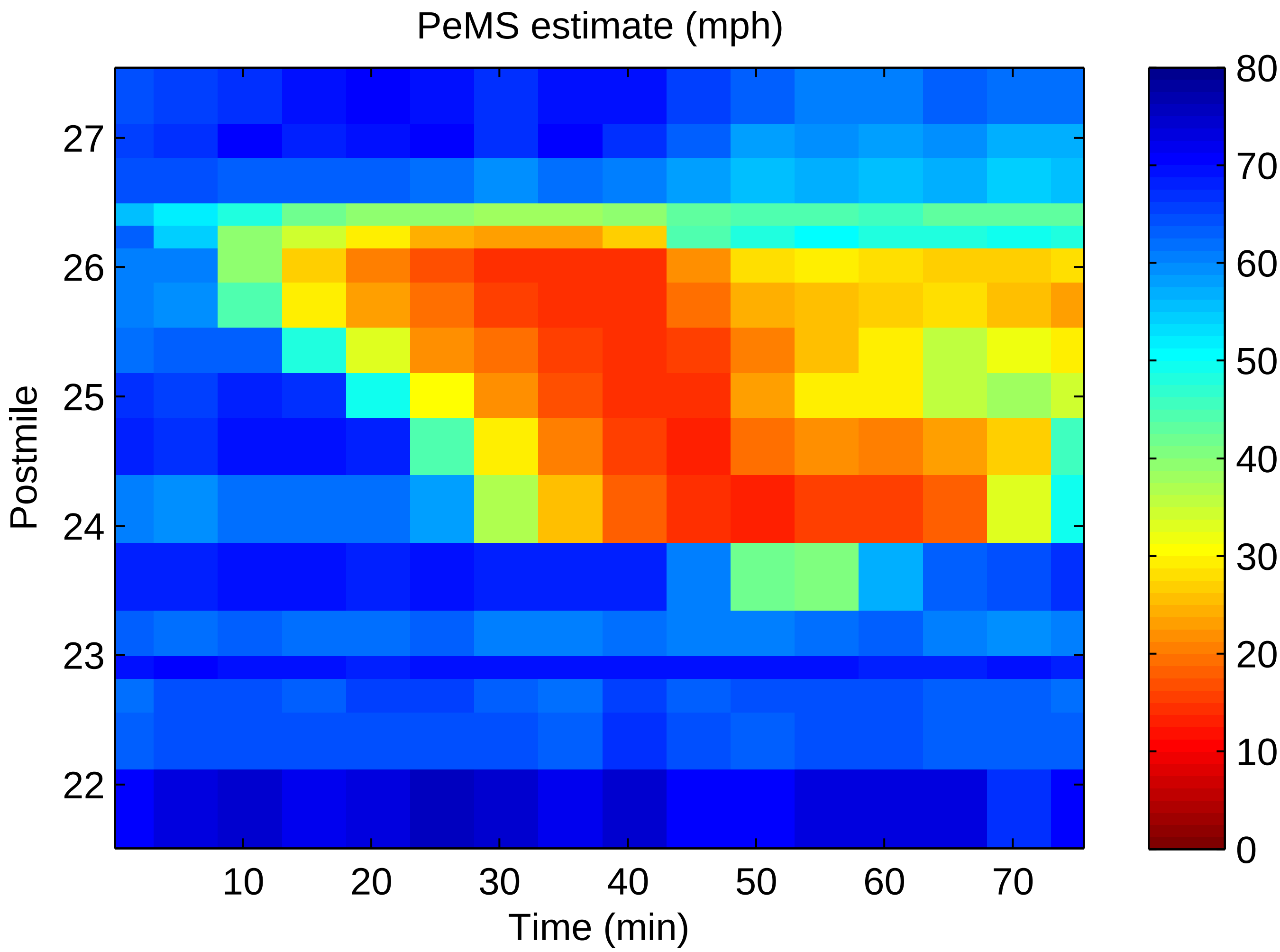

Figure 6. PeMS velocity contour plot, I-880N, Feb. 8, 2008. Color denotes speed in mph, with red denoting slow moving traffic, and blue denoting faster traffic. Vehicles travel from down to up. x-axis: time in minutes past 10:12am. y-axis: postmile between Decoto Rd. to the south (bottom) and Winton Ave. to the north (top).

[2] 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.

[3] 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*, Breckenridge, CO, June 17-18 2008, pp. 15–28.

[4] 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," submitted for publication, *Transportation Research Part C, 2009*.

[5] 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*, vol. 229, no. 1178, pp. 317–345, 1955.

[6] P. I. Richards, "Shock waves on the highway," *Operations Research*, vol. 4, no. 1, pp. 42–51, 1956.

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

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

[9] 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*, vol. 16, pp. 733–748, 2006.

[10] C. F. Daganzo, "The cell transmission model: a dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transportation Research Part B*, vol. 28, no. 4, pp. 269–287, 1994.

[11] C. F. Daganzo, "The cell transmission model, part II: network traffic," *Transportation Research Part B*, vol. 29, no. 2, pp. 79–93, 1995.

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

[13] 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*, vol. 3, Boston, MA, 2004, pp. 2098 – 2103.

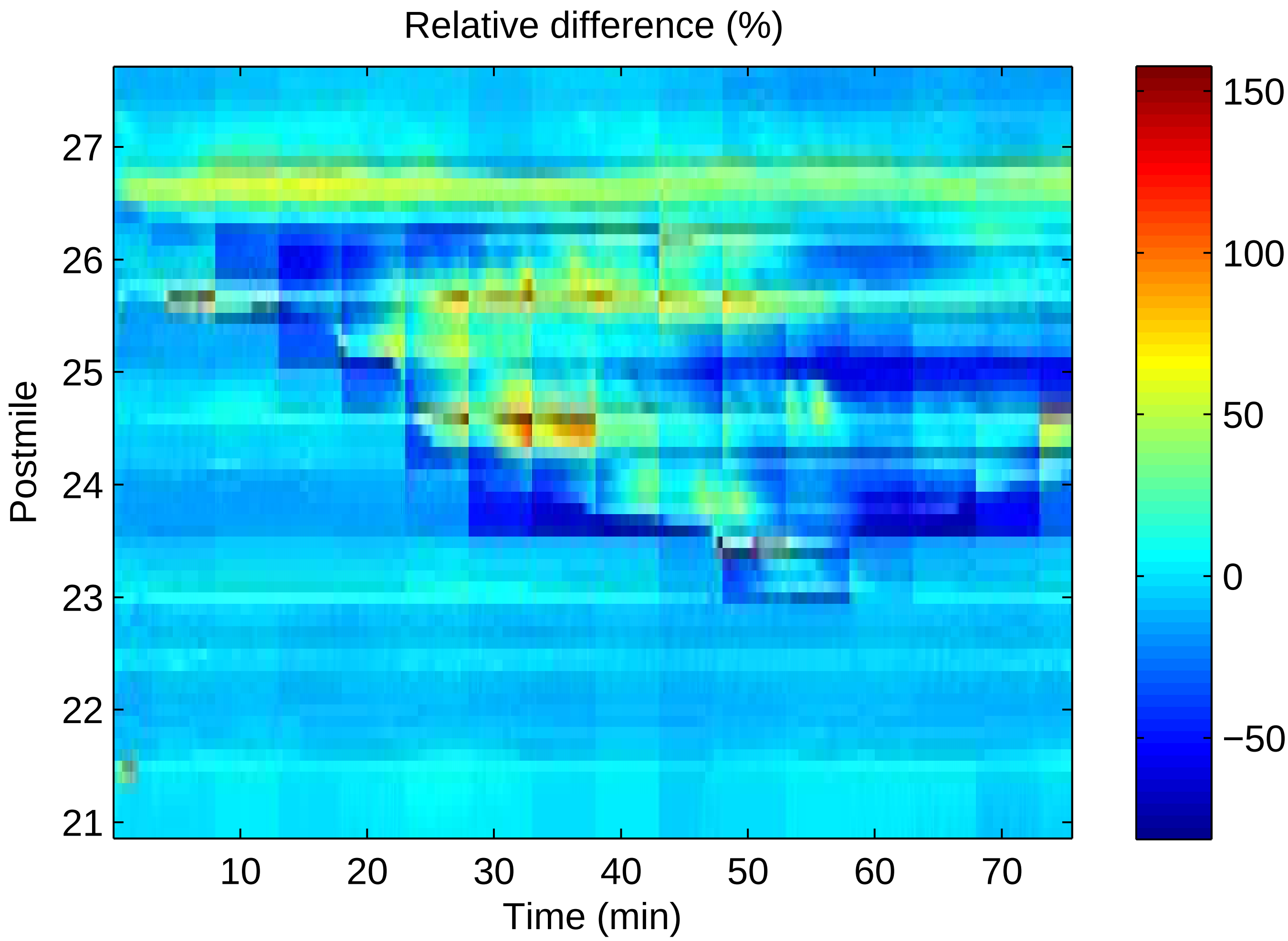

Figure 7. PeMS-EnKF relative difference plot, I-880N, Feb. 8, 2008. Color denotes speed in mph, with red denoting slow moving traffic, and blue denoting faster traffic. Vehicles travel from down to up. x-axis: time in minutes past 10:12am. y-axis: postmile between Decoto Rd. to the south (bottom) and Winton Ave. to the north (top).

[14] 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.: Transportation Research Board, Jan. 12-17 2008.

[15] Y. Wang and M. Papageorgiou, "Real-time freeway traffic state estimation based on extended Kalman filter: a general approach," *Transportation Research Part B*, vol. 39, no. 2, pp. 141–167, 2005.

[16] 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*, Sevilla, Spain, 2005, pp. 2164–2169.

[17] 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*, Minneapolis, MN, 2006, pp. 5213–5218.

[18] L. Mihaylova, R. Boel, and A. Hegyi, "Freeway traffic estimation within recursive bayesian framework," *Automatica*, vol. 43, no. 2, pp. 290–300, 2007.

[19] J. Sau, N. 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*, vol. 12, 2007.

[20] A. Hegyi, L. Mihaylova, R. Boel, and Z. Lendek, "Parallelized particle filtering for freeway traffic state tracking," in *Proc. of the European Control Conference*, Kos, Greece, July 2007, pp. 2442–2449.

[21] A. Alessandri, R. Bolla, and M. Repetto, "Estimation of freeway traffic variables using information from mobile phones," in *Proc. American Control Conference the 2003*, vol. 5, Denver, CO, June 2003, pp. 4089–4094.

[22] 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*, 2006, pp. 1047–1052.

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

[24] L. Ambrosio, N. Fusco, and D. Pallara, *Functions of Bounded Variation and Free Discontinuity Problems*. Oxford University Press, USA, 2000.

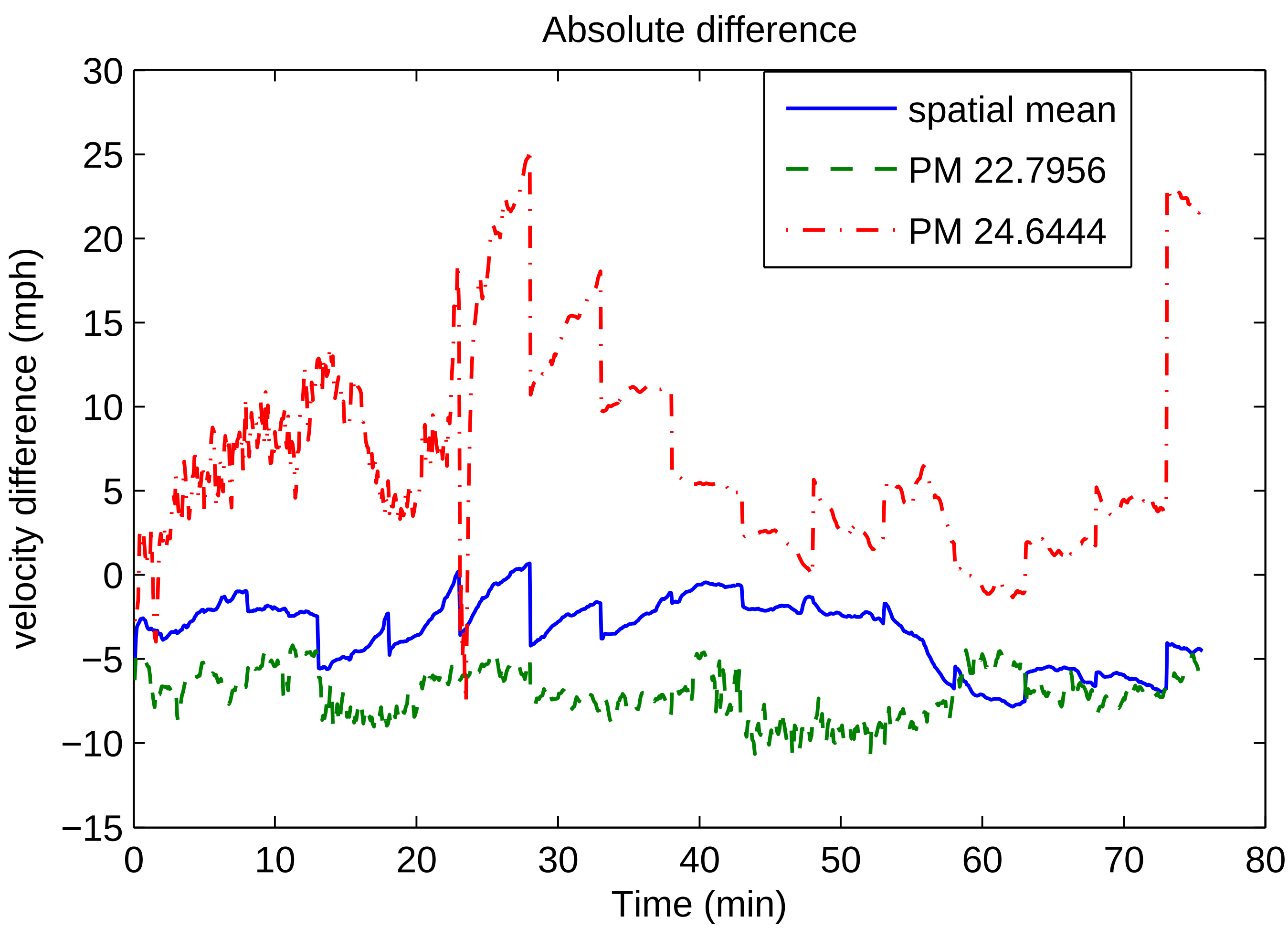

Figure 8. PeMS-EnKF absolute speed difference, I-880N, Feb. 8, 2008. Absolute difference in the velocity estimates between EnKF and PeMS spatially averaged across the network (solid blue), at postmile 22.8 with low relative error (dashed green), and postmile 24.6 with high relative error (dash dot red) as a function of time. x-axis: time in minutes past 10:12am. y-axis: absolute speed difference between EnKF and PeMS.

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

[26] C. Bardos, A. Y. Leroux, and J. C. Nedelec, "First order quasilinear equations with boundary conditions," *Communications in partial differential equations*, vol. 4, no. 9, pp. 1017–1034, 1979.

[27] B. Greenshields, "A study of traffic capacity," *Highway Research Board*, vol. 14, pp. 448–477, 1935.

[28] B. Kerner, "Phase transitions in traffic flow," *Traffic and Granular Flow*, pp. 253–284, 1999.

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

[30] S. Godunov, "A difference method for the numerical calculation of discontinuous solutions of hydrodynamic equations," *Mathematics Sbornik*, vol. 47, no. 3, pp. 271–306, 1959.

[31] 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*, Cancun, Mexico, December 2008, pp. 2141–2147.

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

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

[34] G. Burgers, P. Jan van Leeuwen, and G. Evensen, "Analysis scheme in the ensemble Kalman filter," *Monthly Weather Review*, vol. 126, no. 6, pp. 1719–1724, 1998.

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

[36] C. Chen, P. Varaiya, and J. Kwon, "An empirical assessment of traffic operations," in *International Symposium on Transportation and Traffic Theory*, College Park, MD, July 2005, pp. 19–21.

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

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