

Implementation of a density based cell  
transmission model for traffic estimation on  
highway using Ensemble Kalman Filter

Pierre-Antoine Benard and Boris Prodhomme

May 11, 2012

## Introduction

In the transportation engineering area, there has been recently a lot of interest focused on the way to model the state of traffic on highways. Indeed, the traffic congestion on highways has been constantly increasing over the years and researchers would like to find a way to maximize the satisfaction of motorists, or rather in this case minimize their dissatisfaction.

Nowadays, the increasing number of modern capabilities (Bluetooth and GPS for instance) of smartphones and mobile devices gives access to a huge amount of data that was previously inaccessible. This increasing access to measurements of the state of traffic along with the improvement of data assimilation methods as well as the computational power enables more accurate traffic estimation. In the Mobile Millenium project, traffic estimation is done via a velocity-based model which uses measurements coming from GPS, Bluetooth readers and fixed loop-detectors on highways. In this process, the density based mathematical model needs to be inverted to reach a velocity formulation leading to approximation.

In this report, we focus on the implementation of a density-based cell transmission model combined with Ensemble Kalman filtering algorithm for density data assimilation that estimates traffic on highways. We use loop detectors measurements to compute density values and integrate them in the model. After validating our model by a quantitative analysis that uses a known analytical solution for comparison with the output of our algorithm, we then compare results obtained with real data to Bluetooth data that we consider as ground truth.

# Contents

<b>1</b>	<b>Literature review</b>	<b>5</b>
1.1	Traffic theory . . . . .	5
1.1.1	Macroscopic model of traffic . . . . .	5
1.1.2	The LWR equation . . . . .	5
1.1.3	Fundamental diagram . . . . .	6
1.2	Analytical solution of the LWR PDE . . . . .	8
1.3	The Cell Transmission Model . . . . .	9
1.3.1	Godunov Scheme . . . . .	10
1.3.2	CFL condition . . . . .	10
1.3.3	Our specific case . . . . .	10
1.4	Density data assimilation . . . . .	11
1.4.1	Linear case : Kalman filter . . . . .	11
1.4.2	Extended Kalman filter . . . . .	12
1.4.3	Ensemble Kalman filter . . . . .	13
1.5	Extension to networks . . . . .	14
<b>2</b>	<b>Implementation</b>	<b>15</b>
2.1	Density measurements . . . . .	15
2.2	Constructing the EnKF observation matrix . . . . .	15
2.3	Initialization . . . . .	16
2.4	Description of the main algorithm . . . . .	16
<b>3</b>	<b>Analytical Results</b>	<b>18</b>
3.1	Verification of the CTM script . . . . .	18
3.2	EnKF-implementation verification . . . . .	20
<b>4</b>	<b>Bluetooth Validation and Empirical Results</b>	<b>25</b>
4.1	Methodology . . . . .	25
4.2	Bluetooth data extraction . . . . .	26
4.2.1	Data feed . . . . .	26
4.2.2	Filter . . . . .	27
4.2.3	Travel time comparison . . . . .	27
4.3	Results . . . . .	28
4.3.1	Test on I880 . . . . .	28
	<b>References</b>	<b>32</b>

## List of Figures

1	Common Fundamental Diagrams . . . . .	7
2	Speed of shock. . . . .	9
3	Flow chart describing the main algorithm . . . . .	17
4	Analytical solution of the Riemann problem given by Eq.(50) and (51) . . . . .	19
5	Numerical results for the Riemann problem given by Eq.(50) and (51) . . . . .	19
6	Numerical error of the $\rho$ -CTM model for the Riemann problem given by Eq.(50) and (51) . . . . .	20
7	Propagation of the initial condition Eq.(51) with Daganzo flow for $\rho_c = 1$ . . . . .	21
8	Propagation of the initial condition Eq.(51) with Daganzo flow for $\rho_c = 2$ . . . . .	21
9	Test EnKF, influence of the state noise : 1% . . . . .	22
10	Test EnKF, influence of the state noise : 5% . . . . .	22
11	Test EnKF, influence of the state noise : 10% . . . . .	23
12	Test EnKF modified, influence of the state noise : 1% . . . . .	24
13	Test EnKF modified, influence of the state noise : 5% . . . . .	24
14	Test EnKF modified, influence of the state noise : 10% . . . . .	24
15	Test EnKF influence of the quantity of measurements : 10% in t and x . . . . .	25
16	Data Validation Framework . . . . .	26
17	Map of Bluetooth sensors in Highway I-880 . . . . .	27
18	Comparison of the output for two fundamental diagrams . . . . .	29
19	Comparison of model output with Bluetooth output . . . . .	30

# 1 Literature review

## 1.1 Traffic theory

### 1.1.1 Macroscopic model of traffic

A common approach to the macroscopic mathematical representation of traffic has been introduced in the late 1950's by Lighthill, Whitman [7] and Richards [8]. In this approach, the flow of vehicles is modeled as a fluid with three characteristics : density  $\rho$ , flow  $q$  and speed  $v$ . We have the following definitions :

- *Flow*  $q(x, t)$  is the total number of vehicles that pass the point  $x$  during a given time interval containing  $t$ , divided by the length of the time interval. It is usually expressed as an hourly rate, and is easily measured with road sensors.
- *Density*  $\rho(x, t)$  is the number of vehicles occupying a length of freeway about point  $x$  at instant  $t$ .

Density measurements are difficult to obtain directly but with the previous definition we have:

$$\rho(x, t) = \frac{q(x, t)}{v(x, t)} \quad (1)$$

where  $v(x, t)$  is the velocity. We will use this formula to transform our speed and flow measurements into density measurements to input in our mathematical model.

### 1.1.2 The LWR equation

The evolution of the density of vehicles is described by a conservation law called the LWR equation. This conservation law states that the variation of the number of vehicles in a cell of infinitesimal length  $dx$  during an infinitesimal time  $dt$  is equal to difference of between the number of vehicles that enters the cell and the number of vehicles that exits the cell during  $dt$ .

So as to use this model on a bounded portion of highway, it needs boundary and initial conditions. In the particular case of multilane roads, this model can be interpreted as the description of the average density of vehicles on the lanes. Finally the whole macroscopic continuous model boils down to the following equation:

$$\frac{\partial \rho}{\partial t}(x, t) + \frac{\partial q(\rho(x, t))}{\partial x} = 0 \quad (2)$$

Another way to write Equation (2) using the chain rule is :

$$\frac{\partial \rho(x, t)}{\partial t} + Q'(\rho(x, t)) \frac{\partial \rho(x, t)}{\partial x} = 0 \quad (3)$$

With the associated initial and boundary conditions :

$$\begin{cases} \rho(x, 0) = \rho_0(x) \quad \forall x \in [a, b] \\ \rho(a, t) = \rho_a(t), \quad \rho(b, t) = \rho_b(t) \quad \forall t \in [0, T] \end{cases} \quad (4)$$

with  $q$  the flux function of the fundamental diagram of traffic.

### 1.1.3 Fundamental diagram

From experimental observations, it appears that there exists a correlation between speed, flow and density. This was first mentioned by Greenshields in 1934 in [5]. The relation  $q = Q(\rho)$  is commonly called *the fundamental diagram*. Figure 1 shows a sample of the most commonly used.

In this project, we essentially use Greenshields and Daganzo flows. We use Greenshields for verification purposes because we can derive a complete analytical solution of the problem when using Greenshields. Daganzo is one of the most commonly and accurate fundamental that verifies concavity properties. Moreover, we have a well-calibrated Daganzo flow for the route I880 that we wish to study.

Greenshields flow function:

$$v = V_G(\rho) = v_f \left(1 - \frac{\rho}{\rho_j}\right) \quad (5)$$

and the corresponding flux function is:

$$Q_G(\rho) = \rho V_G(\rho) = v_f \left(\rho - \frac{\rho^2}{\rho_j}\right) \quad (6)$$

Daganzo flow function:

$$v = V_{DN}(\rho) = \begin{cases} v_f & \text{if } \rho \leq \rho_c \\ -w \left(1 - \frac{\rho_j}{\rho}\right) & \text{if } \rho > \rho_c \end{cases} \quad (7)$$

and the corresponding flux function is:

$$Q_{DN}(\rho) = \rho V_{DN}(\rho) \begin{cases} v_f \rho & \text{if } \rho \leq \rho_c \\ -w(\rho - \rho_j) & \text{if } \rho > \rho_c \end{cases} \quad (8)$$

where  $v_f$  is the *free-flow speed*,  $\rho_j$  is the *jam density* (i.e  $q(\rho_j) = 0$ ),  $\rho_c$  is the *critical density* (i.e. *free flow* when  $\rho \leq \rho_c$  and *congestion* when  $\rho > \rho_c$ ) and  $w$  is the backwards propagation wave speed.

It is important to notice that both of the previous fundamental diagrams satisfy the following properties:

- (i) Hypothesis of a static flow/density relationship:  $q = Q(\rho(x, t))$ .
- (ii)  $Q(0) = Q(\rho_j) = 0$ .

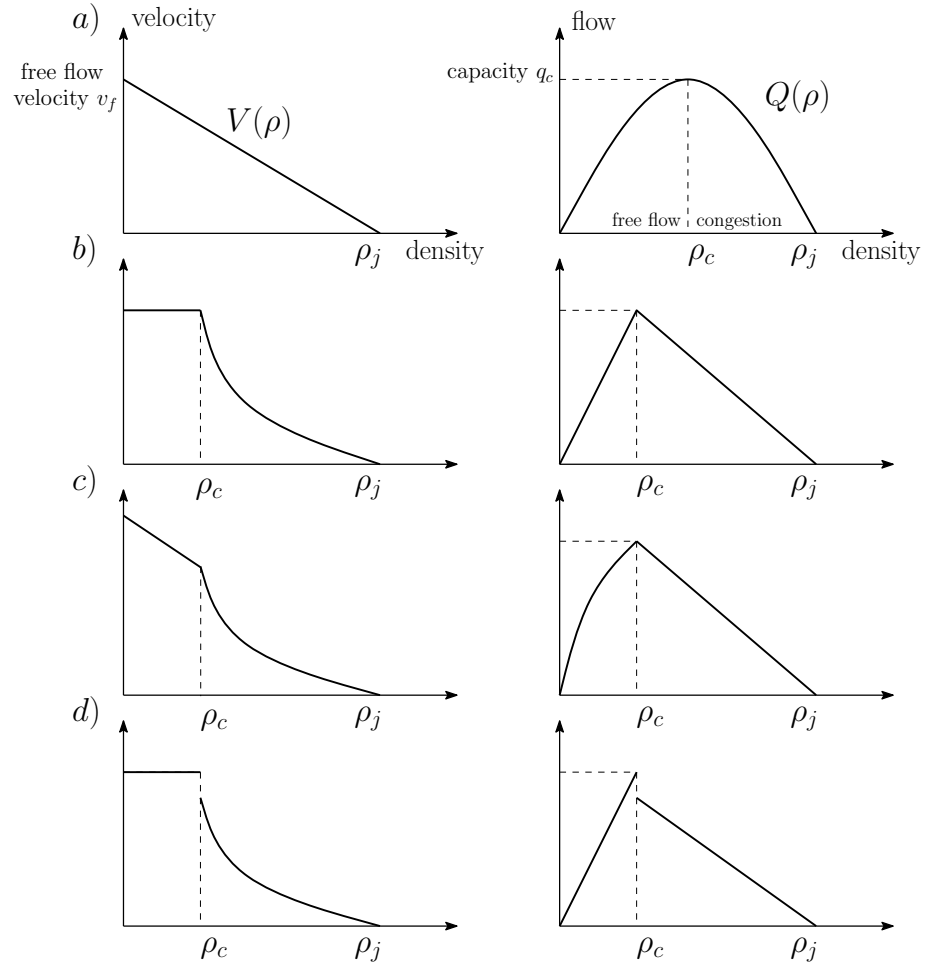


Figure 1: Speed and flow relationships (fundamental diagrams) for Greenshields (a), Daganzo-Newell (b), linear-hyperbolic (c), and discontinuous (d).

- (iii) The continuous portions of  $Q(\rho)$  are concave.
- (iv)  $V(0) = v_f$ , and  $V(\rho_j) = 0$ .
- (v) A critical density  $\rho_c$  can be defined where the maximum flow  $q_c$  is attained. Then,  $Q(\rho)$  is increasing for  $\rho \leq \rho_c$  and decreasing for  $\rho > \rho_c$ .
- (vi) The critical density  $\rho_c$  separates the fundamental diagram into two regimes: *free flow* when  $\rho \leq \rho_c$  and *congestion* when  $\rho > \rho_c$ .

## 1.2 Analytical solution of the LWR PDE

The solutions of this equation can be described in terms of *characteristics*, or trajectories in the time/space plane along which the densities and flows are described by ordinary differential equations. Equation (3) is a relatively simple wave equation: its characteristics are straight lines with slope  $Q'(\rho)$ . Along the characteristics, the densities and flows remain *constant*. The value of the density or flow at any point is therefore equal to a boundary or initial condition to which the point is connected by a characteristic. The boundary value problem is *well-posed* if the family of characteristics emanating from the initial and boundary conditions spans the entire time-space plane. The main complication arises, as described in [7], when characteristics intersect, leading to multiple values at the point of intersection. In this situation, the PDE admits only *weak solutions*, which necessarily contain discontinuities, or *shocks*. The discontinuities are such that the more general integral form of (2) between arbitrary points  $x_1$  and  $x_2$  applies:

$$\frac{d}{dt} \int_{x_1}^{x_2} \rho(x, t) dx + q(x_2, t) - q(x_1, t) = 0 \quad (9)$$

The speed of the shock  $c_s$  can be found by applying (9) to a small interval  $[x_1, x_2]$  containing the moving shock at  $x = s(t)$  [10] (see Figure 2):

$$\begin{aligned} q(x_1, t) - q(x_2, t) &= \frac{d}{dt} \int_{x_1}^{x_2} \rho(x, t) dx \\ &= \frac{d}{dt} \int_{x_1}^{s^-(t)} \rho(x, t) dx + \frac{d}{dt} \int_{s^+(t)}^{x_2} \rho(x, t) dx \\ &= \dot{s} \rho(s^-, t) + \int_{x_1}^{s^-(t)} \frac{\partial \rho}{\partial t} dx - \dot{s} \rho(s^+, t) + \int_{s^+(t)}^{x_2} \frac{\partial \rho}{\partial t} dx \end{aligned} \quad (10)$$

Taking the limit  $x_1 \rightarrow s^-$  and  $x_2 \rightarrow s^+$ :

$$c_s(t) = \dot{s}(t) = \frac{q(s^+, t) - q(s^-, t)}{\rho(s^+, t) - \rho(s^-, t)} = \frac{q^+ - q^-}{\rho^+ - \rho^-} \quad (11)$$

This last equation, called the *Rankine-Hugoniot jump condition*, gives a condition on discontinuities of weak solutions of (3) relating the right and left states with the “speed”  $c_s$  of the “shock”. This last quantity is represented graphically on the fundamental diagram. This is the slope of the chord joining the



upstream and downstream states, as illustrated in Figure 2. It is shown in [10] that the concavity of  $Q(\rho)$  implies that only shocks of increasing density, or *deceleration* shocks, can develop in the traffic stream. This is because conditions of heavy downstream traffic and light upstream traffic generate waves that coalesce, whereas conditions of smaller downstream density lead to diverging waves. Hence, all shocks that form naturally must satisfy:

$$\rho^- \leq \rho^+ \quad (12)$$

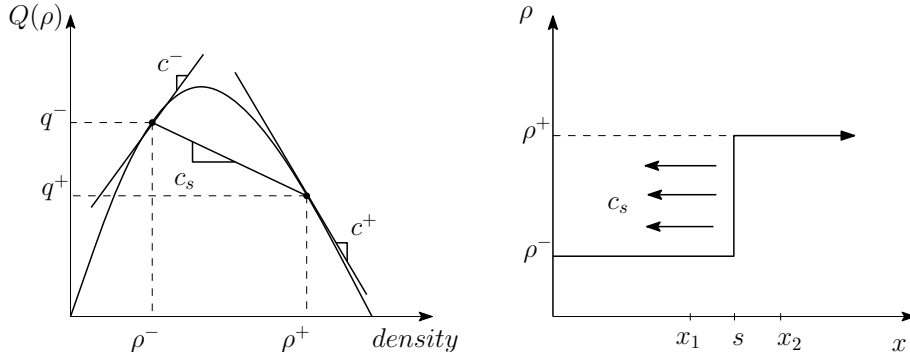


Figure 2: Speed of shock.

Denoting the speed of the characteristics that terminate in the shock with  $c^- = Q'(\rho^-)$  and  $c^+ = Q'(\rho^+)$ , then equations (11), (12), and the concavity of  $Q(\rho)$  imply:

$$c^- > c_s > c^+ \quad (13)$$

This condition, known as the *entropy condition*, along with the *jump condition* solve the problem of *shock waves*. For further details on this problem, which is a specific case to the *Riemann problem*, we refer the reader to [3]. We see in the next section that the *Riemann problem* motivates the choice of specific mathematical schemes and algorithms when it comes to implement the models using numerical approximations. Ansorge [1] found in 1990 a condition of *increasing entropy* that is sufficient to eliminate all of the weak solutions of LWR equation except for the physically relevant one. This justifies why we only work with Greenshields and Daganzo flows in this project.

### 1.3 The Cell Transmission Model

A good numerical method to solve the equations along roads is represented by the Godunov scheme, which is based on exact solutions to Riemann problems [4] and the notion of entropy mentioned in Section 1.2. This leads to the construction of a nonlinear discrete time dynamical system.

### 1.3.1 Godunov Scheme

The Godunov discretization scheme is applied on the LWR PDE, where the discrete time step  $\Delta T$  is indexed by  $n$ , and the discrete space step  $\Delta X$  is indexed by  $i$ :

$$\rho_i^{n+1} = \rho_i^n - \frac{\Delta T}{\Delta X} (G(\rho_i^n, \rho_{i+1}^n) - G(\rho_{i-1}^n, \rho_i^n)) \quad (14)$$

where the Godunov flux  $G(\rho_1, \rho_2)$  is in general defined as:

$$G(\rho_1, \rho_2) = \begin{cases} \min_{\rho \in [\rho_1, \rho_2]} Q(\rho) & \text{if } \rho_1 \leq \rho_2 \\ \max_{\rho \in [\rho_2, \rho_1]} Q(\rho) & \text{if } \rho_2 \leq \rho_1 \end{cases} \quad (15)$$

Similarly, we can apply the scheme at the boundaries (see [9])

$$\rho_0^{n+1} = \rho_0^n - \frac{\Delta T}{\Delta X} (G(\rho_0^n, \rho_1^n) - G(\rho_{-1}^n, \rho_0^n)) \quad (16)$$

$$\rho_{i_{max}}^{n+1} = \rho_{i_{max}}^n - \frac{\Delta T}{\Delta X} (G(\rho_{i_{max}}^n, \rho_{i_{max}+1}^n) - G(\rho_{i_{max}-1}^n, \rho_{i_{max}}^n)) \quad (17)$$

with the ghost cells on the left and the right of the space interval defined as follows:

$$\rho_{-1}^n = \frac{1}{\Delta t} \int_{(n-\frac{1}{2})\Delta t}^{(n+\frac{1}{2})\Delta t} \rho_a(t) dt. \quad (18)$$

$$\rho_{i_{max}+1}^n = \frac{1}{\Delta t} \int_{(n-\frac{1}{2})\Delta t}^{(n+\frac{1}{2})\Delta t} \rho_b(t) dt. \quad (19)$$

### 1.3.2 CFL condition

In order to ensure numerical stability, the time and space steps are coupled by the CFL condition [6]. In the case of our study which is a hyperbolic PDE, the CFL condition becomes:

$$\frac{\Delta T}{\Delta X} \leq \frac{1}{c_{max}} \quad (20)$$

where  $c_{max}$  denotes the maximal characteristic speed.

### 1.3.3 Our specific case

For a family of flux functions  $Q(\rho)$  sharing the same properties (i) to (vi) given in section 1.1.3, the Godunov flux has an explicit formulation. It can also be expressed as the minimum of the *sending flow* from the upstream cell and the *receiving flow* from the downstream cell through a boundary connecting two cells of a homogeneous road. The *sending flow*  $S(\rho)$  is equal to the upstream flow if the upstream traffic is in free flow ( $\rho \leq \rho_c$ ) or the capacity of the upstream section  $q_c$  if the upstream traffic is in congestion ( $\rho > \rho_c$ ); on the other hand, the *receiving flow*  $R(\rho)$  is equal to the capacity of the downstream section if the downstream traffic is in free flow or the downstream flow if the downstream

traffic is in congestion. Notice that this explanation is related to the analytical method of characteristics highlighted in section 1.2 . The explicit formulation of the Godunov flow is given below:

$$G(\rho_1, \rho_2) = \min(S(\rho_1), R(\rho_2)) \quad (21)$$

$$S(\rho) = \begin{cases} Q(\rho) & \text{if } \rho \leq \rho_c \\ q_c & \text{if } \rho > \rho_c \end{cases} \quad (22)$$

$$R(\rho) = \begin{cases} q_c & \text{if } \rho \leq \rho_c \\ Q(\rho) & \text{if } \rho > \rho_c \end{cases} \quad (23)$$

Or, finally,

$$G(\rho_1, \rho_2) = \begin{cases} Q(\rho_2) & \text{if } \rho_c \leq \rho_2 \leq \rho_1 \\ Q(\rho_c) & \text{if } \rho_2 \leq \rho_c \leq \rho_1 \\ Q(\rho_1) & \text{if } \rho_2 \leq \rho_1 \leq \rho_c \\ \min(Q(\rho_1), Q(\rho_2)) & \text{if } \rho_1 \leq \rho_2 \end{cases} \quad (24)$$

In our scripts we use the formulation given by Eq.(24) . Also, for the two Fundamental Diagrams we are working with, Eq. (20) becomes:

Greenshields flow function:

$$\frac{\Delta T}{\Delta X} \leq \frac{1}{v_f} \quad (25)$$

Daganzo flow function:

$$\frac{\Delta T}{\Delta X} \leq \frac{1}{\max(v_f, w)} \quad (26)$$

## 1.4 Density data assimilation

The Kalman filter operates recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying system state. Based on [2], we give here the general formulation of the Kalman Filter for discrete time systems. We explain why we can not use the 2 first approaches in this study. Finally, based on [11], we give a formulation of the Ensemble Kalman Filter for the density-based model.

### 1.4.1 Linear case : Kalman filter

For a discrete-time system:

$$\begin{cases} x(k+1) = \mathbf{M}(k)x(k) + \mathbf{B}_u(k)u(k) + \mathbf{B}_w(k)w(k) \\ y(k) = \mathbf{H}(k)x(k) + v(k) \end{cases} \quad (27)$$

where  $v$  is the measurement noise  $\mathbf{H}(\mathbf{k})$  the observation matrix,  $u$  is the input and  $w$  is the input noise, such that:

$$\begin{cases} v(k) \sim \mathbf{N}(0, R(k)) \text{ zero mean Gaussian white noise with covariance } R(k) \\ w(k) \sim \mathbf{N}(0, Q(k)) \text{ zero mean normal distribution with covariance } Q(k) \\ (x(0), w(1), \dots, w(k), v(1), \dots, v(k)) \text{ are uncorrelated} \end{cases} \quad (28)$$

With these assumptions, it has been shown that it is possible to compute the best estimate of  $x(k)$  in the sense of least squares using a recursive algorithm by Kalman.

- Forecast step (Time-update):

$$x_f(k+1) = \mathbf{M}(k)x_f(k) + \mathbf{B}_u(k)u(k) \quad (29)$$

$$\mathbf{P}_f^n = \mathcal{M}(k)\mathbf{P}_a(k)\mathcal{M}(k)^T + \mathbf{Q}(k) \quad (30)$$

$x_f(k+1)$  (resp.  $x_a(k+1)$ ) : forecast (analyzed) state estimate

- Analysis step (Measurement-update):

$$x_a(k+1) = x_f(k+1) + \mathbf{G}(k+1)(y(k+1) - \mathbf{H}(k+1)x_f(k+1)) \quad (31)$$

$$\mathbf{P}_a(k+1) = \mathbf{P}_f(k+1) - \mathbf{G}(k+1)\mathbf{H}(k+1)\mathbf{P}_f(k+1) \quad (32)$$

$$\mathbf{G}(k+1) = \mathbf{P}_f(k+1)\mathbf{H}(k+1)^T (\mathbf{H}(k+1)\mathbf{P}_f(k+1)\mathbf{H}(k+1)^T + \mathbf{R}(k+1))^{-1} \quad (33)$$

$\mathbf{P}_f(k+1)$  (resp.  $\mathbf{P}_a(k+1)$ ) : error covariance of the forecast (analyzed) state

#### 1.4.2 Extended Kalman filter

In our case we are working with an autonomous equation, i.e. without input, so we can simplify the equations. We now apply every formula found in literature for our specific application. We proceed by analogy with what was done in [] with a velocity formulation of the problem. If equation (14) was differentiable in  $\rho^n$ , so would be the operator  $\mathcal{M}[\cdot]$  in (34), in which case the optimal estimate for the state  $\rho^n$  could be obtained using the traditional Extended Kalman Filter equations. These are simply the Jacobian linearized version of the linear case.

- Forecast step (Time-update):

$$\rho_f^n = \mathcal{M}(\rho_a^{n-1}) \quad (34)$$

$$\mathbf{P}_f^n = \mathcal{M}_L^{n-1}\mathbf{P}_a^{n-1}(\mathcal{M}_L^{n-1})^T + \mathbf{Q}^{n-1} \quad (35)$$

$\rho_f^n$  (resp.  $\rho_a^n$ ) : forecast (analyzed) state estimate

$\mathcal{M}_L^n = \frac{\partial \mathcal{M}(\rho_a^n)}{\partial \rho_a^n}$  : jacobian matrix of  $\mathcal{M}$

- Analysis step (Measurement-update):

$$\rho_a^n = \rho_f^n + \mathbf{G}^n (y^n - \mathbf{H}^n \rho_f^n) \quad (36)$$

$$\mathbf{P}_a^n = \mathbf{P}_f^n - \mathbf{G}^n \mathbf{H}^n \mathbf{P}_f^n \quad (37)$$

$$\mathbf{G}^n = \mathbf{P}_f^n (\mathbf{H}^n)^T \left( \mathbf{H}^n \mathbf{P}_f^n (\mathbf{H}^n)^T + \mathbf{R}^n \right)^{-1} \quad (38)$$

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

### 1.4.3 Ensemble Kalman filter

As mentioned in section 1.4.2, the Extended Kalman filter implementation requires the equation governing the density evolution to be differentiable whereas, in the case of the discretized LWR equation using Godunov scheme, it is not. Indeed, nondifferentiability comes from both the Godunov scheme itself (see  $\min(\cdot)$ ) and the flow function (see Daganzo in (8)). In this case there is no easy way to compute the error covariance matrices in a recursive way as in sections 27 and 1.4.2. To make up for this impossibility, the apriori error covariance is computed by a Monte-Carlo type of approach reasoning on an ensemble of values of  $\rho^n$  instead of one averaged value. Finally, the ensemble Kalman filter algorithm can be summarized as follows [2]:

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

2. *Forecast*: Update each of the  $K$  ensemble members according to the Godunov discretization scheme. Then update the ensemble mean and covariance according to:

$$\rho_f^n(k) = \mathcal{M}(\rho_a^{n-1}(k)) + \eta^n(k) \quad (39)$$

$$\bar{\rho}_f^n = \frac{1}{K} \sum_{k=1}^K \rho_f^n(k) \quad (40)$$

$$\mathbf{P}_{\text{ens},f}^n = \frac{1}{K-1} \sum_{k=1}^K (\rho_f^n(k) - \bar{\rho}_f^n) (\rho_f^n(k) - \bar{\rho}_f^n)^T \quad (41)$$

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

$$\mathbf{G}_{\text{ens}}^n = \mathbf{P}_{\text{ens},f}^n (\mathbf{H}^n)^T \left( \mathbf{H}^n \mathbf{P}_{\text{ens},f}^n (\mathbf{H}^n)^T + \mathbf{R}^n \right)^{-1} \quad (42)$$

$$\rho_a^n(k) = \rho_f^n(k) + \mathbf{G}_{\text{ens}}^n (y^n(k) - \mathbf{H}^n \rho_f^n(k) + \chi^n(k)) \quad (43)$$

4. Return to 2.

where  $\mathcal{M}(\cdot)$  is the Godunov operator,  $\chi^n$  is the measurement noise,  $\eta^n$  the state noise and  $y^n$  the measurements.

## 1.5 Extension to networks

Even though we did not have time to implement the density-based model on a road network, we present here the theory that will enable us to do it in the future. The road network is represented by a directed graph consisting of vertices  $v \in \mathcal{V}$ , which represent the junctions in the network, and edges  $e \in \mathcal{E}$ , which are the links. Each link  $e$  is divided into cells numbered from 0 to  $I_e$ , which gives the density fields for the entire network  $\rho^n = \rho_{i,e}^n, e \in \mathcal{E}, i = 0, \dots, I_e$ , at each time step. For two consecutive cells without any junction the evolution of the density field is governed by the same equations as before.

Let us consider a junction  $v$  with  $\mathcal{I}(v)$  and  $\mathcal{O}(v)$  the sets of incoming and outgoing edges respectively and assume that the traffic entering from an incoming edge is distributed amongst the outgoing edges according to an allocation matrix  $A_v = (\alpha_{e,e'}^v)$  where  $e \in \mathcal{I}(v)$ , and  $e' \in \mathcal{O}(v)$ . For all the vehicles entering the junction  $v$  on edge  $e$ ,  $\alpha_{e,e'}^v$  denotes the proportion of vehicles which will exit the junction through edge  $e'$ . We note that  $\sum_{e' \in \mathcal{O}(v)} \alpha_{e,e'}^v = 1$  for any  $e \in \mathcal{I}(v)$ ,

and the conservation of flux gives  $\sum_{e \in \mathcal{I}(v)} \alpha_{e,e'}^v Q_e = Q_{e'}$  for any  $e' \in \mathcal{O}(v)$ . But strong boundary conditions (i.e. equality) cannot always be imposed for an arbitrary pair  $(\sum_{e \in \mathcal{I}(v)} \alpha_{e,e'}^v Q_e, Q_{e'})$ , so the boundary conditions provide upper bounds on the admissible incoming and admissible outgoing fluxes over which the flow through the incoming edges is maximized:

$$\begin{aligned} & \text{maximize} && 1^T \xi \\ & \text{s.t.} && A_v \xi \leq \delta_{\mathcal{O}(v)} \\ & && 0 \leq \xi \leq \delta_{\mathcal{I}(v)} \end{aligned} \quad (44)$$

where  $\delta_{\mathcal{I}(v)} = (\delta_e)_{e \in \mathcal{I}(v)}$ ,  $\delta_{\mathcal{O}(v)} = (\delta_{e'})_{e' \in \mathcal{O}(v)}$  are the upper bounds on the edges entering and exiting the junction respectively. In the discrete case, they are given by:

$$\delta_e = Q(\rho_{I_e,e}^n) \text{ for any } e \in \mathcal{I}(v) \quad (45)$$

$$\delta_{e'} = Q(\rho_{0,e'}^n) \text{ for any } e' \in \mathcal{O}(v) \quad (46)$$

With the optimal solution to (44), denoted by  $\xi^*$ , we can update  $G(\rho_{-1,e'}^n, \rho_{0,e'}^n)$  and  $G(\rho_{I_e,e}^n, \rho_{I_e+1,e}^n)$  through:

$$G(\rho_{I_e,e}^n, \rho_{I_e+1,e}^n) = \xi_e^* \text{ for any } e \in \mathcal{I}(v) \quad (47)$$

$$G(\rho_{-1,e'}^n, \rho_{0,e'}^n) = \sum_{e \in \mathcal{I}(v)} \alpha_{e,e'}^v \xi_e^* \text{ for any } e' \in \mathcal{O}(v) \quad (48)$$

## 2 Implementation

### 2.1 Density measurements

Single-loop detector data with a sample period of 30 seconds is used as a source of measurements for the density-based highway model.

Single-loop sensors are essentially metal detectors which measure the flow of traffic by counting the number of vehicles having crossed them during a fixed period of time (hereby referred to as the loop detector’s sample period); an indirect measure of density is also obtained by recording the percentage of time during which the detector was occupied by a vehicle (known as the occupancy of the sensor for the sample period).

Due to their low cost and their relative simplicity, loop detectors constitute the single most abundant source of traffic data. Loop detectors are grouped into stations; the measurements of the sensors at each station are sent periodically for data processing. Typically, each station will describe a single “point” of the highway and will thus have as many sensors as the number of lanes at the highway location considered.

The following pre-processing steps are necessary in order to make the single-loop detector data suitable for use by the density-based highway model:

- Converting the occupancy measurements to spatial densities by using an estimate of the average effective vehicle length (or the g-factor)
- Raw occupancy data is noisy and is filtered using heuristics which determine the quality of a datum. Bad data is discarded, which results occasionally in “holes” in the measurements.
- Finally, each loop detector station describes traffic on the different lanes of the highway: in order to obtain a single-pipe measurement for each time sample and each station, the multi-lane densities are summed.

### 2.2 Constructing the EnKF observation matrix

In its current state, the density-based highway model can only be run on routes. Routes in this report are defined as straight-line stretches of road along the highway going in a single direction; they are devoid of ramps, traffic signals, toll booths, or any of the real-life disturbances which normally affect highway traffic. The data structures used to describe the topology of the highway in the *Mobile Millennium* codebase have been leveraged : routes are composed of Model Graph links going in the same direction (model graph links have a pre-defined length). For the highway model to work correctly, it is imperative that the links be provided in the correct geographical order. The links are then divided into  $m$  cells of length  $l$ . Care should be taken when segmenting the route into cells to ensure that the CFL condition is not broken. The link information (link ID, number of cells, cell length) is stored in a MATLAB struct.

The locations of the loop detector stations along the highway are retrieved from the *MM* database in terms of model graph links and offsets. From this

information, it becomes possible to determine which cells contain sensors. Furthermore, the “deceased” sensor stations (i.e. stations showing no sign of activity for the entire period of time considered) are discarded. The sensor information is stored in a separate MATLAB struct. The density measurements for the sensors under consideration are then retrieved.

Finally, because loop detector stations do not always report measurements, it is necessary to build a sensor activity vector of size  $1 \times m$  at each iteration of the EnKF: if a sensor in cell  $i$  is active (inactive) at timestep  $j$ , then the  $j$ th sensor activity vector will have a value of one (zero) in its  $i$ th position.

For a route with  $m$  cells, the observation matrix  $H_j$  at each timestep  $j$  is constructed in the following way using the collected data:

1. Using the sensor information and the link information, a static  $H$  matrix is built: cells containing a sensor have a value of one in the matrix, whereas cells without a sensor have a value of 0.
2. At each iteration  $j$  of the EnKF, a matrix  $H_j$  is formed by multiplying  $H$  by the corresponding sensor activity column vector: the inactive sensors at timestep  $j$  are effectively “zeroed out”.

## 2.3 Initialization

The following parameters are used in the density based highway model:

- The free-flow velocity and the jam density are set to default values
- The boundary conditions and the initial conditions are set to 0
- The initial ensemble has a zero-mean multivariate normal distribution, with a covariance matrix scaled by  $(\frac{\rho_{jam}}{2})^2$ .
- The state noise has a zero-mean multivariate normal distribution with a covariance matrix scaled by  $(\frac{\rho_{jam}}{2})^2$ . The noise amplitude can be adjusted. The noise is constant at each time step  $j$ .
- The measurement noise has a zero-mean multivariate normal distribution with a covariance matrix  $R_j$  at timestep  $j$ .

## 2.4 Description of the main algorithm

As the cells do not have the same length, we choose to fix  $\Delta t$  to verify the CFL condition for stability of the Godunov numerical scheme for the minimum cell length. Therefore, in the case of a Greenshields flow, we have :

$$\Delta t \leq \frac{\min(\Delta x)}{v} \quad (49)$$

To be more accurate, we fix  $\Delta t$  to a value inferior to the one given by Eq. (49) and that divides the time step for which we have measurements. In our



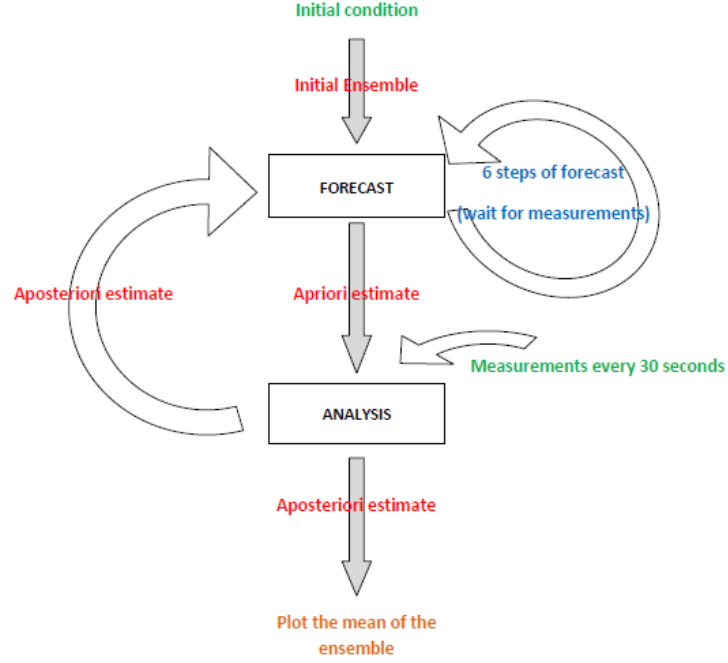


Figure 3: Flow chart describing the main algorithm

specific study, it turns out that we get measurements every 30 seconds and that  $\Delta t = 5$  seconds verifies Eq. (49). Therefore, we can represent the algorithm we implement by the flow chart on Figure 3.

### 3 Analytical Results

#### 3.1 Verification of the CTM script

**Problem description and analytical solution** In section (1.2), we explained how to deal with the Riemann problem analytically. Here, we derive the analytical solution of a well-posed problem so as to use it as a reference to verify our implementation of the algorithms described previously. Our next simulations were done over 100 [units of time]. The full problem for which we derived solution is the following:

$$\begin{cases} \rho_t + q'(\rho)\rho_x = 0 \\ q(\rho) = v\rho(1 - \frac{\rho}{\rho_{max}}) \end{cases} \quad (50)$$

with, for the simulation,  $v = 1$  and  $\rho_{max} = 4$ .  
with the initial conditions :

$$\begin{cases} \rho(x, 0) = 1 & \text{if } x \leq 0 \\ \rho(x, 0) = 1 + x & \text{if } x \in [0, 1] \\ \rho(x, 0) = 2 & \text{if } x \in [1, 10] \\ \rho(x, 0) = 2 + 2(x - 10) & \text{if } x \in [10, 11] \\ \rho(x, 0) = 4 & \text{if } x \in [11, 20] \\ \rho(x, 0) = 1 & \text{if } x \geq 20 \end{cases} \quad (51)$$

In this particular case, the problem is well-posed because the analytical approach enables to work on an infinite space domain. Therefore, the characteristics span the whole time-space plane. After deriving all the equations of the characteristics, shocks and expansion waves, we reach the following closed form of the solution:

$$\begin{cases} \rho(x, 0) = 1 & \forall (x, t) \in abACDE \\ \rho(x, 0) = \frac{1+x-vt}{1-\frac{vt}{2}} & \forall (x, t) \in bAc \\ \rho(x, 0) = 2 & \forall (x, t) \in cACBd \\ \rho(x, 0) = 2(1 + \frac{x-10}{1-vt}) & \forall (x, t) \in dBE \\ \rho(x, 0) = 4 & \forall (x, t) \in eBCDf \\ \rho(x, 0) = 2(1 - \frac{x-20}{vt}) & \forall (x, t) \in EDfF \\ \rho(x, 0) = 1 & \forall (x, t) \in Ffg \end{cases} \quad (52)$$

The plot of this solution is shown on Figure 4.

**Numerical solution** From Figure 5 , we can ensure that the initial condition propagates correctly with the numerical scheme we used. Indeed, the shocks and zone of expansion waves clearly appear. To have a better idea on how the numerical scheme performs, we plot the difference of the numerical and the analytical result on Figure 6 .

We notice from these plots that the numerical solution almost perfectly matches with the analytical approach. The only areas of significant error are the

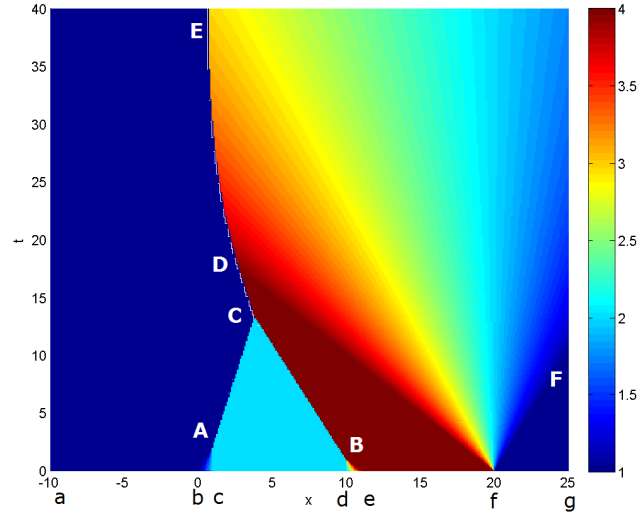


Figure 4: Analytical solution of the Riemann problem given by Eq.(50) and (51)

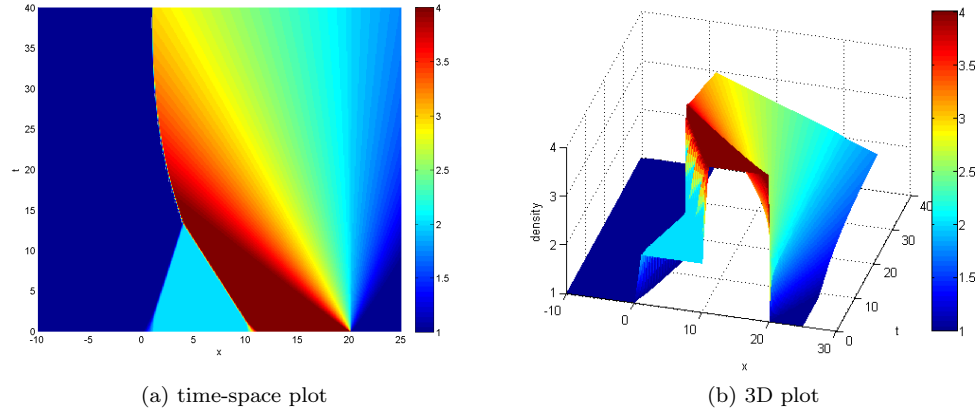


Figure 5: Numerical results for the Riemann problem given by Eq.(50) and (51)

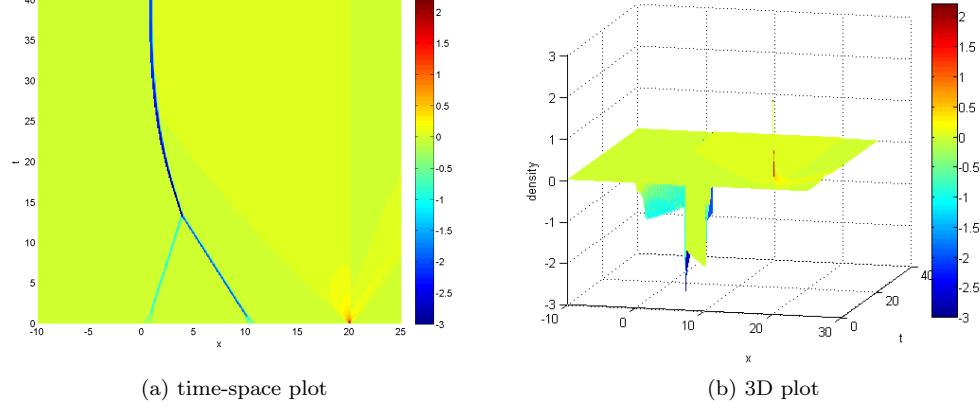


Figure 6: Numerical error of the  $\rho$ -CTM model for the Riemann problem given by Eq.(50) and (51)

areas of the time-space domain where the shocks occur. Indeed, the analytical solution has 'infinite slope' (discontinuity in  $\rho$ ) which cannot be achieved by the numerical scheme. Then, the result obtained is very satisfying. All the more as this simulation was run with an ' $r$ ' parameter of the Godunov scheme close to the limit value defined by the CFL condition. (see (1.3.2))

**Verification of the CTM for Daganzo flow:** In our script, we use an overloaded function `rho_CTM` that executes the CTM for the LWR equation with either Greenshields flow or Daganzo. Figures 7 and 8 show the numerical solutions for different values of critical density for a Daganzo flow. The other values ( $v$  and  $\rho_{max}$ ) are common with the other simulations.

From this test, we can conclude that the overloading of the function `rho_CTM` in our code performs well for both Daganzo and Greenshields.

### 3.2 EnKF-implementation verification

Once our  $\rho$ -CTM implementation proves to be efficient, we can test our implementation of the EnKF and vary some parameters (state noise, initial covariance of the ensemble, initial guess, number of steps) so as to identify their respective influence. In this part, we will use the analytical solution as both measurements and reference to check that the EnKF converges to the analytical solution. So as to make the study more interesting, we will also vary the number of cells in which we have a sensor and the frequency at which we get measurements.

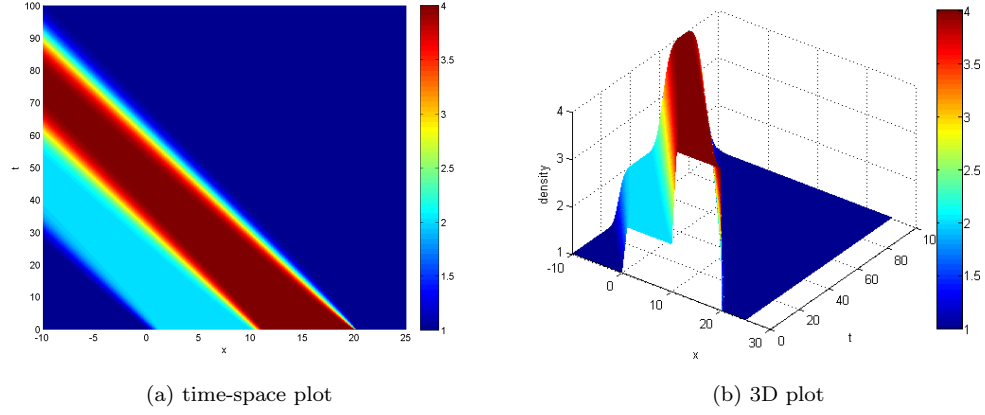


Figure 7: Propagation of the initial condition Eq.(51) with Daganzo flow for  $\rho_c = 1$ .

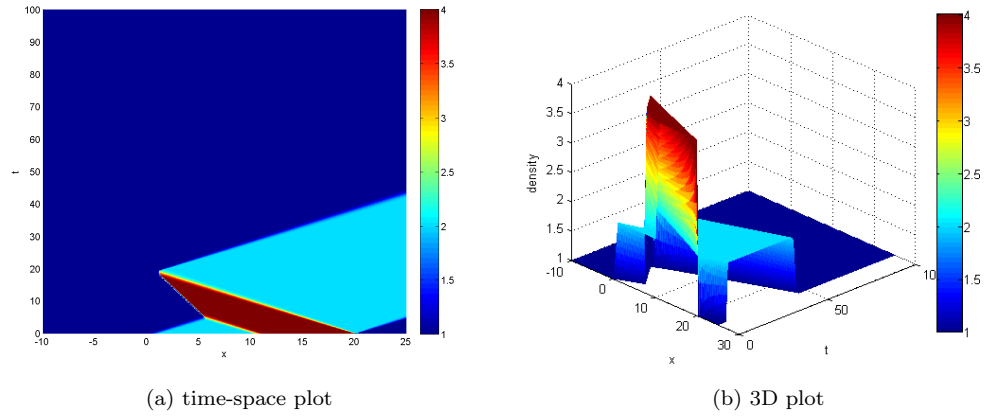
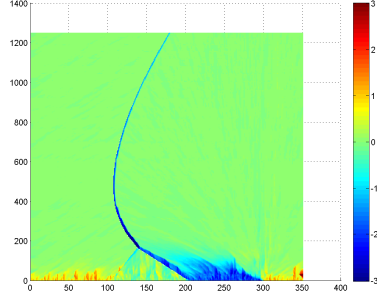
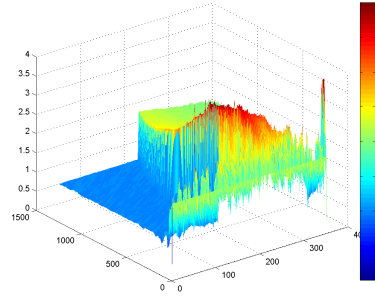


Figure 8: Propagation of the initial condition Eq.(51) with Daganzo flow for  $\rho_c = 2$ .

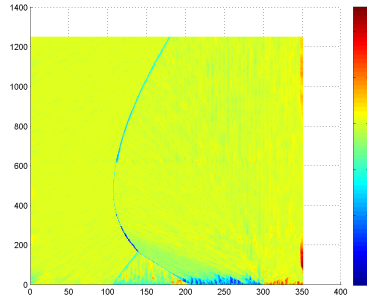


(a) time-space plot

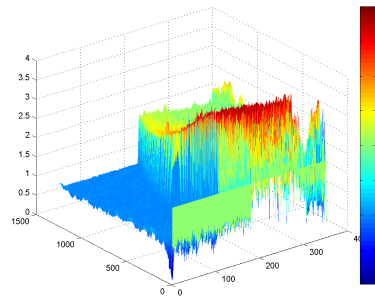


(b) 3D plot

Figure 9: Test EnKF, influence of the state noise : 1%



(a) time-space plot



(b) 3D plot

Figure 10: Test EnKF, influence of the state noise : 5%

**Influence of the state noise:** Figures 9, 10 and 11 show the influence of the state noise on the result. In this case, the percentage of cells which have a sensor is 20%, we have measurements 20% of the time steps, and, the measurement noise and the initial ensemble both have a '5% amplitude'. We simulated for a state noise of 1%, 5% and 10%. (All percentages are standard deviation)

From the observation of these figures, we can infer that it is better to keep the state noise small to reach a 'smooth' numerical solution with a better 'steady-state'. Although the numerical solution may reach the analytical solution faster with a more significant noise, it is better to keep it small. Indeed, the speed of convergence of the numerical simulation with data assimilation can be improved with a better initial guess of density or a bigger initial noise to generate the ensemble - to 'catch' extreme values of density. As regards the measurements noise, we decided to fix it at 5%. In a real-life problem, this value would be imposed by sensors properties, and, therefore, this is not a parameter we can

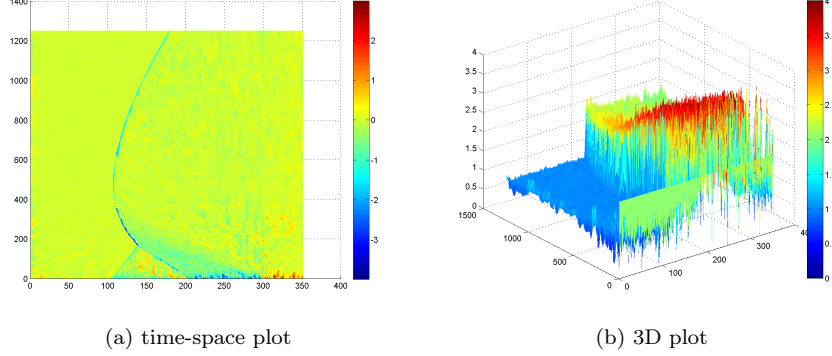


Figure 11: Test EnKF, influence of the state noise : 10%

tune.

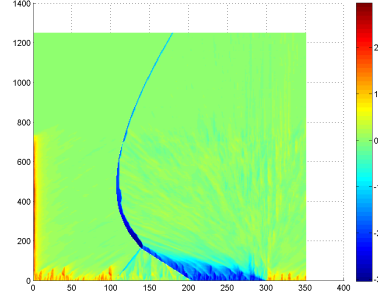
**Modified Forecast** In our first simulations, before having tuned the value of the state noise, we had trouble obtaining 'smooth' solutions. We realized that the state noise was one of the main reason. Indeed, as we make sure that at each time step the density stay between 0 and  $\rho_{max}$ , the addition of a state noise after the forecast may result in a biased noise. Therefore, we tried another approach in which we also propagate the state noise using the CTM. In other words, Eq(39) becomes:

$$\rho_f^n(k) = \mathcal{M}(\rho_a^{n-1}(k) + \eta^n(k)) \quad (53)$$

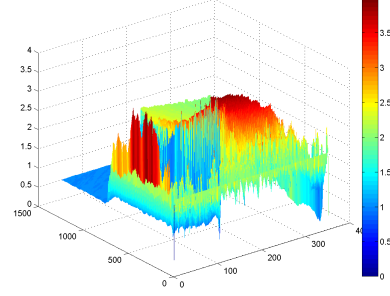
Finally, it turns out that the results are very similar. The modified EnKF even performs worse. Then, the most important factor, to obtain smooth estimates of density, seems to be the covariance of the state noise itself. The presence of the state noise is necessary but its value needs to be very small so as to obtain nice results. Figures 12 , 13 and 14 show the result of the same simulation as the 3 previous Figures, using, this time, the modified EnKF forecast.

As regards the number of measurements and the frequency at which you get them, there are obvious trade-offs between computation and accuracy. However, for small percentages (in time and space) the EnKF still converges to the analytical. Indeed, Figure 15 shows again the efficiency of the Ensemble Kalman Filter.

**Conclusion on the implementation of the EnKF** From all these observations, it seems like high values of noise - in a reasonable range ( $<10\%$ )- enable faster reaction to abrupt changes in density whereas small values guarantee a smoother and more accurate solution after some time. Therefore, for our experiments on real data, we preferred small values of state noise.

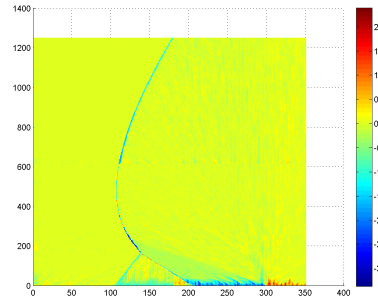


(a) time-space plot

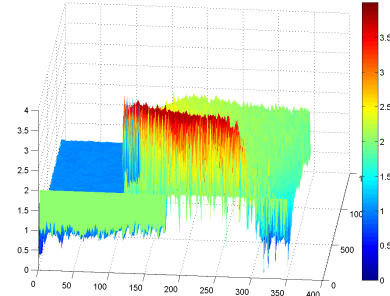


(b) 3D plot

Figure 12: Test EnKF modified, influence of the state noise : 1%

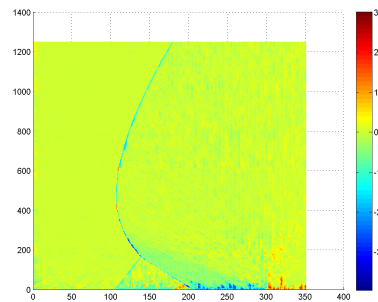


(a) time-space plot

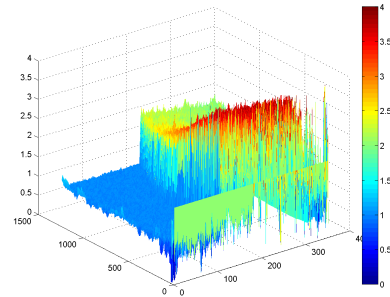


(b) 3D plot

Figure 13: Test EnKF modified, influence of the state noise : 5%



(a) time-space plot



(b) 3D plot

Figure 14: Test EnKF modified, influence of the state noise : 10%



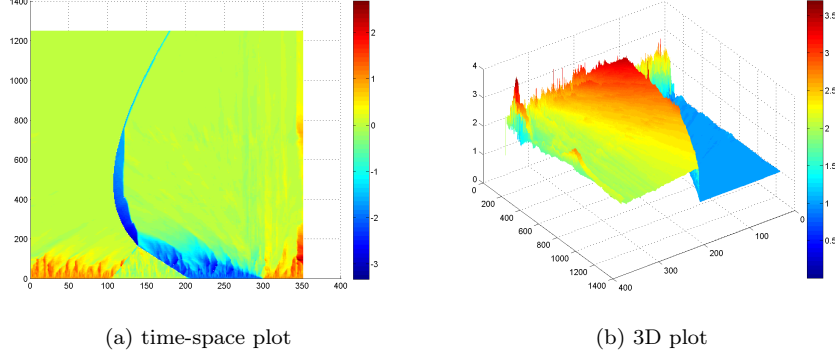


Figure 15: Test EnKF influence of the quantity of measurements : 10% in  $t$  and  $x$

## 4 Bluetooth Validation and Empirical Results

To evaluate the validity of our approach, we use vehicle travel times obtained by Bluetooth sensors. This section details our validation methodology, the Bluetooth data extraction process and the comparison process between our model output and the Bluetooth output.

### 4.1 Methodology

The idea is to compare travel times obtained via Bluetooth sensors on the one hand and via the density map output of our model on the other hand. This density map can be transformed to a velocity map by simply using the inversion formula which is determined by the choice of the fundamental diagram. Let us recall that our traffic estimation model uses traffic counts from loop detectors, which after appropriate filtering enables us to get density data measurements. Because there is no measurable ground truth value for instantaneous point densities, we use travel times obtained via Bluetooth sensors as a proxy. This methodology has been developed as part of the California Center For Innovative Transportation *Hybrid Traffic Data Roadmap project*, in which we were involved as part of our individual research. Figure 16 captures the data validation framework.

One important aspect of this project was the decision on where to implement this methodology. Three highway segments have been spotted that provide areas with varying historical velocity data and varying sensor penetration rates (the percentage of cars that can be detected) allowing us to validate the performance of the density estimation under different conditions :

1. Highway I-880 between Winton Avenue and Stevenson Blvd
2. Highway I-15 between Beech Avenue and Bellegrave Avenue

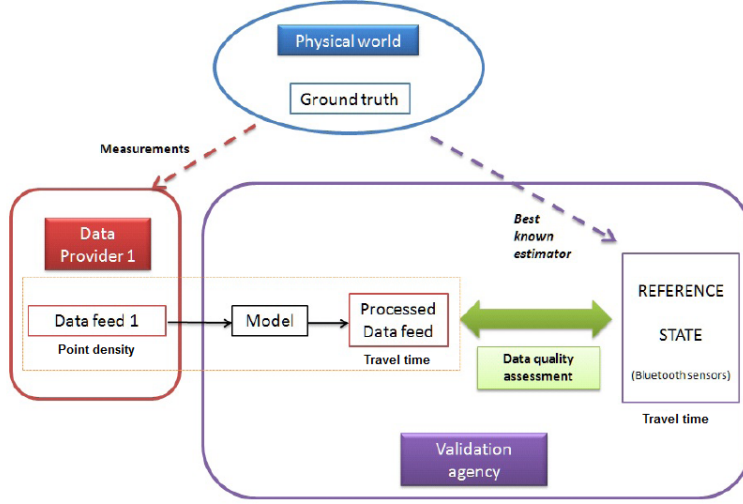


Figure 16: Data Validation Framework

### 3. Highway I-15 between Los Angeles and Victorville

We will show the results for the first experiment in Section 4.3.

For more details about the process see **Validating the use of probe data for highway traffic estimation in the Mobile Millennium highway model using Bluetooth measurements and Implementation of a Reference State using Bluetooth Technology**.

## 4.2 Bluetooth data extraction

### 4.2.1 Data feed

Figure 17 shows where the Bluetooth sensors are on the map.

A Bluetooth data set is defined by :

- a number  $N$  of Bluetooth readers along with their fixed coordinates.
- the start and end dates of the experiment.
- the entire set of sensor's detections of Bluetooth devices.

Every day the Bluetooth sensors send emails with zip files containing all their detections for the day with the corresponding time stamp and MAC id. Their average scan duration is 5 seconds.

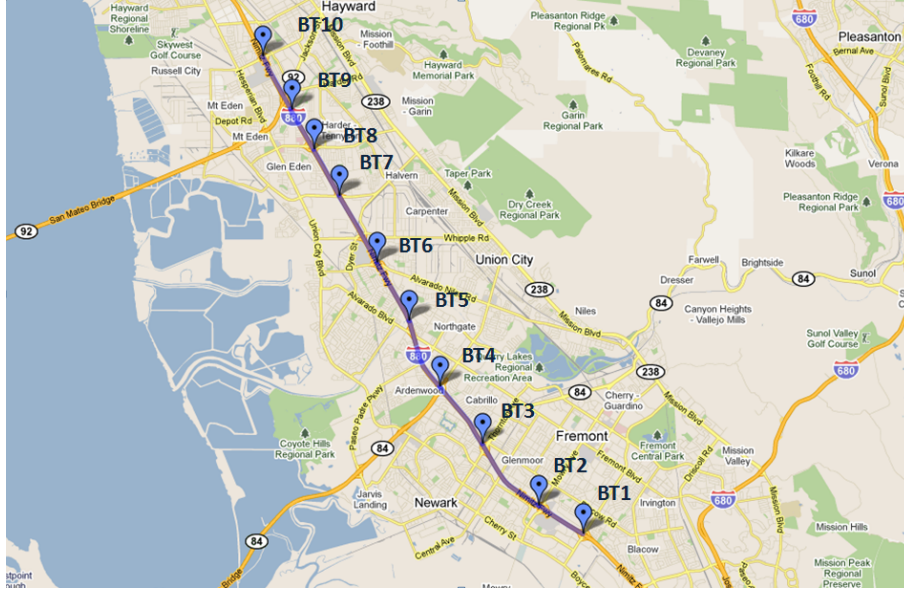


Figure 17: Map of Bluetooth sensors in Highway I-880

#### 4.2.2 Filter

The first filter's responsibility is to couple the detections of the same device by consecutive BLUFAX readers. The result of this matching is the list of all the detected trips. But some of these trips may be meaningless (e.g. a car that has been stopped for a while and then starts off again). The second responsibility is then to filter the trips to get rid of the travel time outliers. For this we use an interquartile range filtering technique. Eventually all the trips detected during the validity period are stored in one table.

#### 4.2.3 Travel time comparison

The task is basically to compute in parallel the aggregated Bluetooth travel times and highway travel times, and, then, to compute the differences between the two. While doing the analysis one should always keep in mind that there are multiple sources of errors :

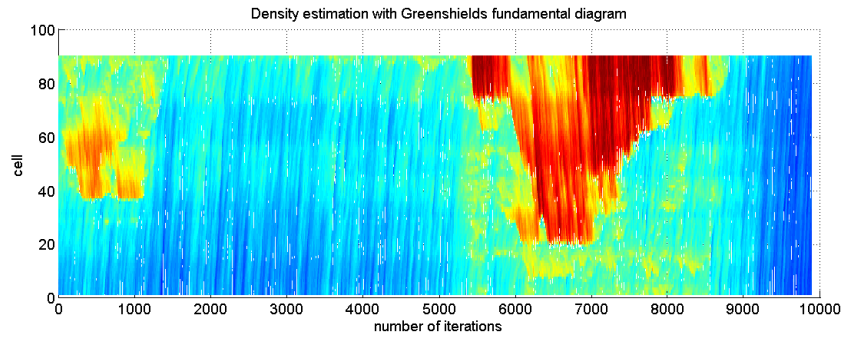
- Model error : The flow theory that gives birth to the LWR equation makes a number of simplifying assumptions that generally do not hold in reality.
- Bluetooth measurement errors can alter travel time estimations and therefore result in errors in the validation process
- Possible bias in Bluetooth equipped vehicles

### 4.3 Results

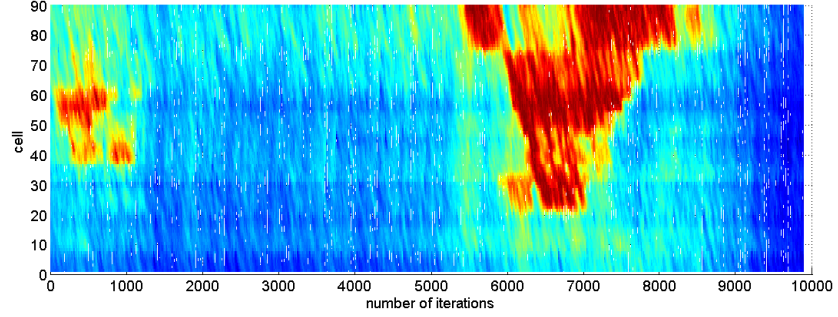
In this section we compare the results from our model to the Bluetooth output. To be able to compare these two, we convert the Bluetooth travel times to Bluetooth mean velocities with a 15-minute aggregation period and we convert the densities to velocities using the Greenshields formula  $v = v_f(1 - \frac{\rho}{\rho_{max}})$ . We do this for the first highway segment mentioned in Section 4 on I-880.

#### 4.3.1 Test on I880

Figure 18 shows our results for two different fundamental diagrams. They look really similar and therefore a quantitative analysis would be needed in order to decide on which fundamental diagram is better to use on a large-scale real-time implementation. Figure 19 shows the accordance between the results of our model and the Bluetooth output. They both show the severe congestion at the evening rush hour on Monday 5th, March 2012. After checking on Google Maps, it actually corresponds to a bottleneck on Highway I880. We can easily see the backward propagation wave which agrees with traffic theoretical principles. The quantitative analysis comparing our results and the Bluetooth travel times is in process and is therefore part of future work.

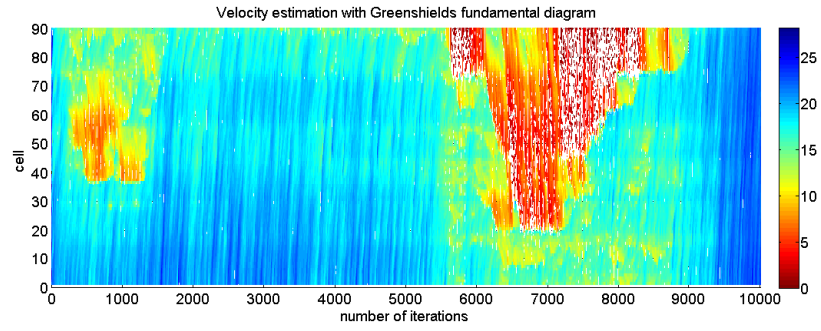


(a) Density output for Greenshields fundamental diagram

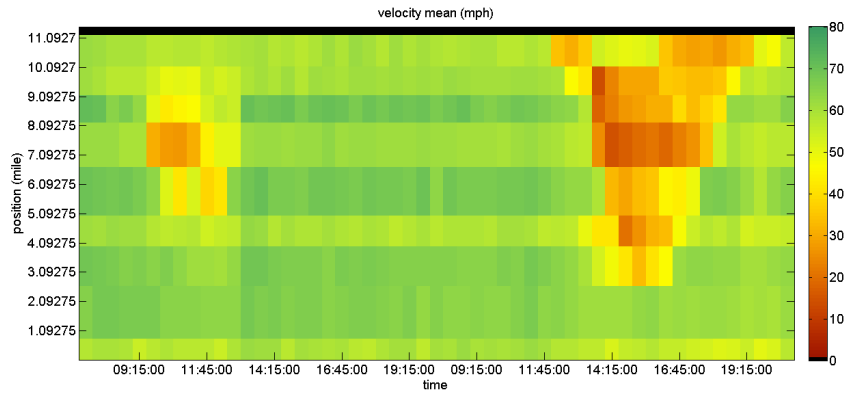


(b) Density output for Daganzo fundamental diagram

Figure 18: Comparison of the output for two fundamental diagrams



(a) Velocity output for Greenshields fundamental diagram



(b) Velocity output for Bluetooth data

Figure 19: Comparison of model output with Bluetooth output

## Conclusion

This has been an exciting project showing a nice application of the distributed parameters system theory. The results that we get are really encouraging and we see that the Ensemble Kalman Filter seems to perform well for real density data assimilation. In this report we first introduced the existing traffic estimation theory with density data assimilation. Then we presented our algorithm and its implementation. We showed exhaustive qualitative and quantitative analysis by comparing our results with a known analytical solution. This analysis validated our implementation and provided interesting insights on the relative influences of the involved parameters. Finally we explained our methodology for an experimental validation of the model on real data that were collected on Californian highways and retrieved from the Mobile Millennium database. Once again the results were encouraging and this the reason why we would like to pursue this project in the future. The next steps are the extension of our implementation to road networks as well as a deeper analysis of the results and tuning of all the parameters involved. We have already developed a script that automatically constructs roads and applies our algorithms, that should enable us to implement large-scale traffic estimation shortly.

## References

- [1] R. Ansorge. What does the entropy condition mean in traffic flow theory? *Transportation Research Part B: Methodological*, 1990.
- [2] G. Evensen. *Data Assimilation: The Ensemble Kalman Filter ()*. Springer, 2007.
- [3] Mauro Garavello and Benedetto Piccoli. *Traffic Flow on Networks*. American Institute of Mathematical Sciences, 2006.
- [4] S.K. Godunov. A finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics. 1959.
- [5] B. D. Greenshields. A study of traffic capacity. *Proceedings of the 14th annual meeting of the Highway Research Board*, 1934.
- [6] R. J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhuser Basel, 1992.
- [7] M. J. Lighthill and G. B. 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*, pages 317–345, 1955.
- [8] P. I. Richards. Shock waves on the highway. *Operations Research*, 4:42–51, 1956.
- [9] I. S. Strub and A. M. Bayen. Weak formulation of boundary conditions for scalar conservation laws: an application to highway traffic modeling. *Int. J. Robust Nonlinear Control*, 2006.
- [10] G. B. Whitham. *Linear and nonlinear waves*. Wiley-Interscience, 1999.
- [11] D. B. Work, S. Blandin, O. Tossavainen, B. Piccoli, and A. M. Bayen. A traffic model for velocity data assimilation. *Applied Mathematics Research eXpress*, 2010.