

A Fluid Dynamic Approach for Traffic Forecast from Mobile Sensor Data

Emiliano Cristiani¹, Corrado de Fabritiis², Benedetto Piccoli³

¹CEMSAC - Università di Salerno, Fisciano (SA), Italy, and IAC-CNR, Rome, Italy
emiliano.cristiani@gmail.com

² Octo Telematics SpA
corrado.defabritiis@octotelematics.com

³ IAC-CNR, Rome, Italy
b.piccoli@iac.cnr.it

Communicated by Nicola Bellomo

Abstract

In this paper we propose an algorithm for vehicular traffic forecast. The mathematical model is developed to exploit a set of experimental data, which comes from a large number of mobile sensors located on cars. Data refer to traffic flow in an urban motorway in Rome, Italy. The model is based on a fluid dynamic approach and is able to make predictions with good accuracy in free, congested, and unsteady conditions. Different kinds of initializations and algorithms are discussed, then forecast is compared with the exact solution. Traffic data were provided by Octo Telematics[©] SpA.

Keywords: traffic flow, fluid dynamic models, scalar conservation laws, vehicular traffic, experimental data.

AMS Subject Classification: 90B20, 35L65

1. Introduction.

Mathematical models for vehicular traffic have been investigated since long time. The traffic simulation problem is challenging due to the rapid changes as well as random events which can happen. It was attacked by means of macro-, meso-, and microscopic models, and a huge bulk of literature is now available, see for example the review [11] and the books [12,15]. In this paper we restrict ourself to macroscopic models, i.e. models where the flow of cars along a road is assimilated to the flow of fluid particles, for which suitable balance or conservation laws can be written. For this reason, macroscopic models are often called in the present context fluid dynamic or hydrodynamic models. This kind of models are particularly suitable to

Received 2010 01 23, in final form 2010 06 12

Published 2010 06 21

deal with large data sets, as it is our case. The interested reader can find an extensive review of the subject in the recent review [20].

Mathematical models have been also extended to graphs. In this case each road is connected to other roads, creating a network. Fluid dynamic models may be formed by a system of scalar conservation laws, and equations are coupled by means of the boundary conditions. Many mathematical results are available to deal with such a systems, from both theoretical and numerical point of view. The interested reader can find extensions to networks of the classical macroscopic approaches in several papers like, among others, [5–7,9,10,14], and in the book [8].

Besides mathematical models, an accurate forecast necessarily relies on large data sets. Indeed, without accurate statistics it is impossible to tune properly the parameters of the model. Focusing on macroscopic models based on first-order scalar conservation laws, the main issue is to compute the *fundamental diagram*, i.e. the relationship between the flux of vehicles (number of cars per unit time) and their density. We also stress that no mathematical model is able to foresee random events, like car accidents. To overcome this issue, forecast algorithms must exchange data with sensors able to measure traffic condition *in real time*.

Many experiments using real data have been reported in the literature. For example, papers [18,19] report the results of an experiment performed in a 6-km segment of an urban motorway in Paris, using a macroscopic approach. Authors use data coming from static sensors placed in 13 locations along the road. Interesting experiments are also reported in [1] and references therein, using static sensors. In [2] the authors perform some experiments in an urban road in Rome. They use data coming from 7 static sensors for each direction, which register flux and velocity of the cars every minute. The road is divided in several segments, and the algorithm is executed in each segment separately (sharing information at the boundaries). In this way it is possible to choose different model parameters in each segment, thus optimizing the algorithm. Finally, let us mention the very recent work [22], in which mobile (Lagrangian) sensors are used. Data come from GPS units installed on mobile phones of the drivers. The road length is about 11 km.

As opposite to previously cited papers, the aim of the present paper is not only to reproduce complex traffic behaviour, but also to make future forecast of traffic conditions: more precisely, in [2,18,22] the proposed algorithms are constantly fed by experimental data while running, so that simulations are corrected at regular time intervals (using a Kalman filter, for example). In this paper we use instead experimental data just to start the algorithm with a fully reliable initial condition, and then we make fore-

casts with no more supporting data. This is made possible thanks to an unprecedented number of data, provided by hundreds of thousands of mobile (Lagrangian) sensors placed directly on cars.

Our results open new possibilities in the field of infomobility. The final goal we have in mind is to provide drivers with real-time traffic forecast directly on their on-board satellite navigator. Moreover, once a reliable forecast about traffic conditions is given, accurate travel time forecast and minimum-time paths can be provided too.

The paper is organized as follows. In the next section we present and discuss the data at our disposal. In Section 3 we introduce the mathematical model and the numerical approximation of the associated equation, discussing in particular the way to compute suitable initial conditions. In Section 4 we present the numerical results and finally some conclusions end the paper.

2. Experimental data.

First, we discuss the experimental data provided by Octo Telematics^{©a} SpA. Octo Telematics[©] mounts on cars a special set of sensors, named *clear box*. Each clear box registers data at some time instants, then sends the information to the base for processing. Final data include position (by a GPS unit) and velocity of the car, as well as registration (measurement) time and processing time. When data used in this paper were collected, about 750,000 cars were equipped with a clear box in Italy, corresponding to more than 2% of the total number of cars.

We ran the algorithm on a batch of data relative to a part of the Grande Raccordo Anulare (GRA) in Rome, Italy. GRA is a circular urban motorway with three lanes for each direction. The whole GRA is in general highly congested, and long queues are usual. The part under consideration spans from exit 10 to exit 19 (18 km), see Fig. 1. The speed limit is 130 km/h. Data were collected on June 19, 2009 (Friday), from 6 a.m. to 11 a.m. We choose as initial time $t = 0$ the time of the registration of the first data. In Fig. 2 we report in the space-time all the positions registered by the sensors for each direction. The size and color of the circles denote the velocity (the smaller and darker the circle the lower the velocity). The data set is remarkably large, much more than real-time data usually available by static sensors along the roads. Here we have 13,728 measurements in the North→South direction and 18,775 measurements in the South→North direction. In both sides several formations of queues are clearly visible, as well as their disappearance.

^awww.octotelematics.it

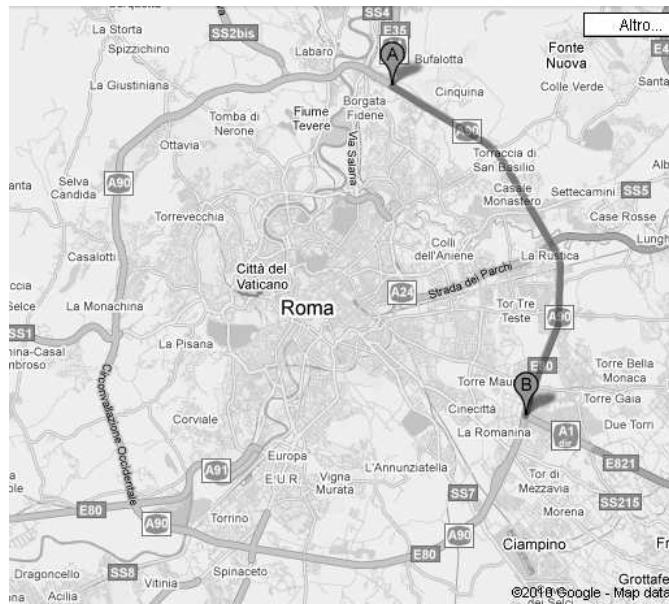


Fig. 1. Considered part of the Grande Raccordo Anulare of Rome, Italy (by Google maps).

In Fig. 3 we track a single car which covers almost the whole road segment. The average delay between the measurement and the processing time is about 10 minutes, with a minimum of 1 minute and a maximum of 1 hour. In Fig. 4 we show the measurements taken between $t = 160$ min and $t = 210$ min. Measurements in black are not yet available at time $t = 210$ min, due to the above-mentioned delay.

It is worth noting that data are affected by noise. Besides the known GPS inaccuracy, we find some still cars as well as cars in some service roads on the side. Nevertheless, we do not make any particular data filtering, in order to show the full potentiality of the model. Finally, note that velocities higher than 130 km/h were truncated to 130 km/h for technical reasons.

We did not include in the simulations the outgoing and incoming roads which heavily influence the traffic in the considered segment. Nevertheless, our results show that the model is quite robust and can provide accurate forecast even neglecting important information. In addition, as recalled in the Introduction, there are no theoretical nor practical difficulties in including a more complex topology in the model, then we expect an even better result including more roads or simply modelling the whole GRA, avoiding in this way the need of boundary conditions.

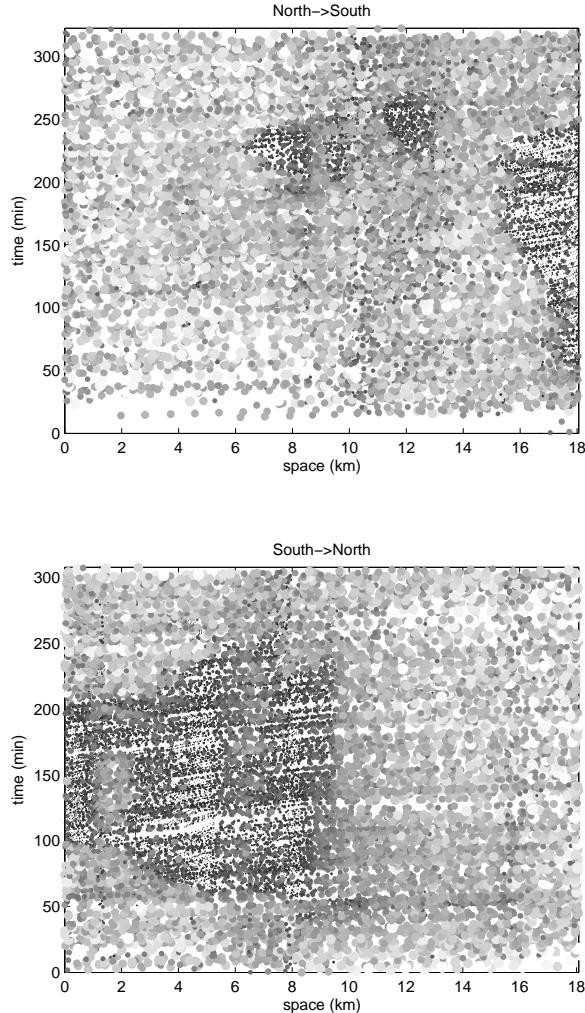


Fig. 2. Data plot, North→South direction (top) and South→North direction (bottom). The smaller and darker the circle the lower the velocity.

3. The mathematical model and its numerical approximation.

The large data set provided by Octo Telematics[©] seems to be particularly suitable for a fluid dynamic model. Indeed, considering that we deal with a three-lane motorway, in a microscopic (i.e. follow-the-leader-like) approach we should consider the three lanes individually (but not independently), including overtaking manoeuvres, and this would make the model more difficult to implement. In addition, we note that if we want to avoid such a fine treatment, an average on the three lanes should be performed,

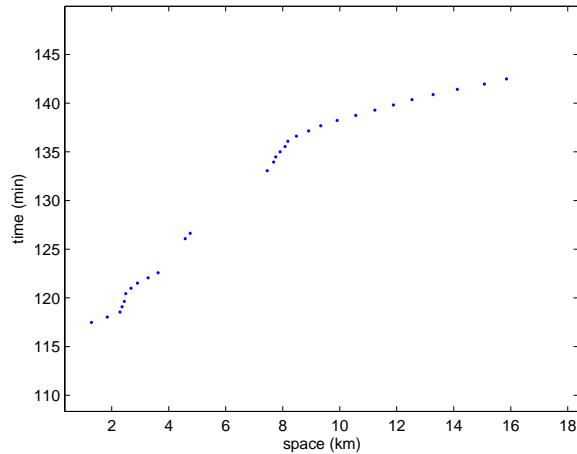


Fig. 3. Trace of a single car on the whole road segment.

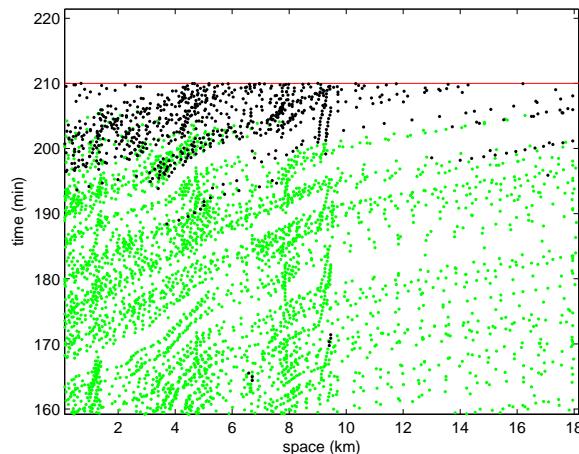


Fig. 4. Data measured between $t = 160$ min and $t = 210$ min (South→North direction). In black, data not yet available at time $t = 210$ min.

making the model closer to a macroscopic one.

We use a model based on the classical Lighthill-Whitham-Richards (LWR) approach [17,21]. The unknown

$$\rho(x, t) : [x_{\text{entry}}, x_{\text{exit}}] \times [0, +\infty) \rightarrow [0, \rho_{\max}]$$

represents the density of vehicles at point x and time t , x_{entry} , x_{exit} being the beginning and the end of the road respectively, and ρ_{\max} the maximal

density. The density function ρ is the solution of the following first-order hyperbolic problem

$$(1) \quad \begin{cases} \rho_t + f(\rho)_x = 0, & x \in [x_{\text{entry}}, x_{\text{exit}}], \quad t > t_0, \\ \rho(x, t_0) = \rho_0(x), & x \in [x_{\text{entry}}, x_{\text{exit}}], \\ \rho(x_{\text{entry}}, t) = \rho_{\text{entry}}(t), & t > t_0, \\ \frac{\partial}{\partial x} \rho(x_{\text{exit}}, t) = 0, & t > t_0, \end{cases}$$

where $f(\rho)$ is the (given) flux, ρ_0 is the initial condition and ρ_{entry} is the inflow boundary condition. Each side of the motorway is modelled separately, since it is assumed that what happens in a direction does not influence cars flowing in the other direction (this is general not true because it is well known that an accident on one side can slow down cars on the other side). As opposite to [2], we consider the whole road as a unique segment, rather than dividing it in several parts. As a consequence, a unique flux $f(\rho)$ must be given. Again, this is done for simplicity and shows the robustness of the model.

3.1. Fundamental diagram

In such a model one of the crucial points is the choice of the flux. The flux $f(\rho)$ is given by

$$(2) \quad f(\rho) = \rho v(\rho),$$

where $v(\rho)$ is the velocity of the cars as a function of the density. A classical choice for the function $v(\rho)$ is the Greenshields velocity function

$$v(\rho) = \frac{v_{\max}}{\rho_{\max}} (\rho_{\max} - \rho)$$

where v_{\max} is the maximal velocity of the cars.

Although the Greenshields function is a reasonable choice, considering the large data set at our disposal we prefer to find $v(\rho)$ by a best-fit interpolation. To this end, we first need to find a suitable value for the maximal density ρ_{\max} in terms of car/km. Unfortunately, we cannot rely on the maximum number of cars which physically stay in 1 km because all cars do not have a transmitting box on board. Then, we introduce a grid in the space-time box of Fig. 2, and we select an apparently fully-congested cell \bar{C} (velocities between 0 and 15 km/h). The number of cars in \bar{C} is then used to define ρ_{\max} , which turns out to be 72 car/km. This value is just a representative constant of the maximal density, but it is not the true value of it. The other model parameters are chosen in accordance to the value of ρ_{\max} . Note that any difference between 72 and the true value only leads to

a rescaling of the function $v(\rho)$. For the best-fit interpolation we used the linear-in-the-parameters regression curve by Matlab, with small by-hand modifications at the left and right endpoints to better fit the data. The two functions $v(\rho)$ and $\rho(v)$ are plotted in Fig. 5, and the function $f(\rho)$ computed by (2) is plotted in Fig. 6. These experimental results can be compared, for example, with those in [2–4,12,13].

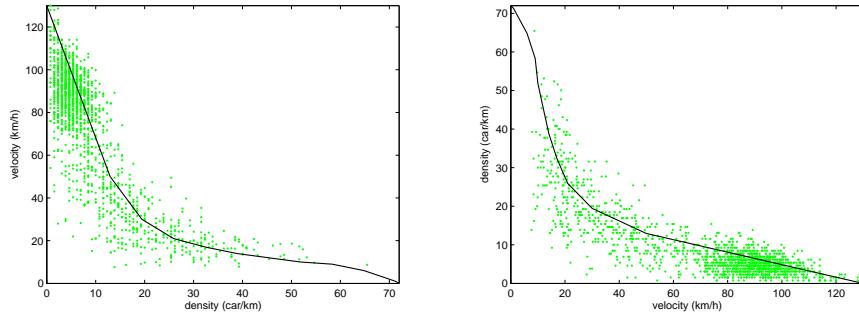


Fig. 5. The functions $v(\rho)$ (left) and $\rho(v)$ (right). Data and best-fit interpolation.

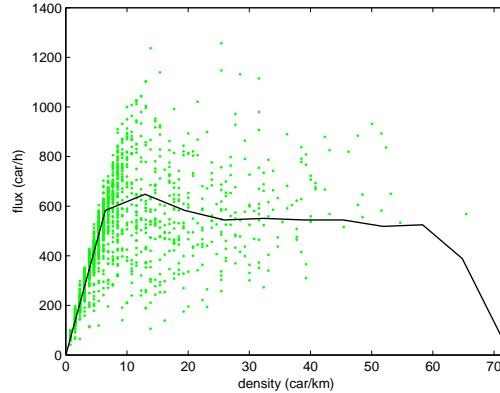


Fig. 6. The function $f(\rho)$. Data and resulting interpolation.

3.2. Numerical approximation

In order to develop a numerical scheme for equation (1), we define as usual a numerical grid in $[x_{\text{entry}}, x_{\text{exit}}] \times [t_0, t_f]$ where t_f is the final time. We denote by Δx the space step size and by Δt the time step size. We denote by N_x and N_t the number of space and time steps respectively, and

by (x_i, t_n) , $i = 1, \dots, N_x$, $n = 1, \dots, N_t$ the grid points. For a function u defined on the grid we write $u_i^n = u(x_i, t_n)$.

Equation (1) is discretized by means of the Godunov scheme. A detailed description of the scheme can be found in several books and papers (see for example [16] or [5]) so we do not report it here. As it is well known, the Godunov scheme needs a CFL condition to be satisfied. Due to the velocities into play (up to 130 km/h), we choose a rather large space step, namely $\Delta x = 700$ m, and a rather fine time step, namely $\Delta t = 9$ sec. Unfortunately, this grid makes it impossible an accurate computation of the numerical error because some cells may have no real data available. To overcome this problem, after computation we average the solution to a rougher grid with $\Delta t = 2.5$ min.

The numerical solution of equation (1) on the grid is denoted by ρ_i^n , and its extension (by linear interpolation) on all over the space-time box is denoted by $\tilde{\rho}(x, t)$. Using the function $v(\rho)$ previously computed, we get the average approximate velocity $\tilde{v}(x, t) = v(\tilde{\rho}(x, t))$.

We also denote by σ_i^n the average density in the cell (i, n) computed by means of the data and of the function $\rho(v)$. Finally, we denote by $\hat{\sigma}_i^n(\bar{n})$ the average density in the cell (i, n) computed by means of the data processed before time $t_{\bar{n}}$. If $\hat{\sigma}_i^n(\bar{n})$ cannot be defined (because of missing data), its value is set to N/A.

3.3. Initial condition

As it can be seen in Fig. 2, if we choose $t_0 = 0$, we do not have enough data to compute a reliable approximation of ρ_0 at every point of the road. This is true also if we postpone the beginning of the simulation at some time $t_0 > 0$. To overcome this issue, we choose $t_0 > 0$, and then we use *all data available at that time* to compute a suitable initial condition ρ_0 . Note that not all data registered within time t_0 are already available at t_0 , due to the delay between measurement and processing. Hereafter we denote by n_0 the time step corresponding to t_0 .

In order to compute ρ_0 , we have tried two different strategies:

1. We start the simulation at time $t = 0$ with a groundless pre-initial condition $\rho(x, 0) \equiv \rho_{\max}/2$. Then, by means of the numerical scheme, we compute the approximate solution until time t_0 , correcting it with experimental data. More precisely, once ρ_i^n is computed, we check if $\hat{\sigma}_i^n(n_0)$ is not N/A, i.e. if there are available data the approximate solution can be corrected with. If this is the case, we set

$$\rho_i^{n,\text{corrected}} = \lambda \hat{\sigma}_i^n(n_0) + (1 - \lambda) \rho_i^n$$

where $\lambda \in [0, 1]$. The weight λ is chosen depending on how many data concurred to the computation of the average value $\hat{\sigma}_i^n(n_0)$ (the more the data the more reliable the value). In this fitting stage, the solution assimilates all the available data, and becomes reliable all over the road. Then, the function $\tilde{\rho}(x, t_0)$ is used as initial condition, and no more data are used to correct the solution.

2. For every $i = 1, \dots, N_x$, we define

$$(3) \quad \phi_i(n, m) := \frac{1}{\sum_{k=0}^m C_i(k, n, m)} \sum_{k=0}^m C_i(k, n, m) \hat{\sigma}_i^k(n)$$

where $C_i(k, n, m)$ is an exponentially decreasing function of $(t_m - t_k)$ if $0 < (t_m - t_k) \leq 60$ min, and $C_i(k, n, m) = 0$ if $t_m - t_k > 60$ min or $\hat{\sigma}_i^k(n)$ is N/A. Roughly speaking, $\phi_i(n, m)$ is a weighted average of the data measured in the hour preceding time t_m and available at time t_n .

Due to the delay between measurement and processing, the most up-to-date density function which can be computed by the data at time t is that at time $t - 10$ min (approximately). To make this evident, in the following we denote by t_0^+ the time $t_0 + 10$ min and by n_0^+ the time step corresponding to t_0^+ . So we refer to t_0^+ as the time at which the simulation is actually done while numerical simulation starts at time $t_0 = t_0^+ - 10$ min, with the initial condition

$$(4) \quad \rho_i^{n_0} = \phi_i(n_0^+, n_0).$$

As a consequence of this delay, the solution at time t_0^+ (current time) is actually a 10-minutes forecast. Similarly, a q -minutes forecast in the future is actually a $(q+10)$ -minutes forecast in the simulation.

Remark 3.1. In the absence of a predictive model like (1), the function ϕ defined in (3) can be used to give short-time forecast of the traffic conditions. More precisely, it is possible to compute the velocity at any time t (past or future) by means of a weighted average of the data measured in the hour preceding t and processed before the current time. We will refer to this quantity as a *static forecast*. This approach is advantageous because it gives results in real time, and, as expected, is quite effective in steady traffic conditions.

3.4. Boundary conditions

The function $\rho_{\text{entry}}(t)$ is not known for $t > t_0$, and it depends on what happens behind the road segment under consideration. As usual in traffic models, the lack of boundary conditions is a relevant issue which cannot be

completely solved, unless we restrict ourself to nonrealistic situations like a circular road with no intersections. We just note that boundary conditions become unimportant in points quite far from the boundaries, because there the density depends only on what happens in the considered road segment (at least for a certain time). In our simulation we simply assume that in $x = x_{\text{entry}} = 0$ the traffic conditions remain unchanged during the simulation, i.e. $\rho_{\text{entry}}(t) = \rho_{\text{entry}}(t_0)$ for $t > t_0$. In addition, we assume Neumann zero boundary condition in $x = x_{\text{exit}} = 18$ km. The latter choice corresponds to the fact that cars touching x_{exit} just exit the road and they do not have effect any more. Numerically speaking, to handle Neumann condition a $(N_x + 1)$ -th ghost node is added at the end of the road and then we set $\rho_{N_x+1}^n = \rho_{N_x}^n$ for every n .

4. Numerical results.

In this section we present numerical results obtained by means of the model described above. Regarding the initial condition, experiments clearly suggest that the second method we described in Section 3.3 performs better than the first one, so we report only the results obtained by the second one. We believe that this is mainly due to the fact that we do not filter data.

The most interesting and challenging events are with no doubt the formation of queues, as well as their disappearance. We test our algorithm on these cases, showing that the model is able to foresee the change of the velocities with good accuracy.

In the following figures the exact solution σ (real data) is plotted with squares, the approximate LWR solution $\tilde{\rho}$ is plotted with a solid line, and the static forecast (see Remark 3.1) is plotted with a cross-dashed line.

Test 1. As first test, we run the algorithm choosing $t_0^+ = 78$ min and the South→North direction, for a 20 minutes forecast (i.e. until $t=98$ min). In Fig. 7-left we show the initial condition, which is relative to the time $t_0 = 68$ min. In Fig. 7-right we show the outcome at the end of simulation. A flattening effect is clearly visible, which tends to make the LWR solution piecewise constant. Nevertheless, it is evident that the scheme is able to foresee the formation of the queue in the first part of the road (compare the result with data plot in Fig. 2-bottom). As expected, the static forecast is completely wrong in the first part of the road while it is quite accurate in second part.

Let us denote by the set of indices $I \subset \{1, 2, \dots, N_x\}$ a portion of the considered road. In order to compute the error made by the algorithm, we define e_I^n as the absolute value of the difference between the approximate

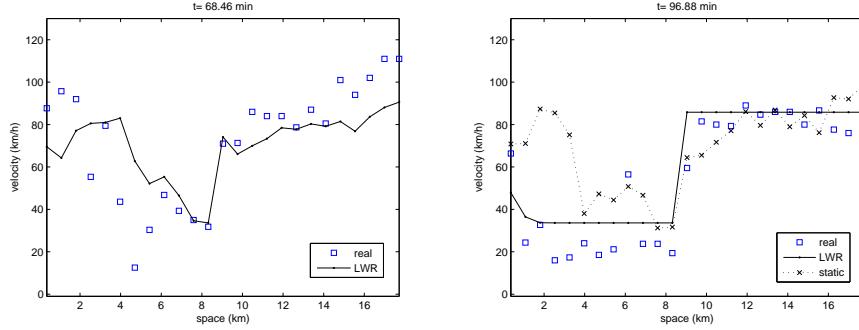


Fig. 7. Test 1. Initial condition at t_0 (left) and solution at $t_0^+ + 20$ min (right).

and the exact velocity at a given time step n , making an average on the portion I of the road,

$$e_I^n := \frac{1}{|I|} \sum_{i \in I} |v(\rho_i^n) - v(\sigma_i^n)|.$$

In Fig. 8-left we show the error e_I^n as a function of n , with $I = \{1, \dots, [\frac{N_x}{2}]\}$ (first half of the road) and in Fig. 8-right with $I = \{[\frac{N_x}{2}] + 1, \dots, N_x\}$ (second half of the road). The black spot on the horizontal axis corresponds to

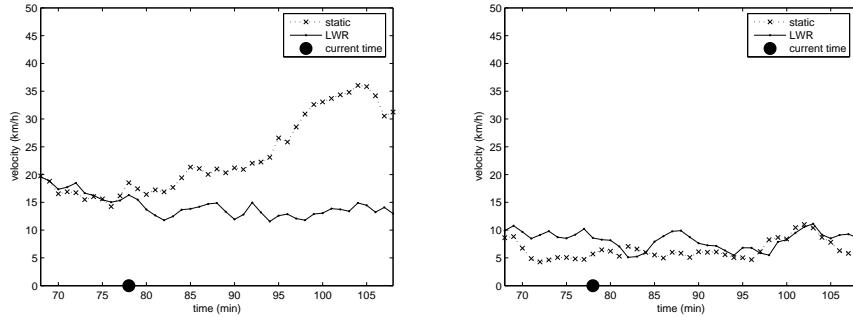


Fig. 8. Error for test 1. First half of the road (left) and second half of the road (right). Time goes from 78-10 min to 108 min.

the time t_0^+ . The error of the LWR forecast is about 15 km/h in the segment where the queue is formed, and about 8 km/h in the steady segment. The static forecast reaches an error of 35 km/h in the unsteady segment, while it behaves like LWR elsewhere. Such a behaviour is found also in the following tests.

Test 2. In the second test we try to foresee the extinction of a queue. We choose again the South→North direction and we set $t_0^+ = 198$ min. We run the algorithm until the final time $t = 238$ min. In Fig. 9 we plot the initial condition and the final result as before. It is evident that the scheme

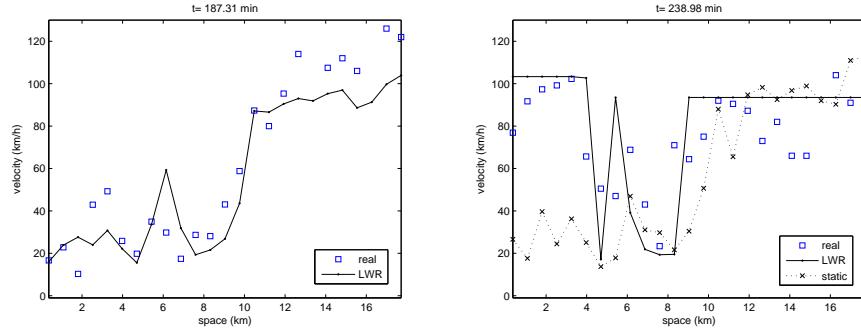


Fig. 9. Test 2. Initial condition at t_0 (left) and solution at $t_0^+ + 40$ min (right).

is able to follow the changes in the traffic condition: in particular we go from a situation where the first part of the road is highly congested while the second part is rather free, to a situation where all the road is free but a small zone immediately before the 8th kilometre. Again, the static forecast shows the expected behaviour.

Test 3. In the third test we focus again on a queue formation. This time we choose the North→South direction and we set $t_0^+ = 126$ min. We run the algorithm until the final time $t = 188$ min. In Fig. 10 we plot the initial condition and the final result. As in test 1, the scheme is able to catch

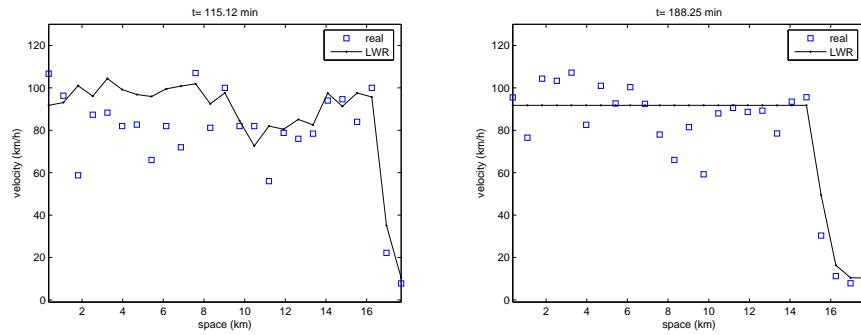


Fig. 10. Test 3. Initial condition at t_0 (left) and solution at $t_0^+ + 62$ min (right).

the formation of the queue at the end of the road. Note that this time the

queue is created quite slowly, and sometimes remains unchanged for 10-20 minutes. Nevertheless, the scheme is quite accurate, this is probably due to the nice choice of the flux $f(\rho)$. In other less congested zones the scheme is not able to catch small perturbation and the flattening effect dominates. Here the static forecast is not available since the simulation runs more than 1 hour in the future.

Distributions of errors. In order to have a global idea of the abilities of the model, we compute the error distribution for several forecasts. In more detail, we run simulations with $t_0^+(p) = 30 + 5p$ minutes, $p = 1, 2, \dots, p_{\max}$. The initial delay of 30 minutes allows to preserve at least 20 minutes old data to compute the initial condition, while p_{\max} is computed in such a way that we have some data for the computation of the error at the end of simulation. Every simulation runs Δn time steps in the future. Denoting by $n_0^+(p)$ the time step corresponding to $t_0^+(p)$, we define an error variable E_i at every space node $i = 1, \dots, N_x$ and $p = 1, \dots, p_{\max}$,

$$E_i^p(\Delta n) := \frac{1}{v\left(\rho_i^{n_0^+(p)+\Delta n}\right)} - \frac{1}{v\left(\sigma_i^{n_0^+(p)+\Delta n}\right)}.$$

$E_i^p(\Delta n)$ is the error made in approximating the time needed to cover 1 km at the velocity computed at node i at the end of simulation. In our opinion this error measurement is particularly meaningful (probably more than the absolute or relative error over the velocity) because it is the information that drivers really pay attention to. In Fig. 11 we report the distributions of the errors $\{E_i^p(\Delta n)\}_{i,p}$ for four choices of Δn , corresponding to a 0, 15, 30 and 45 minutes LWR forecast, respectively. As it can be seen, the distribution has a classical Gaussian behaviour, rather concentrated around 0. As the forecast becomes more challenging (30 and 45 minutes) the graph becomes less symmetric, but it shows in any case a peak in 0.

Travelling times. Once a forecast of the traffic condition is available, a forecast of the travelling time can be made. We have in mind a scenario where a driver asks for the time she/he needs to reach some destination, and the answer is given by means of a real-time traffic forecast. To this end, it is important to note that even if we can rely on an accurate forecast of the average velocity (in space and/or in time), this is not enough to get an accurate estimate of the travelling time. Let us assume for example that a driver has to cover 10 km. If the estimated average velocity is 60 km/h, the estimated travel time is 10 min. On the other hand, if the real velocity

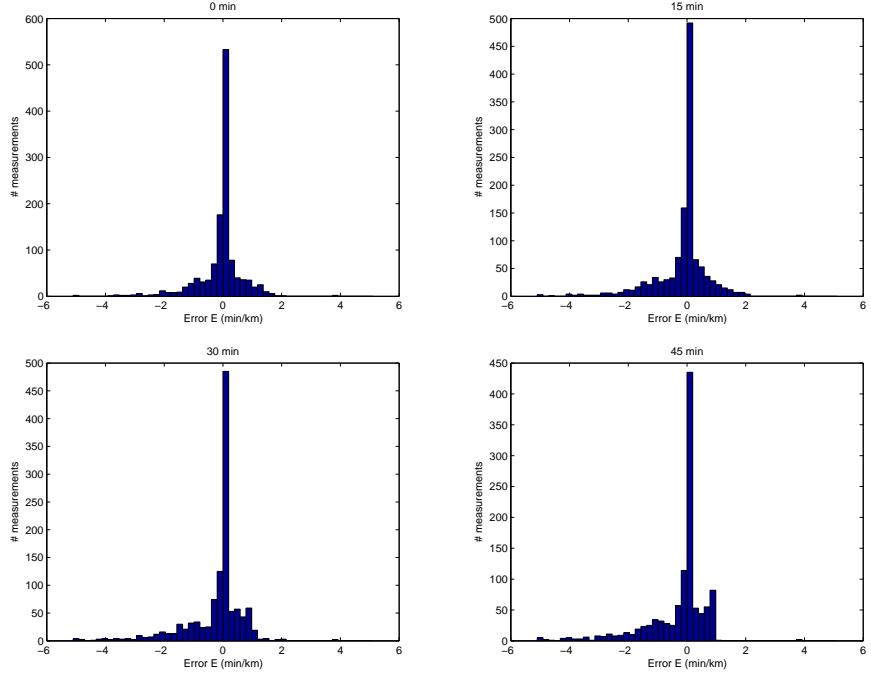


Fig. 11. Travelling time error distribution for a 0 (top-left), 15 (top-right), 30 (bottom-left) and 45 (bottom-right) minutes LWR forecast.

is 20 km/h for the first half of the road and 100 km/h for the second half (60 km/h in average), the travel time is 18 min (8 minutes difference). It is clear how important it is to compute the velocity on a space grid which is as fine as possible. Moreover, the prediction itself is crucial, because traffic conditions can change during the journey, leading to large changes in the travelling time.

In order to test this feature, we use the results in Test 1 and 2 to make a prediction of the travelling time needed to cover 14 km (from the 2nd km to 16th km of the road). The equation to solve is

$$\begin{cases} \dot{x}(t) = \tilde{v}(x(t), t) \\ x(0) = 2 \text{ km.} \end{cases}$$

until $x \leq 14$ km. The velocity is explicitly time-dependent, because the traffic conditions change in time. The equation is solved by the explicit Euler scheme. Table 1 summarizes the results. As expected, the static forecast underestimates the travelling time in case of queue formation while overestimates the travelling time in case of queue extinction. The LWR forecast gives better results, especially in the latter case. This is probably due to

Table 1. Exact and estimated travelling times.

Parameters	exact (min)	static forecast (min)	LWR forecast (min)
t_0^+ as in Test 1	24	13	17
t_0^+ as in Test 2	17	23	16

the fact that in a congested situation (low velocities into play), a small error over the velocity forecast can lead to a large error over travelling time forecast. This effect is instead mitigated in uncongested situations (high velocities into play).

Conclusions

In this paper we have performed traffic forecast up to about 70 minutes, using data from Octo Telematics[©] mobile sensors. The large data set allowed us to have an accurate and reliable initial condition for the algorithm, and then to run the simulation for long time in the future. Other recent experiments of this kind [2,22] are instead limited to model the traffic behaviour feeding regularly the algorithm with new incoming data. The proposed model is able to foresee the changes in traffic conditions, thus allowing to make travelling time forecasts. CPU time is less than 20 seconds for Matlab simulation of 1-hour in real time.

REFERENCES

1. J. H. Banks, M. R. Amin, M. Cassidy, and K. Chung, Validation of Daganzo's Behavioral Theory of Multi-Lane Traffic Flow: Final Report, UC Berkeley, Research Reports, California Partners for Advanced Transit and Highways (PATH), Institute of Transportation Studies, 2003.
2. S. Blandin, G. Bretti, A. Cutolo, and B. Piccoli, Numerical simulations of traffic data via fluid dynamic approach, *Appl. Math. Comput.*, vol. 210, pp. 441–454, 2009.
3. I. Bonzani and L. Mussone, Stochastic modelling of traffic flow, *Math. Comput. Modelling*, vol. 36, pp. 109-119, 2002.
4. I. Bonzani and L. Mussone, From experiments to hydrodynamic traffic flow models. I. Modelling and parameter identification, *Math. Comput. Modelling*, vol. 37, pp. 1435-1442, 2003.
5. G. Bretti, R. Natalini, and B. Piccoli, Fast Algorithms for the Approximation of a Traffic Flow Model on Networks, *Discrete and Continuous Dynamical Systems*, vol. 6, pp. 427–448, 2006.
6. G. Bretti, R. Natalini, and B. Piccoli, Numerical approximations of a

- traffic flow model on networks, *Networks and Heterogeneous Media*, vol. 1, pp. 57-84, 2006.
7. C. F. Daganzo, On the variational theory of traffic flow: well-posedness, duality and applications, *Networks and Heterogeneous Media*, vol. 1, 601–619, 2006.
 8. M. Garavello and B. Piccoli, *Traffic flow on networks*, AIMS Series on Applied Mathematics, 2006.
 9. D. Helbing, J. Siegmeier, and S. Lämmer, Self-organized network flows, *Networks and Heterogeneous Media*, vol. 2, 193-210, 2007.
 10. H. Holden and N. H. Risebro, A mathematical model of traffic flow on a network of unidirectional roads, *SIAM Journal on Mathematical Analysis*, vol. 26, pp. 999-1017, 1995.
 11. S. P. Hoogendoorn and P. H. L. Bovy, State-of-the-art of vehicular traffic flow modelling, *J. Syst. Cont. Eng.*, vol. 215, pp. 283-303, 2001.
 12. B. S. Kerner, *The Physics of Traffic*, Springer, Berlin, 2004.
 13. B. S. Kerner and S. L. Klenov, A microscopic model for phase transitions in traffic flow, *J. Phys. A*, vol. 35, pp. L31-L43, 2002.
 14. J. P. Lebacque and M. M. Khoshyaran, First order macroscopic traffic flow models for networks in the context of dynamic traffic assignment, in *Proceedings of the 6th Meeting of the EURO Working Group on Transportation* (M. Patriksson and M. Labb, eds.), vol. 64, pp. 119-140, 2002.
 15. W. Leutzbach, *Introduction to the theory of traffic flow*, Springer-Verlag, New York, 1988.
 16. R. J. LeVeque, *Numerical methods for conservation laws*, Birkhäuser, 1992.
 17. M. J. Lighthill and G. B. Whitham, On kinetic waves. II. Theory of traffic flows on long crowded roads, *Proceedings of Royal Society of London Series A*, vol. 229, pp. 317-345, 1955.
 18. M. Papageorgiou, J.-M. Blosseville, and H. Hadj-Salem, Modelling and real-time control of traffic flow on the southern part of boulevard périphérique in Paris: Part I: modelling, *Transpn. Res. A*, vol. 24A, pp. 345–359, 1990.
 19. M. Papageorgiou, J.-M. Blosseville, and H. Hadj-Salem, Modelling and real-time control of traffic flow on the southern part of boulevard périphérique in Paris: Part II: Coordinated on-ramp metering, *Transpn. Res. A*, vol. 24A, pp. 361–370, 1990.
 20. B. Piccoli and A. Tosin, Vehicular traffic: A review of continuum math-

- ematical models. In *Encyclopedia of Complexity and Systems Science* (R. A. Meyers, ed.), vol. 22, pp. 9727–9749. Springer New York, 2009.
21. P. I. Richards, Shock waves on the highway, *Operation Research*, vol. 4, pp. 42–51, 1956.
 22. D. B. Work, S. Blandin, O.-P. Tossavainen, B. Piccoli, and A. M. Bayen, A traffic model for velocity data assimilation, *Appl. Math. Res. Express*, vol. 2010, pp. 1–35, 2010.