

MIT Open Access Articles

*Real-Time Multi-Sensor Multi-Source Network Data Fusion Using
Dynamic Traffic Assignment Models*

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: Huang, E. et al. "Real-time multi-sensor multi-source network data fusion using dynamic traffic assignment models." Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on. 2009. 1-6. © 2009 IEEE

Published Version: <http://dx.doi.org/10.1109/ITSC.2009.5309859>

Publisher: Institute of Electrical and Electronics Engineers

Permanent Link: <http://hdl.handle.net/1721.1/54705>

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

Terms of use: Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.



Real-Time Multi-Sensor Multi-Source Network Data Fusion Using Dynamic Traffic Assignment Models

E. Huang, C. Antoniou, Y. Wen, M. Ben-Akiva

Intelligent Transportation Systems Laboratory
Massachusetts Institute of Technology, United States
{enyang; costas; wenyang; mba} @mit.edu

J. Lopes, J. Bento

Brisa, Instituto Superior Técnico
Technical University of Lisbon, Portugal
{jlopes, jbento} @brisa.pt

Abstract — This paper presents a model-based data fusion framework that allows systematic fusing of multi-sensor multi-source traffic network data at real-time. Using simulation-based Dynamic Traffic Assignment (DTA) models, the framework seeks to minimize the inconsistencies between observed network data and the model estimates using a variant of the Hooke-Jeeves Pattern Search. An empirical validation is provided on the Brisa A5 Inter-City Motorway in the West coast of Portugal. The real-time network data provided by loop detectors, video cameras and toll counters is collected and fused within DynaMIT, a state-of-the-art DTA system. State estimation is first performed, yielding consistent approximation of the network condition. This is then followed by network state forecast, showing significantly improved Normalized Root Mean Square Error (RMSN) over alternative predictive systems that do not use real-time information to correct themselves.

Keywords: Multi-Sensor Fusion, Simulation and Modeling, Travel Information and Guidance, Traffic State Analysis and Prediction

I. INTRODUCTION

Recent developments in sensor technologies and their applications have fundamentally changed the landscape of traffic state analysis using DTA models. Modern sensors deployed throughout the network provide rich set of information on traffic network conditions. There are usually different types of sensors, each providing local inferences on certain factors for specific parts of a network.

In order to take full advantage of all available data, one must develop appropriate methods within DTA models to fuse the data into information so it can be used by travelers. However, conventional DTA predictive systems face great challenges when dealing with large arrays of modern traffic sensors. This is because existing systems need to work with a network of sensors that have the following characteristics: 1) Deployed with uneven density within a network. 2) Heterogeneous; 3) Provide highly correlated data; 4) Report at non-uniformed resolutions; 5) Report at different frequencies.

This paper describes a data fusion framework using DTA models. The framework allows systematic combining of multiple sensor sources to generate complete traffic network state using micro-simulation. An implementation of the framework is developed and validated on a real world inter-city highway in Portugal.

The remaining part of this paper is organized as follows. Section II presents a literature review of the state-of-the-art. Section III discusses the proposed data fusion framework using DTA model. Section IV presents a case study and experiment results. Section V concludes this study.

II. RELATED WORKS

The topic of multi-sensor data fusion within DTA models has been discussed extensively in the literature. Ashok and Ben-Akiva [1][2] formulated the real-time OD estimation and prediction problem as a state-space model and solved it using a Kalman Filtering algorithm. The authors' use of deviations of OD flows from their historical values provides an elegant framework for incorporating structural OD information (generated during off-line calibration) into the on-line process. The approach has been implemented in the DynaMIT DTA system (Antoniou et al. [3]; Ben-Akiva et al. [4]). Bierlaire and Crittin [5] outlined an efficient solution algorithm for the OD estimation problem. Van der Zijpp [6] combined volume counts with trajectory information obtained from automated license-plate surveys for the estimation of OD flows. A measurement equation for the trajectory counts is specified and split probabilities are estimated from combined link volume counts and trajectory counts.

Antoniou et al. [7] presented a methodology for the incorporation of AVI information into the OD estimation and prediction framework, which was later extended by Antoniou et al. [8] to allow for the consideration of any type of available surveillance data. Zhou and Mahmassani [9] developed a non-linear ordinary least-squares estimation model to combine and fuse AVI counts, link counts and historical demand information and solved this as an optimization problem.

Antoniou et al. [10] formulated the problem of on-line calibration of the speed-density relationship as a flexible state-space model and presented applicable solution approaches. Three of the solution approaches [Extended Kalman Filter (EKF), Iterated EKF, and Unscented Kalman Filter (UKF)] are implemented and applications of the methodology with freeway sensor data from two networks in Europe and the U.S. are presented.

Antoniou [11] developed an approach that jointly formulates the on-line calibration problem as a state-space model comprising transition and measurement equations. A priori values provide direct measurements of the unknown

parameters (such as origin–destination flows, segment capacities and traffic dynamics models’ parameters), while surveillance information (for example, link counts, speeds and densities) is incorporated through indirect measurement equations. The state vector is defined in terms of deviations of the calibration parameters and inputs from available estimates. Applicable solution algorithms are presented and compared in Antoniou et al. [12].

Balakrishna [13] and Balakrishna et al. [14] formulated the off-line calibration problem of DTA models as an optimization problem that jointly estimates demand and supply parameters. Furthermore, Balakrishna proposed the SPSA algorithm for solving this problem. The advantage of SPSA is that it offers superior computational performance and excellent scalability properties.

Vaze et al. [15] applied Simultaneous Perturbation and Stochastic Approximation (SPSA) method to the off-line DTA calibration problem. In their study sensor data involving vehicle count and Advanced Vehicle Identification (AVI) are used.

Wang and Papageorgiou [16] presented a general approach to the real-time estimation of the complete traffic state in freeway stretches. They use a stochastic macroscopic traffic flow model in a state-space formulation, which they solve using an Extended Kalman Filter. The formulation allows dynamic tracking of time-varying model parameters by including them as state variables to be estimated. Random walk is used as transition equations for the model parameters. A detailed case study of this methodology is presented in [17].

III. MODEL-BASED DATA FUSION

The difficulties of fusing sensor data in a complex traffic network often come from heterogeneity, correlation, non-uniform distribution of/among the sensors. To develop a framework of multi-sensor multi-source data fusion, we begin by presenting the simulation-based DTA models that are used to realistically estimate and predict traffic conditions. These simulation models work in real-time and support dynamic traffic control, incident management, route guidance, and demand prediction. Figure 1 shows a typical DTA simulation system.

The system accepts three types of inputs: 1) Surveillance information. These are the real-time sensor data from multiple sources; 2) A-priori parameter values. These are the default model parameters that are obtained prior to the deployment of the system. Such parameters are usually obtained from a process of model training that is undertaken before the deployment of the system, and are therefore an indication of the model’s long-term average performance. 3) Network representation and historical data. These are the specifications of the network, sensors, OD-demand, travel time, flow rate, and social economic factors, etc.

At the center of the system is the state estimation and prediction. These processes typically involve a number of iterations of interactions between demand simulator and supply simulator. The demand simulator consists of the Origin to Destination (OD) flows and the models that capture travel demand choice behavior (e.g. route choice, mode

choice, and departure time). The supply simulator is usually a meso-scope traffic simulation model that loads the output of demand simulator and produces simulated ground-truth outputs such as simulated travel time, point vehicle count, link/segment speed, and link/segment density.

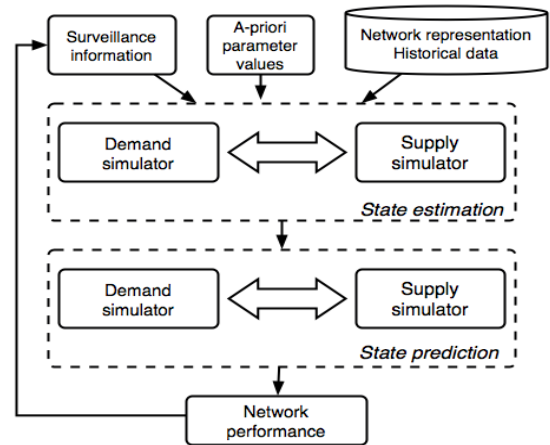


Figure 1. A typical DTA system. The system takes surveillance information, a priori parameters, and historical data and conducts state estimation and state prediction [4].

A. Framework

The inputs described above are used to “correct” results computed by the demand and supply simulators. The corrections take place when there are observed inconsistencies between the inputs and their *simulated counter-parts*. Example of such observed inconsistencies are: 1) observed sensor link speed and simulated link speed, 2) observed sensor point to point travel time and simulated point to point travel time, 3) observed sensor vehicle count and simulated vehicle count. Such inconsistencies are due to two types of errors in the model: 1) Errors between the true model parameters and the default model parameters, such as true link free flow speeds and the model’s default link free flow speeds. 2) Errors between true model input value and the actual value used, such as the true OD demand level and the actual level used in the model.

For real-time multi-sensor multi-source data fusion, the first type of inconsistencies and its remedies are of particular interests. Conventional approaches that only address the second type of error at real-time tend to perform poorly compared to the state-of-the-art, which jointly minimizes both types of errors [11][13]. To construct the correction procedure, DTA models first need to be modified such that each sensor source has its simulated counter-part. This is done by first examining the sensors that are already deployed on a network, and then add implementations of “virtual sensor” that are functionally identical to these deployed sensors within the DTA model. For example, if there is a sensor on a network that reports point vehicle counts, one shall add an implementation of a “virtual sensor” at the same location in the DTA model that reports point vehicle counts.

The DTA model often comprises a large number of parameters and inputs to adjust. These parameters and inputs affect directly the performance of the model, and therefore, affect the reported values from the “virtual sensors”. To

correct the inconsistencies between real-time sensor data and those reported by our “virtual sensors”, one will need to alter the model parameters and inputs accordingly. Formally, the framework can be formulated in the following way:

$$(\mathbf{x}_1, \dots, \mathbf{x}_n, \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_m) = \arg \min [z_1(\mathbf{M}_h, \mathcal{M}_h) + z_2(\mathbf{x}_h, \mathbf{x}_h^a) + z_3(\boldsymbol{\beta}_h, \boldsymbol{\beta}_h^a)] \quad (1)$$

$$\mathcal{M}_h = f(\mathbf{x}_1, \dots, \mathbf{x}_n, \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_m, G_1, \dots, G_o) \quad (2)$$

$$\left. \begin{array}{l} l_i^x \leq \mathbf{x}_i \leq u_i^x \\ l_i^\beta \leq \boldsymbol{\beta}_i \leq u_i^\beta \end{array} \right\} \forall h \in \{1, 2, \dots, H\} \quad (3)$$

In words, we are estimating values for the input data $\mathbf{x}_1, \dots, \mathbf{x}_n$ and model parameters $\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_m$ at time interval h , so that the sum of the following three components is minimized: 1) Inconsistencies between sensory observations \mathbf{M}_h and their simulated model counter-part \mathcal{M}_h , which is obtained from a DTA model that takes input data, model parameters and real-time surveillance information; 2) Inconsistencies between input data $\mathbf{x}_1, \dots, \mathbf{x}_n$ and their a priori beliefs; 3) Inconsistencies between the model parameters $\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_m$ and their a prior beliefs. In addition, each \mathbf{x} and $\boldsymbol{\beta}$ are constraint to a feasible region for each time interval h .

The use of a priori values in a real-time data fusion framework is worth explanations. The prior beliefs of the input data $\mathbf{x}_1^a, \dots, \mathbf{x}_n^a$ and the model parameters $\boldsymbol{\beta}_1^a, \dots, \boldsymbol{\beta}_m^a$ are usually obtained by a process known as *offline calibration* [13][14]. In offline calibration, surveillance data from a long period of time is used. The surveillance log is thus expected to reflect the typical network behaviors.

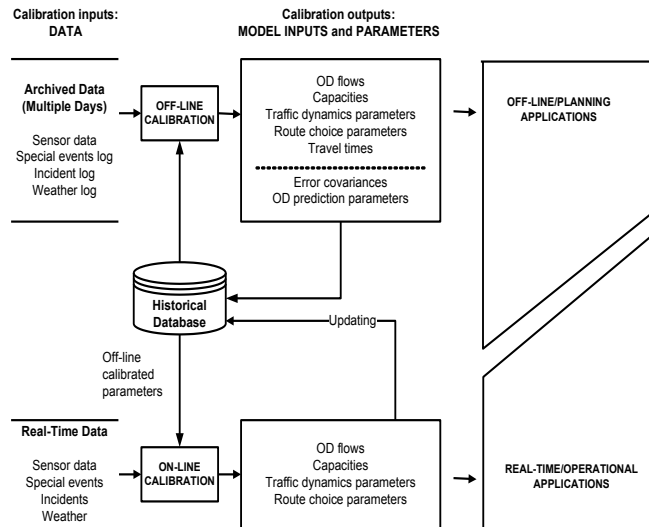


Figure 2. The process flow diagram of off-line calibration and on-line calibration of DTA models [18].

However, this is not good enough. The input data and model parameters generated by offline-calibration process represents a long-term average of network conditions and are therefore insensitive to prevailing conditions. Such conditions, such as peak-hour traffic, road incidents, special events, adverse weathers conditions, usually last for a relatively shorter period. To make the predictive system sensitive to fluctuation of local traffic conditions, *real-time surveillance data* must be collected and fused within a DTA

model. This is done by applying equations (1) - (3) at operational time. This process of combining various sensor sources at operational time to obtain consistent network state estimation is known as model *online calibration*. Figure 2 shows schematic of both off-line and on-line calibrations.

Equations (1) - (3) may be solved by a number of direct optimization algorithms. These methods evoke on the objective equation and outputs optimized model inputs and parameters. A list of these methods and their application to the DTA model calibration problem can be found at [15]. This paper, however, implements the proposed framework using a variant of the heuristic-based Pattern Search algorithm.

B. Solution Approach

The optimization procedure that solves equations (1) - (3) is based on Hooke and Jeeves [19]. The algorithm only requires function evaluations and does not require derivatives. The algorithm works by creating a set of search directions iteratively. (The created search direction spans across the entire search space). In an N-dimensional inputs/parameters calibration problem, this requires at least N linearly independent search directions. Among the N possible combinations of searches some combinations might reach the minima faster than others. The algorithm comprises of two types of moves 1) Exploratory move and 2) Pattern move. The exploratory move systematically finds the best point in the vicinity of the current point. Results obtained by exploration are used to perform pattern moves. The following procedures are adopted from [20].

1) Exploratory move

Assuming that a current solution (the initial vector of parameters and inputs subject to calibration) is obtained and denoted by x^c , the perturbation amount for variable i (x_i^c) is denoted by Δ_i . The exploratory move algorithm is given in Algorithm A.

ALGORITHM A THE EXPLORATORY MOVE THAT EXPLORES THE VICINITY OF THE CURRENT POINT IN SEARCH SPACE

Calculate $f = f(x)$, $f_+ = f(x_i + \Delta_i)$ and $f_- = f(x_i - \Delta_i)$	Step 1
Find $f_{\min} = \min(f, f_+, f_-)$. Set X corresponds to f_{\min} .	Step 2
Is $i = N$? If no, set $i = i + 1$ and go to step 1; Else x is the result and go to step 4	Step 3
If $x \neq x^c$, success; Else failure	Step 4

Figure 3. In the exploratory move, the current point is perturbed in positive and negative directions along each variable one at a time, and the best point is recorded. The current point is changed to the best point at the end of each variable perturbation. If the best point is the same point as the beginning point the exploration move is a failure. If the best point is a different point then the exploration move returns success.

2) Pattern move

The pattern move calculates a new point from the current best point along a direction connecting the previous best point $x^{(k-1)}$ and the current base point $x^{(k)}$ as follows:

$$x_p^{(k-1)} = x^{(k)} + (x^{(k)} - x^{(k-1)}) \quad (5)$$

The newly calculated point may result in better or worse objective function values. If the new point results in a better value the pattern move is a success and the pattern move process is repeated with the new point. Otherwise the new point is rejected and Δ_i is reduced. The exploratory process restarts with a new Δ_i . The entire Pattern Search algorithm is given in Algorithm B.

ALGORITHM B	PATTERN MOVE ALGORITHM MOVES CURRENT POINT TOWARDS THE DIRECTION OF THE EXPLORED POINT	
Choose starting point x^0 , variable increments Δ_i ($i=1,2,\dots,N$), step reduction factor $a>1$, termination parameter, ε . Set $k = 0$.		Step 1
Performe an exploratory move with x^k as the base point. Let x be the outcome of the exploratory move. If exploratory move is a success, set $x^{k+1} = x$ and go to step 4; Otherwise go to step 3.		Step 2
If $\ \Delta\ < \varepsilon$ terminate and return the best x found. Otherwise $\Delta_i = \Delta_i / a$ for $I = 1, 2, \dots, N$ and go to step 2.		Step 3
Set $k = k+1$ and perform the pattern move: $x_p^{(k-1)} = x^{(k)} + (x^{(k)} - x^{(k-1)})$		Step 4
Perform another exploratory move using $x_p^{(k-1)}$ as the base point. Let the result be $x^{(k-1)}$		Step 5
If $f(x^{(k-1)}) < f(x^k)$ go to step 4; Otherwise go to step 3.		Step 6

Figure 4. The search strategy is simple and the algorithm has low memory requirements; Only two points need to be stored at any iteration, x^k and x^{k+1} .

IV. CASE STUDY AND RESULTS

A. The Brisa A5 Motorway

The network used in this analysis is the Brisa A5 motorway. (A5 - Auto-estrada da Costa do Estoril) It is a 25 km inter-urban expressway section between Lisbon and Cascais. The motorway is primarily equipped with toll collection systems, surveillance camera counters and inductive loop detectors. The schematics of these sensor locations are shown in Figure 5.

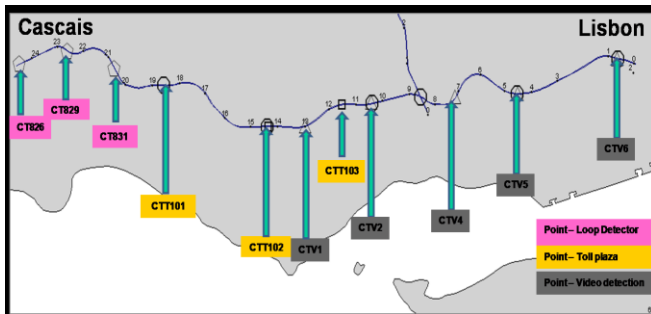


Figure 5. This figure shows sensor types and sensor locations. The three types of sensors are: loop detectors (CT826 - 831), toll collectors (CTT101 - 103), and surveillance video camera (CTV1 - 6).

B. Data Collection

Data from multiple types of sensors for two specific days were collected – April-28-2009 (Collection A) and April-27-2009 (Collection B). In addition, we are in the possession of Collection C, which was recorded in April 2008. The data

collected in 2009 contains a whole day collection of sensor CT826-831, CTT101-103. The data collected in 2008 contains a whole day collection of sensor CT826-831, CTT 101-103 and CTV1-6. The data from both collections are then manually aggregated into fixed processing intervals of 10 minutes. The loop detectors' counts are aggregated every 10 minutes and reported. The same aggregation process is applied to the counters deployed at toll gate and video surveillance stations. The choice of relatively long aggregation interval such as 10 minutes is motivated by a number of practical considerations, such as the accuracy and stability of OD flow estimations and speed-density relationships [13].

C. Choice of DTA System

DynaMIT is a state-of-the-art DTA system for traffic estimation and prediction developed at MIT Intelligent Transportation System Laboratory. The system is composed of two sub-systems. The first sub-system, state estimation combines the available surveillance data with historical information to estimate the current state of the entire network. Pre-Trip demand is simulated. This allows drivers with different characteristics to dynamically change their departure time, travel mode, and trip route. This is followed by OD flow estimation and network estimation. The estimated network condition is then compared to surveillance information. Inconsistent estimations are rejected and new iterations of estimation are initiated. The second sub-system is state prediction. During the pre-trip prediction stage, the DTA model predicts future traffic patterns while accounting for the drivers' responses to the provided guidance and traffic information. The predictions are based on estimates generated from the state estimation stage. The pre-trip prediction stage is then followed by a stage of OD prediction and network state prediction. The disseminated network information includes predicted flow rate, travel time, link speed and density.

D. Objectives

The first objective of this experiment is to demonstrate the effectiveness of the proposed data fusion framework by showing consistently low RMSN from a set of experimental trials. The second objective of this experiment is to demonstrate the advantage of using real-time data fusion to correct for local traffic perturbation in DTA models so that DTA models can yield more accurate forecasts versus conventional offline calibrated models which do not account for real time sensory data. To achieve these objectives, a number of experiments using the collected data were performed. RMSN is used as the performance descriptor in the experiments. RMSN measures the discrepancy between DynaMIT's estimates and the real observations of the sensor data. The formula is as follows.

$$RMSN = \sqrt{\frac{N \sum_N (y - \hat{y})^2}{\sum_N y}} \quad (6)$$

Figure 6. RMSN: Where y is the observed sensor value and \hat{y} is the simulated sensor value from DynaMIT, N is the total number of observation over the analysis period.

E. Offline Calibration Using Collection A

DynaMIT is first calibrated offline using data collection A. The offline calibrated model is then tested on the same collection without out using DynaMIT's online calibration capabilities. As expected, the model showed low RMSN during the intervals used for the experiment from 13:00 to 14:00. The data fusion model (DynaMIT with real-time sensor data feed in) was also tested on the same data collection for the same time period. The data fusion version of DynaMIT outperformed the offline only version in state estimation. However as expected, the predictions' RMSNs for both models are low and comparable. (See Table I)

TABLE I. RMSN STATISTICS OF OFFLINE ONLY AND DATA FUSION MODELS FROM THEIR TRAINING DATA SET (COLLECTION A)

	RMSN	Estimation	10-mins	20-mins	30-mins
Offline Only (%)	15.7%	15.5%	16.8%	17.0%	
Data Fusion (%)	8.8%	15.2%	13.7%	15.1%	
Improvement (%)	44.1%	1.86%	18.5%	11.1%	

F. Online Validation Using Collection B

TABLE II. ESTIMATION RMSN OF OFFLINE ONLY MODEL ACROSS CT AND CTT SENSORS USING VALIDATION DATA COLLECTION B

Sensor ID	CT826	CT829	CT831	CTT101	CTT102	CTT103
Offline Only	63.0%	34.5%	31.3%	6.4%	25.1%	12.4%

TABLE III. PERCENTAGE DIFFERENCE BETWEEN TWO RAW DATA COLLECTIONS FOR SIX SENSOR VALUES BETWEEN 13:00 TO 14:00

	13:00 13:10	13:10 13:20	13:20 13:30	13:30 13:40	13:40 13:50	13:50 14:00
CT826	64.0%	1.1%	22.5%	-53.1%	62.9%	10.1%
CT829	57.0%	7.9%	13.6%	-64.7%	58.3%	1.0%
CT831	54.7%	2.1%	20.3%	-121.3%	59.1%	7.0%
CCT101	-16.7%	1.3%	-18.1%	17.6%	-1.2%	6.1%
CCT102	-13.9%	7.3%	-10.0%	14.5%	-2.0%	2.3%
CCT103	-9.8%	5.2%	-9.6%	10.2%	-1.6%	1.6%

Next, we took both models and validated them using a data set (Collection B) that is different from the training set (Collection A). The offline only model that was calibrated using collection A was first ran. The estimation RMSN was then calculated for each of the sensors. For sensor CT829, CT831 CTT102 and CTT103, the estimations shown in Table II suggest high RMSN for the offline only model, suggesting deteriorated performance in estimation. The inconsistencies between the offline only model's estimates and the real world observed sensor data can be explained by the natural traffic fluctuations between the two days when data was collected. Practically speaking, the traffic fluctuation between two consecutive weekdays can be significant. To demonstrate this nature, Table III presents the variability of the raw sensor observations in terms of the percentage differences between the two days of data collection for each of the six sensors.

The overall RMSN for the state estimation and the three step predictions for both the offline only and the data fusion models are summarized in Table IV. The Offline calibrated model performed poorly overall due to traffic pattern shifts. However, the data fusion version of DynaMIT yielded much lower RMSN, demonstrated robust performance in overcoming changes in traffic conditions. In addition, the RMSN for the data fusion model is comparable to that reported in Table I.

TABLE IV. RMSN STATISTICS OF OFFLINE AND DATA FUSION MODELS FROM VALIDATION DATA SET (COLLECTION B)

	RMSN	Estimation	10-mins	20-mins	30-mins
Offline Only (%)	19.6	19.7	21.6	21.8	
Data Fusion (%)	8.9	15.2	14.1	16.3	
Improvement (%)	54.8	23.1	34.4	25.1	

G. Further Validation Using Random Perturbations

The difference between data collection A and B suggests significant traffic pattern fluctuations between days. As demonstrated above, DTA models must be able to correct themselves frequently while online in order to maintain accurate forecasts. In fact, significant fluctuations not only exist between normal days, but can also be caused by a number of complex external factors that are unobservable to the model. To examine the performance of data fusion models under such traffic fluctuations, an artificial perturbation experiment was developed using Collection C. The experiment consisted of 200 simulation iterations of the data fusion model. During each iteration, an artificially generated sensor data matrix was used for the data fusion model to produce RMSN statistics.

Let kernel matrix x be the K by M sensor data matrix from Collection C, where K is the number of time intervals in this analysis and M is the number of sensors sources. Let w_i be an M by M diagonal perturbation matrix whose diagonal consists of M uniform random numbers between 0.25 and 1.25 each generated independently for iteration i . (Perturbations of sensory observation in the range of -75% to +125%) The final constructed perturbation observation matrix for iteration i is given by $x w_i$. The sample average and variance RMSN of the 200 iterations for both models are shown in Table V and VI.

TABLE V. MEAN RMSN COMPARISON BETWEEN OFFLINE ONLY AND DATA FUSION MODELS WITH 200 RANDOM PERTURBATIONS

RMSN Mean	Estimation	10-mins	20-mins	30-mins
Offline Only (%)	23.0%	23.0%	22.9%	23.1%
Data Fusion (%)	12.9%	17.5%	18.8%	19.7%
Improvement(%)	44.0%	24.0%	18.1%	14.6%

TABLE VI. COMPARISON THE VARIANCE OF RMSN BETWEEN OFFLINE ONLY AND DATA FUSION MODELS WITH 200 RANDOM PERTURBATIONS

RMSN Variance	Estimation	10-mins	20-mins	30-mins
Offline Only (%)	0.19%	0.24%	0.22%	0.21%
Data Fusion (%)	0.11%	0.12%	0.11%	0.11%

V. CONCLUSION AND PROSPECTS

In this paper, a DTA model based multi-sensor multi-source data fusion framework is presented and empirically validated on the Portuguese A5 motorway.

In order to systematically fuse surveillance data from multiple sources and multiple observations over a network, a novel method is developed. Our contribution is in developing a way to use detailed simulation-based DTA models, implement corresponding “virtual sensor” in the model, and then subsequently compare the inconsistencies between data obtained from real network sensors and those “virtual sensors”. In this way, we transformed the original data fusion problem into an optimization task, with the model inputs and parameters being the subjects. The key advantage of our approach is that we are able to accept any types of sensor, that are deployed anywhere in the network, and that report at any time interval.

The case study demonstrated an application of the proposed framework in combining multi-sensor and multi-source traffic data from a real-world inter-city motorway. The case study suggests that DTA models with real-time data fusion capability tend to yield 1) Consistently low RMSN 2) Lower RMSN when compared to DTA models that are calibrated offline without online model adjustment. In addition, this result appears to be consistent in both artificial and real data trials.

Finally, the data-fusion framework was executed in real time. This requires efficient optimization algorithms within the framework. Although demonstrated real-time performance in a medium and non-trivial real-world network, the Pattern Search algorithm is not very efficient in terms of function evaluations. Application of this framework to larger network requires effective generalization of the optimization procedure to higher dimensional space. Hence alternative, less computationally cumbersome methods must be investigated in different framework implementations. The immediate next step of this study will research and compare different methods under the proposed data fusion framework.

ACKNOWLEDGEMENT

This research was made possible by the generous support of the Government of Portugal through the Portuguese Foundation for International Cooperation in Science, Technology and Higher Education and was undertaken in the MIT-Portugal Program.

REFERENCE

- [1] Ashok, K. and Ben-Akiva, M. (1993) “Dynamic O-D matrix estimation and prediction for real-time traffic management systems”. In Daganzo, C., editor, *Transportation and Traffic Theory*, pp. 465-484. Elsevier Science Publishing.
- [2] Ashok, K. and Ben-Akiva, M. (2000) “Alternative approaches for real-time estimation and prediction of time-dependent origin-destination flows”. *Transportation Science*, vol. 34, no.1, pp. 21-36.
- [3] Antoniou, C., Ben-Akiva, M., Bierlaire, M., and Mishalani, R. (1997) “Demand simulation for dynamic traffic assignment”. In *Proceedings of the 8th IFAC Symposium on Transportation Systems*, Chania, Greece.
- [4] Ben-Akiva, M., Bierlaire, M., Koutsopoulos, H. N., and Mishalani, R. (2002) “Real-time simulation of traffic demand-supply interactions within DynaMIT”. In Gendreau, M. and Marcotte, P., editors, *Transportation and network analysis: current trends*, pp.19–36. Kluwer Academic Publishers. Miscellanea in honor of Michael Florian.
- [5] Bierlaire, M. and Crittin, F. (2004) “An efficient algorithm for real-time estimation and prediction of dynamic OD tables”. *Operations Research*, vol. 52, no. 1.
- [6] Van der Zijpp, N. J. (1997) “Dynamic OD-Matrix Estimation from Traffic Counts and Automated Vehicle Identification Data”. *Transportation Research Record*, no. 1607, pp. 87–94.
- [7] Antoniou, C., Ben-Akiva, M., and Koutsopoulos, H. N. (2004) “Incorporating automated vehicle identification into origin-destination estimation”. *Transp. Res. Rec.: J. Transp. Res. Board*, vol. 1882, pp. 37–44.
- [8] Antoniou, C., Ben-Akiva, M. and Koutsopoulos, H. N. (2006) “Dynamic Traffic Demand Prediction Using Conventional and Emerging Data Sources”. *IEEE Proceedings Intelligent Transport Systems*, vol. 153, no. 1, pp. 97-104.
- [9] Zhou, X. and Mahmassani, H. S. (2006) “Dynamic origin destination demand estimation using automatic vehicle identification data”. *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 105-114.
- [10] Antoniou, C., Ben-Akiva, M. and Koutsopoulos, H. N. (2005) “On-Line Calibration of Traffic Prediction Models”. *Transportation Research Record* 1934, pp. 235-245.
- [11] Antoniou, C. (2004) “On-line calibration for Dynamic Traffic Assignment”. PhD Dissertation, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology.
- [12] Antoniou, C., Ben-Akiva, M. and Koutsopoulos, H. N. (2007) “Non-linear Kalman Filtering Algorithms for On-line Calibration of Dynamic Traffic Assignment Models”. *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 4, pp. 661 - 670.
- [13] Balakrishna, R. (2006) “Off-line Calibration of Dynamic Traffic Assignment Models”. Ph. D. thesis, Massachusetts Institute of Technology.
- [14] Balakrishna, R., Ben-Akiva, M. and Koutsopoulos, H. N. (2007) “Off-line Calibration of Dynamic Traffic Assignment: Simultaneous Demand-Supply Estimation”. *Transportation Research Record*, no. 2003, pp. 50-58.
- [15] Vaze, V. S., Antoniou, C., Yang, W., and Ben-Akiva, M. (2009) “Calibration of Dynamic Traffic Assignment Models with Point-to-Point Traffic Surveillance”. *Transportation Research Board Annual Meeting Paper #09-0732*
- [16] Wang, Y. and Papageorgiou, M. (2005) “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.
- [17] Wang, Y., Papageorgiou, M., and Messmer, A. (2007) “Real-Time Freeway Traffic State Estimation based on Extended Kalman Filter: A Case Study”. *Transportation Science*, vol. 41, pp. 167-181.
- [18] Antoniou, C., Balakrishna, R., Koutsopoulos, H. N. and Ben-Akiva, M. (2009) “Off-line and on-line calibration of Dynamic Traffic Assignment Systems”. *Proceedings of the 12th IFAC Symposium on Control in Transportation Systems*, Redondo Beach, CA, USA.
- [19] Hooke, R. and Jeeves, T. (1961) “Direct search solutions of numerical and statistical problems”. *Journal of the Association for Computing Machinery* vol. 8, no. 2, pp. 212–229.
- [20] Deb, K. (1998) *Optimization for Engineering Design*. Edition 8, PHI Learning.