TRAFFIC NETWORK STATE ESTIMATION USING EXTENDED KALMAN FILTERING AND DSMART

Frank Zuurbier.
P.O. Box 5048
2600 GA Delft
The Netherlands
f.s.zuurbier@tudelft.nl
+31 15 27 83346

Hans van Lint
P.O. Box 5048
2600 GA Delft
The Netherlands

Victor Knoop
P.O. Box 5048
2600 GA Delft
The Netherlands

All authors affiliated to Delft University of technology, Transport & Planning Department.

Abstract:
For effective (network) traffic control, such as route guidance, one of the key requirements is to derive an estimate of the current state in the traffic network. A principle method of doing so, is to combine a traffic network model with available measurement data by means of an Extended Kalman filter (EKF). One of the virtues of using an EKF is that aside from an estimate of the *mean* traffic state on each link also an estimate of the state *estimation error covariance matrix* is obtained, which reflects the uncertainty in the state estimates. This paper describes how this can be done *analytically* in the case of a first order traffic flow model and discusses some preliminary results. *Copyright©2006 IFAC*

Keywords: Extended Kalman Filtering, Traffic control, State Estimation, Networks

1. INTRODUCTION

Transportation systems are reaching the limits of their existing capacities due to the increasing demand for transportation caused by changes in activities, increasing prosperity and economic growth (Middelham F. 2005). As traffic networks around the globe become more and more congested the demand for effective and real-time traffic management is increasing. The components of an ideal network-wide dynamic traffic management (DTM) system can be roughly decomposed into three components:

a. state estimation – translating the available measurements on network links into a consistent network wide estimate of traffic conditions (e.g. in terms of flows, densities or speeds)
b. state prediction – extrapolating the most recent state estimates over time
c. Optimization – based on (the preferable predicted) state derive the optimal set of traffic control settings, where optimal is a function of a certain pre defined objective.

Where components (b) and (c) are strongly intertwined. This paper addresses the first of these components; state estimation using point measurements in the form of induction loop data and an extended Kalman filter.

Various attempts in the literature exist using Kalman filtering techniques for different kinds of state estimates. The most recent is the renaissance project (Wang Y., Papageorgiou M. et al. 2006) In this research the 2nd order traffic flow model METANET (Messmer A. and Papageorgiou M. 1990) is used in an extended Kalman filter. The traffic model is linearized numerically in order to implement it in the EKF. Another example is given in (Antoniou C., M. et al. 2004) where a flexible state space model is used to describe the density – speed relationship. This model is then estimated successfully online using an EKF. In this paper a network wide state estimate methodology is formulated based on the 1st order traffic flow model DSMART approximated by means of the Godunov scheme. Due to its simplicity, the model lends itself for analytical derivative calculation, allowing it to be implemented in an EKF. Interestingly, sometimes meaningful partial derivatives can be calculated from a cell to itself, a stream upward and stream downward cell. By also linearizing the node progression scheme, cell derivatives may also be calculated from cells from one link to those of another connected link. By doing so, the Kalman equations updating the error covariance matrix will lead to an estimate of the total network estimation error, including those links where no measurement data is available.

This paper continues as follows: first the model DSMART-SC will be explained in more detail. Following the extended Kalman filter equations are presented with a discussion on how to interpret the different variables used in the network state estimation context. The paper then continues by summarizing how the partial derivatives of the 1st order traffic flow model can be calculated in order to be used in the EKF. Finishing with a discussion of some preliminary results obtained in an experimental network setup.

2. DSMART-SC TRAFFIC MODEL

The traffic model DSMART-SC used for this study is based on the 1st order traffic flow model DSMART as described in (Zuurbier F.S. 2005) implemented in Matlab. The dynamic traffic assignment (DTA) is based on the kinematic wave model, which is a combination of the conservation of vehicle equation and the (linear) fundamental diagram. A numerical approximation of the kinematic wave model is done by means of the Godunov scheme. (Lebaque J.P. 1996). Using this approximation, links are divided into a number of cells ρ^j and the simulation is divided in a total number of T discrete time steps k with duration Δt [hrs]. Traffic within a cell is considered homogeneous and expressed in density [veh/km]. The traffic model in this study is an adapted version of the original DSMART in that it uses *turn fractions* instead of *destination specific split fractions*, hence the addition SC; single commodity.

1.1 Network description

The network G consists of a set of transport links a^t and nodes r which make up for the total $G(a^t, r)$. Based on the demand described in the OD matrix, some nodes in the network are also connected by origin links a^o and destination links a^d. These links are used to handle the OD demand enabling them to be used in the same node progression scheme as transport links. All links in the network $G(a^t, r)$ are discretized into cells based on the simulation time step duration Δt, the maximum allowed speed $V^{i,Max}$ on a link i with a cell length L^i as in (1).

$$L^i = V^{i,\max}\Delta t \qquad (1)$$

1.1 Four modelling steps

DSMART-SC uses four modelling steps to calculate the progression of flows:

i. Progress OD demand
ii. Progress links
iii. Progress Nodes
iv. Check Events

Step 1. OD demand. The demand D_k^o originating from node o, at time k expressed in [veh/hr] is

calculated based on the static $OD^{o,d}$, a periodic time slice factor τ_p and a uniform distributed error term ε as in (2).

$$D_k^o = \sum_d OD^{o,d} \tau_p \varepsilon_k \qquad (2)$$

The calculated demand is assigned to the cell belonging to the origin link at time step *k* and is progressed on the network in step 3.

Step 2. Link progression. The second step deals with progressing link density according to the Godunov scheme. Based on the cell densities ρ^i the corresponding flows f_k^i can be calculated by the fundamental diagram $f_k^i = FD(\rho_k^i)$. These flows are then used to construct the demand D_k^i and supply S_k^i values for each cell i at time k using (3),(4)

$$D_k^i = \begin{cases} f_k^i, & \rho \le \rho^{crit} \\ Q^i, & \rho > \rho^{crit} \end{cases} \qquad (3)$$

$$S_k^i = \begin{cases} f_k^i, & \rho \ge \rho^{crit} \\ Q^i, & \rho < \rho^{crit} \end{cases} \qquad (4)$$

The actual flow out of a cell i is calculated as (5):

$$q_k^{i(out)} = \min(D_k^i, S_k^{i+1}) = q_k^{i+1(in)}, \qquad (5)$$

Which is the same as the inflow for cell $i+1$. These flows are then used to update the cell density in (6)

$$\rho_{k+1}^i = \rho_k^i + \frac{(q_k^{i(in)} - q_k^{i(out)})\Delta t}{L^i} \qquad (6)$$

Step 3. Node progression The node progression scheme in is illustrated below in figure 1:

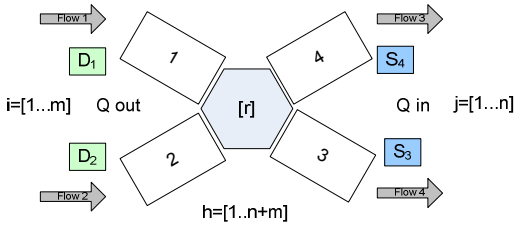

Fig. 1. Example Node in DSMART-SC

First, the total summed demand $D_{k,r}^j$ for all links j at time k connected to node r is calculated based on the summed demand $D_{k,r}^i$ of all links i connected to node r at time k and their corresponding turn fractions $\psi_{k,r}^{i,j}$ by (7).

$$D_{k,r}^j = \sum_{i=1}^{m} D_{k,r}^i \psi_{k,r}^{i,j} \qquad (7)$$

This link specific demand $D_{k,r}^j$ is compared to the supply of each link $S_{k,r}^j$ and the minimum flow between them is calculated. This flow is used to determine the multiplier $M_{k,r}^j$ in (8).

$$M_{k,r}^j = \frac{\min(D_{k,r}^j, S_{k,r}^j)}{D_{k,r}^j} \qquad (8)$$

$M_{k,r}^j$ describes the fraction of link demand which *can* flow from links i to destination links j and is used in calculating the total flow matrix in (9).

$$\mathbf{Q}_{k,r}^{i,j} = D_{k,r}^i \psi_{k,r}^{i,j} M_{k,r}^j \qquad (9)$$

Given this matrix, a summation over j will result in the total outflow $q_{k,r}^{i(out)}$ of a link i and a summation over i will result in the total inflow $q_{k,r}^{j(in)}$ of a link. Both flows are used in (10),(11) to update the cell densities.

$$\rho_{k+1}^i = \rho_k^i - q_{k,r}^{i(out)} \frac{\Delta t}{L^i} \qquad (10)$$

$$\rho_{k+1}^j = \rho_k^j + q_{k,r}^{j(in)} \frac{\Delta t}{L^j} \qquad (11)$$

Step 4. Event handling This step basically increases the time step k and period p and can be used for special simulation events. E.g. an accident can be simulated by forcing the maximum demand value for a cell to some value, and then gradually increasing it to represent different phases of an accident, as done in (Zuurbier F.S. 2006). The next section discusses how DSMART-SC can be linearized and put to use in an extended Kalman filter (EKF).

2. EXTENDED KALMAN FILTER

In general the Kalman filter as was introduced in 1960 (Kalman R.E. 1960), consists of two steps; the first is a *prediction* step which uses the model to make a prediction about the next state. The second step is used to correct this prediction by means of the measurements resulting in the actual estimate for the new state. When using a non linear process model, an

extended Kalman filter (EKF) may be used, clearly illustrated in fig 2 (Welch G. 2001).

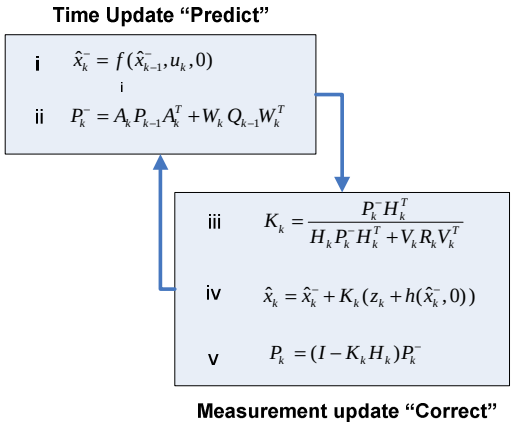

Fig. 2. Extended Kalman filter equations

The DSMART-SC model is used in i. to make a prediction about the expected state. Using a linearized version of this model, the expected change in the error covariance method is calculated in ii. Given this, an update to the Kalman gain is calculated in iii. with information about the variance of measurements. The Kalman gain is the used in conjunction with the measurements in iv. to make a correction to the original state prediction and finishes by correcting the error covariance in the last step v. The state x_k of a network in DSMART-SC contains only the densities of all cells in the network, has size n and is equal to the number of cells used in the model (including origin and destination cells). The measurement size m equals the number of detector measurements used. The DSMART-SC model $f(\hat{x}_{k-1}, u_k, 0)$ uses the previous state x_{k-1} together with external control u_k (edge demand (2), turn fractions, fundamental diagram) to estimate the next state x_k. The error covariance matrix P_k expresses the covariance in errors between cells. A_k Is the linearization of $f(\hat{x}_{k-1}, u_k, 0)$ calculated as a partial derivative to the state x_{k-1} as discussed in the next section. It is here where the effect of turn fractions is explicitly implemented in the Kalman filter. The linearization of the process and measurement error W_k and V_k are not used, and thus the *process noise* Q_k and *measurement noise* R_k matrices remain unchanged. For the measurement data, values on the diagonal of R_k are interpreted as the variation or error belonging to each induction loop. The process noise is considered fixed and based on modelling error in the *triangular fundamental diagram*. In this study a fixed diagonal matrix is used to account for this error. It might prove however beneficial to make the actual error a function of the density; in that case lower variance values may be used in the free flow regime and larger variance values in the congested regime. Measurements are expressed in density, which can be debated off course. For simplicity it was chosen this way. By doing so, measurements can be easily related to the state by $z_k^- = h(\hat{x}_k^-) = \hat{x}_k^-$ .The linearization of this mapping is 1 which form the only non zero elements of H_k. The Kalman gain K_k weighs the densities predicted by the DSMART-SC model with the measurement data and computes the actual state estimate.

The next section is dedicated to describing the linearization of the process model which is treated in a similar way dealing with link and node progression.

3. LINEARIZATION OF DSMART-SC

This section describes the results of linearizing equations (6), (10), (11) around the state x_k. The derivation process itself is discussed in detail in (Zuurbier F.S. 2006).

3.1 Linearizing link progression

The main derivatives which are obtained when linearizing (6) are:

$$\frac{\partial \rho_{k+1}^i}{\partial \rho_k^{i-1}} = \frac{\Delta t}{L}\left(+\frac{\partial q_k^{i(in)}}{\partial \rho_k^{i-1}}\right) \tag{12}$$

$$\frac{\partial \rho_{k+1}^i}{\partial \rho_k^{i}} = 1 + \frac{\Delta t}{L}\left(\frac{\partial q_k^{i(in)}}{\partial \rho_k^{i}} - \frac{\partial q_k^{i(out)}}{\partial \rho_k^{i}}\right) \tag{13}$$

$$\frac{\partial \rho_{k+1}^i}{\partial \rho_k^{i+1}} = \frac{\Delta t}{L}\left(-\frac{\partial q_k^{i(out)}}{\partial \rho_k^{i|+1}}\right) \tag{14}$$

And dependant on the derivatives of the inflow and outflow to the state x_k. Recall that the inflow is the same as the outflow of the previous cells rendering it necessary to calculate only the outflow derivatives. Basically two cases exist: a). $D_k^i > S_k^{i+1}$ the demand of cell i is larger then supply of cell $i+1$, so the supply determines the out flow $q_k^{i(out)} = S_k^{i+1}$ (15)

$$\frac{\partial q_k^{i(out)}}{\partial \rho_k^{i}} = 0, \text{ and } \frac{\partial q_k^{i(out)}}{\partial \rho_k^{i+1}} = \frac{dS_k^{i+1}}{d\rho_k^{i+1}} \tag{15}$$

Or $D_k^i \le S_k^{i+1}$ the demand of cell i is smaller than supply of cell $i+1$, so the demand determines the out flow $q_k^{i,out} = D_k^i$. Resulting in (16):

$$\frac{\partial q_k^{i(out)}}{\partial \rho_k^i} = \frac{dD_k^i}{d\rho_k^i}, \text{ and } \frac{\partial q_k^{i(out)}}{\partial \rho_k^{i+1}} = 0 \qquad (16)$$

The actual value of the derivative is dependant on the *fundamental diagram* used and equations (3),(4)

3.1 Linearizing Node progression

Based on the density updating formulas (10), (11) the linearization of the outflow can be calculated as:

$$\frac{\partial \rho_{k+1}^i}{\partial \rho_k^h} = \begin{cases} 1 - \dfrac{\partial q_{k,r}^{i(out)}}{\partial \rho_k^h}\dfrac{\Delta t}{L^h} & , h = i \\ -\dfrac{\partial q_{k,r}^{i(out)}}{\partial \rho_k^h}\dfrac{\Delta t}{L^h} & , h \neq i \end{cases} \qquad (17)$$

Where h requires some explanation. The actual calculation of the flow matrix $\mathbf{Q}_{k,r}^{i,j}$ in (9) is dependant on the demand D_k^i of the cells of outlinks i and the multiplier $M_{k,r}^j$ (8), which may also be determined by the supply $S_{k,r}^j$ and thus by the cells of inlinks j. This way, the partial derivates may be non zero for the cells of links i and j; index h is simply the collection of the cells of links i and j. Given the demand equation in (7) and the multiplier in (8) we can calculate the derivative of the flow out (the product of both formulas) of a link as:

$$\frac{\partial q_{k,r}^{i(out),j}}{\partial \rho_k^h} = \psi_k^{i,j} M_{k,r}^j \frac{dD_{k,r}^i}{d\rho_k^i} + (D_{k,r}^i \psi_{k,r}^{i,j})\Theta_{k,r}^{j,h} \qquad (18)$$

Where $\Theta_{k,r}^{j,h} = \dfrac{\partial M_{k,r}^j}{\partial \rho_k^h}$ is the derivative of the multiplier to the cells h and is calculated by:

$$\Theta_{k,r}^{j,h} = \begin{cases} \dfrac{-S_{k,r}^j}{{D_{k,r}^j}^2}\dfrac{\partial D_{k,r}^i}{\partial \rho_k^i} & \forall\ h \in i \\ \dfrac{1}{D_{k,r}^j}\dfrac{\partial S_{k,r}^j}{\partial \rho_k^j} & \forall\ h \in j \end{cases} \qquad (19)$$

and thus based on the derivative of supply and demand, determined by (3), (4) and the fundamental diagram that is being used.

4. EXPERIMENTAL SETUP

The linearized model as described in the previous section has been implemented in Matlab® together with the EKF equations described by fig 2. And is tested using the network from fig 3. which consists of 7 links with a length of 10 [km], a maximum speed of 80 [km/hr] and a jam density of 125 [veh/km]. Given the simulation time step $\Delta t = 30$ [sec] and (1) all links are divided in 15 cells with a length of 2/3 [km]. The network is *richly* equipped with detectors, but in this paper only some are used. The detectors in *red* on links 4 and 7 are used for *verification*, whereas the *green* detectors on links 1 and 2 are used for measurement input to the Kalman Filtering equations.

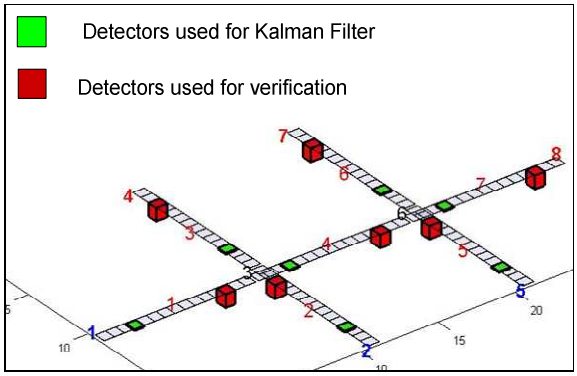

Fig. 3. Test Network for the EKF

4.1 Experiment setup

In Step 1 a DTA is performed on the network using DSMART-SC where the static OD is *dynamized* by (2). The resulting assignment is *measured* (in density) using the red *and* green detectors. In Step 2. the EKF is used to make a state estimate, but this time the OD is kept static with a $\tau_p = 0.5$ and $\varepsilon_t = 0$, and so structurally underestimating edge demand. In addition the measurement data from the *green* detectors is *bootstrapped* based on a normal distribution using the actual measured cell densities as the average μ and a standard deviation $\delta = 0.1\mu$. The EKF is then used, based solely on the measurement data of the green detectors on links 1 and 2 to *reconstruct* the state and compared based on the detector data of the red detectors on link 4 and 7. The OD demand pattern is 3x500 [veh/hr] will want to depart from nodes 1&2 to nodes 4,7 and 8. And 2x500 [veh/hr] will want to depart from node 5 to nodes 7 & 8. The time slice values τ_p are [0.5 0.75 0.85 1 1 0.85 0.75 0.5].

5. RESULTS

In fig 4. the measurement data at two detector locations (link 4 and 7) is compared. For this state reconstruction both the process and measurement covariance Q_k and R_k where set to 5. As a result of the OD demand pattern, the detector on link 4 measures a structural higher density than that of link 7. The induced error on the measurements by bootstrapping the dataset is responsible for the higher stochasticity in the measured values for the reconstructed state.

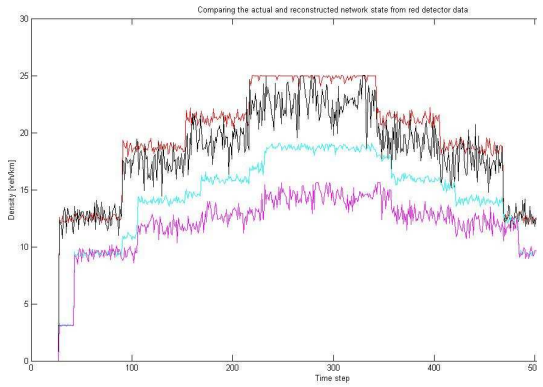

Fig. 4. Detector comparison on links 4&7 in [veh/km]

The corresponding error covariance matrix P_k is investigated at the end of the simulation, by plotting the diagonal values (error variance of a cell) onto the corresponding cells in the network.

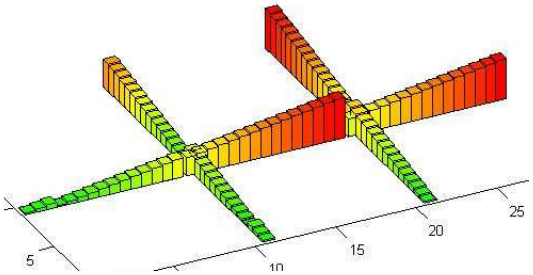

Fig. 6 Error variance matrix P diagonal values plotted on corresponding cells (values in [4 .. 125])

What we see in this diagram is five points where the error is smallest; corresponding to the three origins at the beginning of links 1,2 and 5, and two detectors on cells 4 and 20. The fact that origin nodes have a zero error is the results of implementation. Because traffic flows into one direction, we see that the error variance on the cells in that direction increasing step wise. This is because there is no measurement data available on these cells, only the process model with presumed error $Q = 5$ which is exactly the amount of increase per step. The influence of the node progression scheme becomes visible on the nodes. We see that the error variance at the starting cells of the inlinks of a node relate to the values of the error variance on the last cells of the outlinks and the corresponding turn fractions on those links.

5. 1 Discussion

The EKF with the linearization of the Godunov scheme is in fact capable of reconstructing a network state. The network reconstruction is strongly influenced by a) the different values for R and Q. b) the quality of the prediction of the traffic model and c) the detector availability.
The node progression schemes allows the error covariance to propagate through the network. This fact can be used to determine the certainty of the network estimate which can be valuable information in e.g. route guidance generation or evaluation of a network in terms of where to place new detectors.

In this study detector data was generated and reconstructed in both cases using the same traffic flow model. Further research should point out how well DSMART will perform once it is used on real traffic network data. Another consideration is the fact that in this study density was used as the measurement input. In future research it is intended to extended this methodology using a two phase fundamental diagram, adaptive process variance values, measurement flow and speed as input. And further research will be done in how trajectory data might be implemented. The network state estimate will function as a primary component in the generation of route guidance. The work presented here is funded by the ATMA project of the TRANSUMO research program,

REFERENCES

Antoniou C., B.-A. M., et al. (2004). On line calibration of traffic prediction models. 84th Annual meeting of the transportation research board, Washington DC.

Kalman R.E. (1960). "A new approach to linear filtering and prediction Problems." Transactions of the ASME - Journal of basic engineering **82 (series D)**: 35-45.

Lebaque J.P. (1996). The Godunov scheme and what it means for first order traffic flow models. Proceedings of the 13th international symposium of transportation and traffic theory, Lyon.

Messmer A. and Papageorgiou M. (1990). "METANET: A macroscopic simulation program for motorway networks." Traffic Engineering & Control **31**: 466-470.

Middelham F. (2005). "State of Practice In Dynamic Traffic Management in The Netherlands. ." Transport Research Centre (AVV), Ministry of Transport Public Works and Watermanagement, Rotterdam: 24.

Wang Y., Papageorgiou M., et al. (2006). "A real time freeway network traffic surveillance tool." IEEE Transactions on control systems technology **14**(1).

Welch G., B. G. (2001). An introduction to the Kalman Filter, Course 8, University of north Carolina at Chapel Hill, department of computer science.

Zuurbier F.S. (2005). A generic approach to a route guidance strategy. Transportation and Planning, Technical University Delft. **Msc:** 180.

Zuurbier F.S. (2006). Optimal rerouting during emergency situations by means of prescriptive advice on VMS display. . Submitted for 11th meeting on EURO WG 2006.

Zuurbier F.S., L. H., Knoop V.L., (2006). "State estimation using an extended kalman filter and the first order traffic flow model DSMART." Internal Report section Transportation and Planning.