Using Aurora Road Network Modeler for Active Traffic Management

Alex A. Kurzhanskiy and Pravin Varaiya

***Abstract*— Active Traffic Management (ATM) is the ability to dynamically manage recurrent and nonrecurrent congestion based on prevailing traffic conditions. Focusing on trip reliability, it maximizes the effectiveness and efficiency of freeway corridors. ATM relies on fast and trustworthy traffic simulation software that can assess a large number of control strategies for a given road network, given various scenarios, in a matter of minutes. Effective traffic density estimation is crucial for the successful deployment of feedback algorithms for congestion control.**

Aurora Road Network Modeler (RNM) is an open-source macrosimulation tool set for operational planning and management of freeway corridors. Aurora RNM employs Cell Transmission Model (CTM) for road networks extended to support multiple vehicle classes. It allows dynamic filtering of measurement data coming from traffic sensors for the estimation of traffic density. In this capacity, it can be used for detection of faulty sensors. The virtual sensor infrastructure of Aurora RNM serves as an interface to the real world measurement devices, as well as a simulation of such measurement devices.

I. INTRODUCTION

Traffic congestion is a world-wide source of productivity and efficiency loss, wasteful energy consumption and excessive air pollution. Texas Transportation Institutes 2009 Urban Mobility Report [18] estimates that congestion imposed on urban Americans 4.2 billion hours of delay and an extra 2.9 billion gallons of fuel for a cost of $78 billion. Active Traffic Management (ATM) is the ability to dynamically manage recurrent and nonrecurrent congestion based on prevailing traffic conditions. Focusing on trip reliability, it maximizes the effectiveness and efficiency of freeway corridors.

The task of implementing ATM lies with a Traffic Management Center (TMC). Fig. 1 shows what TMC is expected to provide. The core of a TMC is a Control Center and a communication network that feeds real-time measurements from field elements in the road network to the Control Center and transfers commands generated by the control algorithms. The other entities in the diagram embody the "smarts". One key entity is the "trusted, fast corridor simulator". The simulator is trusted because it is founded on sound theory of traffic flow; it is parsimonious, only including parameters that can be estimated; and it is tested for reliability. Real-time measurements and predicted short-term future traffic demands are continuously fed to a module that estimates the current traffic state and makes short-term predictions, using predictor-corrector estimation techniques. The simulator periodically simulates a set of scenarios involving events and changes in demand pattern, and calculates the resulting congestion, capacity loss and potentially serious stresses. The scenarios incorporate known events such as a football game or a confirmed accident, or future events considered plausible on the basis of statistical inference and learning techniques that combine historical data with the current estimate of the state of the system (e.g., likelihood of an accident occurring in a certain location, conditional on the current congestion state, rain or fog conditions). If a scenario results in an unacceptable condition, an alarm is created. The operator in the Supervisory Control and Data Acquisition (SCADA) system studies the alarm and decides whether to counter the unacceptable hypothetical future with one of the available control policies in case the scenario indeed materializes.

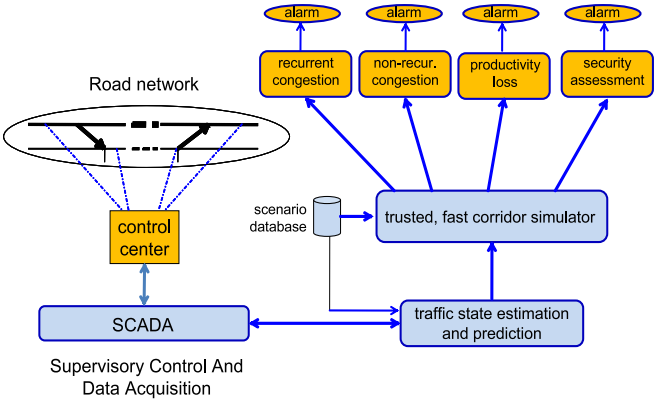

Fig. 1. Structure of Traffic Management Center.

Aurora Road Network Modeler (RNM) [1] is an open-source macrosimulation tool set for operational planning and management of freeway corridors. Aurora RNM simulation is based on the Cell Transmission Model (CTM) [2], [3], which, compared with microscopic simulators, requires negligible computational effort. The model is calibrated following the procedure in [6], and the missing demand data is imputed using the techniques described in [12]. Various event classes allow the user to generate simulation scenarios, such as incidents, demand change and road closures. Control strategies involving local and coordinated ramp metering, signal coordination at intersections, variable speed limit and lane management can be evaluated in terms of performance benefits they provide [9].

This paper focuses on other elements of Aurora RNM

This research is supported by the California Department of Transportation through the California PATH program.

A. A. Kurzhanskiy is with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, USA akurzhan@eecs.berkeley.edu

P. Varaiya is with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, USA varaiya@eecs.berkeley.edu

necessary for ATM. It describes the handling of multiple vehicle types, particularly, how HOV[1] lanes are modeled; extends set-valued traffic density estimation [8] to road networks; and introduces the concept of virtual sensors that serve as interfaces to real world measurement devices as well as simulation of such measurement devices.

The rest of the paper is organized as follows. Section II explains the traffic flow model supporting multiple vehicle types employed by Aurora RNM. Section III describes how HOV lanes are simulated. Section IV is dedicated to dynamic filtering for traffic density estimation. Section V introduces the concept of virtual sensors. Finally, Section VI concludes the paper.

II. MULTI-TYPE VEHICLE FLOW MODEL

Previous work on the first order macroscopic modeling of heterogeneous traffic includes the extension of the LWR model to incorporate multiple lanes and different vehicle types [7], [20], and a discrete freeway model of two vehicle types with a subset of lanes reserved for one of the types based on Godunov scheme applied to the LWR equation [4] used for simulation [5]. In this section we discuss the multi-type vehicle flow model employed by Aurora RNM.

As described in [9], the road network consists of links and nodes, where links represent stretches of roads and nodes connect the links. A node must always have at least one input and at least one output link. A link is called an ordinary link, if it has both begin and end nodes. A link with no begin node is referred to as *source link* or *source*, and a link with no end node is called *destination link* or *sink*. Each link is characterized by its length and the fundamental diagram (capacity F, free flow speed v and congestion wave speed w), which takes into account the number of lanes. Sources are the links through which the vehicles enter the system, and therefore have demand profiles assigned to them. Each node is characterized by a split ratio matrix that determines how the incoming flows are distributed among the output links. Nodes may be used not only to represent intersections, merge, or diverge points, but also to split long links into smaller sections.

The state of the system at each simulation step is described by the vehicle density in each link. For each link, the density $\vec{\rho} = (\rho^1, \cdots, \rho^K)^T$ is K-dimensional vector, with K being the number of modeled vehicle types, and ρ^k representing density of vehicles of type k, $k = 1..K$. Total vehicle density for a link is computed simply as $\rho_T = \sum_{k=1}^{K} \rho^k$. The state update equation for a link is

$$\vec{\rho}(t+1) = \vec{\rho}(t) + \frac{\Delta t}{\Delta x}\left(\vec{f}^u(t) - \vec{f}^d(t)\right), \qquad (1)$$

where Δt is the sampling period, Δx is the link length, and K-dimensional vectors $\vec{f}^u = (f^{u1}, \cdots, f^{uK})^T$ and $\vec{f}^d = (f^{d1}, \cdots, f^{dK})^T$ are upstream (entering the link) and downstream (exiting the link) flows respectively, where elements f^{uk} and f^{dk} correspond to flows of vehicles of type k, $k = 1..K$. For sources, $\vec{f}^u(t) = \vec{r}(t)$, where $\vec{r}(t)$ is K-dimensional demand vector at time step t. For sinks, $\vec{f}^d(t) = v\vec{\rho}(t)\min\left\{1, \frac{F}{v\rho_T(t)}\right\}$. Otherwise, $\vec{f}^u(t)$ is determined by the begin node, and $\vec{f}^d(t)$ is determined by the end node. Total input or output flow is defined as $f_T^{\times} = \sum_{k=1}^{K} f^{\times k}$, where "$\times$" should be replaced by index "u" or "d" respectively.

A node with m input and n output links has $mK \times n$ split ratio matrix $\mathscr{B} = \{\beta_{(k-1)m+i,j}\}_{k=1..K,i=1..m}^{j=1..n}$. This matrix is nonnegative, its elements lying in the interval $[0,1]$, and the sum of the elements in each row equals 1. Element $\beta_{(k-1)m+i,j}$ defines the portion of the flow of vehicles of type k coming from input link i that has to be directed into output link j. At each time step t, the input flow vectors $\vec{f}_i^d(t)$ and output flow vectors $\vec{f}_j^u(t)$, $k = 1..K$, $i = 1..m$, $j = 1..n$, for the node are computed as follows.

1) Compute supply for each output:

$$s_j(t) = \min\left\{F_j, w_j\left(\overline{\rho}_j - \rho_{Tj}(t)\right)\right\}, \quad j = 1..n, \quad (2)$$

where F_j is the capacity, w_j is the congestion wave speed, $\overline{\rho}_j$ is the jam density, and $\rho_{Tj}(t)$ is the total density of the output link j.

2) Compute input demands:

$$\tilde{d}_{(k-1)m+i}(t) = v_i\rho_i^k(t)\min\left\{1, \frac{F_i}{v_i\rho_{Ti}(t)}\right\}, \quad k = 1..K,\ i = 1..m, \qquad (3)$$

where F_i is capacity, v_i is free flow speed, and $\rho_{Ti}(t)$ is the total density of the input link i. Quantity $\tilde{d}_{(k-1)m+i}(t)$ represents the desired flow of vehicles of type k from the input link i.

3) Compute output demands:

$$d_j(t) = \sum_{i=1}^{m}\sum_{k=1}^{K}\beta_{(k-1)m+i,j}(t)\tilde{d}_{(k-1)m+i}(t), \quad j = 1..n. \qquad (4)$$

Quantity $d_j(t)$ represents the total flow that desires to enter the output link j.

4) Initialize the scaling factors $\tilde{\delta}_{(k-1)m+i}(t) = 1$ for $k = 1..K$, $i = 1..m$.

   a) For $q = 1..n$ update scaling factor according to the output supply, if necessary:

$$\tilde{\delta}_{(k-1)m+i}(t) \leftarrow \begin{cases} \tilde{\delta}_{(k-1)m+i}(t), \text{ if } \forall k,\ \beta_{(k-1)m+i}(t) = 0 \\ \min\left\{\tilde{\delta}_{(k-1)m+i}(t), \frac{s_q(t)}{d_q(t)}\right\}, \text{ otherwise} \end{cases}, \quad i = 1..m,\ k = 1..K; \qquad (5)$$

   b) recompute input demands

$$\tilde{d}_{(k-1)m+i}(t) \leftarrow \tilde{d}_{(k-1)m+i}(t)\tilde{\delta}_{(k-1)m+i}(t) \qquad (6)$$

   and recompute output demands $d_j(t)$, $j = 1..n$, according to (4).

This step implements the proportional priority rule for merging links and the first-in-first-out rule for diverging links as they are stated in [3].

[1]High Occupancy Vehicle

5) Vector of flows leaving the input link i is

$$\vec{f}_i^{\,d}(t) = \begin{pmatrix} \tilde{d}_i(t) \\ \tilde{d}_{m+i}(t) \\ \vdots \\ \tilde{d}_{(K-1)m+i}(t) \end{pmatrix}, \quad i = 1..m. \qquad (7)$$

6) Vector of flows entering the output link j is

$$\vec{f}_j^{\,u}(t) = \begin{pmatrix} \sum_{i=1}^{m} \beta_{i,j}\tilde{d}_i(t) \\ \sum_{i=1}^{m} \beta_{m+i,j}\tilde{d}_{m+i}(t) \\ \vdots \\ \sum_{i=1}^{m} \beta_{(K-1)m+i,j}\tilde{d}_{(K-1)m+i}(t) \end{pmatrix}, \quad j = 1..n. \qquad (8)$$

Aurora RNM simulation starts with given initial density vectors on all the links. At every step of the simulation, at first the input/output flow vectors are computed on all the nodes, then the state is updated on all the links using (1).

The presented vector structure for densities and flows allows arbitrary number of vehicle types to be modeled. Vehicle types can be used to represent vehicle classes, such as cars, trucks or buses; certain vehicle properties, such as occupancy (high or single), electric or hybrid; route information; or combination of these.

If type k represents vehicles of certain size, trucks for example, one should replace ρ^k with $\alpha_k\rho^k$, f^{uk} with $\alpha_k f^{uk}$, and f^{dk} with $\alpha_k f^{dk}$ in the definition of $\vec{\rho}$, $\vec{f}^{\,u}$ and $\vec{f}^{\,d}$, where the weight α_k specifies how many cars are equivalent to one vehicle of type k.

If a type represents vehicles that should follow a particular route, the split ratios for this vehicle type are fixed and constant directing the corresponding flow along the given route. Aurora RNM simulation input is normally given in the form of demand profiles for source links and split ratio profiles for nodes. The ability of vehicle types to carry route information allows Aurora RNM to accept the simulation input in the form of origin-destination (OD) matrices.

III. HOV MODELING

To model HOV lane on freeways, we introduce two vehicle types ($K = 2$): HOV (type $k = 1$) and SOV[2] (type $k = 2$). Since Aurora RNM allows multiple links to connect the same pair of nodes in the same direction, we replace each freeway link with two, freeway HOV and freeway mainline (ML), at locations where HOV lane is present (Fig. 2). HOV and ML links are parallel, if they connect the same pair of nodes. Parallel HOV and ML links may have different per lane fundamental diagrams. Vehicles can change between HOV and ML at nodes. If it is desirable to enable switching between HOV and ML in the middle of a long parallel link pair, it is possible to create a node that would split this link pair, as does node 3 in Fig. 2.

HOV lane is active, if only vehicles of type 1 are allowed into HOV links. There are two types of HOV facilities: a dedicated HOV lane that is active all the time, and time dependent HOV lane that is active only at certain hours during the day.

[2]Single Occupancy Vehicle

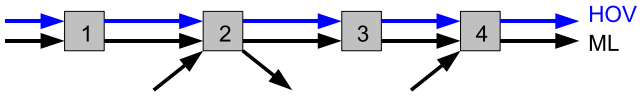

Fig. 2. Freeway is represented as a set of parallel link pairs: mainline (ML) and HOV.

A. Dedicated HOV lane

Dedicated HOV lanes are present on freeways in Southern California. These HOV lanes are separated from ML by double solid line or a concrete divider. Vehicles of type 1 can cross between HOV and ML at special areas, *HOV gates*. In Aurora RNM configuration, nodes should be placed at or near HOV gate locations. If, on the other hand, a node is present where there is no HOV gate, its split ratio matrix should indicate that no crossing between HOV and ML is allowed.

For example, suppose there is no HOV gate near node 1 in Fig. 2, the first input and the first output links for this node are HOV, then the split ratio matrix $\mathscr{B} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$ will ensure that no crossing between HOV and ML occurs at node 1. The fact that all SOVs are forced from HOV into ML ($\beta_{32} = 1$) is not a problem, since there should be zero SOV flow from the HOV input link.

A more interesting question is how to specify the split ratio matrix at nodes where the crossing between HOV and ML is allowed. When the split ratio matrix is well defined, that is, all its elements are nonnegative and each row sums up to 1, it determines the demand for each of the output links. This demand is determined *apriori*, meaning that no matter what the traffic state is, given portion of the flow must be directed into the specific output. In some cases (recall our example with node 1) such interpretation of the split ratio matrix is perfectly adequate. In that case, no switching between HOV and ML is allowed no matter what the traffic conditions in the downstream links.

Consider another example. Suppose, node 3 in Fig. 2 is placed at the HOV gate, and vehicles of type 1 *have a choice*, which of the downstream links, HOV or ML, to take. This choice mainly depends on the traffic conditions downstream. Any *fixed* matrix $\mathscr{B}$ eliminates the possibility of such choice. Therefore, Aurora RNM allows split ratio matrices with *undefined* elements that are computed at simulation run time based on the traffic state in the downstream links. For node 3, the split ratio matrix is partially defined: $\mathscr{B} = \begin{pmatrix} -1 & -1 \\ -1 & -1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$. Elements in the first two rows of matrix $\mathscr{B}$ correspond to the vehicle flows of type 1, and their negative sign indicates that they are undefined and the corresponding input-to-output flows must be assigned by the simulation. The details of this flow assignment are given in [10].

B. Time dependent HOV lane

Time dependent HOV lanes are present on freeways in Northern California. These HOV lanes are not separated from ML by any special divider, and switching between the two is allowed everywhere. Moreover, HOV lane is active only at certain time periods during the day. At other time, both vehicle types can enter HOV links.

There are two ways of modeling time dependent HOV in Aurora RNM. The obvious way, is to repeat what was done for the dedicated HOV lane with exception that now *all* nodes must act as HOV gates, and at times when HOV lane is inactive, the structure of the split ratio matrix should permit vehicles of type 2 to switch between HOV and ML. In the example with node 3 (Fig. 2), the split ratio matrix will be time dependent:

$$\mathscr{B}(t_{active}) = \begin{pmatrix} -1 & -1 \\ -1 & -1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}, \ \mathscr{B}(t_{inactive}) = \begin{pmatrix} -1 & -1 \\ -1 & -1 \\ -1 & -1 \\ -1 & -1 \end{pmatrix}.$$

Here the subscripts "active" and "inactive" indicate active or inactive HOV. This approach is straight forward, since Aurora RNM admits time variant split ratio matrices.

The second way is to close HOV links at times when HOV lane is inactive while simultaneously increasing the capacity of ML: $F_{ML}(t_{inactive}) = F_{ML}(t_{active}) + F_{HOV}$. As HOV lane activates, the system returns to the usual dedicated HOV lane operation — HOV links are opened and ML capacity is reduced: $F_{ML}(t_{active}) = F_{ML}(t_{inactive}) - F_{HOV}$. Aurora RNM allows to change fundamental diagrams on links at any given time during the simulation by means of events. In the example with node 3 (Fig. 2), the time dependent split ratio matrix for HOV active time will be the same as before, and for the rest of the time it will direct all the traffic to ML:

$$\mathscr{B}(t_{active}) = \begin{pmatrix} -1 & -1 \\ -1 & -1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}, \ \mathscr{B}(t_{inactive}) = \begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}.$$

IV. DYNAMIC FILTERING OF MEASUREMENT DATA

The set-valued estimation of freeway traffic density using cell transmission model is explained in [8]. Those results can be extended to the case of road network.

We assume that demands $\vec{r}(t)$ at source links are known with some uncertainty, more precisely, they are constrained by a box $\vec{r}^-(t) \le \vec{r}(t) \le \vec{r}^+(t)$ ($\vec{r}^-(t)$ and $\vec{r}^+(t)$ are known)[3]. The problem of demand prediction can be addressed by various available approaches, such as [11], [13], [17]. Systems such as PeMS [15] provide historic measurement data. The other assumption is that the link capacities F lie within given intervals $F^- \le F \le F^+$ (F^- and F^+ are known). Accordingly, denote

$$\overline{\rho}^+ = \frac{F^+}{w} + \frac{F^+}{v} \ \text{ and } \ \overline{\rho}^- = \frac{F^-}{w} + \frac{F^-}{v}. \tag{9}$$

[3]Vectors $x, y \in \mathbf{R}^K$ are partially ordered, $x \le y$, iff $x^k \le y^k$, $k = 1..K$.

Noisy measurements of the total output flow

$$y_f(t) = f_T^d(t) + \omega_f(t), \tag{10}$$

and speed

$$y_V(t) = V(t) + \omega_V(t), \tag{11}$$

are available at each link. Here $\omega_f(t) \in [-\omega_f^0(t), \omega_f^0(t)]$ is the flow measurement noise, $\omega_V(t) \in [-\omega_V^0(t), \omega_V^0(t)]$ is the speed measurement noise, and bounds $\omega_f^0(t)$ and $\omega_V^0(t)$ are known. Thus, for each link we get an estimate of the total density coming from the measurements:

$$\hat{\rho}_T^-(t) \le \hat{\rho}_T(t) \le \hat{\rho}_T^+(t), \tag{12}$$

where

$$\hat{\rho}_T^-(t) = \frac{y_f(t) - \omega_f^0(t)}{y_V(t) + \omega_V^0(t)}, \ \ \hat{\rho}_T^+(t) = \frac{y_f(t) + \omega_f^0(t)}{y_V(t) - \omega_V^0(t)}. \tag{13}$$

Define state bounds update equations for each link:

$$\vec{\rho}^-(t+1) = \vec{\rho}^-(t) + \frac{\Delta t}{\Delta x}\left(\vec{f}^{u-}(t) - \vec{f}^{d+}(t)\right), \tag{1$^-$}$$

and

$$\vec{\rho}^+(t+1) = \vec{\rho}^+(t) + \frac{\Delta t}{\Delta x}\left(\vec{f}^{u+}(t) - \vec{f}^{d-}(t)\right). \tag{1$^+$}$$

For sinks, $\vec{f}^{d-}(t) = v\vec{\rho}^-(t)\min\left\{1, \frac{F^-}{v\rho_T^-(t)}\right\}$ and $\vec{f}^{d+}(t) = v\vec{\rho}^+(t)\min\left\{1, \frac{F^+}{v\rho_T^+(t)}\right\}$. Otherwise, $\vec{f}^{u-}(t)$ and $\vec{f}^{u+}(t)$ are determined by the begin node, and $\vec{f}^{d-}(t)$ and $\vec{f}^{d+}(t)$ are determined by the end node. Lower input/output flow bounds $\vec{f}^{d-}(t)$ and $\vec{f}^{u-}(t)$ are computed using steps 1-6 from Section II with slight modifications. For a node with input links $i = 1..m$ and output links $j = 1..n$, the algorithm is as follows.

1) Compute lower supply bounds for each output:
$$s_j^-(t) = \max\left\{0, \min\left\{F_j^-, w_j\left(\overline{\rho}_j^- - \rho_{Tj}^+(t)\right)\right\}\right\}, \quad j = 1..n. \tag{14}$$

2) Compute lower input demand bounds:
$$\tilde{d}_{(k-1)m+i}^-(t) = v_i\rho_i^{-k}(t)\min\left\{1, \frac{F_i^-}{v_i\rho_{Ti}^-(t)}\right\}, \quad k = 1..K, \ i = 1..m, \tag{15}$$

3) Compute lower output demand bounds:
$$d_j^-(t) = \sum_{i=1}^{m}\sum_{k=1}^{K}\beta_{(k-1)m+i,j}(t)\tilde{d}_{(k-1)m+i}^-(t), \quad j = 1..n. \tag{16}$$

4) Initialize the scaling factors $\tilde{\delta}_{(k-1)m+i}(t) = 1$ for $k = 1..K$, $i = 1..m$.
   a) For $q = 1..n$ update scaling factor according to output supply, if necessary:
$$\tilde{\delta}_{(k-1)m+i}(t) \leftarrow \begin{cases} \tilde{\delta}_{(k-1)m+i}(t), \text{ if } \forall k,\ \beta_{(k-1)m+i}(t) = 0 \\ \min\left\{\tilde{\delta}_{(k-1)m+i}(t), \frac{s_q^-(t)}{d_q^-(t)}\right\}, \text{ otherwise} \end{cases}, \quad i = 1..m, \ k = 1..K; \tag{17}$$
   b) recompute lower input demand bounds
$$\tilde{d}_{(k-1)m+i}^-(t) \leftarrow \tilde{d}_{(k-1)m+i}^-(t)\tilde{\delta}_{(k-1)m+i}(t) \tag{18}$$

and recompute output lower demand bounds $d_j^-(t)$, $j = 1..n$, according to (16).

5) Vector of lower input flow bounds from link i is

$$\vec{f}_i^{d-}(t) = \begin{pmatrix} \tilde{d}_i^-(t) \\ \tilde{d}_{m+i}^-(t) \\ \vdots \\ \tilde{d}_{(K-1)m+i}^-(t) \end{pmatrix}, \quad i = 1..m. \tag{19}$$

6) Vector of lower output flow bounds to link j is

$$\vec{f}_j^{u-}(t) = \begin{pmatrix} \sum_{i=1}^m \beta_{i,j} \tilde{d}_i^-(t) \\ \sum_{i=1}^m \beta_{m+i,j} \tilde{d}_{m+i}^-(t) \\ \vdots \\ \sum_{i=1}^m \beta_{(K-1)m+i,j} \tilde{d}_{(K-1)m+i}^-(t) \end{pmatrix}, \quad j = 1..n. \tag{20}$$

Upper input/output flow bounds $\vec{f}^{d+}(t)$ and $\vec{f}^{u+}(t)$ are obtained through the same procedure by replacing "$-$" superscripts with "$+$" and vice versa.

By definition of $\vec{\rho}(t)$ in (1), $\vec{\rho}^-(t)$ in (1^-), and $\vec{\rho}^+(t)$ in (1^+), if $\vec{\rho}^-(0) \le \vec{\rho}(0) \le \vec{\rho}^+(0)$, then $\vec{\rho}^-(t) \le \vec{\rho}(t) \le \vec{\rho}^+(t)$ for $t \ge 0$ in every link. Note that $\vec{\rho}^-(t)$ and $\vec{\rho}^+(t)$ are restricted so that $0 \le \rho_T^-(t)$ and $\rho_T^+(t) \le \overline{\rho}^+$. These restrictions are not satisfied automatically in (1^-), (1^+), and must be imposed explicitly every time step. Boundary trajectories $\vec{\rho}^-(\cdot)$ and $\vec{\rho}^+(\cdot)$ are used to filter the incoming measurement data.

Systems (1^-) and (1^+) start evolving at time $t = 0$ with initial conditions $\vec{\rho}^-(0) \le \vec{\rho}^+(0)$ for every link, that possibly result from previous estimation. If no initial conditions are available, take the flow and speed measurements $y_f(0)$ and $y_V(0)$ for each link, determine $\hat{\rho}_T^-(0)$ and $\hat{\rho}_T^+(0)$ from (13), and set

$$\rho^{-k}(0) = \frac{\hat{\rho}_T^-(0)}{K}, \quad \rho^{+k}(0) = \frac{\hat{\rho}_T^+(0)}{K}, \quad k = 1..K.$$

Systems (1^-) and (1^+) evolve in time until time step $\tau > 0$ when the results of new measurements, $\hat{\rho}_T^-(\tau)$ and $\hat{\rho}_T^+(\tau)$, arrive. Density bounds $\vec{\rho}^-(\tau)$ and $\vec{\rho}^+(\tau)$ are adjusted according to these measurements:

$$\vec{\rho}^-(\tau) \leftarrow \max\left\{1, \frac{\hat{\rho}_T^-(\tau)}{\rho_T^-(\tau)}\right\} \vec{\rho}^-(\tau), \tag{21}$$

and

$$\vec{\rho}^+(\tau) \leftarrow \min\left\{1, \frac{\hat{\rho}_T^+(\tau)}{\rho_T^+(\tau)}\right\} \vec{\rho}^+(\tau). \tag{22}$$

These corrections make sense only if

$$\left[\rho_T^-(\tau), \rho_T^+(\tau)\right] \bigcap \left[\hat{\rho}_T^-(\tau), \hat{\rho}_T^+(\tau)\right] \neq \emptyset. \tag{23}$$

Condition (23) must be true in theory. Otherwise, it would mean that we made wrong assumptions about the range of measurement noise. In reality, however, empty intersections do occur. What to do in case condition (23) is not satisfied depends on the specific situation. If we trust our model more than the measurements, we should skip the correction (21)-(22). Moreover, we can use our dynamic filter to detect faulty measurement sensors. If, on the other hand, we believe that the measurement data are reliable, modified corrections (21)-(22) should apply:

$$\vec{\rho}^-(\tau) \leftarrow \frac{\hat{\rho}_T^-(\tau)}{\rho_T^-(\tau)} \vec{\rho}^-(\tau), \tag{21*}$$

and

$$\vec{\rho}^+(\tau) \leftarrow \frac{\hat{\rho}_T^+(\tau)}{\rho_T^+(\tau)} \vec{\rho}^+(\tau). \tag{22*}$$

Once the lower bound $\vec{\rho}^-(\tau)$ is corrected according to (21) or (21*), we must check that for each link

$$\rho_T^-(\tau) \le \overline{\rho}^-. \tag{24}$$

If for some link condition (24) is not fulfilled, there are two alternative ways to proceed. The first way is to modify $\vec{\rho}^-(\tau)$:

$$\vec{\rho}^-(\tau) \leftarrow \frac{\overline{\rho}^-}{\rho_T^-(\tau)} \vec{\rho}^-(\tau). \tag{25}$$

The alternative way is to modify the lower capacity bound of the link:

$$F \leftarrow \frac{\rho_T^-(\tau)}{\overline{\rho}^-} F, \tag{26}$$

and then adjust jam density $\overline{\rho}^-$ according to (9). Once the corrections (21)-(25) are applied, vector densities $\vec{\rho}^-(\tau)$ and $\vec{\rho}^+(\tau)$ in every link can be treated as new initial conditions, and starting at these initial conditions, systems (1^-), (1^+) should run until the next set of measurements arrives, and so on.

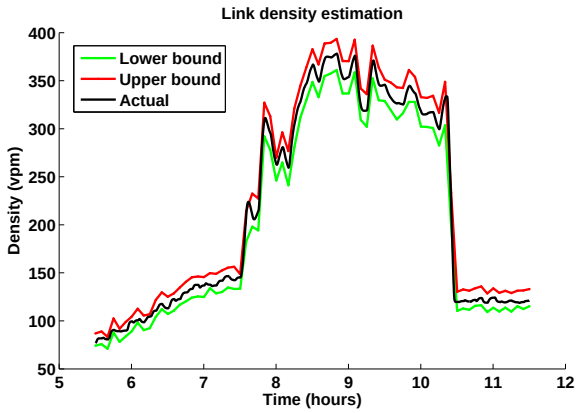

Fig. 3. Actual density with its lower (green) and upper (red) bounds.

Fig. 3 shows the density estimation at one of the links of I-210 West in Southern California on a work day between 5.30 and 11.30 AM using flow and speed measurements from [15] arriving every 5 minutes.

V. VIRTUAL SENSORS

Traffic flow on a road network is a spatio-temporal phenomenon. Accurate and efficient determination of macroscopic traffic flow characteristics is necessary for traffic operations and management. Following are the quantities of interest as documented in [19].

1) Flow, which is direct measure of throughput defined as number of vehicles passing a point on a highway during unit duration of time.

2) Space mean speed, which is useful for computing travel times defined as the harmonic mean of speeds over a given segment of road or an average speed based on the average travel time of vehicles to traverse a given segment of road.
3) Density, which is an indicator of traffic conditions (congested vs. uncongested) defined as number of vehicles observed on a segment over unit length of space.

Among the available sensing techniques, we distinguish between *point sensors* such as loop detectors and wireless sensors, *mobile sensors* such as GPS equipped vehicles and automatic vehicle location techniques, and *space sensors* such as aerial photography and satellite data.

1) Point sensors are fixed in location along a roadway and measure vehicles passing through this location throughout the time for which they are active.
2) Mobile sensors move with the traffic flow in space-time domain and can collect the time stamped position (and hence the speed) of the vehicle.
3) Space sensors can take snapshots of traffic at a given instant of time and repeat such snapshots are multiple time instants.

Aurora RNM implements point sensors and mobile sensors.

Point sensor object is assigned to a link and its position within this link is defined. A link may have multiple point sensors assigned to it. Specific sensor class implementing either loop detector or Sensys wireless sensor [16] inherits from the base point sensor object. The measurements, which loop detector class provides, are vehicle counts and, possibly, speeds. Data coming from Sensys sensors can be processed to obtain traffic density in in a link, as described in [14].

Mobile sensor object is assigned to a route between given origin and destination. It represents a probe vehicle. These probe vehicles are "phantoms" in the sense that they do not affect the density and flow quantities produced by the simulation. The class implementing the mobile sensor reports the speed, the link in which it is currently located and its position within this link. The intended use of mobile sensors in Aurora RNM is to compute actual travel time for certain routes as simulation runs.

Sensor objects have two modes of operation.

1) *Standalone simulation*. Aurora RNM runs traffic simulation and sensor objects simulate the work of measurement devices reporting vehicle counts and/or speeds from particular road network locations based on simulation link data. Within each such sensor object the user can tweak the noise level and choose the sensing quality in the range from excellent to unsatisfactory. The data reported by these sensors is used by traffic responsive controllers in the simulation. Thus, Aurora RNM allows to simulate the impact of feedback traffic control when the measurement data is less than perfect.
2) *Dynamic filtering*. Sensor objects act as interfaces to the real measurement devices communicating directly to the equipment in the field or to the TMC software and collecting raw measurement data in real time, while Aurora RNM simulation running in the background is used to filter these data before feeding back to the traffic responsive controllers. In this mode Aurora RNM can also be used to detect faults in the sensing equipment.

VI. CONCLUSION

Aurora RNM is the first open-source traffic macrosimulation tool set that could be used not just for operational planning, but also for active traffic management. It is computationally efficient, allows the user to create scenarios, and provides means for simulation of complex control strategies. This paper shows how Aurora RNM accounts for multiple vehicle types; describes modeling of HOV; explains its function as a dynamic filter for real-time measurements; and presents virtual sensors, software objects that simulate real measurement devices or serve as interfaces to these devices.

REFERENCES

[1] Aurora RNM Homepage. http://code.google.com/p/aurorarnm.
[2] C. F. Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research, B*, 28(4):269–287, 1994.
[3] C. F. Daganzo. The cell transmission model II: Network traffic. *Transportation Research, B*, 29(2):79–93, 1995.
[4] C. F. Daganzo. A continuum theory of traffic dynamics for freeways with special lanes. *Transportation Research, B*, 31(2):83–102, 1997.
[5] C. F. Daganzo, W.-H. Lin, and J. M. Del Castillo. A simple physical principle for the simulation freeways with special lanes and priority vehicles. *Transportation Research, B*, 31(2):103–125, 1997.
[6] G. Dervisoglu, G. Gomes, J. Kwon, A. Muralidharan, and P. Varaiya. Automatic calibration of the fundamental diagram and empirical observations on capacity. *88 Annual Meeting of the Transportation Research Board, Washington, D.C., USA*, 2008.
[7] E. N. Holland and A. W. Woods. A continuum model for the dispersion of traffic on two-lane roads. *Transportation Research, B*, 31(6):473–485, 1997.
[8] A. A. Kurzhanskiy. Set-valued estimation of freeway traffic density. *Proceedings of the 12th IFAC Symposium on Control in Transportation Systems*, 2009.
[9] A. A. Kurzhanskiy, J. Kwon, and P. Varaiya. Aurora Road Network Modeler. *Proceedings of the 12th IFAC Symposium on Control in Transportation Systems*, 2009.
[10] A. A. Kurzhanskiy and A. Muralidharan. Macroscopic modeling of multiple vehicle types and freeways with HOV lanes. 2009. Online: http://path.berkeley.edu/topl/docs.html.
[11] W. H. Lin. Spillovers, merging traffic and the morning commute. *Proceedings of the 4th IEEE Intelligent Transportation Systems Conference, Oakland, CA*, 2001.
[12] A. Muralidharan, G. Dervisoglu, and R. Horowitz. Freeway traffic flow simulation using the cell transmission model. American Control Conference, 2009.
[13] I. Okutani and Y. J. Stephanedes. Dynamic prediction of traffic volume through kalman filtering theory. *Transportation Research, B*, 18:1–11, 1984.
[14] M. Papageorgiou and P. Varaiya. Link vehicle-count — the missing measurement for traffic control. *Proceedings of the 12th IFAC Symposium on Control in Transportation Systems*, 2009.
[15] PeMS Homepage. http://pems.eecs.berkeley.edu.
[16] Sensys Networks. http://www.sensysnetworks.com.
[17] B. L. Smith, B. M. Williams, and R. K. Oswald. Comparison of parametric and nonparametric models for traffic flow forcasting. *Transportation Research, C*, 10:303–321, 2002.
[18] Texas Transportation Institute. Annual Urban Mobility Report., 2009. http://mobility.tamu.edu/ums.
[19] Transportation Research Bord. Highway capacity manual. *National Research Council, Washington, D.C.*, 2000.
[20] H. M. Zhang and W. L. Jin. Kinematic wave traffic flow model for mixed traffic.