State Estimation for Polyhedral Hybrid Systems and Applications to the Godunov Scheme for Highway Traffic Estimation

Jérôme Thai and Alexandre M. Bayen

***Abstract*—This paper investigates the problem of estimating the state of discretized hyperbolic scalar partial differential equations. It uses a Godunov scheme to discretize the so-called *Lighthill–Whitham–Richards* equation with a triangular flux function, and proves that the resulting nonlinear dynamical system can be decomposed in a *piecewise affine* manner. Using this explicit representation, the system is written as a switching dynamical system, with a state space partitioned into an exponential number of polyhedra in which one mode is active. We propose a feasible approach based on the interactive multiple model (IMM) which is a widely used algorithm for estimation of hybrid systems in the scientific community. The number of modes is reduced based on the geometric properties of the polyhedral partition. The *k-means* algorithm is also applied on historical data to partition modes into clusters. The performance of these algorithms are compared to the extended Kalman filter and the ensemble Kalman filter in the context of Highway Traffic State Estimation. In particular, we use sparse measurements from loop detectors along a section of the I-880 to estimate the state density for our numerical experiments.**

***Index Terms*—Lighthill–Whitham–Richards (LWR), partial differential equations (PDEs).**

I. Introduction

PARTIAL differential equations (PDEs) are often used in traffic as density based traffic models because they provide a concise mathematical model to capture essential properties of a wide variety of phenomena such as fluid flow, heat, and electrodynamics. Based on the conservation of flow, the *Lighthill–Whitham–Richards* (LWR) PDE [21], [25] and its discretization using the Godunov scheme [14], [18], [27] have been widely used in the scientific community for modelling traffic, they also known as the *Cell Transmission Model* (CTM) [6], [7] in the transportation literature. State of the art traffic estimation techniques for this model include the application of the *extended Kalman filter* (EKF) to the LWR PDE by Schreiter *et al.* [26], and to non-scalar traffic model by Papageorgiou [24]. The application of the EKF to the LWR PDE model is problematic due to the non-differentiability of its discretization, a problem which has been partially addressed in [3] and [29]. The *ensemble Kalman filter* (EnKF) has also been applied to a velocity-based model in [30], in order to circumvent the difficulties of non-differentiability of numerical solutions to these PDEs such as the one presented in this paper.

The Godunov scheme applied to the LWR model for a triangular flux function can be proven to lead to a *piecewise affine* (PWA) hybrid system, which is one of the contributions of this paper. Each cell of the discretized system switches between several linear models. We define this new class of systems as *multicellular hybrid systems*. The resulting switching-mode dynamical system combines discrete dynamics modeled by a finite automaton for the *transitions* between the modes and continuous dynamics in the form of linear discretized dynamical systems. Estimation of hybrid systems has been widely studied in past work [16], [17]. In particular, such techniques have been successfully used for aircraft tracking in [13] in which Bar–Shalom's *interacting multiple model* (IMM) algorithm was used [1]. Similar hybrid estimation algorithms and their applications are described in [12], [23], [28]. While the IMM algorithm seems a natural approach for the estimation of hybrid systems, it is intractable when applied to the discretized LWR PDE (thus highway models) because the combination of the modes of each cell induce an exponential number of modes. *A priori*, each cell of the discretized model can be in seven different modes, which leads to 7^n modes, where n is the dimension of the state thus creating serious computational challenges in the estimation problem. One possible way to address this is with the *mixture Kalman filter* algorithm [5] which handles this complexity by randomly sampling in the space of modes.

Our work contains four contributions. To the best of our knowledge, this is the first time that an explicit piecewise affine decomposition of the Godunov is formulated. 1) For a fixed mode vector $\boldsymbol{m}$, the Godunov scheme is locally affine, and we have an explicit formulation of the linear dynamics. 2) The domains of the mode vectors $\mathrm{Dom}(\boldsymbol{m})$ are also expressed with explicit linear constraints, and they form a polyhedral partition of the state space. Even though the IMM is a natural algorithm for hybrid estimation, it is not tractable because of the exponential number of modes. Hence, the second contribution consists in proposing two methods: 3) The first one takes advantage of the geometric properties of the space of

Manuscript received October 24, 2013; revised April 29, 2014 and June 20, 2014; accepted July 7, 2014. Date of publication July 23, 2014; date of current version January 21, 2015. Recommended by Associate Editor C. Prieur.

J. Thai is with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94709 USA (e-mail: jerome.thai@berkeley.edu).

A. M. Bayen is with the Systems Engineering, Department of Electrical Engineering and Computer Sciences, Department of Civil and Environmental Engineering, University of California, Berkeley, CA 94720-1764 USA (e-mail: bayen@berkeley.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2014.2342151

modes to reduce the set of modes to the mode of the current estimate and its adjacent modes. 4) The second one uses a clustering algorithm on historical data to reduce the set of modes to a representative sets: then the reduced model only switches between these modes.

The rest of the paper is organized as follows: Section II presents the mathematical model used and unravels the PWA expression of the Godunov scheme. Section III presents the polyhedral properties of the space of modes. Section IV shows that the IMM applied to the discretized system is not tractable. Section V presents feasible algorithms inspired from IMM using the PWA character of the Godunov scheme and *k-means*.[1]

II. Hybrid Automaton

A. LWR Model

Lighthill, Whitham in 1955 [21], and Richards in 1956 [25] introduced a macroscopic dynamic model of traffic based on conservation of vehicles, using Greenshields' hypothesis [10] of a static flow/density relationship (1), known as the *flux function*:

$$q(x,t) = Q\left(\rho(x,t)\right) \tag{1}$$

where $\rho(x,t)$ and $q(x,t)$ denote the density and the flow of vehicles at location x and time t respectively. The flux function Q is assumed to be a function of the density only. The conservation of mass can be rewritten as follows:

$$\begin{aligned} &\frac{\partial \rho(x,t)}{\partial t} + \frac{\partial Q\left(\rho(x,t)\right)}{\partial x} = 0, \quad \forall (x,t) \in [0,L] \times \mathbb{R}_+ \\ &\rho(0,t) = u(t), \quad \rho(L,t) = d(t) \quad \forall t \in \mathbb{R}_+ \\ &\rho(x,0) = \rho_0(x), \quad \forall x \in [0,L] \end{aligned} \tag{2}$$

where $u(t), d(t)$ are the upstream and downstream densities, respectively, and $\rho_0(x)$ is the initial state [21], [25]. This equation is commonly known as the *Lighthill–Whitham–Richards*, or LWR, model. Different flux functions have been suggested.

At each boundary, the ability to prescribe the value of the solution depends on the sign of the characteristic curve (if it is entering the domain, it can be done in the strong sense, otherwise it cannot be done). Thus, in order for the problem to be well posed, one needs to prescribe the boundary conditions in the weak sense, and they can either apply at the two boundaries, at one boundary or at none of the boundaries, depending on the value of the function in the interior of the domain. This result is described in detail in [2] for a compact domain. It was later instantiated for specific PDEs, in particular in the work of [15], and in the specific case of traffic (concave flux function) in [27].

B. Assumptions and Notations

In the rest of this paper, we will focus on the analysis of the Godunov scheme, which is a conservative numerical scheme for solving PDE. We assume that traffic densities are between 0 and ρ_{jam}, i.e., the density $\rho(x,t)$ is in $[0, \rho_{\mathrm{jam}}]$ for all x, t.

[1]Code available here: https://github.com/jeromethai/hybrid-LWR-estimation

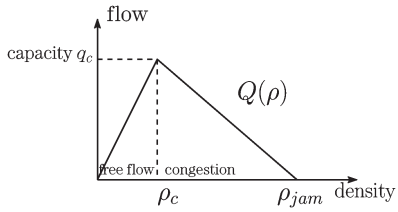

Fig. 1. Speed and flow relationships for triangular flux function.

The widely-used *triangular flux function* described in [6] is also chosen for our dynamic model and results are derived from it. It is a function of the density ρ. It assumes a constant velocity in free-flow and a hyperbolic velocity in congestion as shown in Fig. 1:

$$Q(\rho) = \begin{cases} v_f \rho & \text{if } \rho \le \rho_c \\ -\omega_f(\rho - \rho_{\mathrm{jam}}) & \text{if } \rho > \rho_c \end{cases} \tag{3}$$

where $\omega_f = v_f \rho_c / (\rho_{\mathrm{jam}} - \rho_c)$ is the backward propagation wave speed.

We also assume for simplicity and clarity that the segment of road we are modeling is homogeneous, i.e., the parameters of the flux function $\omega_f, v_f, \rho_{\mathrm{jam}}, \rho_c, q_c$ are uniform along the cells of the discretized road. All the results derived in the rest of this paper still remain valid for an heterogeneous road.

C. Godunov Scheme

A seminal numerical method to solve the above equations is given by the Godunov scheme, which is based on exact solutions to Riemann problems [8], [9]. This leads to the construction of a nonlinear discrete time dynamical system. The Godunov discretization scheme is applied on the LWR PDE, where the discrete time step Δt is indexed by t, and the discrete space step Δx is indexed by i:

$$\rho_i^{t+1} = \rho_i^t - \frac{\Delta t}{\Delta x}\left(G\left(\rho_i^t, \rho_{i+1}^t\right) - G\left(\rho_{i-1}^t, \rho_i^t\right)\right), \quad i = 1, \cdots, n. \tag{4}$$

In order to ensure numerical stability, the time and space steps are coupled by the CFL condition [18]: $c_{\max}(\Delta t/\Delta x) \le 1$ where $c_{\max}$ denotes the maximal characteristic speed.

The Godunov flux can be expressed as the minimum of the *sending flow* $S(\rho)$ from the upstream cell and the *receiving flow* $R(\rho)$ from the downstream cell through a boundary connecting two cells of a homogeneous road (i.e., the upstream and downstream cells have the same characteristics). For the triangular flux function

$$\begin{aligned} G(\rho_1, \rho_2) &= \min\left(S(\rho_1), R(\rho_2)\right) \\ S(\rho) &= \begin{cases} Q(\rho) = v_f \rho & \text{if } \rho \le \rho_c \\ q_c & \text{if } \rho > \rho_c \end{cases} \\ R(\rho) &= \begin{cases} q_c & \text{if } \rho \le \rho_c \\ Q(\rho) = -\omega_f(\rho - \rho_{\mathrm{jam}}) & \text{if } \rho > \rho_c \end{cases} \end{aligned} \tag{5}$$

where ρ_1 is the density of the cell upstream and ρ_2 is the density of the cell downstream.

As shown in Fig. 2(a), the application of the Godunov scheme to the flux functions introduces intuitive concepts of *supply* and *demand* at the boundary connecting two cells.

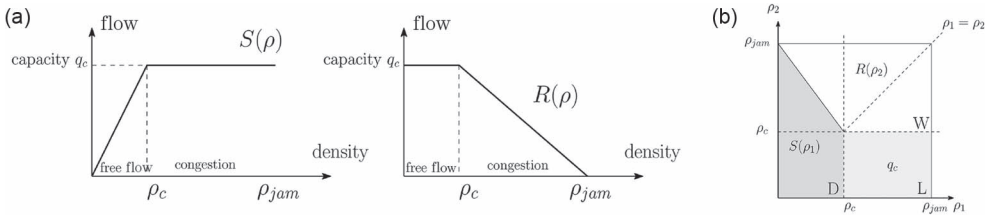

Fig. 2. (a) Sending and receiving flows for triangular flux function. (b) Values of $G(\rho_1, \rho_2)$ in the space $[0, \rho_{\rm jam}]^2$.

Given the partition of the space $[0, \rho_{\rm jam}]^2$ in different regions $\mathbf{W}$, $\mathbf{L}$, and $\mathbf{D}$ as shown in Fig. 2(b), the function $G(\rho_1, \rho_2)$ takes different values.

Lemma 1: With a triangular flux function, the Godunov flux $(\rho_1, \rho_2) \in [0, \rho_{\rm jam}]^2 \mapsto G(\rho_1, \rho_2)$ is piecewise affine:

$$G(\rho_1,\rho_2)=\begin{cases}-\omega_f(\rho_2-\rho_{\rm jam}) & \text{if } (\rho_1,\rho_2)\in\mathbf{W}\\ q_c & \text{if } (\rho_1,\rho_2)\in\mathbf{L}\\ v_f\rho_1 & \text{if } (\rho_1,\rho_2)\in\mathbf{D}\end{cases}$$

$$\mathbf{W}:=\left\{(\rho_1,\rho_2)|\rho_2+\frac{v_f}{w_f}\rho_1>\rho_{\rm jam},\rho_2>\rho_c\right\}$$
$$\mathbf{L}:=\{(\rho_1,\rho_2)|\rho_1>\rho_c,\rho_2\le\rho_c\}$$
$$\mathbf{D}:=\left\{(\rho_1,\rho_2)|\rho_2+\frac{v_f}{w_f}\rho_1\le\rho_{\rm jam},\rho_1\le\rho_c\right\}. \quad (6)$$

Proof: We recall that $(\rho_1, \rho_2) \in [0, \rho_{\rm jam}]^2$. Equations (5) imply

$$\begin{aligned}\rho_1,\rho_2\le\rho_c &\Longrightarrow G(\rho_1,\rho_2)=\min(v_f\rho_1,q_c)=v_f\rho_1\\ \rho_1,\rho_2\ge\rho_c &\Longrightarrow G(\rho_1,\rho_2)=\min\left(q_c,-w_f(\rho_2-\rho_{\rm jam})\right)\\ &\quad=-w_f(\rho_2-\rho_{\rm jam})\\ \rho_1\ge\rho_c,\rho_2\le\rho_c &\Longrightarrow G(\rho_1,\rho_2)=\min(q_c,q_c)=q_c\\ \rho_1\le\rho_c,\rho_2\ge\rho_c &\Longrightarrow G(\rho_1,\rho_2)=\min\left(v_f\rho_1,-w_f(\rho_2-\rho_{\rm jam})\right).\end{aligned}$$

The third implication proves our result for the region $\mathbf{L}$. Then, given $\rho_1 \le \rho_c, \rho_2 \ge \rho_c$, $G(\rho_1, \rho_2) = v_f\rho_1 \Longleftrightarrow v_f\rho_1 \le -w_f(\rho_2 - \rho_{\rm jam}) \Longleftrightarrow \rho_2 + (v_f/w_f)\rho_1 \le \rho_{\rm jam}$. Finally, we note that $\{\rho_1 \le \rho_c, \rho_2 \le \rho_c\} \cup \{\rho_2 + (v_f/w_f)\rho_1 \le \rho_{\rm jam}\} = \{\rho_2 + (v_f/w_f)\rho_1 \le \rho_{\rm jam}, \rho_1 \le \rho_c\}$, hence the definition of $\mathbf{D}$ in (6). The result for $\mathbf{W}$ follows similarly. □

D. Godunov Scheme as a Hybrid Automaton

We now consider an entire link divided into n cells and *we add two ghost cells on the left and right sides of the domain*. Hence, the discrete state space is indexed by $i = 0, 1, \cdots, n+1$, the state of the system is $\boldsymbol{\rho} = [\rho_0, \cdots, \rho_{n+1}]^T \in [0, \rho_{\rm jam}]^{n+2}$, and the dimension is $n + 2$. The density at cell i and time t is then ρ_i^t, the ith entry of vector $\boldsymbol{\rho}$, and the values of ρ_0^t and ρ_{n+1}^t are given by the prescribed boundary conditions to be imposed on the left and right side of the domain, respectively, i.e., $\rho_0^t = u(t)$ and $\rho_{n+1}^t = d(t)$ for all t where $u(t)$ and $d(t)$ are the upstream and downstream densities, respectively.

In the rest of this section we present a simple analysis for the formulation of the discretized system as a *piecewise affine* autonomous hybrid automaton. We will sometimes use the lighter notation $\rho_i^+ = \rho_i - \alpha(G(\rho_i, \rho_{i+1}) - G(\rho_{i-1}, \rho_i))$ for the Godunov scheme (4) with $\alpha = \Delta t/\Delta x$. We rewrite (6) in the state space $[0, \rho_{\rm jam}]^{n+2}$:

$$G(\rho_i,\rho_{i+1})=\begin{cases}-\omega_f(\rho_{i+1}-\rho_{\rm jam}) & \text{if } \boldsymbol{\rho}\in\mathbf{W}_{i+1/2}\\ q_c & \text{if } \boldsymbol{\rho}\in\mathbf{L}_{i+1/2}\\ v_f\rho_i & \text{if } \boldsymbol{\rho}\in\mathbf{D}_{i+1/2}\end{cases}\quad \text{for } i=0,\cdots n \quad (7)$$

where $\mathbf{W}_{i+1/2}, \mathbf{L}_{i+1/2}, \mathbf{D}_{i+1/2}, i = 0, \cdots, n$, are $3(n+1)$ polyhedra in $[0, \rho_{\rm jam}]^{n+2}$:

$$\begin{aligned}\mathbf{W}_{i+1/2}&=\left\{\boldsymbol{\rho}\in[0,\rho_{\rm jam}]^{n+2}|\rho_{i+1}+\frac{v_f}{w_f}\rho_i>\rho_{\rm jam},\rho_{i+1}>\rho_c\right\}\\ \mathbf{L}_{i+1/2}&=\left\{\boldsymbol{\rho}\in[0,\rho_{\rm jam}]^{n+2}|\rho_i>\rho_c,\rho_{i+1}\le\rho_c\right\}\\ \mathbf{D}_{i+1/2}&=\left\{\boldsymbol{\rho}\in[0,\rho_{\rm jam}]^{n+2}|\rho_{i+1}+\frac{v_f}{w_f}\rho_i\le\rho_{\rm jam},\rho_i\le\rho_c\right\}.\end{aligned} \quad (8)$$

We note that we can express the polyhedra $\mathbf{W}_{i+1/2}, \mathbf{L}_{i+1/2}$, $\mathbf{D}_{i+1/2}$ in vector form:

$$\begin{aligned}\mathbf{W}_{i+1/2}&=\left\{\boldsymbol{\rho}|\mathbf{d}(1)\cdot[\rho_i,\rho_{i+1},1]^T>0,\mathbf{d}(3)\cdot[\rho_i,\rho_{i+1},1]^T>0\right\}\\ \mathbf{L}_{i+1/2}&=\left\{\boldsymbol{\rho}|\mathbf{d}(2)\cdot[\rho_i,\rho_{i+1},1]^T>0,\mathbf{d}(3)\cdot[\rho_i,\rho_{i+1},1]^T\le0\right\}\\ \mathbf{D}_{i+1/2}&=\left\{\boldsymbol{\rho}|\mathbf{d}(1)\cdot[\rho_i,\rho_{i+1},1]^T\le0,\mathbf{d}(2)\cdot[\rho_i,\rho_{i+1},1]^T\le0\right\}\end{aligned} \quad (9)$$

with coefficients

$$\begin{aligned}\mathbf{d}(1)&=\left[\frac{(\rho_{\rm jam}-\rho_c)}{\rho_c},1,-\rho_{\rm jam}\right]\\ \mathbf{d}(2)&=[1,0,-\rho_c]\\ \mathbf{d}(3)&=[0,1,-\rho_c].\end{aligned} \quad (10)$$

Combining the Godunov scheme (4) and the Godunov flux in PWA form (7):

Lemma 2: With a triangular flux function, the Godunov scheme at cell $i \in \{1, \cdots, n\}$ can be formulated as a hybrid automaton with linear components:

- mode $m_i \in Q$ with $Q := \{1, \cdots, 9\}^2$
- state $\rho_i \in [0, \rho_{\rm jam}]$
- inputs $(\rho_{i-1}^t, \rho_{i+1}^t) \in [0, \rho_{\rm jam}]^2$, $t \ge 0$
- discrete dynamics $\rho_i^{t+1} = L(m_i)\cdot[\rho_{i-1}^t, \rho_i^t, \rho_{i+1}^t]^T + w(m_i)$ if $(\rho_{i-1}^t, \rho_i^t, \rho_{i+1}^t) \in P(\mathrm{Dom}(m_i))$ where $L(\cdot) : Q \mapsto \mathbb{R}^3$

[2]In this description, the mode m_i takes on values in a finite set $Q = \{1, \cdots, 9\}$ for completeness. We will see in Section III that the modes $m_i = 8$ and $m_i = 9$ are *not accepted*.

TABLE I
GODUNOV SCHEME W.R.T. DISCRETE STATES m_i AT CELL i, E.G., IF $\boldsymbol{\rho} \in \mathrm{Dom}(\{m_i = 4\}) = \mathbf{L}_{i-1/2} \cap \mathbf{D}_{i+1/2} = \{\boldsymbol{\rho} | \rho_{i-1} > \rho_c, \rho_i \leq \rho_c, \rho_{i+1} + (v_f/w_f)\rho_i \leq \rho_{\mathrm{jam}}\}$, THEN $\rho_i^{t+1} = L_4 \cdot [\rho_{i-1}^t, \rho_i^t, \rho_{i+1}^t]^T + w_4 = (1 - \alpha v_f)\rho_i^t + \alpha v_f \rho_c$

m_i	$\mathrm{Dom}(m_i)$	$\boldsymbol{L}(m_i)$	$w(m_i)$	$\rho_i^{t+1} = \boldsymbol{L}(m_i) \cdot [\rho_{i-1}^t, \rho_i^t, \rho_{i+1}^t]^T + w(m_i)$
1	$\mathbf{W}_{i-1/2} \cap \mathbf{W}_{i+1/2}$	$\boldsymbol{L}(1) = [0,\ 1 - \alpha w_f,\ \alpha w_f]$	$w(1) = 0$	$\rho_i^{t+1} = (1 - \alpha\omega_f)\rho_i^t + \alpha\omega_f \rho_{i+1}^t$
2	$\mathbf{W}_{i-1/2} \cap \mathbf{L}_{i+1/2}$	$\boldsymbol{L}(2) = [0,\ 1 - \alpha w_f,\ 0]$	$w(2) = \alpha w_f \rho_c$	$\rho_i^{t+1} = (1 - \alpha\omega_f)\rho_i^t + \alpha\omega_f \rho_c$
3	$\mathbf{L}_{i-1/2} \cap \mathbf{W}_{i+1/2}$	$\boldsymbol{L}(3) = [0,\ 1,\ \alpha w_f]$	$w(3) = -\alpha w_f \rho_c$	$\rho_i^{t+1} = \rho_i^t + \alpha\omega_f \rho_{i+1}^t - \alpha\omega_f \rho_c$
4	$\mathbf{L}_{i-1/2} \cap \mathbf{D}_{i+1/2}$	$\boldsymbol{L}(4) = [0,\ 1 - \alpha v_f,\ 0]$	$w(4) = \alpha v_f \rho_c$	$\rho_i^{t+1} = (1 - \alpha v_f)\rho_i^t + \alpha v_f \rho_c$
5	$\mathbf{D}_{i-1/2} \cap \mathbf{W}_{i+1/2}$	$\boldsymbol{L}(5) = [\alpha v_f,\ 1,\ \alpha w_f]$	$w(5) = -\alpha w_f \rho_{\mathrm{jam}}$	$\rho_i^{t+1} = \alpha v_f \rho_{i-1}^t + \rho_i^t + \alpha\omega_f \rho_{i+1}^t - \alpha\omega_f \rho_{\mathrm{jam}}$
6	$\mathbf{D}_{i-1/2} \cap \mathbf{L}_{i+1/2}$	$\boldsymbol{L}(6) = [v_f,\ 1,\ 0]$	$w(6) = -\alpha v_f \rho_c$	$\rho_i^{t+1} = \alpha v_f \rho_{i-1}^t + \rho_i^t - \alpha v_f \rho_c$
7	$\mathbf{D}_{i-1/2} \cap \mathbf{D}_{i+1/2}$	$\boldsymbol{L}(7) = [v_f,\ 1 - \alpha v_f,\ 0]$	$w(7) = 0$	$\rho_i^{t+1} = \alpha v_f \rho_{i-1}^t + (1 - \alpha v_f)\rho_i^t$
8	$\mathbf{W}_{i-1/2} \cap \mathbf{D}_{i+1/2}$	$\boldsymbol{L}(8) = [0,\ 1 - \alpha v_f - \alpha w_f,\ 0]$	$w(8) = \alpha w_f \rho_{\mathrm{jam}}$	$\rho_i^{t+1} = (1 - \alpha v_f - \alpha w_f)\rho_i^t + \alpha w_f \rho_{\mathrm{jam}}$
9	$\mathbf{L}_{i-1/2} \cap \mathbf{L}_{i+1/2}$	$\boldsymbol{L}(9) = [0,\ 0,\ 0]$	$w(9) = 0$	$\rho_i^{t+1} = \rho_i^t$

and $w(\cdot) : Q \mapsto \mathbb{R}$ are defined in Table I, and $P(\cdot)$ is the projection operator onto $Vect(e_{i-1}, e_i, e_{i+1})$.

- domain of the modes $\mathrm{Dom}(m_i)$ defined in the Table I and (8).

We note that $\mathrm{Dom}(m_i)$ refers to the subset of $\mathbb{R}^{n+2}$ in which the mode of cell i is m_i. Since the linear constraints that define $\mathrm{Dom}(m_i)$ (see Table I) only concern variables $\rho_{i-1}, \rho_i, \rho_{i+1}$, the projection onto $Vect(e_{i-1}, e_i, e_{i+1})$ contains all the information on the shape of $\mathrm{Dom}(m_i)$.

Proof: We prove the result for $m_i = 4$, the other cases follow similarly. When $\boldsymbol{\rho} \in \mathrm{Dom}(\{m_i = 4\}) = \mathbf{L}_{i-1/2} \cap \mathbf{D}_{i+1/2}$ following the definition of $\mathrm{Dom}(m_i)$ in Table I, we have $G(\rho_{i-1}, \rho_i) = q_c$ and $G(\rho_i, \rho_{i+1}) = v_f \rho_i$ from (7) then

$$\begin{aligned}\rho_i^+ &= \rho_i - \alpha \left(G(\rho_i, \rho_{i+1}) - G(\rho_{i-1}, \rho_i)\right) \\ &= \rho_i - \alpha(v_f \rho_i - q_c) = (1 - \alpha v_f)\rho_i + \alpha q_c\end{aligned}$$

hence $\rho_i^+ = L(4) \cdot [\rho_{i-1}, \rho_i, \rho_{i+1}]^T + w(4)$ with $L(4) := [0, 1 - \alpha v_f, 0]$ and $w(4) := \alpha v_f \rho_c$ following the definitions of $L(m_i)$ and $w(m_i)$ in Table I. □

We note that the condition $(\rho_{i-1}^t, \rho_i^t, \rho_{i+1}^t) \in P(\mathrm{Dom}(m_i))$ in the discrete dynamics is a reset relation at each time step: the mode at time t is directly given by state $\boldsymbol{\rho}^t$.

E. Discretized System as a Hybrid System

The mode of each cell can be listed in a *vector* $\boldsymbol{m} \in \{1, \cdots, 9\}^n$ in which the ith entry is the discrete state at cell i. We call it the *mode vector*. As a result, the domain of the mode vector $\boldsymbol{m} \in \{1, \cdots, 9\}^n$ is

$$\mathrm{Dom}(\mathbf{m}) = \bigcap_{i=1}^{n} \mathrm{Dom}(m_i). \tag{11}$$

For example, if $n = 2$, then the state $\boldsymbol{\rho} = [\rho_0, \rho_1, \rho_2, \rho_3]$ is in $[0, \rho_{\mathrm{jam}}]^4$ with boundary cells ρ_0 and ρ_3 and the mode vector $\boldsymbol{m}$ is in $\{1, \cdots, 9\}^4$. More specifically

$$\begin{aligned}&\mathrm{Dom}\left(\{\mathbf{m} = (2,3)\}\right) \\ &= \mathrm{Dom}\left(\{m_1 = 2\}\right) \cap \mathrm{Dom}\left(\{m_2 = 3\}\right) \\ &= (\mathbf{W}_{\frac{1}{2}} \cap \mathbf{L}_{1+1/2}) \cap (\mathbf{L}_{1+1/2} \cap \mathbf{W}_{2+1/2}) \\ &= \mathbf{W}_{1/2} \cap \mathbf{L}_{1+1/2} \cap \mathbf{W}_{2+1/2} \\ &= \left\{ \boldsymbol{\rho} \in [0, \rho_{\mathrm{jam}}]^4 | \rho_1 + \frac{v_f}{w_f}\rho_0 > \rho_{\mathrm{jam}}, \rho_1 > \rho_c, \rho_2 \leq \rho_c, \right. \\ &\qquad \left. \rho_3 + \frac{v_f}{w_f}\rho_2 > \rho_{\mathrm{jam}} \right\}.\end{aligned} \tag{12}$$

We will show later that the subsets $\mathrm{Dom}(\boldsymbol{m})$'s form a partition of $[0, \rho_{\mathrm{jam}}]^{n+2}$.

For each mode vector $\boldsymbol{m}$, we construct the matrix $A_{\boldsymbol{m}} \in \mathbb{R}^{(n+2)\times(n+2)}$, and the row vectors $b_{\boldsymbol{m}}, c^t \in \mathbb{R}^{n+2}$ in the form

$$A_{\mathbf{m}} = \begin{bmatrix} 0 & \cdots & 0 \\ L(m_1) & & \\ & \ddots & \\ & & L(m_n) \\ 0 & \cdots & 0 \end{bmatrix},\ b_{\mathbf{m}} = \begin{bmatrix} 0 \\ w(m_1) \\ \vdots \\ w(m_n) \\ 0 \end{bmatrix},\ c^t = \begin{bmatrix} u(t) \\ 0 \\ \vdots \\ 0 \\ d(t) \end{bmatrix} \tag{13}$$

where $L(m_i), w(m_i)$ are defined in Table I, and $u(t), d(t)$ are the upstream and downstream densities, respectively. This leads to one of the main results of the paper:

Proposition 1: The discretized LWR equation using the Godunov scheme and with a triangular flux function is an autonomous hybrid automaton with affine components:

- discrete state $\boldsymbol{m} \in \{1, \cdots, 9\}^n$;
- state $\boldsymbol{\rho}^t \in [0, \rho_{\mathrm{jam}}]^{n+2}$ at time t;
- inputs $(u(t), d(t)) \in [0, \rho_{\mathrm{jam}}]^2$;
- discrete dynamics $\boldsymbol{\rho}^{t+1} = A_{\boldsymbol{m}}\boldsymbol{\rho}^t + b_{\boldsymbol{m}} + c^t \quad if \quad \boldsymbol{\rho}^t \in \mathrm{Dom}(\boldsymbol{m})$;
- domain of the discrete states $\mathrm{Dom}(\boldsymbol{m})$ defined in (11).

Proof: The formulation as a hybrid automaton is obtained by stacking the states and modes in the hybrid automaton formulation of the Godunov scheme into a vector, and the linear transformations into a matrix. □

Finally, we note that the condition $\boldsymbol{\rho}^t \in \mathrm{Dom}(\boldsymbol{m})$ in the discrete dynamics is a reset relation at each time step: the mode at time t is directly given by state $\boldsymbol{\rho}^t$.

Algorithm 1 Find the mode vector: $\mathbf{rho2m}(\boldsymbol{\rho})$. The parameters $\boldsymbol{d}(1), \boldsymbol{d}(2), \boldsymbol{d}(3) \in \mathbb{R}^3$ in (10) describe the domain of each mode vector [see Table I and (17), (10), (11)]

Require: current state $\boldsymbol{\rho} = [\rho_0, \cdots, \rho_{n+1}] \in [0, \rho_{\text{jam}}]^{n+2}$
1: **for** $i \in \{0, \cdots, n\}$ **do**
2: $x = [\rho_i, \rho_{i+1}, 1]^T$
3: $\boldsymbol{I} = [\boldsymbol{d}(1)x > 0, \boldsymbol{d}(2)x > 0, \boldsymbol{d}(3)x > 0] \in \{0,1\}^3$
4: **if** $\boldsymbol{I}(1) \wedge \boldsymbol{I}(3)$ **then** $s(i) = W \quad \boldsymbol{\rho} \in \mathbf{W}_{i+1/2}$
5: **if** $\boldsymbol{I}(2) \wedge \neg\boldsymbol{I}(3)$ **then** $s(i) = L \quad \boldsymbol{\rho} \in \mathbf{L}_{i+1/2}$
6: **if** $\neg\boldsymbol{I}(1) \wedge \neg\boldsymbol{I}(2)$ **then** $s(i) = D \quad \boldsymbol{\rho} \in \mathbf{D}_{i+1/2}$
7: **end for**
8: **for** $i = \{1, \cdots, n\}$ **do**
9: **if** $\{s(i) = W\} \wedge \{s(i+1) = W\}$ **then** $m_i = 1$
10: **if** $\{s(i) = W\} \wedge \{s(i+1) = L\}$ **then** $m_i = 2$
11: **if** $\{s(i) = L\} \wedge \{s(i+1) = W\}$ **then** $m_i = 3$
12: **if** $\{s(i) = L\} \wedge \{s(i+1) = D\}$ **then** $m_i = 4$
13: **if** $\{s(i) = D\} \wedge \{s(i+1) = W\}$ **then** $m_i = 5$
14: **if** $\{s(i) = D\} \wedge \{s(i+1) = L\}$ **then** $m_i = 6$
15: **if** $\{s(i) = D\} \wedge \{s(i+1) = D\}$ **then** $m_i = 7$
16: **end for**
17: **return** $\boldsymbol{m} = [m_1, \cdots, m_n] \in \{1, \cdots, 7\}^n$

III. Description of the Mode Vectors

A. Accepted Mode Vectors

The following analysis is motivated by the fact that $\text{Dom}(\boldsymbol{m}) = \emptyset$ for some values of $\boldsymbol{m}$, which means that some of the mode vectors $\boldsymbol{m}$'s are not *accepted* by the system.

Definition 1: We say that a mode vector $\boldsymbol{m}$ is *accepted* by the system if and only if its domain $\text{Dom}(\boldsymbol{m})$ is not empty.

Proposition 2: The mode vector $\boldsymbol{m} \in \{1, \cdots, 9\}^n$ is accepted by the system if and only if we have the following two conditions:

$$m_i \in \{1, \cdots, 7\}, \quad \forall i \in \{1, \cdots, n\} \tag{14}$$

$$\forall i \in \{1, \ldots, n-1\}, \quad m_{i+1} \in \begin{cases} \{1,2\} & if\ m_i \in \{1,3,5\} \\ \{3,4\} & if\ m_i \in \{2,6\} \\ \{5,6,7\} & if\ m_i \in \{4,7\}. \end{cases} \tag{15}$$

Proof: From (8), it is easy to see that $\mathbf{W}_{i-1/2} \cap \mathbf{D}_{i+1/2} = \mathbf{L}_{i-1/2} \cap \mathbf{L}_{i+1/2} = \emptyset$ for all $i = 1, \ldots, n$. Hence, $\text{Dom}(\{m_i = 8\}) = \text{Dom}(\{m_i = 9\}) = \emptyset$ (see Table I). In other words, $\boldsymbol{m}$ is not accepted if it has an entry in {8,9} which gives the first condition.

We note that for a fixed i, the polyhedra $\mathbf{W}_{i+1/2}, \mathbf{L}_{i+1/2}, \mathbf{D}_{i+1/2}$ partition $[0, \rho_{\text{jam}}]^{n+2}$. Since $\text{Dom}(m_i) \cap \text{Dom}(m_{i+1})$ is of the form

$$\text{Dom}(m_i) \cap \text{Dom}(m_{i+1})$$
$$= (\mathbf{P}_{i-1/2} \cap \mathbf{P}_{i+1/2}) \cap (\mathbf{P}'_{i+1/2} \cap \mathbf{P}_{i+1+1/2}) \subset \mathbf{P}_{i+1/2} \cap \mathbf{P}'_{i+1/2}$$

with $\mathbf{P}_{i+1/2}, \mathbf{P}'_{i+1/2} \in \{\mathbf{W}_{i+1/2}, \mathbf{L}_{i+1/2}, \mathbf{D}_{i+1/2}\}$, then $\boldsymbol{m}$ is accepted if $\mathbf{P}_{i+1/2} = \mathbf{P}'_{i+1/2}$. In other words, $\text{Dom}(m_i) = \mathbf{P}_{i-1/2} \cap \mathbf{P}_{i+1/2}$ and $\text{Dom}(m_{i+1}) = \mathbf{P}'_{i+1/2} \cap \mathbf{P}_{i+1+1/2}$ must overlap. This gives condition (15).

Reciprocally, if $\boldsymbol{m}$ satisfies conditions (14) and (15), then we have overlaps between $\text{Dom}(m_i)$ and $\text{Dom}(m_{i+1})$. Hence, $\text{Dom}(\boldsymbol{m})$ is of the form

$$\text{Dom}(\mathbf{m}) = \bigcap_{i=0}^{n} \mathbf{P}_{i+1/2}$$

$$\mathbf{P}_{i+1/2} \in \{\mathbf{W}_{i+1/2}, \mathbf{L}_{i+1/2}, \mathbf{D}_{i+1/2}\},\ i = 0, \ldots, n. \tag{16}$$

The intersection of any pair of two consecutive polyhedra in (16) has to be among the first seven subsets in Table I. Hence, for all $i = 1, \ldots, n$, the projection of $\text{Dom}(\boldsymbol{m})$ onto $Vect(e_{i-1}, e_i, e_{i+1})$ is one of the seven subsets of $\mathbb{R}^3$ shown in Fig. 3, which are all nonempty. Hence, $\text{Dom}(\boldsymbol{m})$ is the product of nonempty spaces; hence, it is nonempty. □

From the analysis above, we also conclude that under conditions (14) and (15), the domain of an accepted mode vector can be decomposed in the form (16). This is illustrated in the derivation of $\text{Dom}(\{\boldsymbol{m} = (2,3)\})$ in example (12) above.

From (14), the space of discrete states of the Godunov scheme at each cell is reduced to $\{1, \ldots, 7\}$ and the space in which the mode vector $\boldsymbol{m}$ lies is reduced to $\{1, \ldots, 7\}^n$.

Definition 2: For an accepted mode vector $\boldsymbol{m}$ and the associated $\text{Dom}(\boldsymbol{m}) = \bigcap_{i=0}^{n} \mathbf{P}_{i+1/2}$, a *mode string* $\boldsymbol{s} = s(0)s(1)s(2)\cdots s(n)$ is associated with $\boldsymbol{m}$ if $s(i) = W$ (resp. L, D) if $\mathbf{P}_{i+1/2} = \mathbf{W}_{i+1/2} (resp. \mathbf{L}_{i+1/2}, \mathbf{D}_{i+1/2})$ and a mode string is accepted if and only if $s(i)s(i+1) \in \{WW, WL, LW, LD, DW, DL, DD\}$ for all i, from the analysis done in Proposition 2.

Proposition 3: The number of accepted mode vectors is asymptotically $3.1778 \cdot (2.2470)^n$. (Proof in the Appendix (see Fig. 4)).

Proposition 4: The polyhedra $\text{Dom}(\boldsymbol{m})$ associated with accepted mode vectors $\boldsymbol{m}$ form a partition of $[0, \rho_{\text{jam}}]^{n+2}$.

Proof: Let $\boldsymbol{m}$ and $\boldsymbol{m}'$ be two distinct accepted mode vectors and $\boldsymbol{s}, \boldsymbol{s}'$ the associated strings. We pick $i \in \{0, \ldots, n\}$ such that $s(i) \neq s'(i)$. Then $\text{Dom}(\boldsymbol{m}) \subset \mathbf{P}_{i+1/2}$ and $\text{Dom}(\boldsymbol{m}') \subset \mathbf{P}'_{i+1/2}$, where $\mathbf{P}_{i+1/2}$ and $\mathbf{P}'_{i+1/2}$ are two distinct polyhedra among $\mathbf{W}_{i+1/2}$, $\mathbf{L}_{i+1/2}$, $\mathbf{D}_{i+1/2}$. Hence, $\text{Dom}(\boldsymbol{m})$ and $\text{Dom}(\boldsymbol{m}')$ are disjoint, and for any $\boldsymbol{\rho} \in [0, \rho_{\text{jam}}]^{n+2}$, we can find its associated accepted mode vector $\boldsymbol{m}$ such that $\boldsymbol{\rho} \in \text{Dom}(\boldsymbol{m})$; hence, the different $\text{Dom}(\boldsymbol{m})$ span the whole state space. □

Algorithm 2 mode vector $\boldsymbol{m}$ to mode string: $\mathbf{m2s}(\boldsymbol{m})$

Require: accepted mode vector $\boldsymbol{m}$.
1: **if** $m_1 \in \{1,2\}$ **then** $s(0) = W \quad \mathbf{P}_{1/2} = \mathbf{W}_{1/2}$ in (16)
2: **if** $m_1 \in \{3,4\}$ {**then** $s(0) = L \quad \mathbf{P}_{1/2} = \mathbf{L}_{1/2}$ in (16)
3: **if** $m_1 \in \{5,6,7\}$ **then** $s(0) = D \quad \mathbf{P}_{1/2} = \mathbf{D}_{1/2}$ in (16)
4: **for** $i \in \{1, \ldots, n\}$ **do**
5: $m_i \in \{1,3,5\}$ **then** $s(i) = W \quad \mathbf{P}_{i+1/2} = \mathbf{W}_{i+1/2}$ in (16)
6: $m_i \in \{2,6\}$ **then** $s(i) = L \quad \mathbf{P}_{i+1/2} = \mathbf{L}_{i+1/2}$ in (16)
7: $m_i \in \{4,7\}$ **then** $s(i) = D \quad \mathbf{P}_{i+1/2} = \mathbf{D}_{i+1/2}$ in (16)
8: **end for**
9: **return** the mode string $s(0)s(1)\cdots s(n)$

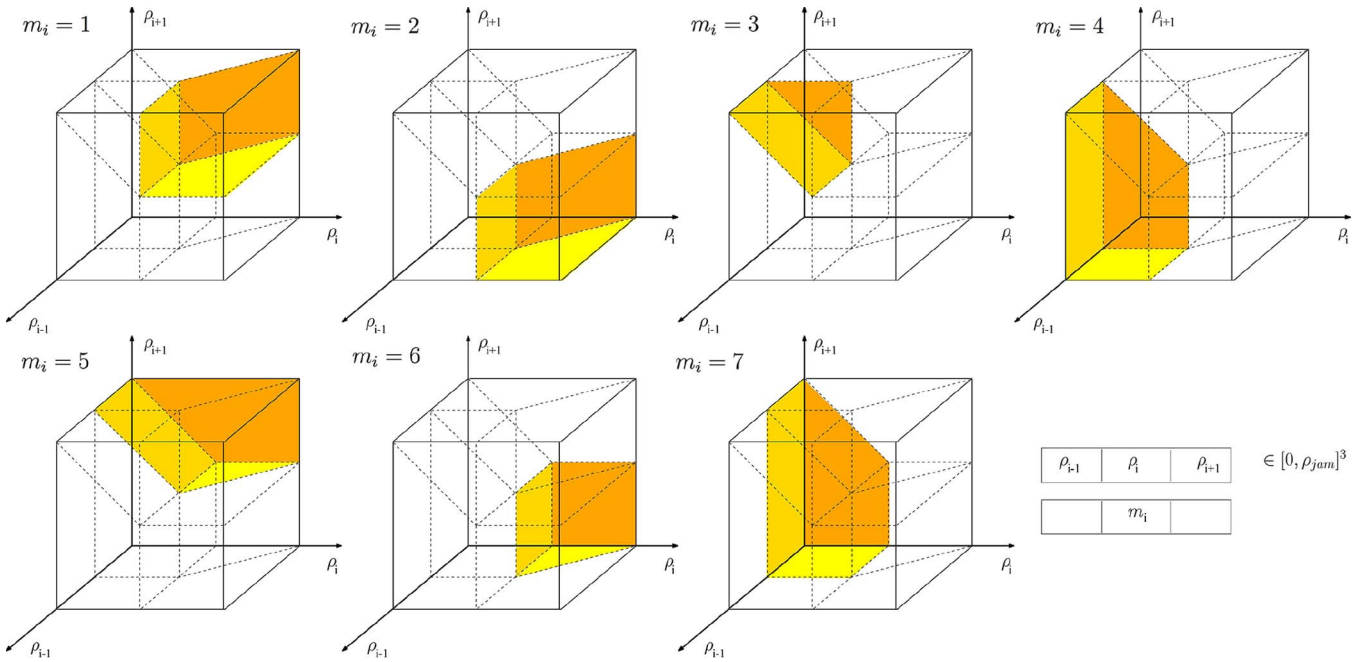

Fig. 3. Projection of $\mathrm{Dom}(m_i)$ onto $Vect(e_{i-1}, e_i, e_{i+1})$ for $i \in \{1, \ldots, 7\}$. For example, in the top left figure, if $(\rho_{i-1}, \rho_i, \rho_{i+1})$ is in the orange polyhedron, then $\boldsymbol{\rho} \in \mathbf{W}_{i-1/2} \cap \mathbf{W}_{i+1/2} = \mathrm{Dom}(\{m_i = 1\})$, the mode m_i is 1 (see Table I).

Algorithm 3 mode string to mode vector: $\mathbf{s2m}(s(0) \cdots s(n))$

Require: accepted mode string $s(0) \cdots s(n)$.
1: apply lines 8 to 16 of Algorithm 1
2: **return** the mode vector $\boldsymbol{m}$

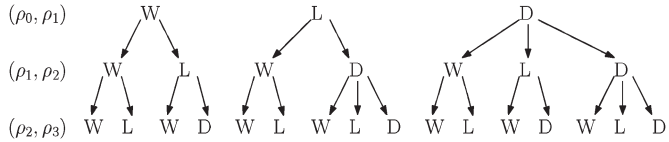

Fig. 4. Sixteen *accepted mode strings* for the first three pairs (ρ_0, ρ_1), (ρ_1, ρ_2), and (ρ_2, ρ_3). For more details, see Propositions 2 and 3.

B. Minimal Representation

We now introduce the concepts of *minimal representation* and *adjacent polyhedra*.

Definition 3 (Faces of a Polyhedron): A supportive hyperplane of a closed convex set $\mathbf{C}$ is a hyperplane $\partial\mathbf{H}$ such that $\mathbf{C} \cap \partial\mathbf{H} \neq \emptyset$ and $\mathbf{C} \subseteq \mathbf{H}$, where $\mathbf{H}$ is one of the two closed half-spaces (associated with the hyperplane). Given a (closed) polyhedron $\mathbf{P}$, the intersection with any supportive hyperplane is a face of $\mathbf{P}$. Moreover, a vertex is a zero-dimension face, an edge a one-dimension face, and a facet is a face of dimension $d-1$ if $\mathbf{P}$ is of dimension d. For a full-dimensional polyhedron, a facet is of dimension $n+1$ (recall that the space $[0, \rho_{\mathrm{jam}}]^{n+2}$ is of dimension $n+2$).

Definition 4 (Minimal H-Representation): There exist infinitely many H-descriptions of a (closed) convex polytope. For a full-dimensional convex polytope, the minimal H-description is unique and is given by the set of the facet-defining half-spaces [11].

We now want to find the minimal representation of $\mathrm{Dom}(\boldsymbol{m}) = \bigcap_{i=0}^{n} \mathbf{P}_{i+1/2}$ for all accepted modes $\boldsymbol{m}$. Each one of the $3(n+1)$ polyhedra $\mathbf{W}_{i+1/2}, \mathbf{L}_{i+1/2}, \mathbf{D}_{i+1/2}, i = 0, \ldots, n$ defined in (8) is intersection of two half-spaces:

$$\begin{aligned} \mathbf{W}_{i+1/2} &= \mathbf{H}_{i+1/2} \cap \mathbf{H}_{i+1} \\ \mathbf{L}_{i+1/2} &= \mathbf{H}_i \cap \mathbf{H}_{i+1}^c \\ \mathbf{D}_{i+1/2} &= \mathbf{H}_i^c \cap \mathbf{H}_{i+1/2}^c \end{aligned} \tag{17}$$

where

$$\begin{aligned} \mathbf{H}_i &= \left\{ \boldsymbol{\rho} \in [0, \rho_{\mathrm{jam}}]^{n+2} | \rho_i > \rho_c \right\}, \quad i = 0, \ldots, n+1 \\ \mathbf{H}_{i+1/2} &= \left\{ \boldsymbol{\rho} \in [0, \rho_{\mathrm{jam}}]^{n+2} | \rho_{i+1} + \frac{v_f}{w_f} \rho_i > \rho_{\mathrm{jam}} \right\}, \ i = 0, \ldots, n \end{aligned} \tag{18}$$

and $\mathbf{H}_i^c$, $\mathbf{H}_{i+1/2}^c$ are the complementary of $\mathbf{H}_i$ and $\mathbf{H}_{i+1/2}$, respectively. The projections of these half-spaces on $Vect(e_i, e_{i+1})$ are illustrated in Fig. 5.

In example (12), we have

$$\begin{aligned} &\mathrm{Dom}\left(\{\mathbf{m} = \{2,3\}\}\right) \\ &\quad = \mathbf{W}_{\frac{1}{2}} \cap \mathbf{L}_{1+1/2} \cap \mathbf{W}_{2+1/2} \\ &\quad = (\mathbf{H}_{\frac{1}{2}} \cap \mathbf{H}_1) \cap (\mathbf{H}_1 \cap \mathbf{H}_2^c) \cap (\mathbf{H}_{2+1/2} \cap \mathbf{H}_3) \\ &\quad = \mathbf{H}_{\frac{1}{2}} \cap \mathbf{H}_1 \cap \mathbf{H}_2^c \cap \mathbf{H}_{2+1/2} \cap \mathbf{H}_3 \\ &\quad = \mathbf{H}_{\frac{1}{2}} \cap \mathbf{H}_1 \cap \mathbf{H}_2^c \cap \mathbf{H}_{2+1/2}. \end{aligned}$$

Since $\mathbf{H}_2^c \cap \mathbf{H}_{2+1/2} \subset \mathbf{H}_3$, we can remove $\mathbf{H}_3$ from the intersection. After removing this redundant constraint, the last equality gives the *minimal representation* of $\mathrm{Dom}(\{\boldsymbol{m} = \{2,3\}\})$.

While finding the minimal representation of a nonempty polyhedron can be difficult in general, it is easy for the

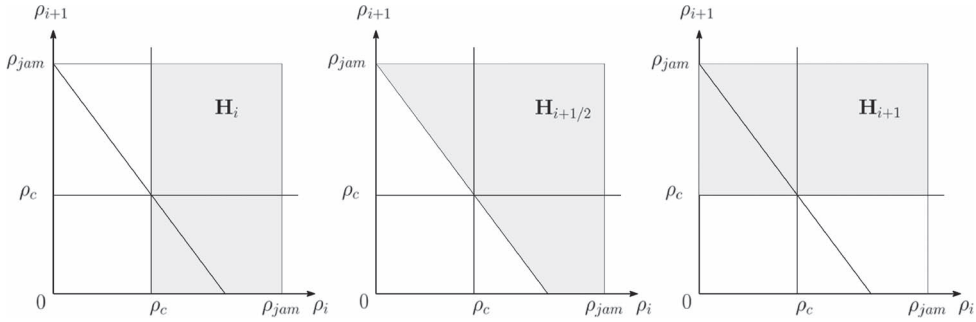

Fig. 5. Projection of the half-spaces $\mathbf{H}_i, \mathbf{H}_{i+1/2}, \mathbf{H}_{i+1}$ on the plane $Vect(e_i, e_{i+1})$.

polyhedra $\mathrm{Dom}(\boldsymbol{m})$ associated with accepted mode vectors $\boldsymbol{m}$. In the form $\mathrm{Dom}(\boldsymbol{m}) = \bigcap_{i=0}^{n} \mathbf{P}_{i+1/2}$, we sequentially derive the minimal representation of each polyhedron of the decreasing sequence $\{\bigcap_{i=0}^{k} \mathbf{P}_{i+1/2}\}_{k\geq 0}$ by successively adding the non-redundant constraints in $\mathbf{P}_{k+1/2} \in \{\mathbf{W}_{k+1/2}, \mathbf{L}_{k+1/2}, \mathbf{D}_{k+1/2}\}$ to the minimal representation of $\bigcap_{i=0}^{k-1} \mathbf{P}_{i+1/2}$. The minimal representation is given by Algorithm 4.

Algorithm 4 Minimum representation of $\mathrm{Dom}(\boldsymbol{m})$: $\mathbf{minRep}(\boldsymbol{m})$

Require: accepted mode vector $\boldsymbol{m}$.
1: $\mathcal{H} = \{\}$
2: **if** $m_1 \in \{1,2\}$ **then** $\mathcal{H} = \mathcal{H} \cup \{\mathbf{H}_{1/2}, \mathbf{H}_1\}$
3: **if** $m_1 \in \{3,4\}$ **then** $\mathcal{H} = \mathcal{H} \cup \{\mathbf{H}_0, \mathbf{H}_1^c\}$
4: **if** $m_1 \in \{5,6,7\}$ **then** $\mathcal{H} = \mathcal{H} \cup \{\mathbf{H}_0^c, \mathbf{H}_{1/2}^c\}$
5: **for** $k \in \{1, \ldots, n\}$ **do**
6: **if** $m_k = 1$ **then** $\mathcal{H} = \mathcal{H} \cup \{\mathbf{H}_{k+1}\}$
7: **if** $m_k = 2$ **then** $\mathcal{H} = \mathcal{H} \cup \{\mathbf{H}_{k+1}^c\}$
8: **if** $m_k = 3$ **then** $\mathcal{H} = \mathcal{H} \cup \{\mathbf{H}_{k+1/2}\}$
9: **if** $m_k = 4$ **then** $\mathcal{H} = \mathcal{H} \cup \{\mathbf{H}_{k+1/2}^c\}$
10: **if** $m_k = 5$ **then** $\mathcal{H} = \mathcal{H} \cup \{\mathbf{H}_{k+1/2}, \mathbf{H}_{k+1}\}$
11: **if** $m_k = 6$ **then** $\mathcal{H} = \mathcal{H} \setminus \{\mathbf{H}_{k-1}^c\} \cup \{\mathbf{H}_k, \mathbf{H}_{k+1}^c\}$
12: **if** $m_k = 7$ **then** $\mathcal{H} = \mathcal{H} \setminus \{\mathbf{H}_{k-1/2}^c\} \cup \{\mathbf{H}_{k+1/2}^c, \mathbf{H}_k^c\}$
13: **end for**
14: **return** the minimal representation $\mathcal{H}$

Proposition 5: For every accepted mode vector $\boldsymbol{m}$, Algorithm 4 returns the minimal representation of the closure of $\mathrm{Dom}(\boldsymbol{m})$. (Proof given in the Appendix)

C. Adjacent Polyhedra

Definition 5 (Adjacent Polyhedra): Two polyhedra$\mathbf{P}$ and $\mathbf{P}'$ in a polyhedral partition of the space are said to be k-*adjacent* if they have a face of dimension k in common, i.e., there exists a supportive hyperplane $\partial\mathbf{H}$ for both $\mathbf{P}$ and $\mathbf{P}'$ and the intersection $\mathbf{P} \cap \mathbf{P}' \cap \partial\mathbf{H}$ is of dimension k. Then $\mathbf{P}$ and $\mathbf{P}'$ are said to be $\partial\mathbf{H}$-adjacent.

For an accepted mode vector $\boldsymbol{m}$ and its associated polyhedron $\mathrm{Dom}(\boldsymbol{m})$, it is of interest to find the polyhedra of the partition adjacent to it. Algorithm 5 returns all the polyhedra of the partition $(\mathrm{n}+1)$-adjacent to $\mathrm{Dom}(\boldsymbol{m})$. First, the mode string $s(0)\cdots s(n)$ and the minimal representation of $\mathrm{Dom}(\boldsymbol{m})$ are computed with Algorithms 2 and 4. Then for all $\mathbf{H} \in \mathcal{H}$, the algorithm computes the mode string of the polyhedron of the partition $\partial\mathbf{H}$-adjacent to $\mathrm{Dom}(\boldsymbol{m})$, and finds the associated mode vector $\boldsymbol{m}_{\mathbf{H}}$ with Algorithm 3 (see Fig. 6 for an illustration of the Algorithm).

Algorithm 5 Find all the polyhedra adjacent to $\mathrm{Dom}(\boldsymbol{m})$: $\mathbf{adj}(\boldsymbol{m})$

Require: accepted mode vector $\boldsymbol{m}$
1: $s(0)\cdots s(n) = \mathbf{m2s}(\boldsymbol{m})$
2: $\mathcal{H} = \mathbf{minRep}(\boldsymbol{m})$
3: **for** $\mathbf{H} \in \mathcal{H}$ **do**
4: $s'(0)\cdots s'(n) = s(0)\cdots s(n)$
5: **for**$i \in \{0, \ldots, n\}$ **do**
6: **if** $\mathbf{H} = \mathbf{H}_i$ **then** $s'(i) = D$
7: **if** $\mathbf{H} = \mathbf{H}_i^c$ **then** $s'(i) = W$
8: **if** $\mathbf{H} = \mathbf{H}_{i+1}$ **then** $s'(i) = L$
9: **if** $\mathbf{H} = \mathbf{H}_{i+1}^c$ **then** $s'(i) = W$
10: **if** $\mathbf{H} = \mathbf{H}_{i+1/2}$ **then** $s'(i) = D$
11: **if** $\mathbf{H} = \mathbf{H}_{i+1/2}^c$ **then** $s'(i) = L$
12: **end for**
13: $\boldsymbol{m}_{\mathbf{H}} = \mathbf{s2m}(s'(0)\cdots s'(n))$
14: **end for**
15: **return** adjacent polyhedra $\{\boldsymbol{m}_{\mathbf{H}}\}_{\mathbf{H}\in\mathcal{H}}$

Definition 6: Two accepted mode vectors $\boldsymbol{m}$ and $\boldsymbol{m}'$ are adjacent if the closures of their respective domain $\mathrm{Dom}(\boldsymbol{m})$ and $\mathrm{Dom}(\boldsymbol{m}')$ are $(\mathrm{n}+1)$-adjacent.

Proposition 6: For every accepted mode vector $\boldsymbol{m}$, Algorithm 5 returns all the accepted mode vectors adjacent to $\boldsymbol{m}$. (Formal proof given in the Appendix.)

Since Algorithm 4 adds at most 2 constraints per iteration, $\mathrm{minRep}(\boldsymbol{m})$ has at most $2(n+1)$ constraints; hence, at most $2(n+1)$ accepted mode vectors are adjacent to $\boldsymbol{m}$.

IV. Hybrid Estimation Algorithms

A. Kalman Filtering Algorithm for Each Mode Vector

In discrete time and space, the dynamics of the traffic flow along a homogeneous section of highway is well described by the Godunov scheme applied to the LWR equation with triangular flux function (see Prop. 1). The small uncertainties on the parameters of the model $A_{\boldsymbol{m}}$ and $b_{\boldsymbol{m}}$ can be reasonably covered

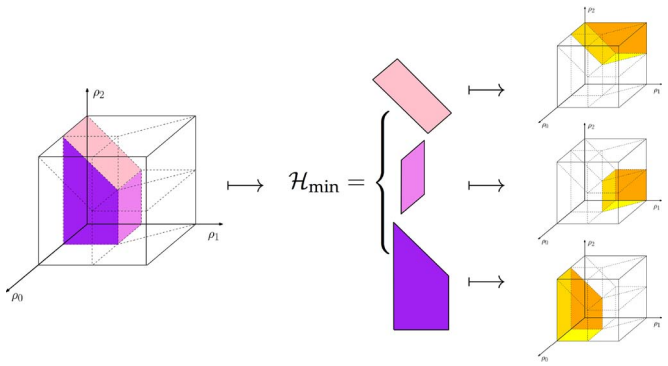

Fig. 6. We want all the polyhedra of the partition adjacent to a fixed polyhedron. First, we find all the K facet-defining hyperplanes of purple, i.e., minimal representation. Then for each facet, we find the only polyhedron that shares this facet with purple. Hence, K is also the number of polyhedra of the partition adjacent to purple.

by a zero-mean Gaussian noise $\boldsymbol{\eta}^t \sim \mathcal{N}(0, Q^t)$ with covariance Q^t. The discrete dynamics in mode vector $\boldsymbol{m}$ become

$$\boldsymbol{\rho}^{t+1} = A_{\mathbf{m}}\boldsymbol{\rho}^t + b_{\mathbf{m}} + c^t + \boldsymbol{\eta}^t. \tag{19}$$

The mode vector $\boldsymbol{m}$ is no longer fixed by $\boldsymbol{\rho}^t$, but a probability distribution over all accepted mode vectors is maintained to take into account the uncertainty in mode estimation; that is, at each time step t, the model is in several different mode vectors with positive probabilities. We add an *observation model*

$$\boldsymbol{z}^t = H^t\boldsymbol{\rho}^t + \boldsymbol{\chi}^t \tag{20}$$

where $\boldsymbol{\chi}^t \sim \mathcal{N}(0, R^t)$ is the zero-mean observation noise with covariance matrix R^t, and H^t is the $d_t \times (n+2)$-dimensional linear observation matrix which encodes the d_t observations (each one of them being at a discrete cell on the discretization domain) for which the density is observed during discrete time step t, and n is the dimensionality of the system. In the traffic case, sensing devices (such as loop detectors) are placed at several locations along a section of highway, and their positions are encoded in the matrix H^t. For example, in the discrete case for $n = 3$, if one sensor is in cell 1 and another in cell 3, then both sensors provide observations $z_1^t = \rho_1^t + \chi_1^t$ and $z_2^t = \rho_3^t + \chi_2^t$, which is in matrix form

$$\begin{pmatrix} z_1^t \\ z_2^t \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \boldsymbol{\rho}^t + \begin{pmatrix} \chi_1^t \\ \chi_2^t \end{pmatrix} \tag{21}$$

where the state is $\boldsymbol{\rho}^t = (\rho_0^t, \rho_1^t, \ldots, \rho_5^t)^T$. In this small example, the observation matrix is $H^t = (0)\ 1000\ 00010$ and the number of observations is $d_t = 2$.

In the rest of the section, we use the standard notations $\boldsymbol{m}_j$ for the different mode vectors, and subscript j denotes quantities that are pertaining to mode $\boldsymbol{m}_j$. Note that $\boldsymbol{m}_j$ refers to the whole *mode vector* $\boldsymbol{m}$ and not the entries of $\boldsymbol{m}$.

Let $\hat{\boldsymbol{\rho}}^{t:t}$ and $P^{t:t}$ be the *a posteriori* state estimate and error covariance matrix at time t. The *predicted* state estimate $\hat{\boldsymbol{\rho}}_j^{t+1:t}$ and covariance estimate $P_j^{t+1:t}$ of the *prediction step* in *mode* $\boldsymbol{m}_j$ are

$$\begin{aligned} \textbf{Prediction}: \quad & \hat{\boldsymbol{\rho}}_j^{t:t+1} = A_j\hat{\boldsymbol{\rho}}^{t:t} + b_j + c^t \\ & P_j^{t:t+1} = A_j P^{t:t}(A_j)^T + Q^t. \end{aligned} \tag{22}$$

The *measurement residual* $\mathbf{r}_j^{t+1}$, *residual covariance* S_j^{t+1}, *Kalman gain* K_j^{t+1}, *updated state estimate* $\hat{\boldsymbol{\rho}}_j^{t+1:t+1}$, and *updated estimate covariance* $P_j^{t+1:t+1}$ of the *update step* in *mode* j are

$$\begin{aligned} \textbf{Residuals}: \quad & \mathbf{r}_j^{t+1} = \boldsymbol{z}^{t+1} - H^{t+1}\hat{\boldsymbol{\rho}}_j^{t:t+1} \\ & S_j^{t+1} = H^{t+1}P_j^{t:t+1}(H^{t+1})^T + R^{t+1} \\ \textbf{Kalman gain}: \quad & K_j^{t+1} = P_j^{t:t+1}(H^{t+1})^T(S_j^{t+1})^{-1} \\ \textbf{Updates}: \quad & \hat{\boldsymbol{\rho}}_j^{t+1:t+1} = \hat{\boldsymbol{\rho}}_j^{t:t+1} + K_j^{t+1}\mathbf{r}_j^{t+1} \\ & P_j^{t+1:t+1} = (I - K_j^{t+1}H^{t+1})P_j^{t+1:t}. \end{aligned} \tag{23}$$

In [19], a measure of the likelihood of the Kalman filter in mode j is given by the *mode likelihood function* Λ_j^{t+1}, where $\mathcal{N}(x; a, b)$ is the probability density function of the normal distribution with mean a and variance b

$$\Lambda_j^{t+1} = \mathcal{N}\left(\mathbf{r}_j^{t+1}; 0, S_j^{t+1}\right). \tag{24}$$

The noise might result in densities outside bounds. We project onto $[0, \rho_{\text{jam}}]^{n+2}$, i.e., the equation is implicitly $\hat{\boldsymbol{\rho}}_j^{t+1:t+1} = \Pi(\hat{\boldsymbol{\rho}}_j^{t:t+1} + K_j^{t+1}\boldsymbol{r}_j^{t+1})$ where $\Pi(\cdot)$ is the projection operator. This is a legitimate because densities cannot be negative nor exceed a maximum value ρ_{jam}.

B. Interactive Multiple Model KF

Let us denote by $\{m(t) = \boldsymbol{m}_j\}$ the event that the system is in the mode $\boldsymbol{m}_j$ at time t. We then assume that the model is a discrete-time stochastic linear hybrid system in which the mode evolution is governed by the finite state Markov chain

$$\mu^{t+1} = \Pi\mu^t \tag{25}$$

where $\pi_{ij} = P(m(t+1) = \boldsymbol{m}_j | m(t) = \boldsymbol{m}_i)$ for all $\boldsymbol{m}_i, \boldsymbol{m}_j \in \mathcal{M}$ is the mode transition matrix, $\mu_j^t = P(m(t) = \boldsymbol{m}_j)$ for all $m_j \in \mathcal{M}$ is the mode probability at time t; and the set of accepted modes is $\mathcal{M}$.

Effective estimation techniques for stochastic hybrid systems are based in multiple models since it is natural to apply a statistical filter for each of the modes. The *Interactive Multiple Model* (IMM) algorithm [1], [4], [20] is a cost-effective (in terms of performance versus complexity) estimation scheme in which there is a *mixing/interacting* step at the beginning of the estimation process, which computes new initial conditions for the Kalman filters matched to the individual modes at each time step as illustrated in Fig. 7.

We consider the IMM algorithm in which $\mathcal{M}^t$ is the set of modes for which the Kalman filter is applied at time step t. The set $\mathcal{M}^t$ is the set of modes $\boldsymbol{m}_j$ with positive mode probabilities $\mathcal{M}^t = \{\boldsymbol{m}_j | \mu_j^t > 0\}$. In the standard IMM, a filter is applied to every mode. The components of the *mixing* step are the *mixing probability* $\mu_{ij}^{t|t+1}$ of being in mode i at time t given that the mode at time $t+1$ is j, the *mixed condition* $\hat{\boldsymbol{\rho}}_{0j}^{t:t}$ and $P_{0j}^{t:t}$ for the state estimate and covariance of mode j at time t, and the "spread-of-the-means" X_j in the expression of $P_{0j}^{t:t}$. They are

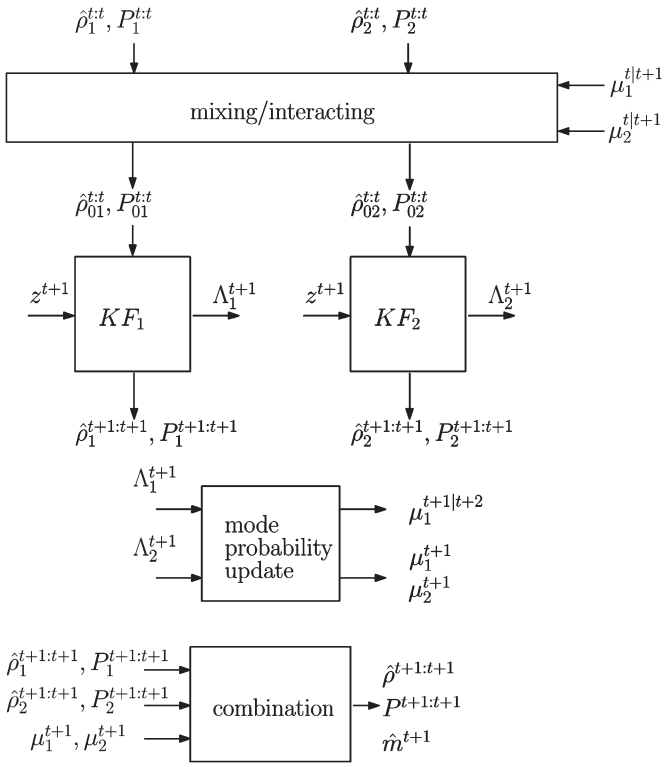

Fig. 7. Illustration of the structure of IMM algorithm for a two-mode system from [27].

computed for $j \in \mathcal{M}^{t+1}$ w.r.t. $\hat{\boldsymbol{\rho}}_i^{t:t}$ and $P_i^{t:t}$, the state estimate and its covariance of Kalman filter i at time t:

$$\begin{aligned}
\mu_{ij}^{t|t+1} &= \frac{1}{Z_j}\pi_{ij}\mu_i^t \quad \text{for } i \in \mathcal{M}^t \quad \text{with} \quad Z_j = \sum_{i\in\mathcal{M}^t} \pi_{ij}\mu_i^t \\
\hat{\boldsymbol{\rho}}_{0_j}^{t:t} &= \sum_{i\in\mathcal{M}^t} \hat{\boldsymbol{\rho}}_i^{t:t}\mu_{ij}^{t|t+1} \\
P_{0_j}^{t:t} &= \sum_{i\in\mathcal{M}^t} P_i^{t:t}\mu_{ij}^{t|t+1} + X_j \\
X_j &:= \sum_{i\in\mathcal{M}^t} \left(\hat{\boldsymbol{\rho}}_i^{t:t} - \hat{\boldsymbol{\rho}}_{0_j}^{t:t}\right)\left(\hat{\boldsymbol{\rho}}_i^{t:t} - \hat{\boldsymbol{\rho}}_{0_j}^{t:t}\right)^T \mu_{ij}^{t|t+1}. \qquad (26)
\end{aligned}$$

We apply the Kalman filter in each mode $j \in \mathcal{M}^{t+1}$ (KF_j) as described with ((22), (23)) and the resulting mode likelihood functions Λ_j^{t+1} are obtained from $\hat{\boldsymbol{\rho}}_j^{t+1:t+1}$ and $P_j^{t+1:t+1}$ with (24). The mode probability $\mu^t = \{\mu_j^t\}$ is then updated through

$$\mu_j^{t+1} = \frac{1}{Z}\Lambda_j^{t+1}\sum_{i\in\mathcal{M}^t}\pi_{ij}\mu_i^t \quad \text{for } j \in \mathcal{M}^{t+1} \qquad (27)$$

where Z is a normalization constant and Λ_j^{t+1} is the mode likelihood function defined in (24). The output of the IMM algorithm are the state estimate $\hat{\boldsymbol{\rho}}^{t+1:t+1}$ which is a weighted sum of the estimates from the Kalman filters in each mode and its covariance $P^{t+1:t+1}$, and the mode estimate $\hat{\boldsymbol{m}}^{t+1}$ is the mode which has the highest mode probability. They are given by the *combination* step

$$\begin{aligned}
\hat{\boldsymbol{\rho}}^{t+1:t+1} &= \sum_{j\in\mathcal{M}^{t+1}} \hat{\boldsymbol{\rho}}_j^{t+1:t+1}\mu_j^{t+1} \\
P^{t+1:t+1} &= \sum_{j\in\mathcal{M}^{t+1}} P_j^{t+1:t+1}\mu_j^{t+1} + X \\
X &:= \sum_{j\in\mathcal{M}^{t+1}} \left(\hat{\boldsymbol{\rho}}_j^{t+1:t+1} - \hat{\boldsymbol{\rho}}^{t+1:t+1}\right) \\
&\quad \times \left(\hat{\boldsymbol{\rho}}_j^{t+1:t+1} - \hat{\boldsymbol{\rho}}^{t+1:t+1}\right)^T \mu_j^{t+1} \\
\hat{\mathbf{m}}^{t+1} &:= \arg\max_{j\in\mathcal{M}^{t+1}} \mu_j^{t+1}. \qquad (28)
\end{aligned}$$

In [13], [19], the IMM algorithm is used as a hybrid estimator for air traffic control (ATC) tracking. The models used include one for the uniform motion and one (or more) for the maneuver. However, the discretized PDE model described in Section II has an exponential number of modes, which induces an exponential time complexity of the IMM.

C. Extended Kalman filter

In the simplest case, we assume that the only possible mode at the next time is the mode $\boldsymbol{m}_j$ of the estimate, i.e., $\mathcal{M}^{t+1} = \{\boldsymbol{m}_j\}$ and $\mu_j^{t+1} = 1$ with $\hat{\boldsymbol{\rho}}^{t:t} \in \mathrm{Dom}(\boldsymbol{m}_j)$. We apply the Kalman filter only to this mode. With $\mathcal{M}^t = \{\boldsymbol{m}_i\}$, (26) become

$$\hat{\boldsymbol{\rho}}_{0_j}^{t:t} = \hat{\boldsymbol{\rho}}_i^{t:t}, \quad P_{0_j}^{t:t} = P_i. \qquad (29)$$

We apply the Kalman filter only to mode $\boldsymbol{m}_j$ to obtain $\hat{\boldsymbol{\rho}}_j^{t+1:t+1}$ and $P_j^{t+1:t+1}$. Finally, the outputs of the *combination step* given by (28) are simply $\hat{\boldsymbol{\rho}}^{t+1:t+1} = \hat{\boldsymbol{\rho}}_j^{t+1:t+1}$, $P^{t+1:t+1} = P_j^{t+1:t+1}$, and $\hat{\boldsymbol{m}}^{t+1} = \boldsymbol{m}_j$.

In this model, the IMM algorithm is exactly an *Extended Kalman filter* (EKF) applied to our discretized system presented in Proposition 1. The linear model in mode $\boldsymbol{m}$ such that $\hat{\boldsymbol{\rho}}^{t:t} \in \mathrm{Dom}(\boldsymbol{m})$ coincides exactly with the linearization of the discrete dynamics around $\hat{\boldsymbol{\rho}}^{t:t}$.

Despite the exponential number of modes, we can compute the *predicted state estimate* $\hat{\boldsymbol{\rho}}_j^{t+1:t}$ and the *predicted covariance estimate* $P_j^{t+1:t}$ in mode $\boldsymbol{m}_j$ [see (22)] in linear time and quadratic time, respectively, without generating any dense matrix because A_j is completely defined by mode vector $\boldsymbol{m}_j$ and A_j is tridiagonal (see Algorithm 7). Hence, the time complexity of the prediction step is $O(n^2)$, with constant space complexity. With d the number of observations (or number of sensors), the time complexity of the *update step* of the Kalman filter given by (23) is $O(dn^2 + d^3 + nd^2)$, and so as the two steps combined of the KF.

In comparison, the *Ensemble Kalman Filter* (EnKF) is a popular estimation algorithm for nonlinear dynamical systems. It is commonly used in the traffic monitoring community [30]. The EnKF is based on a *Monte Carlo approximation* of the Kalman filter which approximates the covariance matrix of the state vector with the sample covariance of the ensemble. The prediction step consists in applying the system's dynamics to each sample, which has complexity $O(Nn^2)$, where N is the number of samples (ensemble members). Mandel's report [22] shows that the computational complexity of the update step of the EnKF algorithm is $O(d^3+d^2N+dN^2+nN^2)$. So the total complexity of the EnKF is $O(d^3+d^2N+dN^2+nN^2+Nn^2)$.

Algorithm 6 describes the EKF. The parameters $\mathbf{L}(1), \ldots, \mathbf{L}(7) \in \mathbb{R}^3$, $w(1), \ldots, w(7) \in \mathbb{R}$, given in Table I describe the linear modes of our hybrid system.

Algorithm 6 (Explicit) Extended Kalman filter

Require: initial state $\boldsymbol{\rho}_0 \in [0, \rho_{\text{jam}}]^{n+2}$, boundary conditions $(u(t), d(t))_{t\geq 0}$, state covariance $\{Q^t\}_{t\geq 0}$, observations $\{\boldsymbol{z}^t\}_{t\geq 0}$, observation matrix $\{H^t\}_{t\geq 0}$, observation covariance $\{R^t\}_{t\geq 0}$.

1: **for** $t \in \{0, 1, 2, \cdots\}$ **do**
2: $\quad \boldsymbol{m} = \mathbf{rho2m}(\hat{\boldsymbol{\rho}}^t)$ {mode estimate, see algo. 1}
3: $\quad (\hat{\boldsymbol{\rho}}^{t+1}, P^{t+1}) = \mathbf{KF}(\boldsymbol{m}, \hat{\boldsymbol{\rho}}^t, P^t, \cdots)$ {KF, see algo. 8}
4: **end for**
5: **return** $(\hat{\boldsymbol{\rho}}^t, P^t)_{t\geq 0}$

Algorithm 7 Prediction step of Kalman filter in mode $\boldsymbol{m}$: $\mathbf{predict}(\boldsymbol{m}, \boldsymbol{\rho}, P, u^+, d^+, Q)$

Require: mode vector $\boldsymbol{m} = [m_1, \ldots, m_n] \in \{1, \ldots, 7\}^n$, current state $\boldsymbol{\rho} = [\rho_0, \ldots, \rho_{n+1}] \in [0, \rho_{\text{jam}}]^{n+2}$, current state estimate covariance P, next boundary conditions $u^+, d^+ \in \mathbb{R}$, current state noise covariance Q.

1: $\rho_0^+ = u^+$
2: $\rho_{n+1}^+ = d^+$
3: **for** $i \in \{1, \ldots, n\}$ **do**
4: $\quad \rho_i^+ = \mathbf{L}(m_i) \times [\rho_{i-1}, \rho_i, \rho_{i+1}]^T + w(m_i)$
5: **end for**
6: $M := zeros(n+2, n+2)$ {create temporary matrix M}
7: **for** $(i, j) \in \{1, \ldots, n\}^2$ **do**
8: $\quad M_{ij} = \mathbf{L}(m_i) \times [P_{i-1,j}, P_{i,j}, P_{i+1,j}]^T$ {do $A \times P$}
9: **end for**
10: **for** $(i, j) \in \{1, \ldots, n\}^2$ **do**
11: $P_{ij}^+ = [M_{i,j-1}, M_{i,j}, M_{i,j+1}] \times \mathbf{L}(m_j)^T$ {do $(AP)A^T$}
12: **end for**
13: $P^+ = P^+ + Q$ {predict state covariance}
14: **return** $\boldsymbol{\rho}^+, P^+$

Algorithm 8 Kalman filter in mode $\boldsymbol{m}$:$\mathbf{KF}(\boldsymbol{m}, \hat{\boldsymbol{\rho}}^t, P^t, u(t+1), d(t+1), Q^t, \boldsymbol{z}^{t+1}, H^{t+1}, R^{t+1})$

Require: mode vector $\boldsymbol{m}$, current state $\hat{\boldsymbol{\rho}}^t$, current state estimate covariance P^t, next boundary conditions $u(t+1), d(t+1)$, current state noise covariance Q^t, next measurement $\boldsymbol{z}^{t+1}$, next observation matrix H^{t+1}, next observation covariance R^{t+1}.

1: $(\hat{\boldsymbol{\rho}}^{t:t+1}, P^{t:t+1}) = \mathbf{predict}(\hat{\boldsymbol{\rho}}^t, P^t, \{\cdots\})$ {see algo. 7}
2: $(\hat{\boldsymbol{\rho}}^{t+1}, P^{t+1}, \Lambda^{t+1}) = \mathbf{update}(\hat{\boldsymbol{\rho}}^{t:t+1}, \{\cdots\})$ {see (23)}
3: **return** $\hat{\boldsymbol{\rho}}^{t+1}, P^{t+1}, \Lambda^{t+1}$

D. Extended Kalman filter: Numerical Results

In traffic estimation, the density measurements along the highway are usually sparse. For example, in the 18-mile stretch of I-880 Northbound in the Bay Area, CA [see Fig. 9(a)], the Mobile Millennium traffic monitoring system receives measurements from 29 loop detectors (PeMS) every 30 s on March 5th, 2012 between 7am and 8am. This section of highway is discretized into cells of length 198 m, hence $n = 148$ and $m = 29$, and the EnKF with 100 ensembles is currently used for traffic estimation, so $N = 100$ and $m \leq \min(n, N)$. Hence, the time complexities of the KF (or EKF) and EnKF are $O(mn^2)$ and $O(n^2N + nN^2)$, respectively. With N large (> 50), the complexity analysis predicts that the EKF should be faster than the EnKF.

The running times of the implementation of both the EKF and the EnKF estimators on an Intel Core i5 480M 2.67 GHz are shown in Fig. 8(a), for increasing portions of the I-880 starting from East Industrial in Fremont, CA. For example, 60 cells ($\sim$7.5 miles) span from East Industrial to Dumbarton Bridge, and 113 cells ($\sim$14 miles) reaches San Mateo Bridge. The EKF is significantly faster than the EnKF with 100 samples, which is implemented in the Mobile Millennium. This confirms our complexity analysis of both algorithms.

Fig. 9(c), (d) shows the contour plots of the output of the EnKF and the EKF estimators, which consists in the density in the time-space domain. The regions with high densities are represented in red and the regions with low densities in blue. Both estimators give very similar higher resolution scalar fields of the density (1440 time steps by 141 cells) by assimilating sparse density measurements (240 time steps by 29 PeMS stations, see Fig. 9(b)). Moreover, by removing measurements at an arbitrary cell, Fig. 8(b) shows that the estimation algorithm performs well since the density estimate is close to the actual measurement.

In summary, the explicit representation as a switched hybrid system gives a powerful framework for tracking the mode evolution and preforming hybrid estimation. For instance, the EKF can be implemented easily by applying the KF in the mode vector of the state estimate. However, straight application of the IMM algorithm [19] is not tractable because the complexity is $O(\tau_n(2.247)^n)$ where τ_n is the complexity of the KF and $(2.247)^n$ is the asymptotic number of modes.

V. Reduced IMM

A. Reduction to Adjacent Modes

We presented an algorithm to construct the minimal representation of $\text{Dom}(\boldsymbol{m})$, which enables to find the adjacent modes. Moreover, two adjacent modes only differ by at most two entries. Hence, when the discretized model is in quasi-steady state, and n is relatively small, only one cell switches mode at the next time step, so the state is most likely to jump to an adjacent mode vector. This suggests to consider only the mode of the state estimate and its adjacent modes. Hence, the number of modes considered is less than $2(n+1)$.

We can further reduce the number of modes by taking into account the state covariance P and the distance between the state estimate and the facets of the polyhedron. Let $\mathcal{H}$ be the minimal representation of the mode vector $\hat{\boldsymbol{m}}$ of the state estimate (i.e., $\hat{\boldsymbol{\rho}} \in \text{Dom}(\hat{\boldsymbol{m}})$), and let $\mathbf{H} \in \mathcal{H}$ with equation $\mathbf{H} = \{\boldsymbol{\rho} | \mathbf{a} \cdot \boldsymbol{\rho} - b \leq 0\}$ and $\|a\|_2 = 1$. Then the distance from the supportive hyperplane $\partial\mathbf{H}$ is: $d(\hat{\boldsymbol{\rho}}, \partial\mathbf{H}) = \min \|\hat{\boldsymbol{\rho}} - \partial\mathbf{H}\|_2 = |b - \mathbf{a} \cdot \hat{\boldsymbol{\rho}}|$.

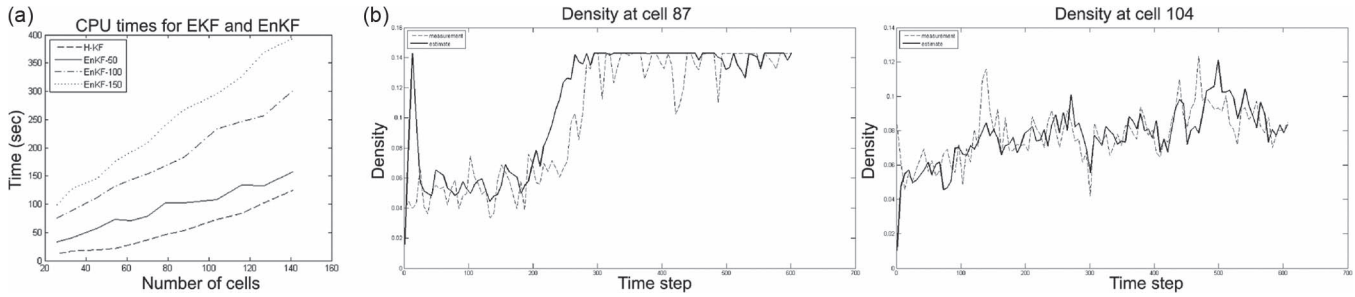

Fig. 8. (a) Computational time for an increasing section of the I-880 (measured in the number of cells) for the EKF (dashed line), the EnKF with 50 ensembles (continuous line), the EnKF with 100 ensembles (dashed-dotted line), and the EnKF with 150 ensembles (dotted line). (b) Comparison between the density measurements (dashed line) and estimates (bold line) at cell 87 and cell 104.

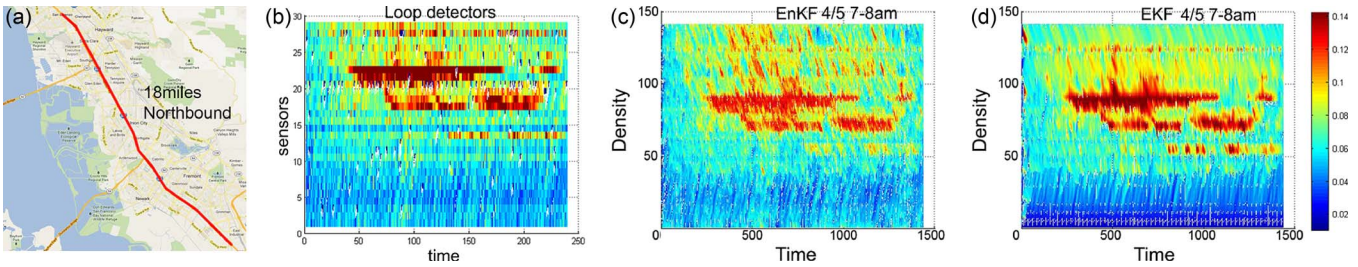

Fig. 9. (a) Experimental data location: 18-mile long stretch of I-880 in the Bay Area on the *Mobile Century* site. (b) Contour plot of the density from the 29 PeMS stations every 30 s on March 5th, 7–8am. Each vertical line in the contour plot reports the measurements from the 29 sensors along the highway at a specific time. (c) Output of the EnKF d) Output of the EKF. The time step is on the x-axis and the number of cells is on the y-axis. Each vertical line of the diagram is a snapshot of the state estimate of the highway at a specific time.

The probability distribution of the state along the normal $\mathbf{a}$ to $\partial\mathbf{H}$ is $Ke^{-((\mathbf{a}\cdot(\boldsymbol{\rho}-\hat{\boldsymbol{\rho}}))^2/2\mathbf{a}^T P\mathbf{a})}$, so the probability that the state is inside of half-space $\mathbf{H}$ along the normal $\mathbf{a}$ is

$$K\int_{-\infty}^{|b-\mathbf{a}\cdot\hat{\boldsymbol{\rho}}|} e^{-\frac{t^2}{2\mathbf{a}^T P\mathbf{a}}}dt = \frac{1}{2}\left(1+\operatorname{erf}\left(\frac{|b-\mathbf{a}\cdot\hat{\boldsymbol{\rho}}|}{\sqrt{2\mathbf{a}^T P\mathbf{a}}}\right)\right)$$

where erf is the *error function*. Since erf is an increasing function, we keep only the $\partial\mathbf{H}$-adjacent modes for which the following quantity is small (see Algorithm 9)

$$r(\hat{\boldsymbol{\rho}},\mathbf{H}) = \frac{|b-\mathbf{a}\cdot\hat{\boldsymbol{\rho}}|}{\sqrt{2\mathbf{a}^T P\mathbf{a}}}, \quad \mathbf{H}\in\mathcal{H}. \tag{30}$$

Algorithm 9 Find all adjacent polyhedra close to $\hat{\boldsymbol{\rho}}$: $\mathbf{adj2}(\boldsymbol{m},\hat{\boldsymbol{\rho}},P,\beta)$

Require: mode estimate $\hat{\boldsymbol{m}}$, state estimate $\hat{\boldsymbol{\rho}}$, state estimate covariance P, tolerance β

1: $s(0)\cdots s(n) = \mathbf{m2s}(\hat{\boldsymbol{m}})$
2: $\mathcal{H} = \mathbf{minRep}(\hat{\boldsymbol{m}})$
3: **for** $\mathbf{H}\in\mathcal{H}$ **do**
4: **if** $\mathbf{H}=\mathbf{H}_i$ **then** $r = |\rho_i-\rho_c|/\sqrt{2P_{ii}}$
5: **if** $\mathbf{H}=\mathbf{H}_{i+1/2}$ **then** $r = |\rho_{i+1}+(v_f/w_f)\rho_i - \rho_{\text{jam}}|/\sqrt{2(v_f/w_f)^2P_{ii}+4(v_f/w_f)P_{i,i+1}+2P_{i+1,i+1}}$
6: **if** $r>\beta$ **then** remove $\mathbf{H}$ from $\mathcal{H}$
7: **end for**
8: execute lines 3 to 14 of Algorithm 5
9: **return** adjacent polyhedra close to state estimate $\{\boldsymbol{m}_{\mathbf{H}}\}_{\mathbf{H}\in\mathcal{H}}$

This is a refinement of the EKF. Instead of relying on one possible mode, we consider a set of possible adjacent modes at time t and apply the KF to each one of them. However, the adjacent modes differ by only one or two entries, so they only represent a restricted set of close possibilities centered around the mode estimate. Hence, the reduced IMM based on adjacent modes is still very similar to the EKF.

B. Representative Mode Vectors With Clustering Algorithm

An intuitive method consists in using a clustering algorithm to reduce the space of modes to a representative set $\mathcal{M}^K$. Historical data of traffic density estimate on March 1st, 2012 [see Fig. 10(i)] provides $T=9355$ observations or samples of the state vector, where T is the number of time steps in the observed data. We partition these T samples into K clusters using the popular *k-means algorithm*. The centroid of each cluster, which may not necessarily be a member of the data set, are density vectors that represent particular states of the highway which are representative of its evolution. They are shown in Fig. 10(a), (d). We have the index of the cluster on the x-axis and the position along the highway on the y-axis. For instance, the first cluster represents a density vector of the highway mostly in free flow whereas the last cluster represents the density vector of the highway mostly in congestion in the top part.

Then, we derive the modes of these K centroids, and we assume that our system can only be in these K modes. They are illustrated in Fig. 10(e). We have the index of the modes on the x-axis, and the position on the highway along the y-axis. Each column represents a modal regime of the highway. For example, in the first mode vector (in the first column),

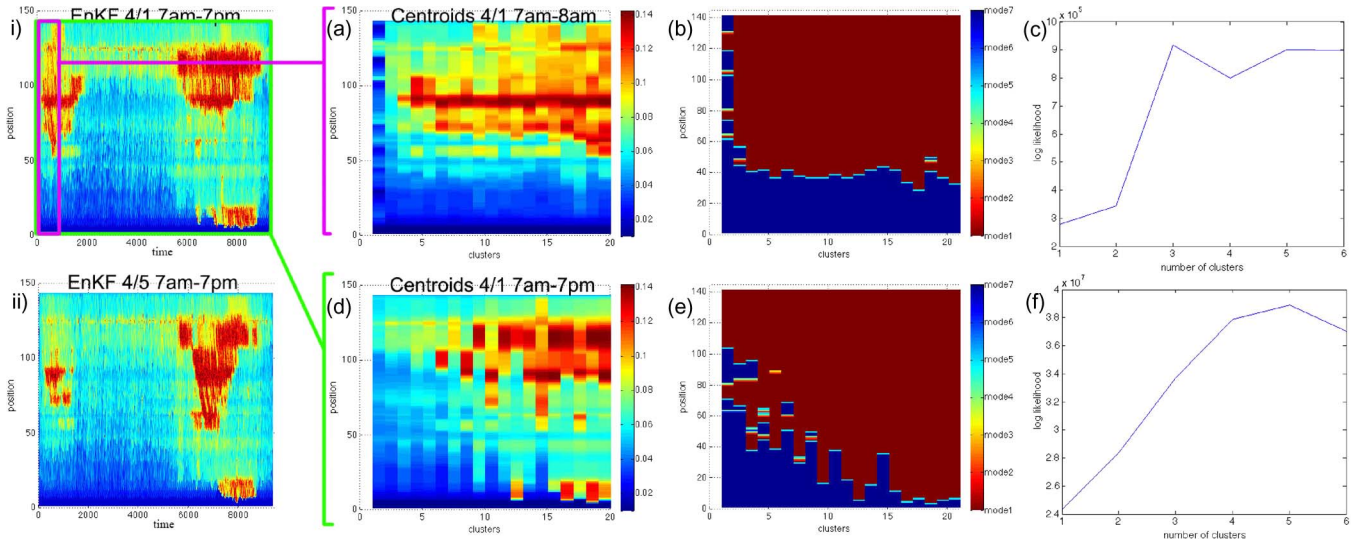

Fig. 10. (i) Traffic density estimate on March 1st, 2012 from 7am to 7pm. (ii) Traffic density estimate on March 5th, 2012 from 7am to 7pm. a,b,c) 20 clusters of the density space using *k-means* on March 1st, 2012 from 7 to 8am, the corresponding modes, and the log likelihood. d,e,f) 20 clusters of the density space using *k-means* on March 1st, 2012 from 7am to 7pm.

the cells are in mode 7 in the upstream part, and the cells in the downstream are in mode 1. When $\rho_{i-1}, \rho_i, \rho_{i+1} > \rho_c$ for a particular cell i, the Godunov flux (7) is in congestion regime at both interfaces $i-1|i$ and $i|i+1$ and cell i is in mode 1 (see Table I). Conversely, $m_i = 7$ when the Godunov flux is in free flow regime at both interfaces: $\rho_{i-1}, \rho_i, \rho_{i+1} < \rho_c$. Hence, the regions in which $m_i = 1$ (resp. $m_i = 7$), colored in red (resp. blue), represent cells that are in congestion (resp. free flow) regime. The cells are in the other modes $\{2, 3, \ldots, 6\}$ correspond to a transition regime between free flow and congestion. We apply the IMM with this reduced set of modes to estimate the traffic on March 5th. This is a valid approach since the traffic conditions are similar during weekdays [see Fig. 10(i), (ii)].

Algorithm 10 Clustering historical data: $\mathbf{cluster}(\{\boldsymbol{\rho}^t\}_{t\in\{1,\ldots,T\}})$

Require: observed data set $\{\boldsymbol{\rho}^t\}_{t\in\{1,\ldots,T\}}$

1: partition $\{\boldsymbol{\rho}^t\}_{t\in\{1,\ldots,T\}}$ into K clusters and get centroids $\{\bar{\boldsymbol{\rho}}^k\}_{k\in\{1,\ldots,K\}}$

2: **for** $k \in \{1, \ldots, K\}$ **do** $\bar{\boldsymbol{m}}^k = \mathbf{rho2m}(\bar{\boldsymbol{\rho}}^k)$; **end for**

3: **return** set of K representative modes $\mathcal{M}^K = \{\bar{\boldsymbol{m}}^k\}_{k\in\{1,\ldots,K\}}$

To determine the optimal number of clusters, we have applied the above procedure to one hour of observed data, on March 1st from 7am to 8am. The density centroids and their mode are shown in Fig. 10(a), (b). Then we applied the IMM algorithm on March 5th from 7am to 8am and compared it against the state estimate given by the EnKF for different numbers of clusters. We have calculated the log-likelihood which is a measure of the performance of the estimation scheme. We see that the optimal number of clusters is 3, because adding more clusters will not increase the performance of the estimation algorithm [see Fig. 10(c)]. We have also applied the procedure to 12 hours of observed data. In this case, the optimal number of clusters increases to 5. This is expected because we have a greater variety of regimes in 12 hours. This proves the efficiency of the IMM algorithm applied with this representative modes, because the complexity is a small factor of the EKF.

C. Implementation and Numerical Results

Algorithm 11 presents the four variants of the IMM algorithm discussed above. With only the mode of the state estimate $(\text{variant} = \text{'EKF'})$, the IMM is reduced to the EKF algorithm. If we add the adjacent modes (RIMM1), we obtain an improvement on the EKF with at most $2(n+1)$ modes. When we only consider the adjacent modes close to the state estimate (RIMM2), then the number of modes depends on the tolerance β in Algorithm 9. In the last variant (RIMM3), discussed in Section V-B, we suppose that the system can only switch between K representative mode vectors. We implement our algorithms on the same experimental data location as in IV-D. As mentioned in [19], the choice of the transition probabilities only affects slightly the performance of the IMM algorithm. The guideline for a proper choice is to match roughly the transition probabilities with the actual mean sojourn time of each mode. In RIMM1 and RIMM2, it is difficult to estimate the transition probabilities because of the exponential number of modes, so we suppose that the system is equally likely to transition to all the modes. In RIMM3, we take sample transition probabilities from the observed data:

$$\tilde{\pi}_{ij} = \frac{\gamma + \sum_{t=1}^{T} \mathbb{I}(\boldsymbol{\rho}^t \in \mathcal{C}_i,\ \boldsymbol{\rho}^{t+1} \in \mathcal{C}_j)}{\gamma K + \sum_{t=1}^{T-1} \mathbb{I}(\boldsymbol{\rho}^t \in \mathcal{C}_i)} \tag{31}$$

where the sets $\{\mathcal{C}_k\}_k$ are the Voronoi cells centered on centroids $\{\tilde{\boldsymbol{\rho}}^k\}$ computed in Algorithm 10, $\mathbb{I}$ is the indicator function, and γ controls the smoothing from the uniform transitions.

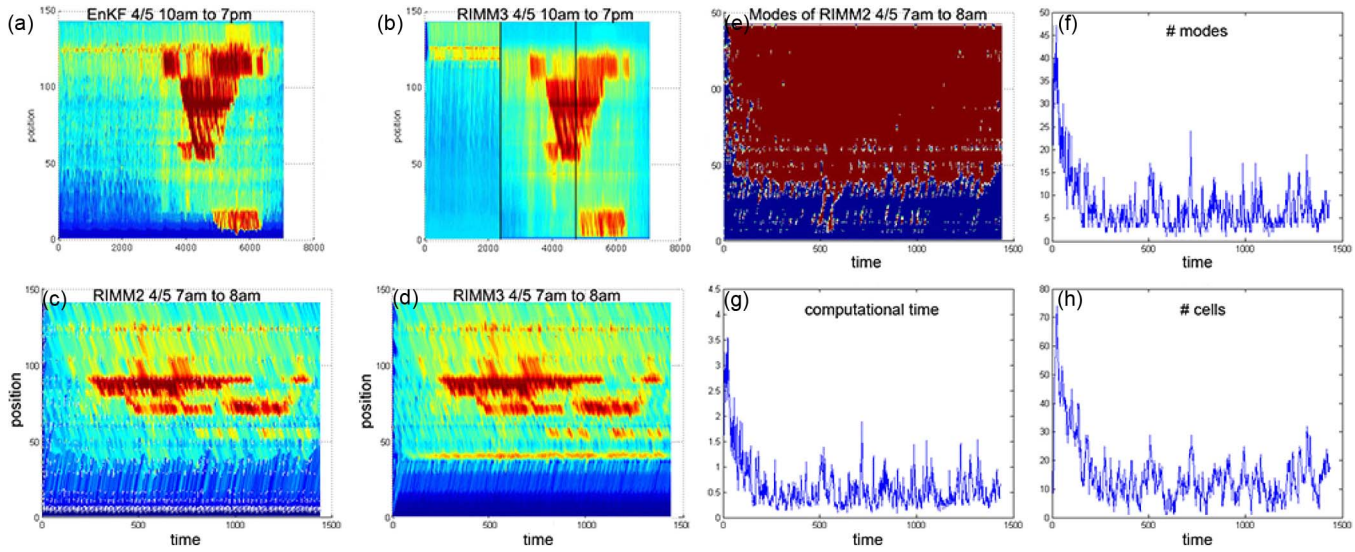

Fig. 11. Contour plot of the density given by (a) the EnKF with 100 ensembles on May 5th, 10am-7pm, (b) the RIMM3 with five clusters on May 5th, at 10am–1pm, 1–4pm, 4–7pm, (c) the RIMM2 with $\beta = 1$ on May 5th, 7–8am, (d) the RIMM3 with 20 clusters using the k-means algorithm on May 5th, 7–8am. Analysis of each time step of the RIMM2 with $\beta = 1$: (e) plot of the mode estimate, (f) number of modes selected by RIMM2, (g) computational time, (h) number of cells with density close to ρ_c.

Algorithm 11 IMM with reduced number of modes: **IMM**(algorithm)

Require: initial state $\boldsymbol{\rho}_0$, boundary conditions $\{u(t), d(t)\}_{t\geq 0}$, state covariance $\{Q^t\}_{t\geq 0}$, observations $\{\boldsymbol{z}^t\}_{t\geq 0}$, observation matrix $\{H^t\}_{t\geq 0}$, observation covariance $\{R^t\}_{t\geq 0}$.

1: $\mathcal{M}^0 = \{\mathbf{rho2m}(\boldsymbol{\rho}_0)\}$ {initial set of modes is the mode of $\boldsymbol{\rho}_0$}
2: **for** $t \in \{0, 1, 2, \cdots\}$ **do**
3: $\quad \boldsymbol{m} = \mathbf{rho2m}(\hat{\boldsymbol{\rho}}^t)$ {Algo 1}
4: $\quad$ **if** 'EKF' **then** $\mathcal{M}^{t+1} = \{\boldsymbol{m}\}$
5: $\quad$ **if** 'RIMM1' **then** $\mathcal{M}^{t+1} = \{\boldsymbol{m}\} \cup \mathbf{adj}(\boldsymbol{m})$ {Algo 5}
6: $\quad$ **if** 'RIMM2' **then** $\mathcal{M}^{t+1} = \{\boldsymbol{m}\} \cup \mathbf{adj2}(\boldsymbol{m}, \hat{\boldsymbol{\rho}}^t, P^t, \beta)$ {Algo 9}
7: $\quad$ **if** 'RIMM3' **then** $\mathcal{M}^{t+1} = \mathcal{M}^K$ {Algo 10}
8: $\quad$ **for** $\boldsymbol{m}_j \in \mathcal{M}^{t+1}$ **do**
9: $\quad\quad (\hat{\boldsymbol{\rho}}^t_{0j}, P^t_{0j}) = \mathbf{mixing}((\hat{\boldsymbol{\rho}}^t_i, P^t_i, \mu^t_i)_{i\in\mathcal{M}^t})$ {see (26)}
10: $\quad\quad (\hat{\boldsymbol{\rho}}^{t+1}_j, P^{t+1}_j, \Lambda^{t+1}_j) = \mathbf{KF}(\boldsymbol{m}_j, \hat{\boldsymbol{\rho}}^t_{0j}, P^t_{0j}, \cdots)$ {Algo. 8}
11: $\quad\quad \mu^{t+1}_j = \mathbf{modeProbUpdate}(\Lambda^{t+1}_j)$ {see (27)}
12: $\quad$ **end for**
13: $\quad (\hat{\boldsymbol{\rho}}^{t+1}, P^{t+1}, \hat{\boldsymbol{m}}^{t+1}) = \mathbf{combination}((\hat{\boldsymbol{\rho}}^{t+1}_j, P^{t+1}_j, \mu^{t+1}_j)_{j\in\mathcal{M}^{t+1}})$ {see (28)}
14: **end for**
15: **return** $(\hat{\boldsymbol{\rho}}^t, P^t)_{t\geq 0}$

The EnKF is a popular estimation algorithm based on *Monte-Carlo approximation* of the Kalman filter. The results are compared with the EnKF estimate presented in Section IV-C. Figs. 9(c), (d), 11(c), (d) present the four estimates which consist in the density in the time-space domain. The regions with high (resp. low) density are represented in red (resp. blue). The estimators give similar higher resolution scalar fields of the density (1440 time steps by 141 cells) by assimilating sparse density measurements (240 time steps by 29 PeMS stations. The shock wave propagation is more noticeable in the output of RIMM estimators in the congested regions.

The density centroids have also been computed to get a set of 5 representative mode vectors for each of the 10am–1pm, 1–4pm, 4–7pm time periods on March 1st, and we applied RIMM3 to estimate the density on March 5th at the same time periods. The estimates are very similar [see 11(a), (b)].

Fig. 11(e) shows the mode estimate computed in the *combination step* of the IMM. Each column represents a modal regime of the highway at a specific time. Finally, Fig. 11(f), (g), (h) show that the number of modes selected, the computational times, and the number of cells with density close to ρ_c at each time step are proportional.

VI. Conclusion

We introduce a new class of algorithms to estimate discretized hyperbolic PDEs. When the Godunov scheme is used to discretize the LWR PDE with a triangular flux function, we showed that the resulting nonlinear dynamical system can be decomposed in *piecewise affine* components. While the IMM seems a natural approach, it becomes intractable because of the exponential number of modes. We then study the validity domain of each mode and show that the state space can be divided into an exponential number of polyhedra.

Feasible heuristics are suggested for the estimation of hybrid multi-cellular systems based on the reduction of the number of modes. On one hand, we take advantage of the geometric properties of the partition into polyhedra to reduce the set of modes to a feasible set of adjacent modes centered around the mode of the state estimate. On the other hand, a clustering algorithm is applied in order to group the modes in clusters, and we show that when the historical data is chosen properly, the optimal number of clusters is less than 5. Hence, the

implementation of the reduced IMM algorithm shows that it is tuned to the discretized model due to its PWA structure.

We have constructed a framework for the estimation of the discretized LWR PDE which enables: 1) the use of Kalman filtering on each of the linear modes, 2) the use of statistical analysis in the space of modes to make the IMM tractable. We believe this work offers several directions of future research: it can inspire other feasible heuristics for the state estimation in other multi-cellular hybrid systems models. In particular, piecewise affine approximations of other flux functions combined with the Godunov scheme induce similar hybrid systems with an exponential number of modes and an adjacent polyhedral representation of the state space. Such representation is similar to the binary space partition (BSP) widely used in computational geometry, which could be used to determine the mode of the estimate.

APPENDIX

A. Proof of Proposition 3

We count the number of accepted mode strings recursively on the length k of the string. Let N_k be the number of accepted strings, Fig. 4 shows the 16 accepted strings of length 3. Let us denote by w_k (resp. l_k, d_k) the number of accepted strings which last element is W (resp. L, D). Then for all $k \geq 0$

$$\begin{aligned} &w_0 = l_0 = d_0 = 1 \\ &w_{k+1} = w_k + l_k + d_k \\ &l_{k+1} = w_k + d_k \\ &d_{k+1} = l_k + d_k \\ &\Longrightarrow \begin{bmatrix} w_k \\ l_k \\ d_k \end{bmatrix} = A^k \times \begin{bmatrix} w_0 \\ l_0 \\ d_0 \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \end{aligned} \tag{32}$$

hence, $N_k = w_k + l_k + d_k = e^T A^k e with e^T = [111]$. Diagonalizing the matrix A gives $A = VDV^{-1}$ with $D := diag(\lambda_1, \lambda_2, \lambda_3)$ where $\lambda_1, \lambda_2, \lambda_3$ are the eigenvalues of A in increasing order. Since λ_3 is the only eigenvalue above 1 in absolute value, we have

$$\begin{aligned} &D^k = \operatorname{diag}\left(\lambda_1^k, \lambda_2^k, \lambda_3^k\right) \sim \operatorname{diag}\left(0, 0, \lambda_3^k\right) \quad \text{when } k \longrightarrow +\infty \\ &\text{hence } e^T A^k e \approx e^T V \operatorname{diag}\left(0, 0, \lambda_3^k\right) V^{-1} e = \lambda_3^k (V^T e)_3 (V^{-1} e)_3 \\ &\approx 3.1778 \cdot (2.2470)^k. \end{aligned}$$

B. Proof or Proposition 5

In this section and the following one, we consider the closure $\overline{\mathbf{H}_k}, \overline{\mathbf{H}_{k+1/2}}, \overline{\mathbf{H}_k^c}, \overline{\mathbf{H}_{k+1/2}^c}$ of the half-spaces defined in (18), and for simplicity we still denote them *without bar*.

Lemma 3: The following inclusions for the half-spaces $\mathbf{H}_k$, $\mathbf{H}_{k+1/2}$, $\mathbf{H}_{k+1}$ for $k \in \{0, \ldots, n\}$ hold:

$$\begin{aligned} \boldsymbol{H}_k \cap \boldsymbol{H}_{k+1} &\subset \boldsymbol{H}_{k+1/2} \\ \boldsymbol{H}_k^c \cap \boldsymbol{H}_{k+1/2} &\subset \boldsymbol{H}_{k+1} \\ \boldsymbol{H}_{k+1/2}^c \cap \boldsymbol{H}_{k+1} &\subset \boldsymbol{H}_k^c \\ \boldsymbol{H}_k^c \cap \boldsymbol{H}_{k+1}^c &\subset \boldsymbol{H}_{k+1/2}^c. \end{aligned} \tag{33}$$

Proof: The proof is easy. See Fig. 5 for an illustration of these inclusions. □

Proof of Proposition 5: We fix an accepted mode vector $\boldsymbol{m}$ and we fix the associated decomposition (16), which gives the sequence $\mathbf{P}_{k+1/2}, k \in \{0, \ldots, n\}$. Let $\mathcal{H}$ be the set of half-spaces (or linear inequalities) in $[0, \rho_{\text{jam}}]^{n+2}$ defined in algorithm 4. First, we express $\mathrm{Dom}(\boldsymbol{m})$ in the form $\mathrm{Dom}(\boldsymbol{m}) = \bigcap_{i=0}^n \mathbf{P}_{i+1/2}$, then we prove by induction that at the kth iteration of the for loop in algorithm 4, the intersection of all the half-spaces in the current $\mathcal{H}$ is the minimal representation of $\bigcap_{i=0}^k \mathbf{P}_{i+1/2}$.

Initialization $k = 1$: if $m_1 \in \{1, 2\}$, then we have $\mathcal{H} = \{\mathbf{H}_{1/2}, \mathbf{H}_1\}$ from the algorithm and $\mathbf{P}_{1/2} = \mathbf{W}_{1/2}$ from Table I. The expression $\cap_{\mathbf{H} \in \mathcal{H}} \mathbf{H} = \mathbf{H}_{1/2} \cap \mathbf{H}_1$ is clearly the minimal representation of $\mathbf{W}_{1/2}$ from (17). The cases $m_1 \in \{3, 4\}$ and $m_1 \in \{5, 6, 7\}$ follow similarly.

Step k: The algorithm provides $\mathcal{H}^-$ and $\mathcal{H}$ which are the minimal representations of $\bigcap_{i=0}^{k-2} \mathbf{P}_{i+1/2}$ and $\bigcap_{i=0}^{k-1} \mathbf{P}_{i+1/2}$, respectively. We want to show that the algorithm updates $\mathcal{H}$ to $\mathcal{H}^+$, such that $\mathcal{H}^+$ is the minimal representation of $\bigcap_{i=0}^k \mathbf{P}_{i+1/2}$. We have seven cases:

a) If $m_k = 1$, then from Table I and (17)

$$\begin{aligned} \mathbf{P}_{k-1/2} &= \mathbf{W}_{k-1/2} = \mathbf{H}_{k-1/2} \cap \mathbf{H}_k \\ \mathbf{P}_{k+1/2} &= \mathbf{W}_{k+1/2} = \mathbf{H}_{k+1/2} \cap \mathbf{H}_{k+1} \end{aligned}$$

in the expression (16), hence $\mathcal{H} \subset \mathcal{H}^- \cup \{\mathbf{H}_{k-1/2}, \mathbf{H}_k\}$. In this case, algorithm 4 adds constraint $\mathbf{H}_{k+1}$ to $\mathcal{H}$, so $\mathcal{H}^+ \subset \mathcal{H}^- \cup \{\mathbf{H}_{k-1/2}, \mathbf{H}_k, \mathbf{H}_{k+1}\}$. Then

$$\begin{aligned} \mathbf{H}_{k-1/2} \cap \mathbf{H}_k \cap \mathbf{H}_{k+1} &= (\mathbf{H}_{k-1/2} \cap \mathbf{H}_k) \cap (\mathbf{H}_{k+1} \cap \mathbf{H}_{k+1/2}) \\ &= \mathbf{W}_{k-1/2} \cap \mathbf{W}_{k+1/2} \end{aligned}$$

where the third equality is from $\mathbf{H}_k \cap \mathbf{H}_{k+1} \subset \mathbf{H}_{k+1/2}$ in (33). Hence, $\mathcal{H}^+$ is a representation of $\cap_{i=0}^k \mathbf{P}_{i+1/2}$. Finally, $\mathcal{H}^+$ is minimal because the added constraint $\mathbf{H}_{k+1}$ is the only constraint on ρ_{k+1}, so it is not redundant with the constraints in $\mathcal{H}$.

b) If $m_k = 2$, then $\mathbf{P}_{k-1/2}$ and $\mathbf{P}_{k+1/2}$ in the expression (16) are

$$\begin{aligned} \mathbf{P}_{k-1/2} &= \mathbf{W}_{k-1/2} = \mathbf{H}_{k-1/2} \cap \mathbf{H}_k \\ \mathbf{P}_{k+1/2} &= \mathbf{L}_{k+1/2} = \mathbf{H}_k \cap \mathbf{H}_{k+1}^c \end{aligned}$$

and constraint $\mathbf{H}_{k+1}^c$ is added to $\mathcal{H}$ in algorithm 4, so $\mathcal{H}^+ \subset \mathcal{H}^- \cup \{\mathbf{H}_{k-1/2}, \mathbf{H}_k, \mathbf{H}_{k+1}^c\}$, and

$$\begin{aligned} \mathbf{H}_{k-1/2} \cap \mathbf{H}_k \cap \mathbf{H}_{k+1}^c &= (\mathbf{H}_{k-1/2} \cap \mathbf{H}_k) \cap \left(\mathbf{H}_k \cap \mathbf{H}_{k+1}^c\right) \\ &= \mathbf{P}_{k-1/2} \cap \mathbf{P}_{k+1/2} \end{aligned}$$

so $\cap_{\mathbf{H} \in \mathcal{H}^+} \mathbf{H} = \cap_{i=0}^k \mathbf{P}_{i+1/2}$, i.e., $\mathcal{H}^+$ is a representation of $\cap_{i=0}^k \mathbf{P}_{i+1/2}$. This is the minimal representation because the added constraint $\mathbf{H}_{k+1}^c$ is the only constraint on ρ_{k+1}.

c) If $m_k = 3$, the analysis is similar to case $m_k = 1$.

d) If $m_k = 4$, the analysis is similar to case $m_k = 2$.

e) If $m_k = 5$, then $\mathbf{P}_{k-1/2} = \mathbf{D}_{k-1/2} = \mathbf{H}_{k-1}^c \cap \mathbf{H}_{k-1/2}^c$ and $\mathbf{P}_{k+1/2} = \mathbf{W}_{k+1/2} = \mathbf{H}_{k+1/2} \cap \mathbf{H}_{k+1}$ in expression (16). Algorithm 4 adds constraints $\mathbf{H}_{k+1/2}, \mathbf{H}_{k+1}$ to $\mathcal{H}$; hence, $\mathcal{H}^+$ is a representation of $\cap_{i=0}^k \mathbf{P}_{i+1/2}$. It

is easy to see that the constraints $\mathbf{H}_{k+1/2} \cap \mathbf{H}_{k+1}$ are not redundant; hence, $\mathcal{H}^+$ is the minimal representation of $\cap_{i=0}^{k} \mathbf{P}_{i+1/2}$.

f) If $m_k = 6$, then $\mathbf{P}_{k-1/2} = \mathbf{D}_{k-1/2} = \mathbf{H}_{k-1}^c \cap \mathbf{H}_{k-1/2}^c$ and $\mathbf{P}_{k+1/2} = \mathbf{L}_{k+1/2} = \mathbf{H}_k \cap \mathbf{H}_{k+1}^c$ in expression (16). We have $\mathcal{H} \subset \mathcal{H}^- \cup \{\mathbf{H}_{k-1}^c, \mathbf{H}_{k-1/2}^c\}$. Algorithm 4 removes constraint $\mathbf{H}_{k-1}^c$ from $\mathcal{H}$ (if $\mathcal{H}$ contains it) and adds constraints $\mathbf{H}_k, \mathbf{H}_{k+1}^c$, hence $\mathcal{H}^+ \subset \mathcal{H}^- \cup \{\mathbf{H}_{k-1/2}^c, \mathbf{H}_k, \mathbf{H}_{k+1}^c\}$. The only potential redundancies in $\mathcal{H}^+$ would be between $\mathbf{H}_{k-1/2}^c$ and the newly added constraints $\mathbf{H}_k, \mathbf{H}_{k+1}^c$. It is easy to verify that there is no redundant constraint in $\mathcal{H}^+$. Finally, since we have the inclusion $\mathbf{H}_{k-1/2}^c \cap \mathbf{H}_k \subset \mathbf{H}_{k-1}^c$ from (33)

$$\begin{aligned}\mathbf{H}_{k-1/2}^c \cap \mathbf{H}_k \cap \mathbf{H}_{k+1}^c &= \left(\mathbf{H}_{k-1}^c \cap \mathbf{H}_{k-1/2}^c\right) \cap \left(\mathbf{H}_k \cap \mathbf{H}_{k+1}^c\right) \\ &= \mathbf{D}_{k-1/2} \cap \mathbf{L}_{k+1/2}\end{aligned}$$

g) hence $\mathcal{H}^+$ is the minimal representation of $\cap_{i=0}^{k} \mathbf{P}_{i+1/2}$.

If $m_k = 7$, the analysis is similar to case $m_k = 6$.

$$\begin{aligned}\mathbf{H}_{k-1}^c \cap \mathbf{H}_k^c \cap \mathbf{H}_{k+1/2}^c &= \left(\mathbf{H}_{k-1}^c \cap \mathbf{H}_{k-1/2}^c\right) \cap \left(\mathbf{H}_k^c \cap \mathbf{H}_{k+1/2}^c\right) \\ &= \mathbf{D}_{k-1/2} \cap \mathbf{D}_{k+1/2}.\end{aligned}$$

hence $\mathcal{H}^+$ is the minimal representation of $\cap_{i=0}^{k} \mathbf{P}_{i+1/2}$. This finishes the proof.

C. Proof of Proposition 6

Lemma 4: The following inclusions for the half-spaces $\mathbf{H}_i$, $\mathbf{H}_{i+1/2}$, $\mathbf{H}_{i+1}$ defined in (18) for $i \in \{0, \ldots, n\}$ hold:

$$\begin{aligned}&\boldsymbol{H}_{i+1/2}^c \cap \boldsymbol{H}_{i+1} \subset \boldsymbol{D}_{i+1/2} && \boldsymbol{H}_{i+1/2} \cap \boldsymbol{H}_{i+1}^c \subset \boldsymbol{L}_{i+1/2} \\ &\boldsymbol{H}_i^c \cap \boldsymbol{H}_{i+1}^c \subset \boldsymbol{D}_{i+1/2} && \boldsymbol{H}_i \cap \boldsymbol{H}_{i+1} \subset \boldsymbol{W}_{i+1/2} \\ &\boldsymbol{H}_{i+1/2} \cap \boldsymbol{H}_i^c \subset \boldsymbol{W}_{i+1/2} && \boldsymbol{H}_{i+1/2}^c \cap \boldsymbol{H}_i \subset \boldsymbol{L}_{i+1/2}.\end{aligned} \tag{34}$$

Proof of Proposition 6: We fix the mode vector $\boldsymbol{m}$, its domain $\mathrm{Dom}(\boldsymbol{m}) = \cap_{i=0}^{n} \mathbf{P}_{i+1/2}$, its mode string $\boldsymbol{s}$, and its minimal representation $\mathcal{H}$. Let $\mathbf{H} \in \mathcal{H}$. Since algorithm 4 (minRep) only adds constraints of the form $\mathbf{H}_i, \mathbf{H}_{i+1/2}, \mathbf{H}_i^c, \mathbf{H}_{i+1/2}^c$ in (18) when finding the minimal representation, and the minimal representation is unique, we must have $\mathbf{H} \in (\cup_{i=0}^{n+1} \{\mathbf{H}_i, \mathbf{H}_i^c\}) \cup (\cup_{i=0}^{n} \{\mathbf{H}_{i+1/2}, \mathbf{H}_{i+1/2}^c\})$. We have 4 different cases (the 4 cases when $\mathbf{H}$ is a constraint at the boundaries are not covered)

a) Suppose $\mathbf{H} = \mathbf{H}_i$ for $i \in \{1, \ldots, n\}$. Since $\mathbf{H}_i$ only appears in $\mathbf{W}_{i-1/2}$ and $\mathbf{L}_{i+1/2}$ in (17), then algorithm 4 only adds $\mathbf{H}_i$ to the minimal representation if $\mathbf{P}_{i-1/2} = \mathbf{W}_{i-1/2} = \mathbf{H}_{i-1/2} \cap \mathbf{H}_i$ and $\mathbf{P}_{i+1/2} = \mathbf{L}_{i+1/2} = \mathbf{H}_i \cap \mathbf{H}_{i+1}^c$. We define the polyhedron

$$\mathbf{P} = \bigcap_{j \in \{0,\ldots,n\} \setminus \{i-1,i\}} \mathbf{P}_{j+1/2} \cap \left(\mathbf{H}_i^c \cap \mathbf{H}_{i+1}^c\right) \cap \left(\mathbf{H}_{i-1/2} \cap \mathbf{H}_i^c\right)$$

by substituting the constraint $\mathbf{H}_i$ in $\mathrm{Dom}(\boldsymbol{m}) = \cap_{i=0}^{n} \mathbf{P}_{i+1/2}$ by the opposite constraint $\mathbf{H}_i^c$. Since $\mathbf{H}_i$ is in the minimal representation, $\mathbf{P}$ must be proper (otherwise $\mathbf{H}_i$ would be redundant).

We have $\mathbf{P} \subset j \in \{0, \ldots, n\} \setminus \{i-1, i\} \cap \mathbf{P}_{j+1/2} \cap (\mathbf{L}_{i-1/2} \cap \mathbf{D}_{i+1/2})$ from the inclusions in (34); hence, the subset on the right-hand side is proper. Since this subset is also of the form $\cap \mathbf{P}_{j+1/2}$, it is the domain of an accepted mode vector, and so a polyhedron of the partition. By construction, it is $(\mathrm{n}+1)$-adjacent to $\mathrm{Dom}(\boldsymbol{m})$. Its associated mode string is obtained from $\boldsymbol{s}$ by replacing entry $s(i-1) = W$ with L when line 8 in the for loop is executed at iteration $i-1$, and by replacing $s(i) = L$ with D when line 6 is executed at iteration i. The case $\mathbf{H} = \mathbf{H}_i^c$ for $i \in \{1, \ldots, n\}$ is similar.

b) Suppose $\mathbf{H} = \mathbf{H}_{i+1/2}$ for $i \in \{0, \ldots, n\}$. Since $\mathbf{H}_{i+1/2}$ only appear in $\mathbf{W}_{i+1/2}$, we have $\mathbf{P}_{i+1/2} = \mathbf{W}_{i+1/2} = \mathbf{H}_{i+1/2} \cap \mathbf{H}_{i+1}$. Let $\mathbf{P} := j \neq i \cap \mathbf{P}_{j+1/2} \cap (\mathbf{H}_{i+1/2}^c \cap \mathbf{H}_{i+1}) \subset j \neq i \cap \mathbf{P}_{j+1/2} \cap \mathbf{D}_{i+1/2}$, then $j \neq i \cap \mathbf{P}_{j+1/2} \cap \mathbf{D}_{i+1/2}$ is proper, and it is a polyhedron of the partition $(\mathrm{n}+1)$-adjacent to $\mathrm{Dom}(\boldsymbol{m})$ by construction. Its associated mode string is obtained by replacing $s(i) = W$ with D when line 10 of the algorithm is executed at iteration i. The case $\mathbf{H} = \mathbf{H}_{i+1/2}^c$ for $i \in \{0, \ldots, n\}$ is similar. This finishes the proof.

ACKNOWLEDGMENT

The authors would like to thank Professor Claire Tomlin from University of California, Berkeley for insightful discussions on hybrid systems and the members of the staff of the California Institute for Innovative Transportation for its contributions to develop, build, and deploy the Mobile Millennium system on which this paper relies.

REFERENCES

[1] Y. Bar-Shalom and X. R. Li, *Estimation and Tracking: Principles, Techniques, and Software*. Norwood, MA, USA: Artech House, 1993.

[2] C. Bardos, A. Y. Leroux, and J. C. Nedelec, "First order quasilinear equations with boundary conditions," *Commun. Partial Different. Equat. 4*, vol. 9, pp. 1017–1034, 1979.

[3] S. Blandin, A. Couque, A. Bayen, and D. Work, "On sequential data assimilation for scalar macroscopic traffic flow models," *Physica D*, 2012.

[4] H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Trans. Autom. Control*, vol. 33, no. 8, pp. 780–783, Aug. 1988.

[5] R. Chen and J. S. Liu, "Mixture Kalman filters," *R. Statist. Soc.*, vol. 62, pp. 493–508, 2000.

[6] C. F. Daganzo, "The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory," *Transport. Res. Part B 28*, vol. 28, no. 4, pp. 269–287, 1994.

[7] C. F. Daganzo, "The cell transmission model, part II: Network traffic," *Transport. Res. Part B 29*, vol. 29, no. 2, pp. 79–93, 1995.

[8] E. Godlewski and P.-A. Raviart, *Numerical Approximation of Hyperbolic Systems of Conservation Laws*. New York, NY, USA: Applied Mathematical Sciences, 1996.

[9] S. K. Godunov, "A finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics," *Math. Sbornik*, vol. 47, pp. 271–306, 1959.

[10] B. D. Greenshields, "A study of traffic capacity," *Proc. 14th Annu. Meeting Highway Res. Board*, vol. 14, pp. 448–477, 1934.

[11] B. Grünbaum, *Convex Polytopes*. New York, NY, USA: Springer, 2003.

[12] R. M. Hawkes and J. B. Moore, "Performance bounds for adaptive estimation," *Proc. IEEE*, vol. 64, no. 8, pp. 1143–1150, Aug. 1976.

[13] I. Hwang, H. Balakrishnan, and C. Tomlin, "State estimation for hybrid systems: Applications to aircraft tracking," *IEE Proc. Control Theory Applicat.*, vol. 153, no. 5, pp. 556–566, Sep. 2006.

[14] J. P. Lebacque, "The Godunov scheme and what it means for first order traffic flow models," in *13th Int. Symp. Transport. Traffic Theory*, 1996, pp. 647–677.

[15] P. LeFloch, "Explicit formula for scalar non-linear conservation laws with boundary condition," *Math. Meth. Appl. Sci.*, vol. 10, pp. 265–287, 1988.
[16] M. D. Lemmon, K. X. He, and I. Markovsky, "Supervisory hybrid systems," *IEEE Control Syst. Mag.*, vol. 19, no. 4, pp. 42–55, Aug. 1999.
[17] B. Lennartson, M. Tittus, B. Egardt, and S. Petterson, "Hybrid systems in process control," *IEEE Control Syst. Mag.*, vol. 16, no. 5, pp. 45–56, 1996.
[18] R. J. LeVeque, *Numerical Methods for Conservation Laws*. Basel, Switzerland: Birkhäuser, 1992.
[19] X. R. Li and Y. Bar-Shalom, "Design of an interacting multiple model algorithm for air traffic control tracking," *IEEE Trans. Control Syst. Technol.*, vol. 1, no. 3, pp. 186–194, Sep. 1993.
[20] X. R. Li and Y. Bar-Shalom, "Performance prediction of the interacting multiple model algorithm," *IEEE Trans. Aerosp. Electron. Syst*, vol. 29, no. 3, pp. 755–771, Jul. 1993.
[21] M. J. Lighthill and G. B. Whitham, "On kinematic waves II. A theory of traffic flow on long crowded roads," *Proc. R. Soc. London. Series A, Math. Phys. Sci.*, vol. 229, pp. 317–345, 1955.
[22] J. Mandel, "Efficient implementation of the ensemble Kalman filter," CCM Rep. No. 231, 2006.
[23] P. S. Maybeck, *Stochastic Models, Estimation, and Control*, vol. 2. New York, NY, USA: Academic, 1982.
[24] M. Papageorgiou, J.-M. Blosseville, and H. Hadj-Salem, "Modelling and real-time control of traffic flow on the southern part of Boulevard Peripherique in Paris: Part I: Modelling," *Transport. Res.*, vol. 24, no. 5, pp. 345–359, 1990.
[25] P. I. Richards, "Shock waves on the highway," *Operat. Res.*, vol. 4, pp. 42–51, 1956.
[26] T. Schreiter, C. van Hinsbergen, F. Zuurbier, H. van Lint, and S. Hoogendoorn, "Data-model synchronization in extended Kalman filters for accurate online traffic state estimation," in *Proc. Traffic Flow Theory Conf.*, Annecy, France, 2010.
[27] I. S. Strub and A. M. Bayen, "Weak formulation of boundary conditions for scalar conservation laws: An application to highway traffic modeling," *Int. J. Robust Nonlinear Control*, vol. 16, pp. 733–748, 2006.
[28] D. D. Sworder and J. E. Boyd, *Estimation Problems in Hybrid Systems*. Cambridge, U.K.: Cambridge Unive. Press, 1999.
[29] J. Thai, B. Prodhomme, and A. M. Bayen, "State estimation for the discretized LWR PDE using explicit polyhedral representations of the Godunov scheme," in *Proc. Amer. Control Conf.*, 2013.
[30] D. B. Work, S. Blandin, O. Tossavainen, B. Piccoli, and A. M. Bayen, "A traffic model for velocity data assimilation," *Appl. Math. Res. eXpress*, 2010.

Jérôme Thai received the Engineering degree in applied mathematics from Ecole Polytechnique, Paris, France, in May 2010, the M.S. degree in operations research from Columbia University, New York, NY, USA, in December 2011. He is currently pursuing the Ph.D. degree in the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, USA.

His research interests are at the intersection of machine learning, game theory, and estimation and control of cyber-physical systems, with applications to transportation.

Alexandre M. Bayen received the Engineering degree in applied mathematics from Ecole Polytechnique, Paris, France, in July 1998 and the M.S. and Ph.D. degrees in aeronautics and astronautics from Stanford University, Stanford, CA, USA, in June 1999 and December 2003, respectively.

He was a Visiting Researcher at NASA Ames Research Center from 2000 to 2003. Between January 2004 and December 2004, he worked as the Research Director of the Autonomous Navigation Laboratory at the Laboratoire de Recher-ches Balistiques et Aerodynamiques, (Ministere de la Defense, Vernon, France), where he holds the rank of Major. He is an Associate Professor in the Department of Electrical Engineering and Computer Sciences, and the Department Civil and Environmental Engineering at the University of California, Berkeley, Berkeley, CA, USA. He has authored one book and over 100 articles in peer-reviewed journals and conferences.

Dr. Bayen is the recipient of the Ballhaus Award from Stanford University, 2004, of the CAREER award from the National Science Foundation, 2009 and he is a NASA Top 10 Innovators on Water Sustainability, 2010. His projects Mobile Century and Mobile Millennium received the 2008 Best of ITS Award for "Best Innovative Practice" at the ITS World Congress and a TRANNY Award from the California Transportation Foundation, 2009. He is the recipient of the Presidential Early Career Award for Scientists and Engineers (PECASE) award from the White House, 2010. Mobile Millennium has been featured more than 100 times in the media, including TV channels and radio stations (CBS, NBC, ABC, CNET, NPR, KGO, the BBC), and in the popular press (Wall Street Journal, Washington Post, LA Times).