

# Freeway Traffic Control from Linear Temporal Logic Specifications

Samuel Coogan  
scoogan@eecs.berkeley.edu

Murat Arcak  
arcak@eecs.berkeley.edu

Department of Electrical Engineering and Computer Sciences  
University of California, Berkeley  
Berkeley, CA 94720

## ABSTRACT

We propose a methodology for synthesizing ramp metering control strategies for freeway networks from linear temporal logic specifications. Such specifications allow very rich control objectives constructed from temporal operators such as “always” and “eventually” combined with Boolean operators and encompass *e.g.* safety, reachability, and liveness conditions. We use the cell transmission model of traffic flow on freeway networks to obtain a piecewise affine model of the traffic network, and we apply recent results on control of such systems from temporal logic specifications to synthesize ramp metering strategies that are correct by construction. We demonstrate our approach on several examples.

## Categories and Subject Descriptors

I.2.8 [ARTIFICIAL INTELLIGENCE]: Problem Solving, Control Methods, and Search—*Control theory*; D.2.4 [SOFTWARE ENGINEERING]: Software/Program Verification—*Formal methods*

## Keywords

Traffic networks, ramp metering, linear temporal logic, piecewise affine dynamical systems

## 1. INTRODUCTION

Increased travel demand coupled with decreased availability of funds and space for expanded roadway has made traffic management an increasingly important research domain [15]. Methods for mitigating freeway

traffic congestion often rely on ramp metering control strategies that control the flow of vehicles entering the freeway network at onramps, see [20] for an overview of various approaches to ramp metering.

Existing ramp metering approaches often have fairly simple objectives. Examples in recent literature include [6] in which the control objective is to minimize a linear combination of total travel time and total distance traveled by all vehicles in the network over a finite horizon. In [7], the authors construct ramp metering strategies to induce an equilibrium in which the freeway is not congested and throughput is increased compared to the equilibrium with no ramp metering. In [22], the authors modify a standard integral-action control law at a single ramp so that the desired set-point is adaptively adjusted to maximize throughput. In [9], the authors apply model predictive control to obtain a ramp metering strategy for a freeway network and variable mainline speed limits to minimize total time spent by all vehicles in the network. In [11], the authors propose an iterative learning algorithm for obtaining a (typically) daily control profile by considering effectiveness of control profiles used on previous days.

In this work, we consider a much richer class of control objectives for ramp metering. In particular, we consider control specifications expressible in *linear temporal logic (LTL)*, see [2] and [1] for overviews. LTL allows control specifications that include safety conditions such as “always avoid traffic congestion”, reachability properties such as “eventually attain acceptable traffic throughput”, liveness conditions such as “always eventually discharge long ramp queues”, sequentiality conditions such as “avoid congestion until onramp queues are full”, and many combinations of these conditions.

We model freeway networks using the *Cell Transmission Model (CTM)* [3, 4], see also [6, 7]. In the CTM, the fluid-like flow of traffic through junctions is determined by considering the upstream *demand* of vehicles wishing to move through the junction and the downstream *supply* of available road space. The demand of traffic

wishing to exit a link is linear in the density of vehicles on that link up to a saturating flow value. Likewise, supply of road space decreases linearly up to a jam density for which no road capacity is available. Furthermore, flow of an upstream link at a junction is divided among downstream links via fixed *turn ratios*, and supply of a downstream link is divided among upstream links via fixed *supply ratios*. Combined, these factors lead to a *piecewise affine (PWA)* traffic network model.

By considering each partition of the PWA traffic network model to be a single state in a finite state machine, we create an abstraction of the traffic network dynamics amenable to the powerful results available from the formal methods domain of computer science. In particular, we leverage results developed in [27] for synthesizing control strategies of PWA systems to achieve objectives expressible as LTL formulae, see [14, 12] for related theory on controller synthesis. In [27], the authors provide a software package called *conPAS2* to automate the controller synthesis process. We employ a modified version of conPAS2 that allows for input constraints that are based on the current state and can be calculated using polytopic operations, see Section 5.3.

An important contribution of [27] is the ability to find controllers even when the finite state abstraction of the traffic network is not deterministic; this novelty is made possible by the algorithm for solving *Rabin games* on graphs developed in [10]. The approach taken to controller synthesis in this paper and in [27] is related to the *bisimulation* approach employed in [23], however the abstractions obtained here and in [27] are typically non-deterministic and therefore only a *simulation* [1] of the PWA dynamics.

This paper is organized as follows: In Section 3, we review necessary preliminaries. In Section 4, we derive the piecewise affine CTM model of traffic flow on freeway networks. In Section 5, we describe the controller synthesis process, and we suggest a method for refining the partitioning of the traffic network PWA model to decrease conservativeness of the synthesis method and to allow for richer controller specifications. We present examples in Section 6 and provide concluding remarks in Section 7.

## 2. NOTATION

Let  $\mathbb{N} \triangleq \{0, 1, 2, \dots\}$ . For a function  $x[t]$  with  $t \in \mathbb{N}$ , we interchangeably refer to  $x[t]$  as a *sequence* or a *signal*. In the context of finite state machines or transition systems, such sequences are sometimes called *strings* or *words*. We often drop the index variable when it is apparent from context, *i.e.* we write  $x$  for  $x[t]$ .

For a set  $Q$ ,  $2^Q$  denotes the set of all subsets of  $Q$ ,  $|Q|$  denotes the cardinality of  $Q$ , and  $\text{int}(Q)$  denotes the interior of  $Q$ . For a finite set of points  $Q = \{q_i\}_{i=1}^{|Q|}$  with

$q_i \in \mathbb{R}^n$  for all  $i$ , we let  $\text{hull}(Q)$  denote the convex hull of  $Q$ , that is

$$\text{hull}(Q) = \left\{ \sum_{i=1}^{|Q|} \alpha_i q_i \mid \sum_i \alpha_i = 1, \alpha_i \geq 0 \forall i \right\} \subset \mathbb{R}^n. \quad (1)$$

For two vectors  $x, y \in \mathbb{R}^n$ ,  $x \leq y$  denotes elementwise inequality, and  $x < y$  denotes elementwise strict inequality.

## 3. PRELIMINARIES

### 3.1 Polytopes

A polytope  $\mathcal{X} \subseteq \mathbb{R}^n$  is a convex set representable as the intersection of a finite number of half-spaces, that is

$$\mathcal{X} = \{x \mid Hx \leq K\} \quad (2)$$

for matrix  $H \in \mathbb{R}^{m \times n}$  and vector  $K \in \mathbb{R}^m$  where  $m$  is the number of half-spaces defining  $\mathcal{X}$ . A polytope is *full dimensional* if there exists a point  $x$  such that  $Hx < K$ . Alternatively,  $\mathcal{X}$  can be defined as the convex hull of its vertices. That is,  $\mathcal{X} = \text{hull}\{V(\mathcal{X})\}$  where  $V(\mathcal{X})$  is the set of vertices of  $\mathcal{X}$ .

### 3.2 Piecewise Affine Systems

A discrete-time *piecewise affine (PWA)* dynamical system has dynamics [8]

$$\begin{aligned} x[t+1] &= f(x[t], u[t]) \\ &\triangleq A_p x[t] + B_p u[t] + c_p \quad \text{if } x[t] \in \mathcal{X}_p \end{aligned} \quad (3) \quad (4)$$

where  $x \in \mathbb{R}^n$  is the state,  $u \in \mathcal{U} \subset \mathbb{R}^m$  is the control input which is constrained to lie within the polytope  $\mathcal{U}$ , and  $\mathcal{X}_p$  is a full dimensional polytope for each  $p \in \mathcal{P}$  where  $\mathcal{P}$  is an index set for the partitions defining the system dynamics with the property  $\text{int}(\mathcal{X}_p) \cap \text{int}(\mathcal{X}_q) = \emptyset$  for all  $p, q \in \mathcal{P}$ . The domain of the PWA system is  $\mathcal{X} \triangleq \bigcup_{p \in \mathcal{P}} \mathcal{X}_p$ , which we assume to be non-empty, compact, and connected. The PWA system is *continuous* if  $f(\cdot, \cdot)$  is continuous over  $\mathcal{X} \times \mathcal{U}$ . All PWA systems encountered in this paper are continuous and satisfy  $B_p = B$  for all  $p \in \mathcal{P}$  for some  $B \in \mathbb{R}^{n \times m}$ . Continuity implies there is no ambiguity when  $x[t] \in \mathcal{X}_p \cap \mathcal{X}_q$  for some  $p, q \in \mathcal{P}$ .

### 3.3 Transition Systems and Rabin Automata

A *transition system* is a tuple  $\mathcal{T} = (\mathcal{Q}, \Sigma, \delta, O, \gamma)$  where  $\mathcal{Q}$  is a set of states,  $\Sigma$  is a set of inputs,  $\delta : \mathcal{Q} \times \Sigma \rightarrow 2^{\mathcal{Q}}$  is a transition map,  $O$  is a set of observations, and  $\gamma : \mathcal{Q} \rightarrow O$  is an observation map. Given a state  $q \in \mathcal{Q}$ ,  $\delta(q, \sigma) \subset \mathcal{Q}$  denotes the set of states into which the system may transition under input  $\sigma \in \Sigma$ . If  $\mathcal{Q}$ ,  $\Sigma$ , and  $O$  are finite, we say the transition system is *finite*. All transition systems in this paper are finite. As

in [27], we exclude initial states from the definition of  $\mathcal{T}$  as we will synthesize the initial states along with a control strategy for all transition systems appearing in this paper.

From an initial state  $q_0 \in \mathcal{Q}$  and an input sequence  $\sigma[t]$  for  $t \in \mathbb{N}$  with  $\sigma[t] \in \Sigma$  for all  $t$ , a *trajectory* of  $\mathcal{T}$  is defined to be a sequence  $q[t], t \in \mathbb{N}$  such that  $q[0] = q_0$  and  $q[t+1] \in \delta(q[t], \sigma[t])$  for all  $t$ . Note that given an input sequence  $\sigma[t]$  and initial condition  $q_0$ , trajectories may not be unique. Each trajectory  $q[t]$  of  $\mathcal{T}$  generates an output sequence  $o[t], t \in \mathbb{N}$  defined by  $o[t] \triangleq \gamma(q[t])$  for all  $t$ .

A *Rabin automaton* is a transition system for which only some trajectories are *accepted*. Formally, a Rabin automaton is a tuple  $\mathcal{A} = (\mathcal{Q}, \Sigma, \delta, \mathcal{Q}^0, F)$  where  $\mathcal{Q}^0 \subset \mathcal{Q}$  is a set of initial states, and

$$F = \{(G_1, B_1), \dots, (G_{n_F}, B_{n_F})\} \quad (5)$$

with  $G_i \subseteq \mathcal{Q}, B_i \subseteq \mathcal{Q}$  for  $i = 1, \dots, n_F$  is the *acceptance condition*. A trajectory of  $\mathcal{A}$  is defined analogously to trajectories of a transition system  $\mathcal{T}$ . For a trajectory  $q[t]$  of  $\mathcal{A}$ , let  $\text{Inf}(q[t]) \subset \mathcal{Q}$  denote the set of states that occur infinitely often in the sequence  $q[0], q[1], \dots$ . A trajectory  $q[t]$  of  $\mathcal{A}$  is said to be *accepting* if  $q[0] \in \mathcal{Q}^0$  and there exists some  $i \in \{1, \dots, n_F\}$  such that

$$\text{Inf}(q[t]) \cap G_i \neq \emptyset, \quad \text{and} \quad (6)$$

$$\text{Inf}(q[t]) \cap B_i = \emptyset. \quad (7)$$

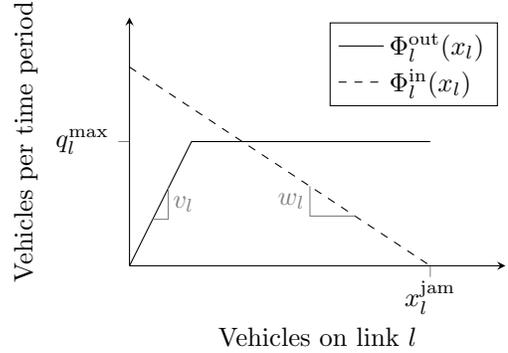
An input signal  $u[t]$  is said to be *accepted* if there exists a trajectory generated by  $u[t]$  with initial state  $q_0 \in \mathcal{Q}^0$  that is accepting. A Rabin automaton is *deterministic* if  $|\delta(q, \sigma)| \leq 1$  for all  $q \in \mathcal{Q}, \sigma \in \Sigma$ .

Despite their similarities, Rabin automata and transition systems will take on very different roles in this paper. Transition systems will be used as finite state abstractions of piecewise affine systems (PWA) with control inputs and the observations will be the set of partitions defining the PWA. Rabin automaton will be used to generate a control strategy for the PWA and thus inputs to a Rabin automaton will be outputs of the transition system obtained from the PWA. A controller maps states of the Rabin automaton to inputs of the PWA abstraction, which in turn will generate inputs to the original PWA, see [27].

## 4. A PIECEWISE AFFINE MODEL OF FREEWAY TRAFFIC FLOW ON NETWORKS

### 4.1 Link Properties and Network Topology

A freeway network is modeled as a directed graph  $\mathcal{G} = (\mathcal{V}, \mathcal{O})$  with *junctions*  $\mathcal{V}$  and *freeway links*  $\mathcal{O}$  along with a set of *onramps*  $\mathcal{R}$  that serve as entry points into the network. For  $l \in \mathcal{O}$ , let  $\sigma(l)$  denote the head vertex



**Figure 1: Supply and demand functions for freeway links.**

of freeway link  $l$  and let  $\tau(l)$  denote the tail vertex of freeway link  $l$ . Each onramp  $l \in \mathcal{R}$  directs an exogenous input flow onto  $\mathcal{G}$  via a junction, and  $\sigma(l) \in \mathcal{V}$  for  $l \in \mathcal{R}$  denotes the entry junction for onramp  $l$ . By convention,  $\tau(l) = \emptyset$  for all  $l \in \mathcal{R}$ . Let  $\mathcal{L} = \mathcal{R} \cup \mathcal{O}$  (jointly called *links*).

Each  $l \in \mathcal{L}$  has time-varying state

$$x_l[t] \in [0, x_l^{\text{jam}}] \quad (8)$$

representing the number of vehicles on link  $l$  at time-step  $t \in \mathbb{N}$  where  $x_l^{\text{jam}}$  is the maximum number of vehicles that can occupy link  $l$ . As is standard in the transportation literature, we refer to  $x_l$  as the *density* of link  $l$ . We denote the collection of link states by  $x[t] \triangleq \{x_l[t]\}_{l \in \mathcal{L}}$ . To aid analysis and when clear from context, we also consider  $x[t] \in \mathbb{R}^{|\mathcal{L}|}$  for some enumeration of  $\mathcal{L}$ . The domain of interest is then

$$\mathcal{X} \triangleq \{x \mid x_l \in [0, x_l^{\text{jam}}] \quad \forall l \in \mathcal{L}\} \subset \mathbb{R}^{|\mathcal{L}|}. \quad (9)$$

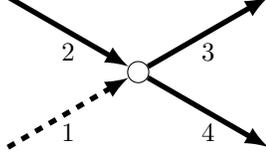
Each link  $l \in \mathcal{O}$  possesses a *demand* function  $\Phi_l^{\text{out}}(\cdot)$  and a *supply* function  $\Phi_l^{\text{in}}(\cdot)$  given by

$$\Phi_l^{\text{out}}(x_l) = \min\{v_l x_l, q_l^{\text{max}}\} \quad (10)$$

$$\Phi_l^{\text{in}}(x_l) = w_l(x_l^{\text{jam}} - x_l) \quad (11)$$

for constants  $v_l > 0, w_l > 0, q_l^{\text{max}} > 0$  that are the *free flow speed*, *congested wave speed*, and *maximum flow*, respectively [7], see Fig. 1. An interpretation of the demand  $\Phi_l^{\text{out}}(\cdot)$  is the number of vehicles per time period wishing to flow from link  $l$  to downstream links, while the supply  $\Phi_l^{\text{in}}(\cdot)$  is the number of vehicles per time period that link  $l$  can accept from upstream links. The flow through a junction is defined subsequently so that neither demand nor supply is exceeded.

For each  $v \in \mathcal{V}$ , we denote by  $\mathcal{L}_v^{\text{in}} \subset \mathcal{L}$  the set of incoming links to node  $v$  and by  $\mathcal{L}_v^{\text{out}} \subset \mathcal{L}$  the set of outgoing links, *i.e.*  $\mathcal{L}_v^{\text{in}} = \{l : \sigma(l) = v\}$  and  $\mathcal{L}_v^{\text{out}} = \{l : \tau(l) = v\}$ . We assume  $\mathcal{L}_v^{\text{in}} \neq \emptyset$  for all  $v \in \mathcal{V}$ . Furthermore, we assume  $\mathcal{L}_{\sigma(l)}^{\text{out}} \neq \emptyset$  for all  $l \in \mathcal{R}$ , *i.e.* onramps



**Figure 2: Flows through junctions with an arbitrary number of incoming links, some of which may be onramps, and outgoing links is determined by comparing the demand of the incoming links with the supply of outgoing links as in (12). Dashed links indicate a controlled onramp.**

always flow into at least one ordinary link downstream. Let  $\mathcal{O}_v^{\text{in}} \triangleq \mathcal{L}_v^{\text{in}} \cap \mathcal{O}$  and  $\mathcal{R}_v^{\text{in}} \triangleq \mathcal{L}_v^{\text{in}} \cap \mathcal{R}$ .

## 4.2 Behavior of Ordinary Links

At each junction  $v \in \mathcal{V}$ , let  $\beta_{lk}^v$  for  $l \in \mathcal{L}_v^{\text{in}}$  and  $k \in \mathcal{L}_v^{\text{out}}$  be the fraction of vehicles flowing out of link  $l$  that are routed to link  $k$ . Collectively,  $\{\beta_{lk}^v\}_{l \in \mathcal{L}_v^{\text{in}}, k \in \mathcal{L}_v^{\text{out}}}$  are the *turn ratios* at junction  $v$ . We have  $\sum_{k \in \mathcal{L}_v^{\text{out}}} \beta_{lk}^v \leq 1$  for each  $l \in \mathcal{L}_v^{\text{in}}$ , and strict inequality implies a fraction of vehicles flowing out of link  $l$  exits the network via an unmodeled offramp at the junction.

Supply of outgoing links is similarly divided among incoming links via the *supply ratios*  $\{\alpha_{lk}^v\}_{l \in \mathcal{L}_v^{\text{in}}, k \in \mathcal{L}_v^{\text{out}}}$  with the property  $\sum_{l \in \mathcal{L}_v^{\text{in}}} \alpha_{lk}^v = 1$  for all  $k \in \mathcal{L}_v^{\text{out}}$ . Such supply ratios are introduced in [16], see [17, 18] for further discussion. We then define the outflow of each freeway link  $l \in \mathcal{O}$  as follows:

$$f_l^{\text{out}}(x) \triangleq \min \left\{ \Phi_l^{\text{out}}(x_l), \min_{k \in \mathcal{L}_{\sigma(l)}^{\text{out}}} \frac{\alpha_{lk}^v}{\beta_{lk}^v} \Phi_k^{\text{in}}(x_k) \right\}. \quad (12)$$

Equation (12) defines the outflow of a link  $l$  to be the minimum of the demand at link  $l$  and the supply available to link  $l$  from any outgoing link  $k$  multiplied by the fraction of vehicles  $\alpha_{lk}$  that will flow from link  $l$  to link  $k$ , see Fig. 2.

## 4.3 Behavior of Onramps

For each  $l \in \mathcal{R}$ , we assume a prescribed constant inflow  $I_l$  enters the  $l$ th onramp queue in each time step. Onramps may be either *controlled* or *uncontrolled*. Furthermore, onramps may behave as either a *symmetric* link which is constrained by supply availability or an *asymmetric* link which is not constrained by downstream supply [6]. Asymmetric onramp models may be appropriate when a relatively minor onramp branch merges with a major mainline link, and onramp vehicles are always capable of joining the mainline unless jam density has been reached.

Let  $\mathcal{R}^c$ ,  $\mathcal{R}^u$ ,  $\mathcal{R}^s$ ,  $\mathcal{R}^a$  denote the set of controlled, uncontrolled, symmetric, and asymmetric onramps, re-

spectively. Note that  $\mathcal{R} = \mathcal{R}^c \cup \mathcal{R}^u$  and  $\mathcal{R} = \mathcal{R}^s \cup \mathcal{R}^a$ . We detail the possible onramp behaviors below.

### 4.3.1 Controlled Onramps

The flow out of each controlled onramp  $l \in \mathcal{R}^c$  is defined to be a controlled quantity

$$u_l[t] \in [0, q_l^{\text{max}}] \quad (13)$$

where  $q_l^{\text{max}} > 0$  is the maximum possible outflow of ramp  $l \in \mathcal{R}$ . Let

$$\mathcal{U} \triangleq \{\{u_l\}_{l \in \mathcal{R}^c} \mid u_l \in [0, q_l^{\text{max}}] \forall l \in \mathcal{R}^c\}. \quad (14)$$

Controlling the outflow of these links to achieve a desired temporal logic property (see Section 5.1) is the main focus of this paper.

### 4.3.2 Uncontrolled Onramps

Uncontrolled onramps possess a demand function  $\Phi_l^{\text{out}}(x_l)$  having the same form (10) where  $v_l > 0$  and  $q_l^{\text{max}} > 0$  are the freeflow speed and the maximum flow, respectively, of link  $l \in \mathcal{R}^u$ . Thus uncontrolled onramps act as ordinary links with a prescribed constant inflow  $I_l$ .

### 4.3.3 Symmetric Onramps

Symmetric onramps are required to obey any downstream supply restrictions, thus for all  $l \in \mathcal{R}^s \cap \mathcal{R}^c$  we require

$$\beta_{lk} u_l \leq \alpha_{lk} \Phi_k^{\text{in}}(x_k) \text{ for all } k \in \mathcal{L}_{\sigma(l)}^{\text{out}} \quad (15)$$

and we define  $f_l^{\text{out}}(x)$  for all  $l \in \mathcal{R}^s \cap \mathcal{R}^u$  as in (12).

For onramps that are also controlled, note that (15) is an affine, state-based constraint on the input  $u_l$ . We provide a technique for incorporating a general class of such restrictions on control that encompasses (15) in Section 5.3.

### 4.3.4 Asymmetric Onramps

Asymmetric onramps are assumed to not be restricted by downstream supply constraints. Thus for controlled onramps, the only restriction on control input is given by (13), however we will see in Section 5 that we require the control to be chosen so that (8) always holds. In this paper, we assume  $\mathcal{R}^a \cap \mathcal{R}^u = \emptyset$  so that (8) is not violated by uncontrolled asymmetric onramp flows, however alternative approaches for accommodating such onramps are possible [6].

## 4.4 Network Dynamics

The number of vehicles on each onramp link then evolves according to the mass conservation equation

$$x_l[t+1] = x_l[t] + I_l - u_l[t] \quad \forall l \in \mathcal{R}^c \quad (16)$$

$$x_l[t+1] = x_l[t] + I_l - f_l^{\text{out}}(x[t]) \quad \forall l \in \mathcal{R}^u \quad (17)$$

and the number of vehicles on ordinary links evolves according to

$$\begin{aligned}
x_l[t+1] &= x_l[t] - f_l^{\text{out}}(x[t]) + \sum_{j \in \mathcal{O}_{\tau(l)}^{\text{in}}} \beta_{jl}^{\tau(l)} f_j^{\text{out}}(x[t]) \\
&+ \sum_{\substack{j \in \\ \mathcal{R}_{\tau(l)}^{\text{in}} \cup \mathcal{R}^{\text{u}}}} \beta_{jl}^{\tau(l)} f_j^{\text{out}}(x[t]) + \sum_{\substack{j \in \\ \mathcal{R}_{\tau(l)}^{\text{in}} \cup \mathcal{R}^{\text{c}}}} \beta_{jl}^{\tau(l)} u_j[t]
\end{aligned} \tag{18}$$

for all  $l \in \mathcal{O}$ . The controller synthesis described in Section 5 ensures that  $u[t] \triangleq \{u_l[t]\}_{l \in \mathcal{R}^{\text{c}}}$  is chosen such that (8) holds for all  $l \in \mathcal{L}$  for all time.

Combined, (10)–(12) coupled with the dynamics (16)–(18) produces a PWA model of traffic flow for arbitrary directed network topologies. We refer to this model as the *traffic network system* or the *traffic network model* and denote the dynamics as

$$x[t+1] = f_{\text{net}}(x[t], u[t]) \tag{19}$$

with  $f_{\text{net}}$  piecewise affine in  $x$  and  $u$ . Note that due to the definition of  $f_l^{\text{out}}$  in (12), the PWA (19) is continuous.

Let  $\mathcal{P}$  be the index set of the partitioning of  $\mathcal{X}$  induced by (10)–(18). For  $p \in \mathcal{P}$ , let  $\mathcal{X}_p$  denote the polytope corresponding to partition index  $p$ , then

$$\mathcal{X} = \bigcup_{p \in \mathcal{P}} \mathcal{X}_p. \tag{20}$$

Observe via (12) that for  $l \in \mathcal{R}^{\text{u}} \cup \mathcal{O}$ ,  $f_l^{\text{out}}(x)$  is the minimum of a demand constraint and  $|\mathcal{L}_{\sigma(l)}^{\text{out}}|$  linear supply constraints, and the demand constraint is itself the minimum of two linear constraints via (10), thus  $f_l^{\text{out}}(x)$  is the minimum of  $|\mathcal{L}_{\sigma(l)}^{\text{out}}| + 2$  linear functions. It follows that

$$|\mathcal{P}| \leq \prod_{l \in \mathcal{O} \cup \mathcal{R}^{\text{u}}} (|\mathcal{L}_{\sigma(l)}^{\text{out}}| + 2). \tag{21}$$

The input constraint (14) is itself a polytopical constraint. We will see in Section 5.3 that a computationally efficient method exists for finding the largest subset  $\mathcal{U}_p \subseteq \mathcal{U}$  such that (15) is satisfied for all  $l \in \mathcal{R}^{\text{s}} \cap \mathcal{R}^{\text{c}}$  and all  $x \in \mathcal{X}_p$  for each  $p \in \mathcal{P}$ .

The partition  $\mathcal{P}$  induces the projection map  $y(\cdot) : \mathcal{X} \rightarrow \mathcal{P}$  defined by the property that for each  $x \in \mathcal{X}$ ,  $x \in \mathcal{X}_{y(x)}$ . We ignore nonuniqueness of  $y(x)$  when  $x$  is on the boundary of two polytopical partitions, however the feedback controller synthesized in Section 5 is valid when  $y(x)$  is taken to be any partition for which  $x \in \mathcal{X}_{y(x)}$ . We refer to  $y(x)$  as the *output*, *observation*, or *measurement* at state  $x$ .

Observe that interpreting  $y(x)$  as a literal measurement available from a traffic network is reasonable. In particular, the partition set  $\mathcal{P}$  is defined by which downstream links may be supply constrained and thus re-

stricting flow from upstream links. Determining the state of freeway traffic using available sensor data is an active area of research with numerous results in the transportation literature, see *e.g.* [25] and [24].

To simplify notation, we denote by  $y[t]$  the signal of observations generated by a trajectory  $x[t]$  of the traffic network system, *i.e.*  $y[t] = y(x[t])$ .

## 5. CONTROLLER SYNTHESIS

We wish to design a feedback ramp metering controller for the discrete-time, PWA traffic network model proposed in Section 4. We assume the signal available for measurement is  $y[t]$  indicating which region of the PWA model the system state is in for all time. However, it is apparent that a single region  $\mathcal{X}_p$  of the PWA can be further partitioned into multiple regions, each with the same dynamics, to obtain a richer measurement signal  $y[t]$ . We explore this idea in detail in Section 5.5.

### 5.1 Temporal Logic Specifications for Traffic Networks

We employ the methodology proposed in [27] to synthesize a ramp metering control strategy that achieves a control objective expressible using *linear temporal logic (LTL)*. LTL formulae are expressed over the set of observations (partitions)  $\mathcal{P}$  and typically denoted by the variable  $\varphi$ . Valid LTL formulae are generated inductively using the Boolean operators  $\vee$  (disjunction) and  $\neg$  (negation) and the temporal operators  $\bigcirc$  (next) and  $\text{U}$  (until), along with the finite set of atomic propositions  $\mathcal{P}$  which correspond to the partitions of the traffic network model. From these basic blocks, very rich specifications can be produced that include derived logical and temporal operators such as  $\wedge$  (conjunction),  $\rightarrow$  (implication),  $\square$  (always),  $\diamond$  (eventually), and others, see [1] for an overview.

Recall that the partitions  $\mathcal{P}$  arise from the requirement that flows through a junction do not exceed upstream demand or downstream supply, which is captured in, *e.g.* the definition (12). Thus examples of LTL formulas possible given the PWA defined in Section 4 include (interpretations are in quotes below each formula):

- $\varphi_1 = \diamond \square (f_l^{\text{out}}(x) = \Phi_l^{\text{out}}(x) \text{ for all } l \in \mathcal{O})$   
“Eventually, the flow of vehicles exiting any ordinary link is always equal to demand (*i.e.*, not restricted by downstream supply)”
- $\varphi_2 = \square ((f_l^{\text{out}}(x) < \Phi_l^{\text{out}}(x)) \rightarrow (f_k^{\text{out}}(x) < \Phi_k^{\text{out}}(x)), l, k \in \mathcal{O})$   
“It is always the case that if link  $l$  is restricted by downstream supply, then link  $k$  must also be restricted by downstream supply”

- $\varphi_3 = (f_l^{\text{out}}(x) = q_l^{\text{max}}) \text{U} (f_l^{\text{out}} < \Phi_l^{\text{out}}(x))$ ,  $l \in \mathcal{O}$   
“The flow exiting link  $l \in \mathcal{O}$  is equal to the maximum flow until a downstream link  $l$  has inadequate supply”
- $\varphi_4 = \diamond (\neg(\beta_{lk} f_l^{\text{out}}(x) = \alpha_{lk} \Phi_k^{\text{out}}(x)))$ ,  $l, k \in \mathcal{O}$   
“Eventually, the supply of link  $k$  is not restricting the flow of vehicles exiting link  $l$ ”

As demonstrated in Sections 5.5 and 6, many other formulae are possible by refining the partitions of the PWA, thereby introducing additional atomic propositions.

When  $f_l^{\text{out}}(x) < \Phi_l^{\text{out}}(x)$  then by (12), there exists  $k \in \mathcal{L}_{\sigma(l)}^{\text{out}}$  such that  $\beta_{lk} f_l^{\text{out}}(x) = \alpha_{lk} \Phi_k^{\text{in}}(x)$ , and link  $k$  is said to be *congested*<sup>1</sup>. Thus we see that the example LTL formulae above all relate to congestion. For example,  $\varphi_1$  can be restated as “eventually all links become uncongested and remain so forever.”

As defined in Section 4.3, we assume that inflows to onramps is a constant quantity and thus the dynamics correspond to a relatively fast time scale. For example, the assumed constant inflows may be valid during *e.g.* rush hour, but not for long periods. Thus temporal logic specifications are interpreted to hold as long as the dynamics remain valid, and therefore “forever” may mean “until rush hour ends.” Indeed, freeway traffic control is often not necessary except during periods of high demand.

For any LTL formula  $\varphi$ , there exists a deterministic Rabin automaton  $\mathcal{A}$  such that the only signals accepted by  $\mathcal{A}$  are output signals that satisfy  $\varphi$ , see [1] for an overview. Rabin automata are similar to the Büchi automata that are more commonly found in the literature, however not all LTL formulae can be transformed into a *deterministic* Büchi automaton and determinism of the specification automaton is important for controller synthesis [27]. The tool `ltl2sdstar` [13] converts an LTL specification into a deterministic Rabin automaton by first converting the specification to a nondeterministic Büchi automaton using, *e.g.*, the tool `ltl2ba` [5].

## 5.2 Controller Definition

Given an LTL specification  $\varphi$  defined over the partitions  $\mathcal{P}$ , we wish to find a set of partitions  $\mathcal{P}^0 \subset \mathcal{P}$  and a controller that chooses input  $u \in \mathcal{U}$  such that  $\varphi$  is satisfied when the initial condition  $x[0]$  satisfies

$$x[0] \in \mathcal{X}^0 \triangleq \bigcup_{p \in \mathcal{P}^0} \mathcal{X}_p \quad (22)$$

<sup>1</sup>Various definitions of congestion exist in the literature. For example, link  $l$  is sometimes said to be *congested* if  $x_l > x_l^{\text{crit}}$  where  $x_l^{\text{crit}}$  is the density such that  $\Phi_l^{\text{out}}(x_l^{\text{crit}}) = \Phi_l^{\text{in}}(x_l^{\text{crit}})$ . We use a definition that is more closely related to the dynamics of the network and define link  $l$  to be *congested* if it has inadequate supply for some upstream link’s demand.

and such that (15) is satisfied for all  $l \in \mathcal{R}^s \cap \mathcal{R}^c$  and all  $x \in \mathcal{X}_p$  for each  $p \in \mathcal{P}$ . Note that calculating the set of partitions  $\mathcal{P}^0$  from which a control strategy can be found for any initial condition satisfying (22) is part of the synthesis procedure and in general,  $\mathcal{P}^0 \neq \mathcal{P}$ . For example, for the specification  $\varphi = \square(f_l^{\text{out}}(x) = \Phi_l^{\text{out}}(x))$  for all  $l \in \mathcal{O}$ , clearly no satisfying controller can be found for any  $x[0]$  such that  $f_l^{\text{out}}(x[0]) < \Phi_l^{\text{out}}(x_l[0])$  for some  $l \in \mathcal{O}$ . In addition, other partitions may be conservatively excluded from  $\mathcal{P}^0$  due to the finite abstraction of the PWA traffic network model introduced in Section 5.4, in particular the abstraction only approximates the dynamics of the PWA system (*i.e.*, it is only a *simulation* [1]).

Given a Rabin automaton  $\mathcal{A}$  with states  $\mathcal{Q}$  corresponding to the LTL specification  $\varphi$ , a *controller* is a function  $g(\cdot, \cdot) : \mathcal{P} \times \mathcal{Q} \rightarrow \mathcal{U}$  that chooses a control input given the state of the Rabin automaton and the observation at time  $t$  for trajectories initialized within  $\mathcal{X}^0$ , *i.e.*  $g(y[t], q[t]) \in \mathcal{U}$  for all  $y[t], q[t]$  resulting from trajectories such that  $x[0] \in \mathcal{X}^0$ . We see that the controller evolves in accordance with the specification automaton and chooses the control input from a lookup table that contains an input for each PWA partition for each state of the specification automaton. The process for synthesizing this controller is described below.

## 5.3 State-based Input Constraint

In order to include input constraints of the form (15), we introduce a modification to the controller synthesis approach proposed in [27]. We begin with the following general result:

LEMMA 1. *Let  $L \in \mathbb{R}^{k \times m}$ ,  $M \in \mathbb{R}^{k \times n}$ ,  $N \in \mathbb{R}^k$  be given matrices and  $\mathcal{X}_p = \text{hull}(\{y \in V(\mathcal{X}_p)\}) \subset \mathbb{R}^n$  a given polytope with vertices  $V(\mathcal{X}_p)$ . Consider  $u \in \mathbb{R}^m$ . Then*

$$Lu \leq Mx + N \quad \text{for all } x \in \mathcal{X}_p \quad (23)$$

*if and only if*

$$Lu \leq My + N \quad \text{for all } y \in V(\mathcal{X}_p). \quad (24)$$

COROLLARY 1. *Given polytope  $\mathcal{U} \subset \mathbb{R}^m$  and polytope  $\mathcal{X}_p \subset \mathbb{R}^n$  with vertices  $\mathcal{V}(\mathcal{X}_p)$ . The set  $\mathcal{U}_p \triangleq \{u \in \mathcal{U} : Lu \leq Mx + N \forall x \in \mathcal{X}_p\}$  for given matrices  $L, M, N$  can be computed by calculating the intersection of a finite number of polytopes. Specifically,*

$$\mathcal{U}_p = \mathcal{U} \cap \bigcap_{y \in \mathcal{V}(\mathcal{X}_p)} \{u : Lu \leq My + N\}. \quad (25)$$

Taking  $\mathcal{U}$  to be the polytope given in (14), Corollary 1 implies that calculating  $\mathcal{U}_p \subseteq \mathcal{U}$  such that (15) is satisfied for all  $l \in \mathcal{R}^s \cap \mathcal{R}^c$  and all  $x \in \mathcal{X}_p$  for each  $p \in \mathcal{P}$  can be accomplished by taking a finite number of polytopic intersections.

## 5.4 Finite State Abstraction and Controller Synthesis from a Rabin Game

To synthesize a control strategy for a given LTL formula  $\varphi$ , we first obtain a finite transition system abstraction of the traffic network model which we denote by

$$\mathcal{T}_{\text{sys}} = (\mathcal{Q}_{\text{sys}}, \Sigma_{\text{sys}}, \delta_{\text{sys}}, O_{\text{sys}}, \gamma_{\text{sys}}) \quad (26)$$

and is defined as follows. We let  $\mathcal{Q}_{\text{sys}} = O_{\text{sys}} \triangleq \mathcal{P}$ , and  $\gamma_{\text{sys}} \triangleq I$  the identity function. We now determine  $\Sigma_{\text{sys}}$  and  $\delta_{\text{sys}}$ . Let

$$\mathcal{U}_p \triangleq \{u \in \mathcal{U} \mid (15) \text{ is satisfied } \forall l \in \mathcal{R}^s \cap \mathcal{R}^c, \forall x \in \mathcal{X}_p\} \quad (27)$$

for all  $p \in \mathcal{P}$ . By Corollary 1,  $\mathcal{U}_p$  is a polytope.

For each  $p \in \mathcal{P}$ ,  $u \in \mathcal{U}$ , and  $\mathcal{P}' \in 2^{\mathcal{P}}$ , define

$$\mathcal{P}_{p \rightarrow} \triangleq \{q \in \mathcal{P} \mid \exists x \in \mathcal{X}_p \text{ s.t. } f_{\text{net}}(x, u) \in \mathcal{X}_q\} \quad (28)$$

$$\mathcal{U}_p^{\mathcal{P}'} \triangleq \{u \in \mathcal{U}_p \mid \mathcal{P}_{p \rightarrow} = \mathcal{P}'\}. \quad (29)$$

In words, for each subset  $\mathcal{P}' \subseteq \mathcal{P}$ ,  $\mathcal{U}_p^{\mathcal{P}'}$  is the set of inputs in  $\mathcal{U}_p$  that will induce a transition to partition  $q$  for some  $x \in \mathcal{X}_p$  for all  $q \in \mathcal{P}'$ . Furthermore,  $\mathcal{U}_p^{\mathcal{P}'}$  is “maximal” in the sense that there does not exist  $u \in \mathcal{U}_p^{\mathcal{P}'}$ ,  $q \in \mathcal{P} \setminus \mathcal{P}'$ , and  $x \in \mathcal{X}_p$  such that  $f_{\text{net}}(x, u) \in \mathcal{X}_q$ . The set  $\mathcal{U}_p^{\mathcal{P}'}$  can be calculated using polytopic operations, see [27].

We then define

$$\Sigma_{\text{sys}} \triangleq \{\mathcal{U}_p^{\mathcal{P}'} \mid \mathcal{U}_p^{\mathcal{P}'} \neq \emptyset\}. \quad (30)$$

Strictly speaking,  $\Sigma_{\text{sys}}$  is a collection of sets, but  $\Sigma_{\text{sys}}$  is to be interpreted as input labels of  $\mathcal{T}_{\text{sys}}$  indicating for each  $\mathcal{U}_p^{\mathcal{P}'}$  in  $\Sigma_{\text{sys}}$  that partition  $p$  may be forced to transition into the set of partitions  $\mathcal{P}'$ . We make this exact by defining

$$\delta_{\text{sys}}(p, \mathcal{U}_p^{\mathcal{P}'}) = \mathcal{P}' \quad (31)$$

for each  $\mathcal{U}_p^{\mathcal{P}'} \in \Sigma_{\text{sys}}$ .

We now provide a brief summary of how a control strategy is generated using  $\mathcal{T}_{\text{sys}}$  and an LTL formula; in particular, we describe the approach used within the software package conPAS2 [27].

As described in Section 5.1, each temporal logic specification  $\varphi$  of the traffic network model is transformed into a Rabin automaton  $\mathcal{A}_\varphi = (\mathcal{Q}_\varphi, \Sigma_\varphi, \delta_\varphi, \mathcal{Q}_\varphi^0, F_\varphi)$ . We assume that the atomic propositions appearing in  $\varphi$  are a subset of  $\mathcal{Q}_{\text{sys}} = \mathcal{P}$ . If this not *a priori* the case, we may introduce additional partitions as described in Section 5.5. We have  $\Sigma_\varphi = O_{\text{sys}} = \mathcal{P}$ , and  $\mathcal{Q}_\varphi, \mathcal{Q}_\varphi^0, \delta_\varphi, F_\varphi$  are obtained via tools such as `ltl2dstar` as described in Section 5.1. It is then standard to “interconnect”  $\mathcal{T}_{\text{sys}}$  and  $\mathcal{A}_\varphi$  to construct a *product automaton* with states  $\mathcal{P} \times \mathcal{Q}_\varphi$  and inputs  $\Sigma_{\text{sys}}$ .

This product automaton is *solved* by finding an initial set  $\mathcal{P}^0 \subset \mathcal{P}$  and a function  $\pi : \mathcal{P} \times \mathcal{Q}_\varphi \rightarrow \Sigma_{\text{sys}}$  such that if the product automaton is initialized with state  $(p[0], q[0]) \in \mathcal{P}^0 \times \mathcal{Q}_\varphi^0$  and, subsequently, input  $\pi(p[t], q[t])$  is chosen at each state  $(p[t], q[t]) \in \mathcal{P} \times \mathcal{Q}_\varphi$ , then an accepting run is always produced, *i.e.* the acceptance condition of the product automaton is satisfied. The product automaton is solved by playing a *Rabin game*, see [10] for details. An important novelty of [27] is the ability to identify and prevent so-called *stuttering inputs* that arise from self-loops present in  $\mathcal{T}_{\text{sys}}$  than cannot be taken infinitely often.

Finding a solution to the product automaton implies that the LTL formula  $\varphi$  is satisfied when  $x[0] \in \mathcal{X}_p$  for any  $p \in \mathcal{P}^0$ , the Rabin automaton is initialized with  $q[0] \in \mathcal{Q}^0$ , and any input  $u \in \pi(y[t], q[t])$  is applied at time  $t$  to the original traffic network PWA model where we utilize  $\mathcal{A}_\varphi$  to “keep track” of  $q[t]$ , *i.e.*  $q[t]$  becomes the state of the controller. The output signal  $y[t]$  is as defined in Section 4.4 and therefore can be interpreted as the output of a trajectory of  $\mathcal{T}_{\text{sys}}$ . Thus we see that any feedback control law satisfying

$$g(y[t], q[t]) \in \pi(y[t], q[t]) \quad (32)$$

where  $q[t]$  is the signal generated by the Rabin automaton  $\mathcal{A}_\varphi$  ensures that the LTL formula  $\varphi$  is satisfied for any  $x[0] \in \mathcal{X}^0$  as defined in (22). Any choice satisfying (32) is acceptable, however since  $\pi(y[t], q[t]) \in \Sigma_{\text{sys}}$ ,  $\pi(y, q)$  has the form  $\mathcal{U}_p^{\mathcal{P}'}$  for some  $p \in \mathcal{P}$ ,  $\mathcal{P}' \subseteq \mathcal{P}$  for each  $y, q$ . Choosing  $g$  to be the center of the largest sphere inscribed within  $\mathcal{U}_p^{\mathcal{P}'}$  is a reasonable choice as it provides a degree of robustness and is the choice made in [27].

## 5.5 Partition Refinement

The formulation above implies  $y[t]$  as defined in Section 4.4 is the signal available for feedback control, however many specifications require a finer partitioning than the partitioning induced by the PWA dynamics of Section 4. For example, the following specifications require a finer partitioning:

- $\varphi_5 = \diamond \square \left( x_l \leq \frac{q_l^{\max}}{v_l} \right) \quad \forall l \in \mathcal{O}$   
 “Eventually, it is always the case that  $x_l \leq \frac{q_l^{\max}}{v_l}$  for all  $l \in \mathcal{O}$ ” (when  $x_l \leq \frac{q_l^{\max}}{v_l}$ , we interpret all vehicles as traveling at the freeflow speed).
- $\varphi_6 = \square ((x_l \geq C_1) \rightarrow \diamond (x_l \leq C_2)), l \in \mathcal{R}^c, C_1, C_2 \in \mathbb{R}$   
 “Always, if  $x_l \geq C_1$  then eventually  $x_l \leq C_2$ .”
- $\varphi_7 = \square (f_l^{\text{out}}(x) \geq C), l \in \mathcal{O}, C \in \mathbb{R}$   
 “Always the flow out of link  $l$  is at least  $C$ .”

- $\varphi_8 = (f_l^{\text{out}}(x) \geq C) \cup (x_k \leq C_k \ \forall k \in \mathcal{R}) \ l \in \mathcal{O}$   
 $C \in \mathbb{R}, C_k \in \mathbb{R} \ \forall k$   
 “The flow out of link  $l$  is at least  $C$  until the queue on each onramp  $k$  is below the threshold  $C_k$ .”

For example, if  $C_1$  denotes a “large” queue length and  $C_2$  a “short” queue, then  $\varphi_6$  is “always, if the queue on link  $l$  is long, eventually it becomes short.”

In addition, we can introduce a partition refinement to decrease conservativeness of the synthesis approach. Indeed, for some systems, a finite partitioning can be found that constitutes a *bisimulation* relation, [21, 23]. Here, as in [26, 27], the finite transition system we obtain is usually only a *simulation* [1] of the original PWA system.

Any partition refinement must be based on information available from sensor data. We do not discuss this requirement further in this work, but we note that determining traffic link densities via, *e.g.* flow measurements at onramps, offramps, and at intervals along freeway mainlines is a well-studied problem, see *e.g.* [25, 19, 15, 24] and references therein.

To obtain a finer partitioning, we first partition the state of each link into intervals. Let  $\mathcal{S}_l$  be an index set of partitions for link  $l$ , and let  $\{\mathcal{I}_s^l\}_{s \in \mathcal{S}_l}$  be the partitioning where  $\mathcal{I}_s^l \subset \mathbb{R}$  for all  $s \in \mathcal{S}^l$  with

$$\bigcup_{s \in \mathcal{S}_l} \mathcal{I}_s^l = [0, x_l^{\text{jam}}], \quad (33)$$

and  $\text{int}(\mathcal{I}_s^l) \cap \text{int}(\mathcal{I}_w^l) = \emptyset$  for all  $s, w \in \mathcal{S}_l, s \neq w$ . For example, if  $x_l^{\text{jam}} = 100$  for some  $l \in \mathcal{L}$ , we may introduce the partitioning  $\mathcal{S}_l \triangleq \{s_1, s_2, s_3\}$  with  $\{\mathcal{I}_{s_1}^l, \mathcal{I}_{s_2}^l, \mathcal{I}_{s_3}^l\} \triangleq \{[0, 25], [25, 75], [75, 100]\}$ , which divides link  $l$  into regions of low, medium, and high density.

We denote the cartesian product of the partition index sets by

$$\mathcal{S} \triangleq \prod_{l \in \mathcal{L}} \mathcal{S}_l. \quad (34)$$

For each  $s = \{s_l\}_{l \in \mathcal{L}} \in \mathcal{S}$  with  $s_l \in \mathcal{S}_l$ , define the function

$$\text{box}(\cdot) : \mathcal{S} \rightarrow \mathbb{R}^{|\mathcal{L}|} \quad (35)$$

$$s \mapsto \prod_{l \in \mathcal{L}} \mathcal{I}_{s_l}^l. \quad (36)$$

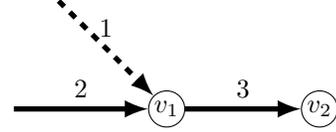
Let  $\mathcal{P}$  denote the partition induced only by the PWA dynamics as defined in (10)–(18). We then introduce a refinement of  $\mathcal{P}$ , denoted by  $\mathcal{P}'$ , as follows:

$$\mathcal{P}' = \{(s, p) \mid s \in \mathcal{S}, p \in \mathcal{P}, \text{box}(s) \cap \mathcal{X}_p \neq \emptyset\} \quad (37)$$

and the corresponding partitions are

$$\mathcal{X}_{(s,p)} \triangleq \text{box}(s) \cap \mathcal{X}_p \quad (s, p) \in \mathcal{P}'. \quad (38)$$

The dynamics of  $\mathcal{X}_{(s,p)}$  are the dynamics of  $\mathcal{X}_p$ .



**Figure 3: A network with one controlled symmetric onramp 1, one uncontrolled symmetric onramp 2, and one ordinary link 3. In Section 6.1.1, we seek a control strategy such that always  $x_1 \geq 75$  implies eventually  $x_1 \leq 25$ . In Section 6.1.2, we seek a control strategy such that eventually link 3 is always uncongested.**

In the sequel, we do not explicitly distinguish between the partition  $\mathcal{P}$  obtained from (10)–(18) and a refinement  $\mathcal{P}'$ , that is, we simply denote the final partitioning by  $\mathcal{P}$ . By an abuse of notation, we sometimes use the notation  $\mathcal{S}_l$  to denote the set of partitions of link  $l$  rather than the index set for the partitions.

## 6. EXAMPLE NETWORKS

We illustrate the controller synthesis procedure via the example network shown in Fig. 3. For this network, we consider two specifications: the first specification induces a partition refinement, and the second specification requires a partition refinement to reduce conservatism. We then suggest scalability of our approach through the example network shown in Fig. 6.

### 6.1 Example 1

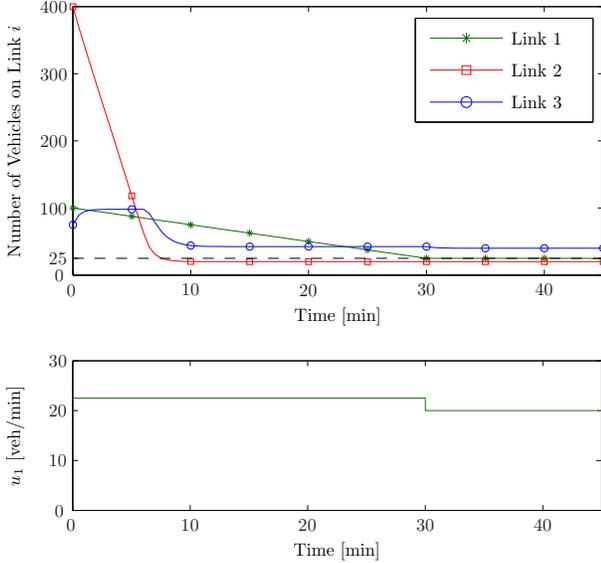
#### 6.1.1 Specification Induced Refinement

Consider the simple merge junction depicted in Fig. 3 with one controlled symmetric onramp 1, one uncontrolled symmetric onramp 2, and one ordinary link 3. We assume  $x_1^{\text{jam}} = 100, x_2^{\text{jam}} = x_3^{\text{jam}} = 400, v_2 = v_3 = 1/2, w_3 = 1/6, q_1^{\text{max}} = 40, q_2^{\text{max}} = q_3^{\text{max}} = 100, I_1 = 10, I_3 = 10, \alpha_{13} = 1/4, \alpha_{23} = 3/4, \text{ and } \beta_{13} = \beta_{23} = 1$ . We assume the system time step is 30 seconds so that, for the parameters given above, the free flow speed of link 2 and 3 is 60 miles per hour (mph), the congested wave speed of link 3 is 20 mph, and 1200 vehicles per hour enter links 1 and 3. This is in accordance with parameters given in *e.g.* [7].

This traffic network has one controlled variable,  $u_1$ , the flow exiting onramp 1. We wish to find a control strategy that ensures whenever the queue on onramp 1 is “large”, it eventually becomes “small.” If we consider  $x_1 \geq 75$  to constitute a large queue and  $x_1 \leq 25$  a small queue, we have the final specification

$$\varphi_1 = \square((x_1 \geq 75) \rightarrow \diamond(x_1 \leq 25)) \quad (39)$$

We transform condition (39) into a deterministic Rabin



**Figure 4: A trajectory of the system given in Section 6.1.1. As required by the specification, eventually  $x_1 \leq 25$  since initially  $x_1 \geq 75$ .**

automaton with two states, one of which must be visited infinitely often. Furthermore, we ensure  $x[t] \in \mathcal{X}$  for all time by restricting our attention to control inputs that render  $\mathcal{X}$  invariant, which is accomplished using only polytopic operations [27].

The PWA dynamics induce 6 partitions, however this partitioning is not sufficiently refined to consider the conditions in (39). To include (39), we segment onramp 2 into three regions:

$$\mathcal{S}_1 \triangleq \{[0, 25], [25, 75], [75, 100]\}. \quad (40)$$

We assume that the measurements available on onramp 1 allow us to determine  $s_1[t] \in \mathcal{S}_1$  such that  $x_1[t] \in s_1[t]$  for all time. We thus obtain a PWA with 18 partitions.

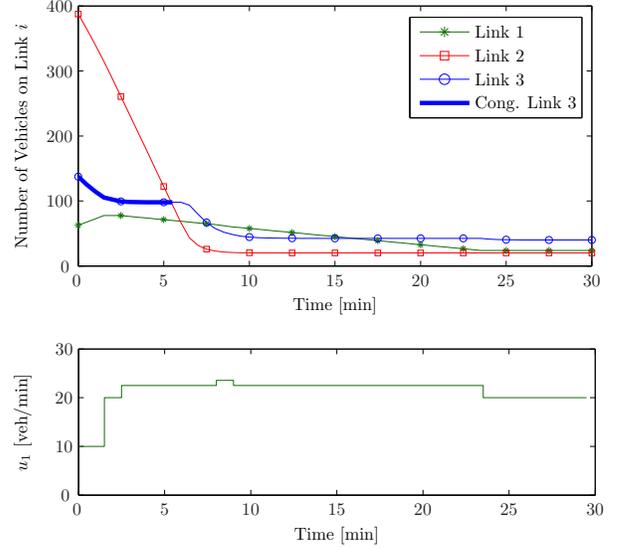
The controller synthesis finds nine regions from which the specification (39) is satisfied and computes a controller to satisfy this specification. Fig. 4 depicts a trajectory resulting from the synthesized controller. In the figure, onramp 1 initially has queue length 100, and eventually settles to a steady state queue length of 24.85, thus satisfying (39).

### 6.1.2 Partition Refinement to Reduce Conservatism

Again consider the network in Fig. 3 with parameters given in Section 6.1.1. Link 3 is congested if

$$\beta_{23}\Phi_2^{\text{out}}(x_2) > \alpha_{23}\Phi_3^{\text{in}}(x_3) \quad (41)$$

and uncongested otherwise. We wish to find a controller that ensures eventually link 3 is always uncongested,



**Figure 5: A trajectory of the system given in Section 6.1.2. The thick blue line indicates when link 3 is congested. As required by the specification, link 3 is eventually always uncongested.**

and thus our specification becomes

$$\varphi_2 = \diamond \square (\text{link 3 is uncongested}) \quad (42)$$

with the implicit requirement  $x[t] \in \mathcal{X}$  for all time. Condition (42) is encoded via a deterministic Rabin automaton with two states. Note that (42) does not directly induce a partition refinement. However, without introducing further refinement, the controller synthesis procedure does not find any satisfying initial states, thus we introduce the following link partitions:

$$\mathcal{S}_1 = \{[0, 25], [25, 50], [50, 75], [75, 100]\} \quad (43)$$

$$\mathcal{S}_2 = \mathcal{S}_3 = \{[0, 25], [25, 50], \dots, [350, 375], [375, 400]\}. \quad (44)$$

The final discrete transition system contains 1088 states, of which a controller satisfying (43) is found from 414 initial states. An example trajectory is shown in Fig. 5.

## 6.2 Example 2

We consider the traffic network shown in Fig. 6 with two onramps and three freeway links. We assume both onramps are controlled and that onramp 1 is symmetric while onramp 2 is asymmetric. Furthermore, we assume the following parameters (the time step is again 30 sec

onds):

$$x_1^{\text{jam}} = 200 \quad x_2^{\text{jam}} = 100 \quad (45)$$

$$x_l^{\text{jam}} = 400, \quad q_l^{\text{max}} = 100 \quad l \in \{3, 4, 5\} \quad (46)$$

$$v_l = \frac{1}{2}, \quad w_l = \frac{1}{6} \quad l \in \{3, 4, 5\} \quad (47)$$

$$q_1^{\text{max}} = 40 \quad q_2^{\text{max}} = 20 \quad (48)$$

$$\beta_{13} = \beta_{15} = \frac{1}{2} \quad \alpha_{34} = \frac{3}{4} \quad (49)$$

$$I_1 = 20 \quad I_2 = 10 \quad (50)$$

Turn ratios  $\beta_{lk}$  and supply ratios  $\alpha_{lk}$  not specified above are assumed to be 1. Note that we do not need to specify  $\alpha_{24}$  as onramp 2 is asymmetric.

We interpret the network in Fig. 6 as follows: links 1, 3, 4 constitute a stretch of mainline freeway, and link 5 is a freeway that diverges from the mainline. Onramp 1 is an upstream highway link whose outflow can be controlled via metering, however the flow must obey downstream supply constraints. Onramp 2 is a highway entrance onramp and can be controlled via metering. The outflow of this onramp is not subject to supply constraints, *i.e.* link 2 is an asymmetric ramp. We furthermore assume the link partitions are

$$\mathcal{S}_1 = \{[0, 50], [50, 100], [100, 150], [150, 200]\} \quad (51)$$

$$\mathcal{S}_2 = \{[0, 25], [25, 75], [75, 100]\} \quad (52)$$

$$\mathcal{S}_l = \{[0, 100], [100, 200], [200, 300], [300, 400]\} \quad l \in \{3, 4, 5\}. \quad (53)$$

Link 4 is congested if  $\beta_{34}\Phi_3^{\text{out}}(x_3) > \alpha_{34}\Phi_4^{\text{in}}(x_4)$  and is uncongested otherwise. We wish to design a control strategy with the following specification:

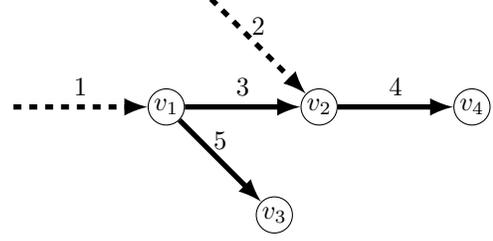
$$\varphi = \diamond \square (\text{link 4 is uncong.} \wedge (x_4 \leq 100) \wedge (x_5 \leq 100)) \quad (54)$$

$$\begin{aligned} & \wedge \square \left( [(x_1 \geq 150 \wedge x_2 \leq 25) \vee (x_1 \leq 50 \wedge x_2 \geq 75)] \right) \\ & \rightarrow \diamond [(x_1 \leq 150) \vee (x_2 \leq 75)] \end{aligned} \quad (55)$$

and (8) holds for all time. Condition (54) indicates that eventually, it is the case for all future time that link 4 is uncongested and the densities on links 4 and 5 are less than one hundred vehicles.

Condition (55) states that “whenever (onramp 1 has a large queue while onramp 2 has a small queue) or (onramp 1 has a small queue while onramp 2 has a large queue) then eventually both onramps have a medium-sized queue or smaller” where small, medium, and large queues are determined by the partitioning  $\mathcal{S}_1$  and  $\mathcal{S}_2$ . Thus (55) stipulates that eventually queues on onramps become equitable.

The piecewise affine dynamics described in Section 4 and the refinement induced by (51)–(53) results in an abstraction  $\mathcal{T}_c$  with  $|\mathcal{P}| = 720$ . Solving the correspond-



**Figure 6: A network with two controlled onramps  $\{1, 2\}$  and three freeway links  $\{3, 4, 5\}$ . Onramp 1 is symmetric while onramp 2 is asymmetric.**

ing Rabin game results in  $|\mathcal{P}^0| = 324$  and a control strategy that satisfies (54)–(55) when the system is initialized within  $\mathcal{X}^0 = \bigcup_{p \in \mathcal{P}^0} \mathcal{X}_p$ .

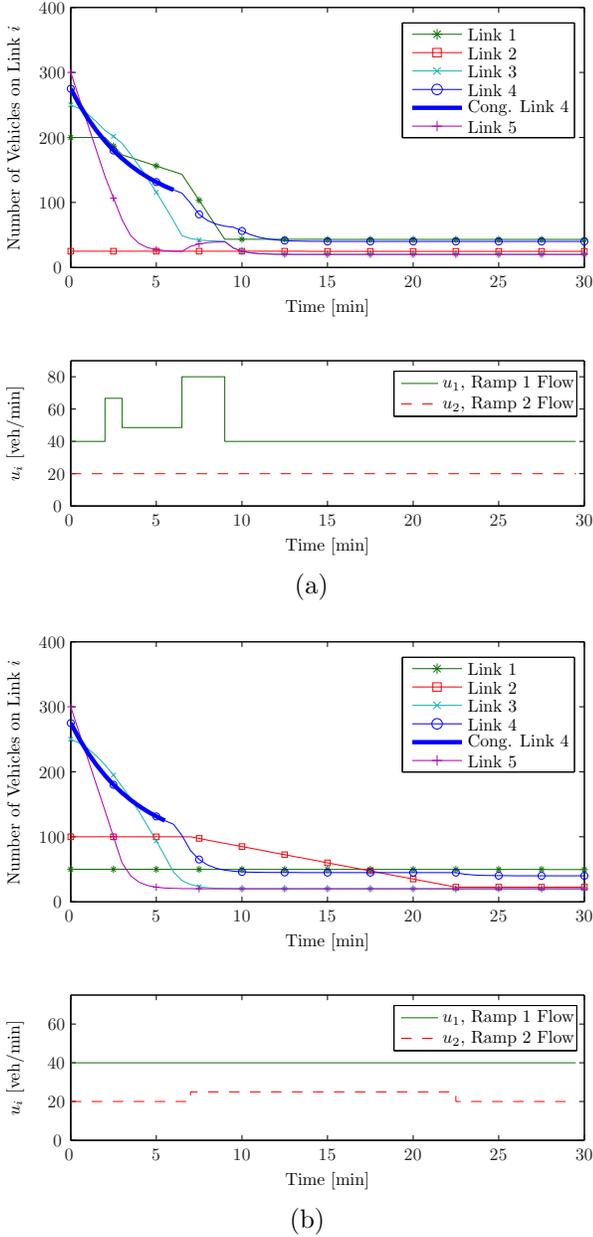
Example trajectories are shown in Fig. 7. Link 4 is originally congested, indicated by the thick blue line. Eventually, link 4 becomes uncongested and the number of vehicles on link 4 and 5 becomes less than 100. In addition, both onramps eventually have medium queues or smaller even though one initially has a large queue and the other a small queue.

The specification (54)–(55) includes competing objectives such as servicing the queues while eliminating congestion, and the controller must balance these goals. The controller adjusts the emphasis on different objectives depending on the conditions of the network, much as one would expect from a hand-tuned control strategy. As seen in Figure 7(a), initially the queue at onramp 1 is full with 200 vehicles and exogenously receives 40 vehicles per minute (vpm), thus the outflow of ramp 1 must be at least 40 vpm. To decrease the densities of the initially crowded links 3, 4, and 5, the controller applies the minimum  $u_1 = 40$  until  $t \approx 1.5$ , when the density of link 5 becomes less than 200. At this point, the controller increases  $u_1$  to decrease the queue length of onramp 1 until the density of link 5 decreases below 100 and the density of link 3 decreases below 200 at time  $t \approx 2.5$ . Then, to accelerate the decongestion of link 4,  $u_1$  drops to 50 until link 4 is decongested at  $t \approx 6$ . Next,  $u_1$  is set to its maximum value  $u_1 = 80$  until both onramps have small queue size, after which  $u_1$  is set back to 40. Setting  $u_2 = 20$  for all time maintains queue size of 25 vehicles on onramp 2, which proves sufficient.

Note that the control is naturally piecewise constant due to the finite state abstraction of the freeway network model. Such control profiles are well-suited for implementation as they do not require continuous adjustment of input flows.

## 7. CONCLUSIONS

We have presented a method for designing a ramp



**Figure 7: Two trajectories and control inputs for the network depicted in Fig. 6 that satisfies condition (54)–(55). The thick blue line indicates when link 4 is congested. In (a), the queue for onramp 1 is initially large while the queue for onramp 2 is small, and eventually both are of medium length or shorter where small, medium, and large are determined by the partitioning  $S_1$  and  $S_2$ . In (b), the opposite initial condition holds for the onramp queue lengths. In both cases, eventually link 4 becomes uncongested and eventually both link 4 and link 5 have fewer than 100 vehicles.**

metering strategy of a freeway network to meet control objectives expressible in linear temporal logic. This method therefore encompasses a rich class of controller design objectives that includes conditions for safety, reachability, liveness, *etc.* which are pertinent for the goals of traffic control but are not typically considered in the transportation literature. We begin with a piecewise-affine model of traffic flow that assumes a constant exogenous input flow for each onramp. This assumption is often reasonable during “rush hour” or times of similarly large traffic demand. Thus the temporal logic specifications for our control approach are applicable for relatively fast timescales, however this is often sufficient as ramp metering is typically only required during periods of high demand.

A number of directions for future research remain. The refinement proposed in Section 5.5 decreases conservatism but does not provide insight into the best partitioning. Furthermore, a more thorough study of scalability remains to be done. In particular, the strong structural properties of traffic networks may allow for approaches that are computationally more efficient. For example, in one time step, the density on a road link can only affect the inflow or outflow of links immediate downstream or upstream. Thus, even large networks possess dynamics with a sparsity property resulting from these local interactions. Exploiting this structure is a direction for future research.

## Acknowledgements

We thank Calin Belta for providing feedback on an early draft of the manuscript and pointing to several future research directions. This research was supported in part by the Air Force Office of Scientific Research under grant FA9550-11-1-0244.

## 8. REFERENCES

- [1] BAIER, C., AND KATOEN, J. *Principals of Model Checking*. The MIT Press, 2008.
- [2] CLARKE, E. M., GRUMBERG, O., AND PELED, D. A. *Model checking*. MIT press, 1999.
- [3] DAGANZO, C. F. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological* 28, 4 (1994), 269–287.
- [4] DAGANZO, C. F. The cell transmission model, part II: Network traffic. *Transportation Research Part B: Methodological* 29, 2 (1995), 79–93.
- [5] GASTIN, P., AND ODDOUX, D. Fast LTL to Büchi automata translation. In *Computer Aided Verification* (2001), Springer, pp. 53–65.
- [6] GOMES, G., AND HOROWITZ, R. Optimal freeway ramp metering using the asymmetric cell

- transmission model. *Transportation Research Part C: Emerging Technologies* 14, 4 (2006), 244–262.
- [7] GOMES, G., HOROWITZ, R., KURZHANSKIY, A. A., VARAIYA, P., AND KWON, J. Behavior of the cell transmission model and effectiveness of ramp metering. *Transportation Research Part C: Emerging Technologies* 16, 4 (2008), 485–513.
- [8] HEEMELS, W. P., DE SCHUTTER, B., AND BEMPORAD, A. Equivalence of hybrid dynamical models. *Automatica* 37, 7 (2001), 1085–1091.
- [9] HEGYI, A., SCHUTTER, B. D., AND HELLENDORF, H. Model predictive control for optimal coordination of ramp metering and variable speed limits. *Transportation Research Part C: Emerging Technologies* 13, 3 (2005), 185 – 209.
- [10] HORN, F. Street games on finite graphs. *Proc. 2nd Workshop Games in Design Verification (GDV)* (2005).
- [11] HOU, Z., XU, J.-X., AND YAN, J. An iterative learning approach for density control of freeway traffic flow via ramp metering. *Transportation Research Part C: Emerging Technologies* 16, 1 (2008), 71–97.
- [12] JIANG, S., AND KUMAR, R. Supervisory control of discrete event systems with CTL\* temporal logic specifications. *SIAM Journal on Control and Optimization* 44, 6 (2006), 2079–2103.
- [13] KLEIN, J., AND BAIER, C. Experiments with deterministic  $\omega$ -automata for formulas of linear temporal logic. *Theoretical Computer Science* 363, 2 (2006), 182–195.
- [14] KLOETZER, M., AND BELTA, C. Dealing with nondeterminism in symbolic control. In *Hybrid Systems: Computation and Control*. Springer, 2008, pp. 287–300.
- [15] KURZHANSKIY, A. A., AND VARAIYA, P. Active traffic management on road networks: A macroscopic approach. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 368, 1928 (2010), 4607–4626.
- [16] LEBACQUE, J. The Godunov scheme and what it means for first order traffic flow models. In *International symposium on transportation and traffic theory* (1996), pp. 647–677.
- [17] LEBACQUE, J. Intersection modeling, application to macroscopic network traffic flow models and traffic management. In *Traffic and Granular Flow'03*. Springer, 2005, pp. 261–278.
- [18] LEBACQUE, J., AND KHOSHYARAN, M. First order macroscopic traffic flow models for networks in the context of dynamic assignment. In *Transportation Planning*. Springer, 2004, pp. 119–140.
- [19] MURALIDHARAN, A., AND HOROWITZ, R. Imputation of ramp flow data for freeway traffic simulation. *Transportation Research Record: Journal of the Transportation Research Board* 2099 (Sept. 2009), 58–64.
- [20] PAPAGEORGIOU, M., AND KOTSIALOS, A. Freeway ramp metering: An overview. In *Intelligent Transportation Systems, 2000. Proceedings. 2000 IEEE* (2000), IEEE, pp. 228–239.
- [21] PAPPAS, G. J. Bisimilar linear systems. *Automatica* 39, 12 (2003), 2035 – 2047.
- [22] SMARAGDIS, E., PAPAGEORGIOU, M., AND KOSMATOPOULOS, E. A flow-maximizing adaptive local ramp metering strategy. *Transportation Research Part B: Methodological* 38, 3 (2004), 251–270.
- [23] TABUADA, P., AND PAPPAS, G. Linear time logic control of discrete-time linear systems. *IEEE Transactions on Automatic Control* 51, 12 (2006), 1862–1877.
- [24] THAI, J., PRODHOMME, B., AND BAYEN, A. M. State estimation for the discretized LWR PDE using explicit polyhedral representations of the Godunov scheme. In *Proceedings of the 2013 American Control Conference* (2013), pp. 2428–2435.
- [25] WANG, Y., AND PAPAGEORGIOU, M. Real-time freeway traffic state estimation based on extended Kalman filter: a general approach. *Transportation Research Part B: Methodological* 39, 2 (2005), 141–167.
- [26] YORDANOV, B., AND BELTA, C. Formal analysis of discrete-time piecewise affine systems. *IEEE Transactions on Automatic Control* 55, 12 (2010), 2834–2840.
- [27] YORDANOV, B., TŮMOVÁ, J., ČERNÁ, I., BARNAT, J., AND BELTA, C. Temporal logic control of discrete-time piecewise affine systems. *IEEE Transactions on Automatic Control* 57, 6 (2012), 1491–1504.