A Provably Correct MPC Approach to Safety Control of Urban Traffic Networks

Sadra Sadraddini and Calin Belta

***Abstract*— Model predictive control (MPC) is a popular strategy for urban traffic management that is able to incorporate physical and user defined constraints. However, the current MPC methods rely on finite horizon predictions that are unable to guarantee desirable behaviors over long periods of time. In this paper we design an MPC strategy that is guaranteed to keep the evolution of a network in a desirable yet arbitrary -safe- set, while optimizing a finite horizon cost function. Our approach relies on finding a robust controlled invariant set inside the safe set that provides an appropriate terminal constraint for the MPC optimization problem. An illustrative example is included.**

I. Introduction

Traffic congestion is a major problem in many cities worldwide. Many methods for more efficient usage of existing physical infrastructure have been proposed including new strategies specialized for controlling traffic lights in an urban network [1]. With the advent of new sensing technologies and improvements in online computation capabilities, traffic responsive strategies are gaining more popularity.

Model predictive control (MPC) is a powerful framework for coordinating urban traffic lights that relies on online optimization while accounting for various constraints in the system [2], [3]. However, MPC is known to exhibit "myopic" behavior that is a result of limited horizon planning. For instance, an MPC traffic light control strategy may lead a network to a state in which undesirable behaviors such as gridlock, spill-back and heavy congestion become inevitable for any future control action. A tempting resolution is elongating the prediction horizon which is often impractical from computational perspective. Some control strategies [3]–[5] have proposed enhancing the control architecture with additional layers that try to detect and avoid undesirable behaviors such as spill-back. However, these methods largely rely on heuristics rather than formal and verifiable measures.

MPC closed-loop strategies that are guaranteed to satisfy a set of constraints are studied extensively in the control theory literature [6]. Using set invariance theories [7] and terminal constraints, MPC strategies have been developed that are able to address stability issues while restricting the system trajectory to a convex safe set. However, applying similar methods to urban traffic models is impractical if not impossible due to the complexity of the constraints, controls and uncertainties. Furthermore, a critical bottleneck in the set invariance theories is the inability to deal with non-convex safe sets [8] [9]. Those approaches are computationally intractable for traffic models which have to take into consideration finite capacity roads, discrete controls (traffic lights), uncertain exogenous inputs (vehicular arrival rates) and non-convex safe sets that are able to specify desirable behaviors such as avoidance of spill-backs and gridlocks

In this paper, we wish to design an MPC strategy that is guaranteed to confine the evolution of an urban traffic network to a user-defined (non-convex) "safe" set. Inspired by recent advances in formal methods approaches to control theory, we propose a new method to overcome the mentioned issues, which is based on abstracting the network system model to a finite state system. Finite state representations of urban traffic networks have been recently investigated in [10], [11]. In a finite system, we can easily solve a "safety game" [12]. We prove that the safe invariant set found in the finite representation corresponds to a robust controlled invariant set in the original system that can be used as a terminal constraint for the MPC optimization problem. We are thus able to guarantee that the evolution of the system remains in the safe set, while planning ahead for optimality. Similar to works in [2], [13], we formulate the MPC optimization problem as a mixed integer linear programming (MILP) problem. We also argue that now being able to use small prediction horizons, we also can rely on full enumeration of possible controls to find the best future control plan instead of solving a possibly large MILP.

The network model and finite abstraction section of this work is almost identical to the work in [11]. However, there are clear differences between the control strategies. The authors in [11] find a control strategy that satisfies a linear temporal logic (LTL) specification directly from solving a Rabin game [14] on a finite state graph, without any attempt to consider optimality. However, we find an optimization based control strategy in the original continuous system subject to the constraints we find in the abstract finite state system. While LTL control in a traffic network typically requires the user to rigorously specify the desired behavior of each road and intersection, MPC naturally selects a (sub)optimal policy in the absence of user-defined constraints.

II. Traffic Network Model

The urban traffic network model used in this paper is adopted from the discrete-time fluid-like flow model in [11]. The network consists of a set of links denoted by $\mathcal{L}$ and a set of intersections denoted by $\mathcal{V}$. Each $l \in \mathcal{L}$ is a oneway link

The authors are with the Department of Mechanical Engineering, Boston University, Boston, MA 02215 {sadra,cbelta}@bu.edu.

This work was partially supported by the NSF under grants CPS-1446151 and CMMI-1400167.

from *tail* intersection $\tau(l) \in \mathcal{V} \cup \emptyset$ toward *head* intersection $\eta(l) \in \mathcal{V}$. The links that flow out of the network are not explicitly modeled. For each link l, the set of *downstream* links is defined as $\mathcal{L}_l^{down} := \{k \in \mathcal{L} : \tau(k) = \eta(l)\}$. Similarly, the set of *upstream* links is $\mathcal{L}_l^{up} := \{i \in \mathcal{L} : \eta(i) = \tau(l)\}$, and the set of *adjacent* links is $\mathcal{L}_l^{adj} := \{j \in \mathcal{L} : \tau(j) = \tau(l)\}$. Links in $\mathcal{L}_l^{local} := \{\mathcal{L}_l^{up} \cup \mathcal{L}_l^{down} \cup \mathcal{L}_l^{adj}\}$ are local to l. The number of vehicles on link l at time $t \in \mathbb{Z}_{\geq 0}$ is denoted by $x_l[t] \in [0, x_l^{cap}]$, where x_l^{cap} is the capacity of link l. The network state at time t is the vector representation $x[t] = \{x_l\}_{l \in \mathcal{L}} \in \mathcal{X} \subset \mathbb{R}^n$ where $n = |\mathcal{L}|$ is the number of links and

$$\mathcal{X} = \prod_{l \in \mathcal{L}} [0, x_l^{cap}]. \tag{1}$$

The vehicular flow of link l at time t is controlled by a binary decision denoted by $u_l[t] \in \{0, 1\}$, where values 0 and 1 represent the red and green traffic lights, respectively. The control decision combining all traffic lights at time t is $u[t] = \{u_l\}_{l \in \mathcal{L}} \in \mathcal{U} \subset \{0, 1\}^n$, where $\mathcal{U}$ is the set of admissible combinations of traffic lights, which is defined with respect to the traffic conventions. For instance, the green/green traffic light combination for two perpendicular links l_1 and l_2 that have a common head intersection is excluded by adding the constraint $u_{l_1}[t] + u_{l_2}[t] \leq 1$, or $u_{l_1}[t] + u_{l_2}[t] = 1$. In the latter case, the red/red combination is also disallowed. Note that the set $\mathcal{U}$ is finite.

Now we describe the network dynamics. When $u_l[t] = 1$, vehicles of link l flow to its downstream links $\mathcal{L}_l^{down}$. Let β_{lk} be the ratio of vehicles turning into link $k \in \mathcal{L}_l^{down}$. Note that $\sum_{k \in \mathcal{L}_l^{down}} \beta_{lk} \leq 1$,where the inequality indicates that some vehicles may flow out of the network. The capacity available at link l to its upstream links at time t is $x_l^{cap} - x_l[t]$. Let $\alpha_{il}^{u[t]}$ be the capacity portion of link l available to link $i \in \mathcal{L}_l^{up}$ when the decision on traffic lights is $u[t]$. We also have $\sum_{i \in \mathcal{L}_l^{up}} \alpha_{il}^{u[t]} = 1$. For simple intersections, it is reasonable to assume $\alpha_{il}^{u[t]}$ is constant if $u_i[t] = 1$ and zero otherwise. Therefore, for simplicity, we drop out the "$u[t]$" superscripts from the capacity ratios in the rest of the paper. The number of vehicles flowing out of link l at time step t is given by the following equation:

$$f_l[t] = u_l[t] \min \left\{ x_l[t], c_l, \min_{k \in \mathcal{L}_l^{down}} \frac{\alpha_{lk}}{\beta_{lk}} (x_k^{cap} - x_k[t]) \right\}, \tag{2}$$

where c_l is the maximum number of vehicles that can flow out of link l in one time step. The one-step evolution of link x_l is given by:

$$\begin{aligned} x_l[t+1] &= \mathcal{F}_l(x_l^{local}[t], u[t], d_l[t]) \\ &= \min \left\{ x_l[t] - f_l[t] + \sum_{i \in \mathcal{L}_l^{up}} \beta_{il} f_i[t] + d_l[t], x_l^{cap} \right\}, \end{aligned} \tag{3}$$

where $x_l^{local} = \{x_m\}, m = \left\{l \cup \mathcal{L}_l^{local}\right\}$ and $d_l[t]$ is the number of vehicles arriving in the link l from outside of the network at time t. We also denote $d[t] = \{d_l\}_{l \in \mathcal{L}} \in \mathcal{D} \subset \mathbb{R}^n$, where $\mathcal{D}$ is assumed to be a known set. We observe that $\mathcal{F}_l$ is a piecewise affine function. The network dynamics is written in the following compact form:

$$x[t+1] = \mathcal{F}\left(x[t], d[t], u[t]\right), \tag{4}$$

where $\mathcal{F} : \mathcal{X} \times \mathcal{D} \times \mathcal{U} \to \mathcal{X}$ is a piecewise affine function.

III. Problem Formulation and Approach

In this section we formulate the problem and briefly explain the approach. First, we define the safety set as a union of hyper-rectangles in the state space $\mathcal{X}$. A hyper-rectangle $\mathcal{H} \subset \mathcal{X}$ is defined with respect to a set of *rectangular* inequalities in the form of:

$$\mathcal{H} := \{x \in \mathcal{X} \mid x_{l_i} \leq r_i\}, i = 1, \cdots, p \tag{5}$$

where $r_i \in (0, x_{l_i}^{cap})$ and $p \leq n$. A *safe set* $\mathcal{S}$ is defined as a union of hyper-rectangles:

$$\mathcal{S} := \bigcup_s \mathcal{H}_s, \ s = 1, \cdots, n_{\mathcal{S}}, \tag{6}$$

where each $\mathcal{H}_s$ is a hyper-rectangle in the form of (5). Note that the safe set is, in general, non-convex. Considering constraints of the form $x_l \leq r_l$ stems from the practical purpose that the safe set should always favor fewer number of vehicles on each link.

From a game theoretical perspective, the exogenous values for d are considered as adversarial inputs. The *safety problem* considered in this paper is finding a control strategy such that for all allowable realizations of adversarial inputs, the evolution of the system remains in the safe set. The controls that ensure safety are often not unique. For practical implementation, (sub)optimal selection of the controls subject to an appropriate cost function is also important. Since the system is complex and various constraints are present, we use MPC strategy to the find the optimal control sequence over a finite horizon. Once an optimal control sequence is found, only the current step control is applied to the system and given the new measurements at next time step, a new optimal control sequence is found accordingly. The finite horizon cost criterion considered in this paper is the total time spent (TTS) of the vehicles in the network, which is used extensively in traffic literature. It can be shown that finite horizon TTS is equivalent to the total number of vehicles [1]:

$$J := \sum_{\tau=t+1}^{t+H} \sum_{l \in \mathcal{L}} x_l[\tau],$$

where H is the length of the prediction horizon. The finite horizon control sequence starting at time t is:

$$u^H[t] := \{u[t], u[t+1], \cdots, u[t+H-1]\}.$$

Given the finite horizon exogenous input sequence $\{d[t], \cdots, d[t+H-1]\}$ and the current system state $x[t]$, one can compute, using (4), the finite horizon evolution of the system $\{x[t+1], \cdots, x[t+H]\}$.

Problem 1: Given an urban traffic network (4) and a safety set $\mathcal{S}$ in the form of (6), find a control strategy that for all allowable sequences of adversarial inputs $d[t] \in \mathcal{D}$,

the evolution of the system is guaranteed to remain in the safe set:

$$x[t] \in \mathcal{S}, \forall t \in \mathbb{Z}_{\geq 0}, \tag{7}$$

and, optimize an estimated finite horizon cost function:

$$J^e = \sum_{\tau=t+1}^{t+H} \sum_{l \in \mathcal{L}} x_l^e[\tau], \tag{8}$$

where x^e are given by the estimated (nominal) finite horizon evolution of the system.

The MPC optimization problem that includes safety only over the finite horizon is:

$$\begin{aligned} u^H[t] = \quad & \text{argmin} \sum_{\tau=t+1}^{t+H} \sum_{l \in \mathcal{L}} x_l^e[\tau] \\ \text{s.t} \quad & x[\tau] \in \mathcal{S}, \\ & x[\tau+1] = \mathcal{F}(x[\tau], d[\tau], u[\tau]), \\ & x^e[\tau+1] = \mathcal{F}(x[\tau], d^e[\tau], u[\tau]), \\ & \forall d[\tau] \in \mathcal{D}, \tau = t, \dots, t+H-1, \end{aligned} \tag{9}$$

where d^e is the estimated value for the vehicular arrivals. Finite horizon safety does not guarantee infinite horizon safety. In other words, it is possible that the MPC optimization problem becomes infeasible at some time. The key contribution of this paper is guaranteeing *recursive feasibility*, which is defined as follows.

Definition 1: An MPC problem is recursively feasible if the application of control $u[t]$ from the solution of the MPC optimization problem at time t guarantees the feasibility of the MPC optimization problem at time $t+1$.

A well known method to guarantee recursive feasibility is adding an appropriate terminal constraint [6] to the MPC problem in the form of:

$$x[t+H] \in \mathcal{T}, \tag{10}$$

where $\mathcal{T} \subseteq \mathcal{S}$ is the *terminal set*. Our approach to Problem 1 involves solutions to the two following subproblems.

Subproblem 1: (Terminal Set) Find a terminal set $\mathcal{T}$ such that adding the terminal constraint (10) to the MPC optimization problem (9) guarantees recursive feasibility.

Subproblem 2: (MPC) Find $u[t]$ by solving the optimization problem (9) with the addition of the terminal constraint (10).

Subproblem 1 is solved once and in offline fashion, while Subproblem 2 is solved online at each time step. It is also reasonable to assume that in the online implementation, more precise knowledge of values of d are available for a finite horizon. However once solving Subproblem 1, the whole set of $\mathcal{D}$ is taken into account. Our approach to Subproblem 1 concerns computing a robust controlled invariant set inside $\mathcal{S}$ that involves abstracting the system into a finite state transition system, which is explained in detail in Section IV. Our solution to Subproblem 2 is based on formulating the problem as a robust optimization problem. The translation of dynamical and set constraints to MILP is explained in Section V.

IV. Terminal Set and Invariance

This section focuses on the solution to Subproblem 1. First, we define the notion of robust controlled invariant set. Next, we provide a summary on how to abstract system (4) to a finite state transition system. We then state, and prove, that the properties of the abstract system can be used to find a solution to Subproblem 1.

A. Robust Controlled Invariant Set

Definition 2: Given the non-deterministic discrete time control system (4), the set $\Omega \subseteq \mathcal{X}$ is *robust controlled invariant* if and only if:

$$\forall x \in \mathcal{X} \ \exists u \in \mathcal{U} \ s.t. \ f(x,d,u) \in \Omega \ \forall d \in \mathcal{D}. \tag{11}$$

It can be shown that if there exists a robust control invariant set, then there exists a unique *maximal robust controlled invariant set* (MRCIS) which is the union of all possible robust control invariant sets [6].

We wish to find the robust controlled invariant set of the traffic network inside the safe set $\mathcal{S}$. The traffic network's MRCIS inside $\mathcal{S}$, which is denoted by $\mathcal{I}$, can be computed using the fixed-point iterative algorithm [6]. The main drawback of this approach is that finite termination of the iterations is not guaranteed. Furthermore, early termination results in an over-approximation of MRCIS which is undesirable. Even if MRCIS is determined in finitely many steps, performing the polyhedral operations for dynamics (4), which is piecewise affine with adversarial inputs, and a non-convex safe set $\mathcal{S}$, is computationally intractable due to the severe limitations in performing polyhedral operations like Pontryagin's difference [6] for non-convex sets (see [8]). Our approach to overcome these issues is to abstract the system as a finite state transition system. In the finite realm, implementation of the fixed-point algorithm, also known as safety game, is easier and finite termination is assured.

B. Finite State Abstraction

Definition 3: (Finite State Transition System) A (non-determistic) finite state transition system is defined as the tuple $\mathcal{FTS} = (Q, \Sigma, \delta, \Lambda, \lambda)$ where Q is a finite set of states, Σ is a finite set of symbols (controls), $\delta : Q \times \Sigma \to 2^Q$ is the transition function, Λ is a finite set of labels and $\lambda : Q \to \Lambda$ is a labeling function.

Constructing a finite state transition system $\mathcal{FTS} = (Q, \Sigma, \delta, \Lambda, \lambda)$ from the original system (4) is treated in [10], [11] and the details are not presented in this section. Instead, we summarize the main points about the properties of the abstract system and provide the necessary notation for the remaining of the paper.

Abstraction involves partitioning the state space into a finite set of hyper-rectangles denoted by Q. The state space corresponding to each link $l \in \mathcal{L}$, $[0, x_l^{cap}]$, is partitioned into N_l intervals:

$$\left\{ [0, x_l^{(1)}], (x_l^{(2)}, x_l^{(3)}], \cdots, (x_l^{(N_l-1}, x_l^{cap})] \right\}. \tag{12}$$

By performing the cartesian product of the sets from (12) for all different $l \in \mathcal{L}$, $|Q| = \prod_{l \in \mathcal{L}} N_l$ hyper-rectangles are

obtained [1] . Each abstract state $q \in Q$ uniquely represents a hyper-rectangle, denoted by $\mathcal{P}(q)$, where the function $\mathcal{P} : Q \to 2^{\mathcal{X}}$ is defined to map each abstract state into its hyper-rectangle representation inside $\mathcal{X}$. Note that these sets provide a partition of $\mathcal{X}$:

$$\bigcup_{q \in Q} \mathcal{P}(q) = \mathcal{X}, \ \mathcal{P}(q_a) \cap \mathcal{P}(q_b) = \emptyset, q_a \neq q_b.$$

The labeling function Λ is defined with respect to the safe set $\mathcal{S}$. Each $q \in Q$ is labeled as $safe$ if the whole hyper-rectangle is inside $\mathcal{S}$ and $unsafe$ otherwise. Let $Q^{\mathcal{S}} \subset Q$ represent the set of states that are labeled safe. If the intervals (12) are initialized with respect to $\mathcal{S}$, it can be shown that:

$$\mathcal{S} = \bigcup_{q \in Q^{\mathcal{S}}} \mathcal{P}(q). \tag{13}$$

In other words, no $q \in Q$ represents a hyper-rectangle that is only partially in the safe set.

The transitions are determined using dynamics (4). Since the set of controls is finite, we let $\Sigma = \mathcal{U}$. For each abstract state $q \in Q$ and control $u \in \mathcal{U}$, the set of one-step reachable abstract states, denoted by $post(q, u)$, is computed by taking into account all allowable adversarial inputs:

$$\begin{aligned} &q' \in post(q, u) \text{ if and only if} \\ &\exists x \in \mathcal{P}(q), \exists d \in \mathcal{D} \ s.t. \ \mathcal{F}(x, u, d) \in \mathcal{P}(q'). \end{aligned} \tag{14}$$

The set $post(q, u)$ often includes more than one state which results in non-determinism in the finite state transition system. The computation of $post$ operation for a piece-wise affine system with exogenous inputs requires intensive polyhedral operations. On the other hand, based on the sparsity and component-wise monotonicity properties of the traffic networks, authors in [10], [11] have introduced a computationally efficient method that, under some mild assumptions, slightly over-approximates $post(q, u)$ by the set $\overline{post}(q, u)$, $post(q, u) \subseteq \overline{post}(q, u)$. Finally, all transition relations $\delta(q, u) = \left\{q' \mid q' \in \overline{post}(q, u)\right\}$ are constructed and the finite state abstraction procedure is completed.

Proposition 1: The abstarct finite system *simulates* the original system, i.e. any transition in the original system is captured by at least one transition in the abstract system:

$$\begin{aligned} &\forall x \in \mathcal{X} \ \forall u \in \mathcal{U} \ \forall d \in \mathcal{D} \ s.t. \ x' = \mathcal{F}(x, u, d), \\ &\exists q, q' \in Q, x \in \mathcal{P}(q), x' \in \mathcal{P}(q') \ s.t. \ q' \in \delta(q, u). \end{aligned} \tag{15}$$

Proof: See [11]. ■

Note that the simulation property does not state that all trajectories in the abstract system are also present in the original system. In fact, the finite state transition system may include *spurious* trajectories that are not present in the original system. Although the presence of these transitions do not affect safety control, they introduce conservatism in optimal planning.

[1] We abuse the notation and call these sets hyper-rectangles even though not all of them contain their boundaries (see (5)).

C. MRCIS for the Abstract System

In this section, we find the abstract system's MRCIS, denoted by $Q^{\mathcal{I}}$, in the safe set $Q^{\mathcal{S}}$. This problem, also known as the "safety game" [12], is based on iteratively removing the states that are absorbed into the unsafe set for all control inputs. The procedure is outlined in Algorithm 1. We define

Algorithm 1 Procedure for MRCIS $Q^{\mathcal{I}}$ inside $Q^{\mathcal{S}}$

```
Q^I = Q^S
while Q^a ≠ ∅ do
    Q^a = ∅
    for q ∈ Q^I do
        if ∀u ∈ U δ(q, u) ⊈ Q^I then
            Q^a ← Q^a ∪ q
        end if
    end for
    Q^I ← Q^I \ Q^a
end while
return Q^I
```

the hyper-rectangle representation of $Q^{\mathcal{I}}$ as

$$\tilde{\mathcal{I}} := \bigcup_{q \in \mathcal{Q}^{\mathcal{I}}} \mathcal{P}(q). \tag{16}$$

Since $Q^{\mathcal{I}} \subseteq Q^{\mathcal{S}}$, it is clear that $\tilde{\mathcal{I}} \subseteq \mathcal{S}$.

Theorem 1: $\tilde{\mathcal{I}}$ is a robust controlled invariant set of the original system (4).

Corollary 1: The abstract system's MRCIS is a subset of the original system's MRCIS. i.e. $\tilde{\mathcal{I}} \subseteq \mathcal{I}$.

Due to the space constraints, we have omitted the proofs for the results of this paper. The interested reader is referred to the extended version [2] of this paper for the proofs.

D. Recursive Feasibility

Replacing the terminal set with a robust controlled invariant set guarantees recursive feasibility:

Theorem 2: The following MPC optimization problem:

$$\begin{aligned} u^{H,opt}[t] = \quad & \text{argmin} \sum_{\tau=t+1}^{t+H} \sum_{l \in \mathcal{L}} x_l^e[\tau] \\ \text{s.t} \quad & x[\tau] \in \mathcal{S}, \\ & x[t+H] \in \tilde{\mathcal{I}}, \\ & x[\tau+1] = \mathcal{F}(x[\tau], d[\tau], u[\tau]), \\ & x^e[\tau+1] = \mathcal{F}(x[\tau], d^e[\tau], u[\tau]), \\ & \forall d[\tau] \in \mathcal{D}, \tau = t, \dots, t+H-1. \end{aligned} \tag{17}$$

is recursively feasible.

Finally, we have found a solution to Problem 1 and we let the terminal constraint to be:

$$\mathcal{T} = \tilde{\mathcal{I}}. \tag{18}$$

[2] http://arxiv.org/abs/1602.01028

E. Discussions

Algorithm 1 may end with an empty set which, unfortunately, does not conclude that $\mathcal{I}$ is also empty. In this case, finer partitions are required to find a nonempty invariant set. Furthermore, refinement may also enlarge $\tilde{\mathcal{I}}$ if it is already nonempty, which leads to less conservative hence more control options. Note that the refinement of partitions outside $Q^{\mathcal{S}}$ is not necessary. However, the number of partitions and transitions in the abstract system grows exponentially with respect to the network size. Therefore, our approach to safety control is restricted to small networks. We also note that if the initial MPC optimization problem is infeasible, for instance x is initially outside $\mathcal{S}$, one can solve the "reachability" game [12] to guide the state of the system into $\tilde{\mathcal{I}}$ and later start implementing the MPC. It is worth to note that the reachability game may also be infeasible from some initial conditions.

V. Model Predictive Control

In this section, we provide the solution to Subproblem 2 based on MILP formulation of the optimization problem (17). We also discuss the limitations of the MILP approach.

A. Mixed Logical Representations of Traffic Networks

The traffic network dynamics (4) is a hybrid system that falls into the class of mixed logical dynamical (MLD) systems which can be encoded using mixed integer constraints [15]. Formally, we have:

Proposition 2: The traffic dynamics (4) can be formulated as a finite set of mixed integer constraints:

$$\begin{aligned} x[t+1] = \quad & \mathcal{F}\left(x[t], d[t], u[t]\right) \\ \Leftrightarrow \quad & x[t+1] + E_x x[t] + E_u u[t] \\ & + E_d d[t] + E_z z[t] + E_\delta \delta[t] \leq e, \end{aligned} \tag{19}$$

where $z[t]$ is the vector of auxiliary continuous variables, $\delta[t]$ is the vector of auxiliary binary variables, E_x, E_u, E_d, E_z, E_δ are appropriately defined constant matrices and e is a vector that is defined such that the set of mixed-integer constraints is *well posed*, i.e. for given values of $x[t], u[t]$ and $d[t]$, the feasible set of $x[t+1]$ is a single point.

B. Robust MPC

A solution to the robust optimization problem requires that the safety and terminal constraints are satisfied for all allowable adversarial inputs $d \in \mathcal{D}$. In this section, we briefly explain how to characterize the H-step reachable sets.

1) One-step reachable set: We assume the set $\mathcal{D}$ is given as a union of hyper-rectangles:

$$\mathcal{D} = \bigcup_{i_d=1} D_{i_d}, i_d = 1, \cdots, n_{\mathcal{D}}, \tag{20}$$

such that each D_{i_d} is a hyper-rectangle in the form of $\left\{d \,|\, \underline{d}^{i_d} \leq d \leq \overline{d}^{i_d}\right\}$, where the inequalities are interpreted element-wise. Note that this assumption is not restrictive as one may over-approximate any bounded set by hyper-rectangles. Given control $u[\tau]$, we wish to compute the one-step reachable set of hyper-rectangle $\prod_{l\in\mathcal{L}} [\underline{x}_l[\tau], \overline{x}_l[\tau]]$, where lower (over) bars stand for lower (upper) bounds of the values x and d, and superscript i_d stands for the values obtained from exogenous inputs in hyper-rectangle $\mathcal{D}_{i_d}$. We denote:

$$\mathcal{R}_{i_d}([\underline{x}[\tau], \overline{x}[\tau]], u[\tau]) = \prod_{l\in\mathcal{L}} \left[\underline{x}_l^{i_d}[\tau+1], \overline{x}_l^{i_d}[\tau+1]\right]. \tag{21}$$

The one step reachable set is thus given by an union of hyper-rectangles:

$$\mathcal{R}^{(1)}(\underline{x}[\tau], \overline{x}[\tau], u[\tau]) = \bigcup_{i_d=1}^{n_{\mathcal{D}}} \mathcal{R}_{i_d}([\underline{x}[\tau], \overline{x}[\tau]], u[\tau]). \tag{22}$$

2) H-step reachable set: Using (22), the H-step reachable set denoted by $\mathcal{R}^{(H)}([\underline{x}[t], \overline{x}[t]], u^H[t])$ is:

$$\bigcup_{i_d^H=1}^{n_{\mathcal{D}}} \cdots \bigcup_{i_d^1=1}^{n_{\mathcal{D}}} \mathcal{R}_{i_d^H}\left(\left[\cdots\left[\mathcal{R}_{i_d^1}([\underline{x}[t], \overline{x}[t]], u[t])\right]\cdots\right], u[t+H-1]\right), \tag{23}$$

where $\underline{x}[t] = \overline{x}[t]$. Note that we may assume bounded uncertainties in online measurements of $x[t]$ but this may cause issues with recursive feasibility guarantee for the MPC problem The number of mixed logical equations in the form of (19) required to encode the H-step reachable set is $2n_{\mathcal{D}}^H$. The MILP problem becomes quickly intractable if $n_{\mathcal{D}} > 1$.

C. Non-Convex Sets and Additional Integer Constraints

The constraints corresponding to the safe set $\mathcal{S}$ and terminal set $\mathcal{T}$ also can be encoded as mixed integer constraints. We do not explain this method as encoding non-convex sets using integer constraints is a well known straightforward procedure [16]. The main issue is that the number of integer constraints describing the sets can be very large, and hence the MILP formulation becomes impractical. One approach to overcome this issue is considering the set constraints as *lazy constraints* [17], i.e. adding them to the MILP problem if they are violated by the relaxed solution. However, this approach may still require incorporating all the constraints.

D. Discussions

The computational complexity of MILPs grow exponentially with respect to the number of integer constraints. Therefore, practical implementation of MILP in our framework is restricted to simple problems. Since the set of controls is finite, a much simpler approach is full enumeration of the all controls over H-step horizon. This approach is appropriate if the size of $\mathcal{U}$ and the horizon H are small, which is the case in small networks.

VI. Example

We implemented our methods on a simple urban traffic network. The results are presented in this section. Consider a network with 9 links and 3 intersections, as illustrated in Fig. 1. The parameters of the model are given as

$x_l^{cap} = 55, l = 1, \cdots, 6,\ x_l^{cap} = 40, l = 7, 8, 9,$
$c_l = 20, l = 1, \cdots, 6,\ c_l = 15, l = 7, 8, 9, \beta_{86} = \beta_{83} = 0.4,$
$\beta_{12} = \beta_{23} = \beta_{45} = \beta_{56} = 0.7, \beta_{72} = 0.5, \beta_{95} = 0.3$

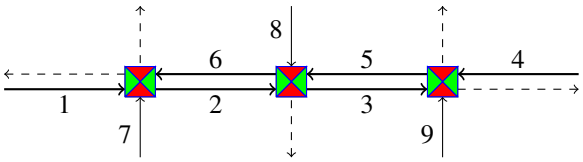

Fig. 1. The urban traffic network studied in this example

and all capacity ratios are one. The safety set is given by the boolean expression $\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4$, where

$$\begin{aligned}\phi_1 &= (x_1 \leq 36) \wedge (x_4 \leq 36),\\ \phi_2 &= (x_2 \leq 44) \vee (x_3 \leq 44),\\ \phi_3 &= (x_5 \leq 44) \vee (x_6 \leq 44),\\ \phi_4 &= (x_7 \leq 32) \vee (x_8 \leq 32) \vee (x_9 \leq 32).\end{aligned}$$

The conversion of a set defined by a boolean expression to the hyper-rectangle form of (6) is straightforward. Informally, ϕ_1 requires that the entry arterials $1, 4$ are never congested, ϕ_2 and ϕ_3 state that if the traffic on a link in the mid-corridor is heavy, the other is light. ϕ_4 prevents the entry side links $7, 8, 9$ from being congested simultaneously. We assume that each intersection has two modes of controls corresponding to the horizontal and vertical flows, $|\mathcal{U}| = 2^3 = 8$. The set characterizing exogenous inputs (arrival rates) is $\mathcal{D} = \left\{d \,|\, \mathbf{0} \leq d \leq (15, 0, 0, 15, 0, 0, 10, 10, 10)^T\right\}$.

We abstract the states corresponding to the links $1, 4, 7, 8, 9$ to 3 intervals and the states of the remaining links into 2, generating 3888 abstract states, where 1664 of them represent the safe hyper-rectangles. Computing the finite state transition system took 434 seconds on a 3 GHz dual core Macbook Pro. Solving the safety game took 31 seconds and the finite state transition system's MRCIS, $Q^{\mathcal{I}}$, consists of 1176 abstract states.

We solved the robust MPC with horizon $H = 3$ relying on full enumeration of the controls. We used (23) to find 3-step reachable set and checked the safety and the terminal constraint for all the reachable set. Fig. 2 b) shows the *robustness* of the trajectory obtained from the MPC solution simulated for 20 time steps. The robustness is computed by measuring the minimum Euclidian distance of the system's state to the safety set's boundaries [3]. The exogenous inputs were randomly chosen from $\mathcal{D}$ with uniform distribution. The values of estimated exogenous inputs for optimal planning were also randomly chosen.

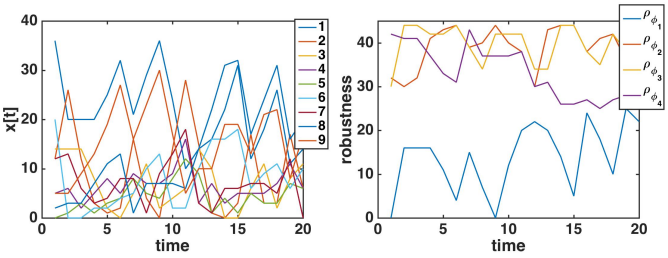

Fig. 2. a) The number of vehicles on each link b) robustness ρ measures the distance to the safety set boundaries. Notice that robustness is always above zero since the trajecory is always in the safe set.

ACKNOWLEDGEMENT

We thank Majid Zamani and Matthias Rungger from TU München for useful discussions on the material of this paper.

REFERENCES

[1] M. Papageorgiou, C. Diakaki, V. Dinopoulou, a. Kotsialos, and Y. Wang, "Review of road traffic control strategies," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043–2067, 2003.

[2] S. Lin, B. D. Schutter, Y. Xi, and H. Hellendoorn, "Model Predictive Control for urban traffic networks via MILP," *American Control Conference (ACC), 2010*, vol. 12, no. 3, pp. 846–856, 2010.

[3]The robustness defined in this section is inspired by the definition of STL robustness (see [18]).

[3] P. Mirchandani and L. Head, "A real-time traffic signal control system: Architecture, algorithms, and analysis," *Transportation Research Part C: Emerging Technologies*, vol. 9, no. 6, pp. 415–432, 2001.

[4] B. Cesme and P. G. Furth, "Self-Organizing Control Logic for Oversaturated Arterials," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2356, no. -1, pp. 92–99, 2013.

[5] E. Christofa, J. Argote, and a. Skabardonis, "Arterial Queue Spillback Detection and Signal Control Based on Connected Vehicle Technology," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2356, no. 2, pp. 61–70, 2013.

[6] E. C. Kerrigan, "Robust Constraint Satisfaction: Invariant Sets and Predictive Control," Ph.D. dissertation, University of Cambridge, 2000.

[7] F. Blanchini, "Set invariance in controla survey," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[8] S. V. Raković, P. Grieder, M. Kvasnica, D. Q. Mayne, and M. Morari, "Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 2. IEEE, 2004, pp. 1418–1423.

[9] M. Rungger, M. Mazo, and P. Tabuada, "Controller synthesis for linear systems and safe linear-time temporal logic," in *Proceedings of the 16th international conference on Hybrid systems: computation and control*. ACM, 2013, pp. 333–342.

[10] S. Coogan and M. Arcak, "Efficient finite abstraction of mixed monotone systems," in *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control, 58-67, 2015*. ACM, 2015, pp. 58–67.

[11] S. Coogan, E. A. Gol, M. Arcak, and C. Belta, "Traffic Network Control from Temporal Logic Specifications," in *arXiv*, Chicago, IL, 2014, pp. 1–21.

[12] P. Tabuada, *Verification and Control of Hybrid Systems* . Springer Science & Business Media, 2008.

[13] M. A. S. Kamal, J.-i. Imura, T. Hayakawa, A. Ohata, and K. Aihara, "Traffic Signal Control of a Road Network Using MILP in the MPC Framework," *International Journal of Intelligent Transportation Systems Research*, vol. 13, no. 2, pp. 107–118, 2014.

[14] B. Yordanov and C. Belta, "Temporal logic control of discrete-time piecewise affine systems," *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, vol. 57, no. 6, pp. 3182–3187, 2009.

[15] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[16] D. Bertsimas, J. N. Tsitsiklis, and J. Tsitsiklis, *Introduction to Linear Optimization*. Athena Scientific Belmont, MA, 1997, vol. 6.

[17] G. O. Inc., "Gurobi Optimizer reference manual," p. 572, 2014.

[18] O. Maler and D. Nickovic, "Monitoring Temporal Properties of Continuous Signals," in *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*. Springer, 2004, pp. 152 – 166.