Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Optimal Control of Mixed Logical Dynamical Systems with Linear Temporal Logic Specifications

Sertac Karaman, Ricardo G. Sanfelice, and Emilio Frazzoli

Abstract— **Recently, Linear Temporal Logic (LTL) has been employed as a tool for formal specification in dynamical control systems. With this formal approach, control systems can be designed to provably accomplish a large class of complex tasks specified via LTL. For this purpose, language generating Buchi automata with finite abstractions of dynamical systems have been used in the literature. In this paper, we take a mathematical programming-based approach to control of a broad class of discrete-time dynamical systems, called Mixed Logic Dynamical (MLD) systems, with LTL specifications. MLDs include discontinuous and hybrid piecewise discrete-time linear systems. We apply these tools for model checking and optimal control of MLD systems with LTL specifications. Our algorithms exploit Mixed Integer Linear Programming (MILP) as well as, in the appropriate setting, Mixed Integer Quadratic Programming (MIQP) techniques. Our solution approach introduces a general technique useful in representing LTL constraints as mixed-integer linear constraints.**

I. Introduction

A recent trend in control theory is to address several different complex properties using high level languages like Linear Temporal Logic (LTL) and design controllers to satisfy such high-level specifications [9], [10], [16]. LTL was introduced in the seminal paper by Pnueli [13] to reason about temporal properties of computer programs. Notable related references include [12] and [4]. Recently, LTL model checking techniques have been extended to controllable linear systems [15] and similar techniques have led to design of control systems which satisfy by construction quite complex properties given as an LTL specification [9], [10], [16]. A traditional approach in the literature is to construct two automata or finite-state machines, one to accept the language formed by all inputs that satisfy the specifications, and the other to process the language formed by all inputs that corresponds to executions of the computer program. This is typical in model checking and software verification, where checking if a computer program satisfies a specification is equivalent to checking whether the "execution" language is included in the "specification" language [5], [4].

In this paper, we propose a class of algorithms that can be used for LTL model checking, satisfiability problems, and optimal control for Mixed Logic Dynamical (MLD) systems, which include discontinuous and hybrid piecewise discrete-time linear systems. For this purpose, we introduce a class of algorithms that employ mixed integer programming techniques known as Mixed Integer Linear Programing (MILP) and Mixed Integer Quadratic Programming (MIQP). MILP has been successfully applied to the satisfiability problem for propositional logic in [3], [6] while MIQP has been employed for model predictive and optimal of a broad class of systems [2], [17]. They rely on a general technique capable of representing LTL specifications as mixed-integer linear constraints, which has applicability to other problems, like those in [7], [8]. We show that, under a finite horizon assumption, model checking of MLD systems satisfying LTL specifications is decidable. Decidability of a model checking algorithm for linear systems with LTL specifications was recently shown in [11], also under a finite horizon assumption. The algorithm in [11] uses a model checker, called LTLC, developed by the authors. Our algorithms can be implemented using off-the-shelf numerical optimization solvers and are applicable to general MLD systems with LTL specifications. To the best of our knowledge, there are no previous results on MILP-based LTL model checking.

The authors are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology
sertac,sricardo,frazzoli@mit.edu

The paper is organized as follows. Section II presents MLD systems and the LTL Language and outlines the problem formulation. In Section IV-A, the problem definition is given. Section IV is devoted to the MILP-based algorithm where Section IV-D gives an illustrative example.

A. Notation

We denote the set of discrete time instances starting from the initial time $t = 0$ and extending to T as $\mathcal{T} = \{0, 1, \ldots, T\} \subset \mathbb{Z}$. When the time instances extend to infinity, we use $\mathcal{T}^\infty$, i.e., $\mathcal{T}^\infty = \{0, 1, \ldots, T, \ldots\}$. We write $\boldsymbol{\rho}$ in bold to refer to all of the values that the variable ρ takes on $\mathcal{T}$, i.e., $\boldsymbol{\rho}$ is equivalent to the sequence $\rho(0), \rho(1), \ldots, \rho(T)$.

II. Mixed Logic Dynamical Systems with Linear Temporal Logic Specifications

A. Mixed Logical Dynamical Systems

Mixed Logical Dynamical (MLD) systems are discrete-time systems involving continuous and discrete-valued states, constraints, nonlinearities, logic statements, etc. Following [2], an MLD system is given by

$$\begin{array}{rcl} x^+ & = & A_t x + B_{1t} u + B_{2t}\delta + B_{3t} z \\ y & = & C_t x + D_{1t} u + D_{2t}\delta + D_{3t} z \\ \text{subject to} & & E_{2t}\delta + E_{3t} z \leq E_{1t} u + E_{4t} x + E_{5t}, \end{array} \tag{1}$$

where $x = [x_c' \;\; x_l']'$, $x_c \in \mathbb{R}^{n_c}$, $x_l \in \{0,1\}^{n_l}$ is the state vector, $y = [y_c' \;\; y_l']'$, $y_c \in \mathbb{R}^{p_c}$, $y_l \in \{0,1\}^{p_l}$ is the output vector, $u = [u_c' \;\; u_l']'$, $u_c \in \mathbb{R}^{m_c}$, $u_l \in \{0,1\}^{m_l}$ is the input vector, $\delta \in \{0,1\}^{r_l}$ is the vector of discrete auxiliary variables, $z \in \mathbb{R}^{r_c}$ is the vector of continuous auxiliary variables, and, for each $t \in \mathcal{T}$, A_t, B_{1t}, B_{2t}, B_{3t},

$C_t, D_{1t}, D_{2t}, D_{3t}, E_{1t}, E_{2t}, E_{3t}, E_{4t}$, and E_{5t} are the system matrices. The subindex c denotes the continuous-valued components while l denotes the discrete-valued ones. Above, x^+ denotes $x(t+1)$.

Notice that integer variables can be used to model not only the logical facts in the system, but also several nonlinearities, like piecewise linear functions, disjunctive constraints, saturation, and many others [2], [17].

Throughout this paper, we assume well posedness of MLD systems as in [2] (see Definition 1 therein) which implies that for each initial condition, solutions and outputs are unique.

B. Linear Temporal Logic

1) Preliminary Definitions: Below, we refer to the MLD system variables used in evaluating the value of a logic statement as *system variables*.

Definition II.1 (Atomic Propositions) *An atomic proposition is a statement on the system variables γ that is either* `True` *(1 or $\top$) or* `False` *(0 or $\bot$) for some given value of the systems variables.*

An atomic proposition is associated to a function π, mapping the domain Γ of the system variables into $\{0, 1\}$. The set of atomic propositions will be denoted with Π.

Definition II.2 (Interpretation, State) *An interpretation of the system variables γ is a mapping that assigns a value to each variable in the domain Γ. Accordingly, a system state is an interpretation of the system variables γ that assigns unique values to all propositions in Π.*

Any interpretation of the system variables γ can be a system state. Notice that different values of γ can correspond to the very same system state. However, it is forbidden for a system state to represent two different values of an atomic proposition. If a system state s_i assigns a proposition p value `True` then it is denoted as $s_i \Vdash p$, otherwise we denote it by $s_i \nVdash p$. Standard notation is to use $s[p]$ to indicate the value of atomic proposition p at system state s.

Definition II.3 (Transition System) *A transition system is a tuple $\mathcal{TS} = (Q, Q_0, \rightsquigarrow, \Pi, \vDash)$ where Q is a set of system states, $Q_0 \subseteq Q$ is a set of initial system states, $\rightsquigarrow \subseteq Q \times Q$ is a transition relation, Π is a set of atomic propositions, and $\vDash: Q \to 2^\Pi$ is a labeling function.*

Intuitively, the transition system identifies a relation between the system states. This relation defines the behavior of the system by posing a constraint on the system states that are reachable from a given system state $q \in Q$. However, since the state space of the system is generally very large, the atomic propositions are used as an abstraction of the large system state space and high level conditions are posed on the atomic propositions using languages like LTL. For this purpose, the transition system labels each state s with the atomic propositions that are true at s. More precisely, a transition system assigns a proposition the value `True` at a given state s_i if and only if $p \in \vDash (s_i)$; the proposition is assigned the value `False` otherwise.

Definition II.4 (Run) *For a given run σ of a transition system $\mathcal{TS}$ is an infinite sequence of states $\sigma = (s_0, s_1, \dots)$ such that $s_i \in Q$ for all $i \in \{1, 2, \dots\}$, $s_0 \in Q_0$, and $(s_i, s_{i+1}) \in \rightsquigarrow$ for all $i \in \{0, 1, 2, \dots\}$.*

Definition II.5 (Evolution) *The evolution of an atomic proposition $p \in \Pi$ in a given run $\sigma = (s_0, s_1, \dots)$ on a transition system $\mathcal{TS}$ is defined as the infinite sequence $\pi = (s_0[p], s_1[p], \dots)$.*

2) LTL Syntax: The syntax of LTL language can be defined recursively as follows. Every atomic proposition $p \in \Pi$ is an LTL formula and if ϕ and ψ are formulae then so are $\neg\phi$, $\phi \vee \psi$, $\bigcirc\phi$ and $\phi\mathcal{U}\psi$, i.e., in BNF,

$$\phi ::= p \mid \neg\phi \mid \phi \vee \phi \mid \bigcirc\phi \mid \phi\mathcal{U}\phi \tag{2}$$

where ϕ is a formula, $\neg$, $\vee$, $\bigcirc$, and $\mathcal{U}$ are the *negation*, *disjunction*, *next*, and *until* operators, respectively.

One can also define operators other than the ones that are used for constructing the grammar. Given the operators *negation* and *disjunction*, the operators *conjunction* ($\wedge$), *implication* ($\Rightarrow$), and *equivalency* ($\Leftrightarrow$) can be defined as $\phi_1 \wedge \phi_2 = \neg(\neg\phi_1 \vee \neg\phi_2)$, $\phi_1 \Rightarrow \phi_2 = \neg\phi_1 \vee \phi_2$, and $\phi_1 \Leftrightarrow \phi_2 = (\phi_1 \Rightarrow \phi_2) \wedge (\phi_2 \Rightarrow \phi_1)$, respectively. Finally, the operators *eventually* ($\Diamond$) and *always* ($\Box$) can be defined as $\Diamond\phi = \top\mathcal{U}\phi$ and $\Box\phi = \neg\Diamond\neg\phi$, respectively.

3) LTL Semantics: Given a transition system $\mathcal{TS}$, if a run σ on $\mathcal{TS}$ satisfies a formula ϕ at some state s_j, this will be denoted by $(\sigma, j) \vDash \phi$.

Let p be an atomic proposition, ϕ and ψ be any two formulae in LTL, then the semantics of LTL are defined as

$$
\begin{aligned}
(\sigma, j) \vDash p \quad & \text{iff} \quad s_j \Vdash p; & (3)\\
(\sigma, j) \vDash \neg\phi \quad & \text{iff} \quad (\sigma, j) \nvDash \phi; & (4)\\
(\sigma, j) \vDash p \vee q \quad & \text{iff} \quad (\sigma, j) \vDash p \text{ or } (\sigma, j) \vDash q; & (5)\\
(\sigma, j) \vDash \bigcirc p \quad & \text{iff} \quad (\sigma, j+1) \vDash p; & (6)\\
(\sigma, j) \vDash p\mathcal{U}q \quad & \text{iff} \quad \exists k \geq j \text{ such that } (\sigma, k) \vDash q, & (7)\\
& \qquad \text{and for all } i, j \leq i < k : (\sigma, k) \vDash p.
\end{aligned}
$$

Even though the above completely define the semantics, the following operators are used for convenience:

$$
\begin{aligned}
(\sigma, j) \vDash p \wedge q \quad & \text{iff} \quad s_j \Vdash p \text{ and } s_j \Vdash q; & (8)\\
(\sigma, j) \vDash \Box p \quad & \text{iff} \quad (\sigma, k) \vDash p \text{ for all } k \geq j; & (9)\\
(\sigma, j) \vDash \Diamond p \quad & \text{iff} \quad \exists k \geq j \text{ such that } (\sigma, k) \vDash p. & (10)
\end{aligned}
$$

Given a transition system $\mathcal{TS}$ and a run σ on $\mathcal{TS}$, σ is said to satisfy an LTL formula ϕ if and only if $(\sigma, 0) \vDash \phi$.

Definition II.6 (LTL Formula) *An LTL formula on Π is a sentence that consists of atomic propositions and operators of LTL and obeys the grammar of LTL given by (2).*

Notice that every formula ϕ is a proposition for which an evolution can be computed and denoted as $\pi_\phi = (s_0[\phi], s_1[\phi], \dots)$, for a given run $\sigma = (s_0, s_1, \dots)$.

Definition II.7 (LTL Subformula) *Given an LTL formula ϕ, a subformula of ϕ is any LTL formula which is a strict subsentence of ϕ that obeys the grammar of LTL.*

Note that the formula ϕ is not a subformula of itself. Also note that an atomic proposition has no subformula.

Definition II.8 (Height of an LTL Formula) *The height of an LTL formula ϕ is defined to be the largest number n such that: (i) ψ_1 is a subformula of ϕ; (ii) ψ_k is a subformula of ψ_{k-1} for all $k \in \{2, \dots, n\}$.*

C. Incorporating LTL specifications in MLD systems

We start by introducing several definitions that take a connective role between MLD systems, as in Section II-A, and the transition system in LTL language described in Section II-B. The following definition specifies the state of the transition system at a given time.

Definition II.9 (Valuation Function) *For a given run $\sigma = (s_1, s_2, \dots)$, a valuation function s is a mapping from the set of time instances ($\mathcal{T}$ or $\mathcal{T}^\infty$) to the set of states Q of the transition system such that $s(t) = s_i$.*

The valuation function is also used to denote the value of an atomic proposition $p \in \Pi$ at a given time t. We use the shorthand notation $s(t)[p]$ to denote the value of the proposition p at time t.

Definition II.10 (Time Evolution of a Proposition) *Given a proposition $p \in \Pi$, its time evolution is given by*

$$\boldsymbol{P}_p = (P_p^0, P_p^1, \dots, P_p^T) = (s(0)[p], s(1)[p], \dots, s(T)[p])$$

Similarly, we will use this notation for a general formula ϕ, in which case we write $\boldsymbol{P}_\phi$ to denote its time evolution.

Then, an MLD system, given as in (1), with an LTL specification given by the formula ϕ with set of atomic propositions $\Pi = \{p_1, \dots, p_m\}$ can be written as

$$x(t+1) = A_t x(t) + B_{1t} u(t) + B_{2t}\delta(t) + B_{3t} z(t), \forall t \in \mathcal{T};$$
$$y(t) = C_t x(t) + D_{1t} u(t) + D_{2t}\delta(t) + D_{3t} z(t), \forall t \in \mathcal{T};$$

subject to

$$\begin{aligned} &E_{2t}\delta(t) + E_{3t} z(t) \le E_{1t} u(t) + E_{4t} x(t) + E_{5t}, \forall t \in \mathcal{T};\\ &\boldsymbol{P}_{p_i} \in F_p(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\delta}, \boldsymbol{z}),\ \forall p \in \Pi;\\ &\boldsymbol{P}_\phi \in G_\phi(\boldsymbol{P}_{p_1}, \dots, \boldsymbol{P}_{p_m});\\ &P_\phi^0 = 1;\end{aligned}$$

where the first constraint corresponds to the MLD constraint, the second one constrains the atomic propositions according to their definitions, the third one constrains the time evolution of the proposition ϕ according to the time evolutions of the atomic propositions, and the fourth one enforces that ϕ holds `True` at the initial time. Along with the last condition, the constraint sets F_p, $p \in \Pi$, and G_ϕ enforce that the given formula is satisfied.

III. Model Checking and Optimal Control of MLD systems with LTL specifications

In this section, we define two problems for MLD systems with LTL specifications. As in [11], we impose the following finite time horizon property: every run of the transition system is such that the evolution of every atomic proposition in the given set Π reaches a final value in finite time and remains constant for all future time, i.e., for some $T' \in \mathbb{Z}$, each $p_i \in \Pi$ is such that $s(t)[p_i] = s(T')[p_i]$ for all $t > T'$. That is, for each given LTL specification ϕ, we enforce the specification $\bigcirc\bigcirc \dots \bigcirc\Box p_{ss} \wedge \phi$, which we denote by ψ_{FH}, where the next operator $\bigcirc$ appears T' times and p_{ss} is an atomic proposition that is `True` if and only if $x(t) = x(t-1)$ and `False` otherwise.

Problem III.1 (Model Checking for MLDs with LTL) *Given an MLD system as in* (1)*, an LTL formula ϕ defined on Π, determine whether or not there exist a control input $\boldsymbol{u}$ and an initial condition $x(0)$ for which the time evolution of the atomic propositions in Π satisfies the LTL formula $\psi_{FH} \wedge \phi$ while the dynamics given by* (1) *hold for all $t \in \mathcal{T}$.*

Problem III.1 is equivalent to checking the emptiness of the initial condition and control input pair on finite horizon:

$$\begin{aligned} &\mathcal{I} := \{(x(0), \boldsymbol{u}) :\\ &x(t+1) = A_t x(t) + B_{1t} u(t) + B_{2t}\delta(t) + B_{3t} z(t),\ \forall t \in \mathcal{T};\\ &E_{2t}\delta(t) + E_{3t} z(t) \le E_{1t} u(t) + E_{4t} x(t) + E_{5t}, \quad \forall t \in \mathcal{T};\\ &\boldsymbol{P}_{p_i} \in F_p(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\delta}, \boldsymbol{z}),\ \forall p \in \Pi;\\ &\boldsymbol{P}_\phi \in G_\phi(\boldsymbol{P}_{p_1}, \dots, \boldsymbol{P}_{p_m}); \quad P_\phi^0 = 1.\}\end{aligned}$$

It answers the question of whether or not a given formula is satisfiable by the given MLD system.

Let $f(\boldsymbol{x}, \boldsymbol{u})$ define the cost function.

Problem III.2 (Optimal Control of MLDs with LTL) *Given an MLD system as in* (1)*, an LTL formula ϕ defined on Π, and an initial condition $x(0)$, find a control law $\boldsymbol{u}$ such that the resulting time evolution of the atomic propositions in Π satisfy the LTL formula $\psi_{FH} \wedge \phi$ and the convex cost function $f(\boldsymbol{x}, \boldsymbol{u})$ is minimized while the dynamics given by* (1) *hold for all $t \in \mathcal{T}$.*

Hence the problem is to solve,

$$\text{minimize} \quad f(\boldsymbol{x}, \boldsymbol{u})$$

subject to

$$\begin{aligned} &x(t+1) = A_t x(t) + B_{1t} u(t) + B_{2t}\delta(t) + B_{3t} z(t),\ \forall t \in \mathcal{T};\\ &E_{2t}\delta(t) + E_{3t} z(t) \le E_{1t} u(t) + E_{4t} x(t) + E_{5t},\ \forall t \in \mathcal{T};\\ &\boldsymbol{P}_{p_i} \in F_p(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\delta}, \boldsymbol{z}),\ \forall p \in \Pi;\\ &\boldsymbol{P}_\phi \in G_\phi(\boldsymbol{P}_1, \dots, \boldsymbol{P}_m); \quad P_\phi^0 = 1.\end{aligned}$$

Notice that Problems III.1 and III.2 are quite similar to each other, and can be thought as a feasibility and optimization problem, respectively, for the same set of constraints. In other words, Problem III.2 seeks the solution with minimum cost among all feasible solutions for Problem III.1. To provide solutions to the problems, our approach is to represent the constraint sets F_p and G_ϕ as mixed-integer linear constraints. In this case, Problem III.1 reduces to checking the emptiness of a finite set of linear constraints of continuous and discrete-valued variables, i.e., a MILP feasibility problem, whereas Problem III.2 becomes an MIQP given that f is convex quadratic.

IV. A Numerically Tractable Solution via MILP/MIQP Formulation

A. Formulation of Atomic Propositions

This section is devoted to formulation of atomic propositions in a MILP framework. Examples of atomic propositions for MLD systems are presented and the corresponding constraint sets F_p are constructed. These sets are defined such that when $\boldsymbol{P}_p \in F_p$ holds then $s(t)[p] = \texttt{True}$ if $P_p^t = 1$ and $s(t)[p] = \texttt{False}$ if $P_p^t = 0$. We impose the following restriction on these constraint sets.

Assumption IV.1 *For each $p \in \Pi$, the set $F_p(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\delta}, \boldsymbol{z})$ consists of a finite set of linear constraints over the variables $\boldsymbol{x}$, $\boldsymbol{u}$, $\boldsymbol{\delta}$, and $\boldsymbol{z}$ possibly together with some other slack variables.*

All the propositions we introduce throughout this section satisfy Assumption IV.1. Perhaps the most plausible proposition is the one that indicates whether the state vector of the MLD system is in a given polyhedral subset of the state space or not. Let us define a polyhedral set as the intersection of a finite number of halfspaces. A halfspace is defined as

$$H = \{x \in \mathbb{R}^n : a'x \leq 1\}, \tag{11}$$

where $a \in \mathbb{R}^n$ is the vector normal to the hyperplane that separates the state space in two halfspaces and one of them is given by (11). We say that a halfspace is defined by a if a is a vector normal to this halfspace. Let H_i, $i \in \{1, 2, \ldots, k\}$, be halfspaces defined by $a_i \in \mathbb{R}^n$; and let $\theta_i^t \in \{0, 1\}$ for all $t \in \mathcal{T}$ be binary variables indicating that the state belongs to H_i, by assuming the value 1. Then, θ_i^t should satisfy

$$\theta_i^t \leq 1 + \tfrac{1}{M}(1 - a'x); \qquad \theta_i^t \geq \tfrac{1}{M}(1 - a'x) + \epsilon; \tag{12}$$

where M is a large enough constant and ϵ is a small number. Then, along with $0 \leq P^t \leq 1$, the variables $P^t \in \mathbb{R}$ for $t \in \mathcal{T}$ should satisfy

$$\begin{array}{lll} P^t \leq \theta_i^t & i \in \{1, \ldots, k\}, & t \in \mathcal{T}; \\ P^t \geq 1 + (\sum_{i=1}^n \theta_i^t - k) & t \in \mathcal{T}. & \end{array} \tag{13}$$

Notice that $P^t = 1$ if and only if the state vector is in the polyhedral set $\cap_i H_i$ and $P^t = 0$ otherwise (with precision ϵ on the bounday), and also that the set F_p is the set of $\boldsymbol{P}_i$ satisfying (12) and (13).

B. Formulation of LTL Sentences

Given a formula ϕ, whose time evolution is given by $\boldsymbol{P}_\phi = (P_\phi^0, \ldots, P_\phi^T)$, the formula $\phi = \neg p$ with only one negation operator, where p is an atomic proposition, the set G_ϕ can be taken to be

$$G_\phi(\boldsymbol{P}_p) = \left\{ \boldsymbol{P}_\phi : P_\phi^t = (1 - P_p^t), \quad t \in \mathcal{T} \right\}.$$

For all the other operators we present the corresponding constraints on $\boldsymbol{P}_\phi$ for the sake of brevity.

Binary conjunction was defined in (8). In the general case of conjunction of p_i for $i = 1, \ldots, k$, i.e., $\phi = \bigwedge_{i=1}^k p_i$, can be modeled with one slack variable satisfying

$$\begin{array}{ll} P_\phi^t \leq P_{p_i}^t, & i \in \{1, \ldots, k\}, \ t \in \mathcal{T}; \\ P_\phi^t \geq \sum_{i=1}^k P_{p_i}^t - (k-1), & t \in \mathcal{T}. \end{array}$$

Similar discussion applies for the binary disjunction defined by (5), where the constraints for $\phi = \bigvee_{i=1}^k p_i$ are

$$\begin{array}{lll} P_\phi^t \leq \sum_{i=1}^k P_i^t, & t \in \mathcal{T}; & \\ P_\phi^t \geq P_i^t, & i \in \{1, \ldots, k\}, & t \in \mathcal{T}. \end{array}$$

Consider the formula $\phi = \bigcirc p$, which was defined by (6).The corresponding constraints are

$$P_\phi^t = P_p^{t+1}, \quad t \in \{0, \ldots, T-1\}; \quad P_\phi^T = P_p^T.$$

In the case of the formula $\phi = \Diamond p$ with an *eventually* operator defined by (10), the corresponding constraints are

$$\begin{array}{lll} P_\phi^t \leq \sum_{\tau=t}^T P_p^\tau & t \in \mathcal{T}; & \\ P_\phi^t \geq P_p^\tau & \tau \in \{t, \ldots, T\}, & t \in \mathcal{T}. \end{array}$$

In the case of the formula $\phi = \Box p$ with an *always* operator as defined in (9), the corresponding constraints are

$$\begin{array}{lll} P_\phi^t \leq P_p^\tau & \tau \in \{t, \ldots, T\}, & t \in \mathcal{T}; \\ P_\phi^t \geq \sum_{\tau=t}^T P_p^\tau - (T-t) & t \in \mathcal{T}. & \end{array}$$

For the *until* operator we define extra slack variables in order to make the constraints linear in terms of the variables. For the formula $\phi = p\mathcal{U}q$, the corresponding constraints are

$$\begin{array}{ll} \alpha_{tj} \geq P_q^j + \sum_{\tau=t}^j P_p^\tau - (j-t+1) & j \in \{t+1, \ldots, T\}, \\ & t \in \{0, \ldots, T-1\}; \\ \alpha_{tj} \leq P_q^j & j \in \{t+1, \ldots, T\}, \ t \in \{0, \ldots, T-1\}; \\ \alpha_{tj} \leq P_p^\tau & \tau \in \{t, \ldots, j\}, \quad j \in \{t+1, \ldots, T\}, \\ & t \in \{0, \ldots, T-1\}; \\ \alpha_{tt} = P_q^t & t \in \mathcal{T}; \\ P_\phi^t \leq \sum_{j=t}^T \alpha_{tj} & t \in \mathcal{T}; \\ P_\phi^t \geq \alpha_{tj} & t \in \mathcal{T}, \quad j \in \{t, \ldots, T\}, \end{array}$$

where α_{tj} are defined for each $t \in \mathcal{T}$, $j \in \{t, \ldots, T\}$.

The algorithm for constructing sets G_ϕ for formulae of finite arbitrary height is presented in Algorithm 1. This algorithm decomposes a formula ϕ into formulae of height one recursively to build G_ϕ. Since we consider formulae of finite height, the algorithm terminates in finite time. Note that the constraint set G_ϕ is composed of a finite collection of inequalities that are linear in the decision variables and the number of constraints is polynomial in the number of operators in the formula ϕ. Moreover, all the intermediate slack variables introduced to represent the subformulae of ϕ and ϕ itself are continuous (not binary) variables. This makes the solution method computationally more tractable.

Theorem IV.2 (Completeness of Algorithm 1) *Given a set of atomic propositions $\Pi = \{p_1, \ldots, p_m\}$ and an LTL formula ϕ, let $\sigma = (s_0, s_1, \ldots, s_T, s_T, \ldots)$ such that*

Algorithm 1: Computation of set G_ϕ

```
1 $k \leftarrow 1$, $G_\phi = \emptyset$.
2 while there exists a subformula $\varphi$ of formula $\phi$ of
  height one do
3   Define the variables $P^t_{\psi_k}$ for all $t \in \{1,\ldots,T\}$.
4   Construct the set $G_\varphi$ for $\varphi$ which has only one
    operator.
5   Add the constraints $\boldsymbol{P}_{\psi_k} \in G_\varphi$ into the set $G_\phi$.
6   Update the formula $\phi$ by substituting the slack
    proposition $\psi_k$ instead of $\varphi$ in the formula $\phi$.
7   $k \leftarrow k+1$.
8 end while
```

$s(t)[p_i] = \texttt{True}$ *if* $P^t_{p_i} = 1$ *and* $s(t)[p_i] = \texttt{False}$ *if* $P^t_{p_i} = 0$ *if* $P^t_{p_i} = 0$ *for all* $p_i \in \Pi$ *and for all* $t \in \mathcal{T}$. *Let, also,* $\boldsymbol{P}_\phi$ *be such that* $\boldsymbol{P}_\phi \in G_\phi(\boldsymbol{P}_{p_1},\ldots,\boldsymbol{P}_{p_m})$ *holds. Then, for any* $t \in \mathcal{T}$, $(\sigma,t) \vDash \phi$ *if and only if* $P^t_\phi = 1$.

C. Model Checking using MILP Feasibility

Given a set of atomic propositions Π and an LTL formula defined on Π, the problem of determining the existence of a run σ for which $(\sigma,0) \vDash \psi_{FH} \wedge \phi$ holds can be posed as a mixed integer linear feasibility check. This is possible by constructing the sets F_{p_i} for all $p_i \in \Pi$ using similar methods to the ones presented in Section IV-A and G_ϕ using Algorithm 1. Then, it is left to check whether the set $\mathcal{I}$ is empty or not. Any feasible solution in this set is indeed an execution of the MLD system satisfying the LTL formula ϕ.

Theorem IV.3 (Decidability) *Suppose Assumption IV.1 holds. Then, given an LTL formula* ϕ *defined on* Π, *solving Problem III.1 is decidable.*

MILP feasibility checks are in general NP-complete, but effective algorithms exists for such purposes mainly relying on branch and bound methods [14]. Moreover, several off-the-shelf commercial optimization solvers are able to handle this problem for fairly large state space dimensions. For the examples given in this section, we employ the commercial solver CPLEX [1].

D. An illustrative example

Let us consider the following discrete-time discontinuous system with input and state constraints.

$$\begin{aligned}
x_1(t+1) &= \begin{cases} x_1(t)+x_3(t)+0.5u_1(t) & \text{if } x_1(t)\geq 1\\ x_1(t)+0.5x_3(t)+0.5u_1(t) & \text{if } x_1(t)<1,\end{cases}\\
x_2(t+1) &= x_2(t)+x_4(t)+0.5u_2(t),\\
x_3(t+1) &= x_3(t)+u_1(t),\\
x_4(t+1) &= x_4(t)+u_2(t),\\
y_1(t) &= x_1(t), \quad y_2(t)=x_2(t),
\end{aligned}$$

with $|u_1| \leq 1$, $|u_2| \leq 1$, $|x_3| \leq 1$, and $|x_4| \leq 1$.

This system corresponds to the discretization of a planar system with decoupled, double integrator dynamics in each orthogonal direction, which is typical in robotic systems. Let us consider three atomic propositions all of which correspond to a region in the state space. Let R_1 be the region which is the convex hull of $\{(1.5,10.5),(2.5,10.5),(1.5,12.5),(2.5,12.5)\}$ in the two dimensional $x_1 - x_2$ plane. Similarly, let R_2 and R_3 be the rectangular regions which are the convex hulls of $(1.2,1.5),(1.3,1.5),(1.2,2.5),(1.3,2.5)$ and $(0.5,5),(6,5),(0.5,7.5),(6,7.5)$ respectively. Let us denote the statement that indicates that the dynamical system is in region R_1 by p_1 and let us also define p_2 and p_3 in the same way for R_2 and R_3. The LTL specification we require to be true in this example is that both R_1 and R_2 will eventually be reached and R_3 will always be avoided. This specification can be written as the following LTL formula

$$\phi = (\Diamond p_1) \wedge (\Diamond p_2) \wedge (\Box \neg p_3)\,.$$

Let us first construct the corresponding MLD system considering the definitions presented in Section II-A. Consider the piecewise linear function (14). Following the method in [2], let us define a binary variable $\delta(t)$ which will be zero if $x_1(t) \geq 1$ and one if $x_1(t) < 1$. Hence $\delta(t)$ has to satisfy

$$\begin{aligned}
\delta(t) &\leq 1-(1/M)(x_1(t)-1), \quad t\in\mathcal{T};\\
\delta(t) &\geq -(1/M)(x_1(t)-1), \quad t\in\mathcal{T};
\end{aligned}$$

where M is a big enough number. Let us define another slack variable $z(t)$ which will be equal to $\delta(t)x_3(t)$. This equality can be obtained using the following linear constraints [2]

$$\begin{aligned}
z(t) \leq M\delta(t), \quad z(t) &\geq M\delta(t), \quad t\in\mathcal{T};\\
z(t) &\leq x_3(t)+M(1-\delta(t)), \quad t\in\mathcal{T};\\
z(t) &\geq x_3(t)-M(1-\delta(t)), \quad t\in\mathcal{T}.
\end{aligned}$$

Then, the dynamics for the state x_1 of the system with the variable z become

$$x_1(t+1) \quad = \quad x_1(t)-0.5z(t)+x_3(t)+0.5u_1(t)$$

with $|u_1| \leq 1$, $|u_2| \leq 1$, $|x_3| \leq 1$, and $|x_4| \leq 1$.

Next step is to construct the constraints corresponding to the atomic propositions, i.e., the sets F_1, F_2, and F_3. These sets are the same as the ones defined in Section IV-A, we will omit rewriting them here.

Let us introduce the binary variable vectors $\boldsymbol{P}_{p_1}$, $\boldsymbol{P}_{p_2}$, and $\boldsymbol{P}_{p_3}$ and employ Algorithm 1 to construct the set $G(\boldsymbol{P}_{p_1},\boldsymbol{P}_{p_2},\boldsymbol{P}_{p_3})$. In the first iteration we start with the one of the subformulae with only one operator, say $\varphi = \Diamond p_1$. Let us denote it as ψ_1. Theset of linear constraints is as follows

$$\begin{array}{ll} P^t_{\psi_1} \leq \sum_{\tau=t}^T P^\tau_{p_1} & t\in\mathcal{T};\\ P^t_{\psi_1} \geq P^\tau_{p_1} & \tau\in\{t,\ldots,T\}, t\in\mathcal{T}.\end{array} \qquad (14)$$

Now substituting ψ_1 into the formula ϕ we have $\phi = \psi_1 \wedge (\Diamond p_2) \wedge (\Box\neg p_3)$. Now we continue with another subformula of height one. Let us pick $\varphi = \Diamond p_2$ and denote it as ψ_2. Then the resulting linear constraints will be

$$\begin{array}{ll} P^t_{\psi_2} \leq \sum_{\tau=t}^T P^\tau_{p_2} & t\in\mathcal{T};\\ P^t_{\psi_2} \geq P^\tau_{p_2} & \tau\in\{t,\ldots,T\}, t\in\mathcal{T}.\end{array} \qquad (15)$$

Then after the second iteration the formula becomes $\phi = \psi_1 \wedge \psi_2 \wedge (\Box\neg p_3)$. For the third iteration let us consider the height one subformula $\varphi = \neg p_3$ which will be denoted by ψ_3. Then the linear constraints for this subformula is

$$\begin{array}{ll} P^t_{\phi_3} \leq P^\tau_{p_3} & \tau\in\{t,\ldots,T\}, t\in\mathcal{T};\\ P^t_{\psi_3} \geq \sum_{\tau=t}^T P^\tau_{p_3} - (T-t) & t\in\mathcal{T}.\end{array} \qquad (16)$$

Refining the formula we have $\phi = \psi_1 \wedge \psi_2 \wedge (\Box\psi_3)$. For the fourth iteration of the algorithm we consider the only height one formula $\varphi = \Box\psi_3$ which will be denoted by ψ_4. In this case the linear constraints are

$$P^t_{\psi_4} = (1 - P^t_p), \quad t \in \mathcal{T}. \tag{17}$$

Then the formula becomes $\phi = \psi_1 \wedge \psi_2 \wedge \psi_4$ after further refinement. The final height one formula is the ϕ itself which leads to the following linear constraints

$$\begin{array}{ll} P^t_\phi \leq P^t_{\psi_i}, & i = 1,2,4,\ t \in \mathcal{T}; \\ P^t_\phi \geq \sum_{i \in \{1,2,4\}} P^t_{\psi_i} - (3-1), & t \in \mathcal{T} \end{array} \tag{18}$$

Note that $P^t_\rho \in \mathbb{R}$, $0 \leq P^t_\rho \leq 1$, for $\rho = \psi_1, \psi_2, \psi_3, \psi_4, \phi$. Notice that the Equations (14)-(18) defines the constraint $\boldsymbol{P}_\phi \in G(\boldsymbol{P}_{p_1}, \boldsymbol{P}_{p_2}, \boldsymbol{P}_{p_3})$. Now a final constraint is to assert that the formula ϕ is true at the initial time, i.e. $P^0_\phi = 1$.

Our model checking algorithm was executed on a computer with two 2.66 GHz processors and 4GB RAM. The computation time to find a feasible solution was less than one second. Figure 1 depicts the resulting solution from $x_1(0) = x_2(0) = 0$, which was further imposed as a constraint. The solution reaches the R_2 first and then reaches R_1 while tightly avoiding R_3. Now, we add the cost function $f(\boldsymbol{x}, \boldsymbol{u}) = \sum_{t=1}^{T} \left(u_1^2(t) + u_2^2(t)\right)$ penalizing control effort. To solve the resulting MIQP problem, we employed AMPL-CPLEX. The resulting optimal solution satisfying the LTL specification is shown in Figure 2. Compared with the solution in Figure 1, the optimal solution has a smaller overshoot to the left when avoiding R_3.

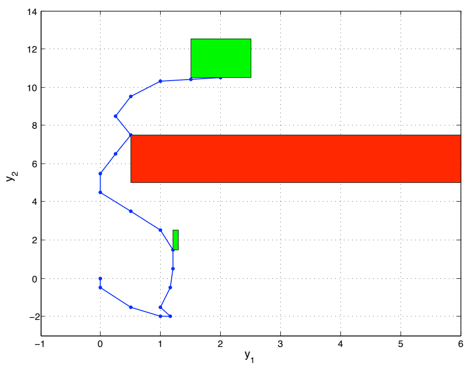

Fig. 1. Feasible solution generated by the model checker. Regions R_1 and R_2 are shown in green where as the region R_3 is shown in red. The discrete-time trajectory of the MLD systems is shown by (blue) dots.

To solve the resulting MIQP problem, we have employed AMPL-CPLEX. The resulting optimal control strategy is shown in Figure 2.

V. Conclusions

This paper presented novel MILP/MIQP based joint model checking and control design for MLD systems with LTL specifications. The algorithms presented are easily implementable using off-the-shelf MILP/MIQP solvers. We have shown that the model checking problem for MLD systems with LTL specifications considered is decidable. By means of an example, we illustrated the applicability of our results.

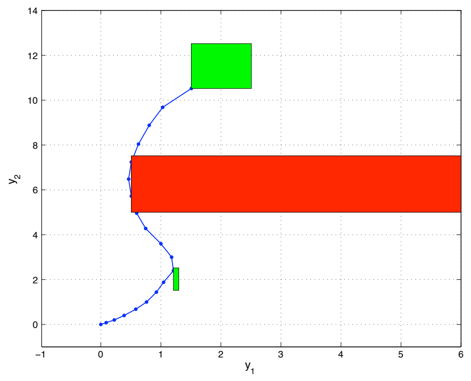

Fig. 2. Optimal control strategy in $y_1 - y_2$ plane. Dots correspond to the trajectory of the discrete-time MLD system.

Acknowledgments

This work was supported in part by the National Science Foundation, through grant #0715025, and by the Michigan/AFRL Collaborative Center on Control Sciences. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the supporting organizations.

References

[1] *CPLEX User's Manual.* ILOG, 2003.
[2] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35:407–427, 1999.
[3] C.E. Blair, R.G. Jeroslow, and J.K. Lowe. Some results and experiments in programming techniques for propositional logic. *Computers and Operations Research*, 13(5):633–645, 1986.
[4] E.M. Clarke, O. Grumberg, and D.A. Pled. *Model Checking.* MIT Press, Cambridge, MA, 1999.
[5] G.J. Holzmann. The model checker SPIN. *IEEE Transactions on Software Engineering*, 23(5):279–295, 1997.
[6] J.N. Hooker and C. Fedjki. Branch-and-cut solution of inference problems in propositional logic. *Annals of Mathematics and Aritificial Intelligence*, 1:123–139, 1990.
[7] S. Karaman and E. Frazzoli. Complex mission optimization using linear temporal logic. In *American Control Conference*, 2008.
[8] S. Karaman and E. Frazzoli. Vehicle routing using linear temporal logic: Applications to multi-UAV mission planning. In *AIAA Conference on Guidance Navigation and Control*, 2008.
[9] M. Kloetzer and C. Belta. A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control*, 53(1):287–297, 2007.
[10] M. Kloetzer and C. Belta. Temporal logic planning and control of robotic swarms by hierachical abstractions. *IEEE Transactions on Automatic Control*, 23(2):320–331, 2007.
[11] Y.M. Kwon and G. Agha. Ltlc: Linear temporal logic for control. In *to appear in Hybrid Systems: Computation and Control*, 2008.
[12] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems*. Springer-Verlag, 1992.
[13] A. Pnueli. The temporal logic of programs. In *18th annual IEEE-CS Symposium on Foundations of Computer Science*, pages 46–57, 1977.
[14] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley and Sons, 1986.
[15] P. Tabuada and G.J. Pappas. *Hybrid Systems: Computation and Control*, volume 2623/2003 of *Lecture Notes in Computer Science*, chapter Model checking LTL over controllable linear systems is decidable, pages 498–513. Springer, 2003.
[16] P. Tabuada and G.J. Pappas. Linear time logic control of discrete-time linear systems. *IEEE Transactions on Automatic Control*, 51(12):1862–1877, 2006.
[17] M.L. Tyler and M. Morari. Propositional logic in control and monitoring problems. *Automatica*, 35:565–582, 1999.