Robust Model Predictive Control for Signal Temporal Logic Synthesis

Samira S. Farahani * **Vasumathi Raman** *
Richard M. Murray **

* *Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA ({farahani,vasu}@caltech.edu)*
** *Department of Control and Dynamical Systems, California Institute of Technology, Pasadena, CA (murray@caltech.edu)*

Abstract: Most automated systems operate in uncertain or adversarial conditions, and have to be capable of reliably reacting to changes in the environment. The focus of this paper is on automatically synthesizing reactive controllers for cyber-physical systems subject to signal temporal logic (STL) specifications. We build on recent work that encodes STL specifications as mixed integer linear constraints on the variables of a discrete-time model of the system and environment dynamics. To obtain a reactive controller, we present solutions to the worst-case model predictive control (MPC) problem using a suite of mixed integer linear programming techniques. We demonstrate the comparative effectiveness of several existing worst-case MPC techniques, when applied to the problem of control subject to temporal logic specifications; our empirical results emphasize the need to develop specialized solutions for this domain.

Keywords: Hybrid systems, model predictive control, uncertain systems, robust optimization, signal temporal logic

1. INTRODUCTION

The focus of this paper is on controlling hybrid systems, i.e., those that contain both continuous and logical (discrete or switching) dynamics. Our aim is to obtain a controller that satisfies desired properties despite a potentially adversarial environment: the controller must therefore be robust to environment actions with regards to which we are uncertain. To this end, model predictive control (MPC) is an appropriate choice, since it is capable of handling constraints on both inputs and outputs in a systematic way, while having other desirable properties such as being easy to tune, enabling tracking of pre-scheduled reference signals, and being applicable to multivariable systems. MPC can incorporate linear or nonlinear hybrid models as well as discrete-time or discrete-event models of the system being controlled; see (Bemporad et al., 2002; De Schutter and van den Boom, 2001; Lazar et al., 2006; Maciejowski, 2002) for examples.

It is always important to be able to concisely and precisely specify the desired system specifications. Formal methods employing temporal logics have recently proven to be useful tools for specifying and designing controllers for intricate tasks and complicated system behavior. Algorithms for verification and synthesis of systems obeying temporal logic specifications are usually categorized based on whether they utilize a discrete abstraction of the system, and whether the environment is assumed to be deterministic or adversarial. Approaches that utilize a discrete abstraction construct discrete supervisory controllers integrated with continuous control during execution (Kloetzer and Belta, 2008; Nuzzo et al., 2014; Fainekos et al., 2009; Wongpiromsarn et al., 2012). In contrast, approaches that do not require discrete abstractions include those based on sampling-based methods (Karaman and Frazzoli, 2009), and mixed integer linear programming (MILP) encodings of temporal logic specifications (Karaman and Frazzoli, 2008, 2011; Kwon and Agha, 2008; Wolff et al., 2014; Raman et al., 2014).

In this work, we adopt the specification language of Signal Temporal Logic (STL) (Maler and Nickovic, 2004), developed specifically for monitoring the expected behavior of physical systems, including constraints on the temporal duration between events. STL allows the specification of properties of dense-time, real-valued signals, and has been applied to the analysis of several types of continuous and hybrid systems. It has also recently been used for controller synthesis in a variety of domains (Raman et al., 2014, 2015). STL has the advantage of naturally admitting a quantitative semantics which, in addition to the binary answer to the satisfaction question, provides a real number that indicates the extent to which the property is satisfied or violated.

The authors in (Raman et al., 2015) recently provided an optimization-based framework for reactive synthesis from STL specifications. They presented a counterexample-guided inductive synthesis scheme (CEGIS) for obtaining a reactive controller capable of responding to an uncertain environment. Informally, the scheme proceeds by iteratively fixing the uncertain environment input to the worst case from the previous step; however, this method is not guaranteed to converge if the environment is not finite. This motivates us to look to solutions that always terminate and still give us the guarantees we seek.

We propose three alternative approaches by reformulating the problem as worst-case MPC optimization. The objective is, as before, to maximize the quantitative satisfaction of the specified properties at each time step in a receding horizon fashion. To automatically encode specifications in an MILP, we make use of the encoding developed in (Raman et al., 2014). This procedure allows us to unify the system dynamics and the temporal logic constraints into a single MILP optimization problem. In the case of reactive synthesis, this is a robust MPC optimization problem. We will approach this problem in a number of ways and demonstrate our findings.

2. PRELIMINARIES

Much of the notation in this section mirrors that in (Raman et al., 2014) and (Raman et al., 2015), and missing details can be found in either work.

2.1 System Model

We consider a continuous-time system Σ of the form

$$\dot{x} = f(x, u, w)$$

where $x \in \mathcal{X} \subseteq (\mathbb{R}^{n_c} \times \{0,1\}^{n_l})$ are the continuous and binary/logical states, $u \in U \subseteq (\mathbb{R}^{m_c} \times \{0,1\}^{m_l})$ are the (continuous and logical) control inputs, $w \in W \subseteq (\mathbb{R}^{e_c} \times \{0,1\}^{e_l})$ are the (possibly adversarial) external inputs or disturbances, and $x_0 \in \mathcal{X}$ is the initial state. We will refer to w as the *environment* input.

As in (Raman et al., 2015), given a sampling time $\Delta t > 0$, we assume that Σ admits a discrete-time approximation Σ_d of the form

$$x(t_{k+1}) = f_d(x(t_k), u(t_k), w(t_k)) \tag{1}$$

where for all $k > 0$, $t_{k+1} - t_k = \Delta t$. A *run*

$$\xi_f = (x_0 u_0 w_0)(x_1 u_1 w_1)(x_2 u_2 w_2)...$$

of Σ_d is a sequence where $x_k = x(t_k) \in \mathcal{X}$ is the state of the system at index k, and for each $k \in \mathbb{N}$, $u_k = u(t_k) \in U$, $w_k = w(t_k) \in W$ and $x_{k+1} = f_d(x_k, u_k, w_k)$. Given $x_0 \in X$, $\mathbf{u} \in U^\omega$ and $\mathbf{w} \in W^\omega$, let $\xi_f(x_0, \mathbf{u}, \mathbf{w})$ denote the run generated following equation (1). The corresponding sequence of states (*discrete-time signal* or just *signal*), is $\mathbf{x} = x_0 x_1 \ldots$. We assume that given an initial state $x_0 \in X$, a control input sequence $\mathbf{u}^N = u_0 u_1 u_2 \ldots u_{N-1} \in U^N$ and a sequence of environment inputs $\mathbf{w}^N = w_0 w_1 w_2 \ldots w_{N-1} \in W^N$, the resulting horizon-$N$ run of a system modeled by (1), denoted by

$$\xi_f(x_0, \mathbf{u}^N, \mathbf{w}^N) = (x_0 u_0 w_0)(x_1 u_1 w_1)(x_2 u_2 w_2)...(x_N u_N w_N)$$

is unique. Finally, we introduce a generic cost function $J(\xi_f(x_0, \mathbf{u}, \mathbf{w}))$ that maps (infinite and finite) runs to $\mathbb{R}$.

2.2 STL and Robust Satisfaction of STL Formulas

We consider STL formulas defined recursively according to the grammar

$$\varphi ::= \pi^\mu \mid \neg\pi^\mu \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \Box_{[a,b]}\ \psi \mid \varphi\ \mathcal{U}_{[a,b]}\ \psi$$

where π^μ is an atomic predicate $\mathbb{R}^n \to \mathbb{B}$ whose truth value is determined by the sign of a function $\mu : \mathbb{R}^n \to \mathbb{R}$ and ψ is an STL formula. A signal $\mathbf{x} = x_0 x_1 x_2 ...$ satisfies φ, denoted by $\mathbf{x} \models \varphi$, if $(\mathbf{x}, t_0) \models \varphi$. Informally, $\mathbf{x} \models \Box_{[a,b]}\varphi$ if φ holds at every time step between a and b, and $\mathbf{x} \models \varphi\ \mathcal{U}_{[a,b]}\ \psi$ if φ holds at every time step before ψ holds, and ψ holds at some time step between a and b. Additionally, we define $\Diamond_{[a,b]}\varphi = \top\ \mathcal{U}_{[a,b]}\ \varphi$, so that $\mathbf{x} \models \Diamond_{[a,b]}\varphi$ if φ holds at some time step between a and b. A full treatment of the semantics of STL is available in (Maler and Nickovic, 2004).

Quantitative or robust semantics define a real-valued function ρ^φ of signal $\mathbf{x}$ and t such that $(\mathbf{x}, t) \models \varphi \equiv \rho^\varphi(\mathbf{x}, t) > 0$. In this work, we utilize a quantitative semantic for *space*-robustness, as defined in (Jin et al., 2013) and reproduced here:

$$\begin{aligned}
\rho^{\pi^\mu}(\mathbf{x}, t) &= \mu(x_t)\\
\rho^{\neg\pi^\mu}(\mathbf{x}, t) &= -\mu(x_t)\\
\rho^{\varphi\wedge\psi}(\mathbf{x}, t) &= \min(\rho^\varphi(\mathbf{x}, t), \rho^\psi(\mathbf{x}, t))\\
\rho^{\varphi\vee\psi}(\mathbf{x}, t) &= \max(\rho^\varphi(\mathbf{x}, t), \rho^\psi(\mathbf{x}, t))\\
\rho^{\Box_{[a,b]}\varphi}(\mathbf{x}, t) &= \min_{t' \in [t+a, t+b]} \rho^\varphi(\mathbf{x}, t')\\
\rho^{\varphi\,\mathcal{U}_{[a,b]}\psi}(\mathbf{x}, t) &= \max_{t' \in [t+a, t+b]} (\min(\rho^\psi(\mathbf{x}, t'), \min_{t'' \in [t, t']} \rho^\varphi(\mathbf{x}, t''))
\end{aligned}$$

To simplify the notation, we denote ρ^{π^μ} by ρ^μ for the remainder of this paper. The robustness of satisfaction for an arbitrary STL formula is computed recursively on the structure of the formula according to the above semantics, by propagating the values of the functions associated with each operand using min and max operators corresponding to the various STL operators. The robustness score $\rho^\varphi(\mathbf{x}, t)$ can be interpreted as *how strongly* the signal $\mathbf{x}$ satisfies φ, and its absolute value can be viewed as the distance of $\mathbf{x}$ from the set of trajectories satisfying or violating φ.

2.3 MILP Encoding for Controller Synthesis

In order to synthesize a run that satisfies an STL formula φ, we add STL constraints to a MILP formulation of the control synthesis problem, as in (Raman et al., 2014). We first represent the system trajectory as a finite sequence of states satisfying the model dynamics in equation (1). Then, we encode the formula φ with a set of MILP constraints; our encoding produces a MILP as long as the functions μ that define the predicates π^μ in φ are piecewise linear or affine.

The system constraints encode valid finite (horizon-N) trajectories for a system of the form (1), and are designed to be satisfied if and only if the trajectory $\xi_f(x_0, \mathbf{u}^N, \mathbf{w}^N)$ obeys the dynamics in (1). We employ the *robustness-based* encoding of STL constraints in the MILP, as defined in (Raman et al., 2014). As described in Section 2.2, the robustness of an STL specification φ can be computed recursively on the structure of the formula. The max and min operations can be expressed in a MILP formulation using additional binary variables and a large constant M (commonly called big-M). The interested reader is referred to (Raman et al., 2014) for details of this encoding, the gist of which follows. For a given formula φ, the MILP is extended with a real-valued variable r_k^φ and an associated set of constraints such that $r_k^\varphi > 0$ if and only if φ holds at time t_k. This is accomplished by recursively generating MILP constraints for every subformula of φ, such that $r_k^\varphi = \rho^\varphi(\mathbf{x}, t_k)$; then r_0^φ determines whether φ holds in the initial state. Maximizing or minimizing the value of

r_0^{φ} allows us to obtain a trajectory that maximizes or minimizes the robustness of satisfaction of φ.

3. PROBLEM STATEMENT: WORST-CASE MODEL PREDICTIVE CONTROL

We will address the problem of synthesizing control inputs for a system operating in the presence of potentially adversarial, uncertain external inputs or disturbances. As in (Raman et al., 2015), the controllers we produce must provide guarantees for specifications of the form $\varphi \doteq \varphi_e \Rightarrow \varphi_s$, where φ_e places assumptions on the external environment, and φ_s specifies desired guarantees on the plant behavior. Here φ_e refers to properties of $\mathbf{w} \in W^{\omega}$ and φ_s refers to properties of $\mathbf{x} \in \mathcal{X}^{\omega}$ and $\mathbf{u} \in U^{\omega}$.

In keeping with recent success in this domain, we will try to synthesize a strategy in a model predictive control (MPC) fashion. In MPC, at each iteration, the optimal control sequence is computed over a finite horizon. MPC uses the receding horizon principle, which means that after computing the optimal control sequence, only the first element is implemented in the current iteration. Subsequently, the horizon is shifted by one time step, and the optimization restarted to include any new information, such as about the disturbance measurements.

Given a system of the form in equation (1), initial state x_0, STL formula φ and a piecewise affine cost function J, at each time step k, the *worst-case MPC* optimization problem is defined as follows:

$$\begin{aligned} \underset{\mathbf{u}^{H,k}}{\operatorname{argmin}} \quad & \max_{\mathbf{w}^{H,k} \in \{\mathbf{w} \in W^H \mid \mathbf{w} \models \varphi_e\}} J(\xi_f(x_k, \mathbf{u}^{H,k}, \mathbf{w}^{H,k})) \\ \text{s.t.} \quad & \forall \mathbf{w} \in W^{\omega}, \qquad \xi_f(x_0, \mathbf{u}, \mathbf{w}) \models \varphi, \end{aligned} \tag{2}$$

where H is a finite horizon, $\mathbf{u}^{H,k}$ is the horizon-H control input computed at each time step k, and $\mathbf{u} = u_0^{H,0} u_0^{H,1} u_0^{H,2} \ldots$ is the infinite sequence of control inputs produced in the receding horizon manner described above.

As proved in (Raman et al., 2015, Theorem 2), the solution of the following finite-horizon version of problem (2), with the right choice of H, can be extended to obtain a solution to the infinite horizon problem for a large class of specifications:

$$\begin{aligned} \underset{\mathbf{u}^{H,k}}{\operatorname{argmin}} \quad & \max_{\mathbf{w}^{H,k} \in \{\mathbf{w} \in W^H \mid \mathbf{w} \models \varphi_e\}} J(\xi_f(x_0, \mathbf{u}^{H,k}, \mathbf{w}^{H,k})) \\ \text{s.t.} \quad & \forall \mathbf{w}^{H,k} \in W^H, \qquad \xi_f(x_0, \mathbf{u}^{H,k}, \mathbf{w}^{H,k}) \models \varphi \end{aligned} \tag{3}$$

In (Raman et al., 2015), the worst-case MPC problem was solved using a CEGIS procedure. Informally, at each time step the algorithm starts by guessing the value of the disturbance signal over the current horizon. It then tries to synthesize a control input that satisfies the specification in the face of this disturbance. Once such control input is found, a new disturbance is sought that thwarts this control input, by minimizing the robustness of satisfaction to a level at which the specification is not satisfied. The process repeats until a control input is found such that there is no disturbance that can prevent the specification from being satisfied. The major disadvantage of this approach is that the CEGIS loop may never terminate if W^H is infinite. Since each step of the MPC involves a CEGIS loop, the chances of getting stuck increase the longer the MPC has to run.

4. APPROACHES TO SOLVING WORST-CASE MPC

In this section, we investigate some alternatives to solving this finite-horizon problem, drawing on the classical control literature. The extension to solve problem (3) is as in (Raman et al., 2015), and hence, the results of Theorem 2 of Raman et al. (2015) still apply. Note that both the cost function $J(\xi_f(x_k, \mathbf{u}^{H,k}, \mathbf{w}^{H,k}))$ and the constraints are piecewise affine in x_k, $\mathbf{u}^{H,k}$, and $\mathbf{w}^{H,k}$). Therefore, we look to approaches for solving the optimization problem (3) that are commonly used in the context of piecewise affine systems.

4.1 Explicit MPC for the inner optimization

For a *given* $\mathbf{u}^{H,k}$, the optimization problem

$$\begin{aligned} & \max_{\mathbf{w}^{H,k} \in \{\mathbf{w} \in W^H \mid \mathbf{w} \models \varphi_e\}} J(\xi_f(x_0, \mathbf{u}^{H,k}, \mathbf{w}^{H,k})) \\ \text{s.t.} \quad & \forall \mathbf{w}^{H,k} \in W^H, \qquad \xi_f(x_0, \mathbf{u}^{H,k}, \mathbf{w}^{H,k}) \models \varphi \end{aligned} \tag{4}$$

can be solved as a multi-parametric MILP (mp-MILP) problem, in which $\mathbf{u}^{H,k}$ is the vector of parameters, using the algorithm in (Dua and Pistikopoulos, 2000). Consider the following lemma:

Lemma 1. (Borrelli, 2003; Kerrigan and Mayne, 2002) Let $V : \mathbb{R}^n \times \mathbb{R}^{n_\theta} \to \mathbb{R}$, $(x, \theta) \mapsto V(x, \theta)$ be a piecewise-affine function and consider the following multi-parametric optimization problem:

$$\begin{aligned} & \max_x V(x, \theta) \\ \text{s.t.} \quad & Sx \leq q + U\theta \end{aligned}$$

where $x \in \mathbb{R}^n$ is the optimization variable, $\theta \in \Theta \subseteq \mathbb{R}^s$ is a vector of parameters, S, U and c, q are appropriately defined matrices and vectors, respectively, and Θ is a bounded polyhedral set. Assuming that for any parameter $\theta \in \Theta$ the above optimization problem has a finite solution, then the solution $V^*(\theta)$ can be obtained by solving of a set of mp-LPs. Moreover, $V^*(\theta)$ is a piecewise-affine function. □

Note that the result of Lemma 1 also holds for the case of an mp-MILP problem (Acevedo and Pistikopoulos, 1999; Dua and Pistikopoulos, 2000).

Let ${\mathbf{w}^*}^{H,k}(\mathbf{u}^{H,k}) = \arg\max_{\mathbf{w}^{H,k}} J(\xi_f(x_k, \mathbf{u}^{H,k}, \mathbf{w}^{H,k}))$ denote the solution of the mp-MILP problem (4). Based on the above lemma, ${\mathbf{w}^*}^{H,k}$ is a piecewise-affine function in $\mathbf{u}^{H,k}$ and hence, the outer optimization problem, i.e.,

$$\begin{aligned} \min_{\mathbf{u}^{H,k}} \quad & J(\xi_f(x_k, \mathbf{u}^{H,k}, {\mathbf{w}^*}^{H,k}(\mathbf{u}^{H,k}))) \\ \text{s.t.} \quad & \forall {\mathbf{w}^*}^{H,k}(\mathbf{u}^{H,k}) \in W^H, \ \xi_f(x_0, \mathbf{u}^{H,k}, {\mathbf{w}^*}^{H,k}(\mathbf{u}^{H,k})) \models \varphi, \end{aligned} \tag{5}$$

can be solved as an MILP optimization problem using an off-the-shelf solver based on, e.g., branch-and-bound or cutting plane algorithms (Atamtürk and Savelsbergh, 2005; Linderoth and Ralphs, 2005).

Note that the available mp-MILP algorithms are not very efficient when the size of the vector of parameters and the prediction horizon H increases. Worse still, the number of integer variables in the MILP adversely affects computation time. Unfortunately, the number of integer variables is significant for non-trivial STL formulas, as binary variables are introduced to encode each min and max operation in the robustness-based encoding.

4.2 Monte Carlo Approach

The second approach we consider is to use Monte Carlo simulation to eliminate the inner optimization problem as follows. Let ${\mathbf{w^{(1)}}}^{H,k}, \dots, {\mathbf{w^{(L)}}}^{H,k}$ denote L different noise realizations belonging to the set $\{\mathbf{w} \in W^H | \mathbf{w} \models \varphi_e\}$ and let

$$t(k) = \max_{{\mathbf{w^{(1)}}}^{H,k},\dots,{\mathbf{w^{(L)}}}^{H,k}} (J(\xi_f(x_k, \mathbf{u}^{H,k}, {\mathbf{w^{(1)}}}^{H,k})), \cdots , J(\xi_f(x_k, \mathbf{u}^{H,k}, {\mathbf{w^{(L)}}}^{H,k}))). \quad (6)$$

The choice of L can be made based on the desired level of accuracy and computational efficiency (Agili et al., 2012). The optimization problem (3) can be then rewritten as

$$\min_{\mathbf{u}^{H,k}, t(k)} t(k) \quad (7)$$

$$\begin{aligned} \text{s.t. } & t(k) \geq J(\xi_f(x_k, \mathbf{u}^{H,k}, {\mathbf{w^{(1)}}}^{H,k})) \quad (8) \\ & \cdots \\ & t(k) \geq J(\xi_f(x_k, \mathbf{u}^{H,k}, {\mathbf{w^{(L)}}}^{H,k})) \\ & \forall \mathbf{w^{(1)}} \in W^H, \quad \xi_f(x_0, \mathbf{u}^{H,k}, \mathbf{w^{(1)}}) \models \varphi, \\ & \cdots \\ & \forall \mathbf{w^{(L)}} \in W^H, \quad \xi_f(x_0, \mathbf{u}^{H,k}, \mathbf{w^{(L)}}) \models \varphi \end{aligned}$$

which can be solved as a single MILP optimization problem. Note that the Monte Carlo approach guarantees the termination of the optimization; however, the satisfaction of the specifications is only guaranteed with a high probability and not with probability one, since we are only looking at a limited, no matter how large L is, number of noise realizations. L can be defined based on the desired level of accuracy and computational efficiency (Agili et al., 2012).

One challenge in applying the Monte Carlo approach to the problem of reactive synthesis from STL specifications is that of sampling disturbances that satisfy φ_e. A naive way to do this is via rejection-sampling: we generate a random sequence of inputs and then determine if it satisfies φ by checking feasibility of a MILP with constraints corresponding to φ_e and inputs fixed to the sampled values at each time step. If this MILP is feasible, we add the sample to the set. Otherwise, we put it in a "reject" pile. Then, for subsequent samples, we can immediately reject any sample that is "too close" to one of our rejected samples, since it means that they will also not satisfy φ_e. Determining a good set of samples to use is a direction of future work, but this seems like another opportunity for exploiting the quantitative semantics of STL.

4.3 Dual formulation of the MILP optimization problem

Another approach to solving optimization problem (3) is to use the dual reformulation. To this end, we first replace the inner optimization problem (4) by its dual in order to obtain a minimization problem instead of a min-max problem (3). If the disturbance does not have any discrete component, the dual problem gives the same result as the primal for the inner problem, since it will be a linear optimization in terms of $\mathbf{w}^{H,k}$. In the general case where the disturbance has both continuous and discrete components, the dual problem gives an upper bound on the inner optimization problem.

First, we rewrite (4) in the following form:

$$\max_{\mathbf{w}^{H,k} \in \{\mathbf{w}: S\mathbf{w} \leq \tilde{q}\}} \quad J(\xi_f(x_k, \mathbf{u}^{H,k}, \mathbf{w}^{H,k})) \quad (9)$$

$$\text{s.t.} \quad R(k)\mathbf{u}^{H,k} + E(k)\mathbf{w}^{H,k} + g(k) \leq 0 \quad (10)$$

where $\mathbf{u}^{H,k}$ is the vector of decision variables and $\mathbf{w}^{H,k}$ is the disturbance vector, both including continuous and binary variables; $R(k)$ and $E(k)$ are inequality constraint matrices; and $g(k)$ is a constant vector. Matrix S and vector $\tilde{q}$ define the polyhedral set to which the disturbance vector $\mathbf{w}^{H,k}$ belongs. Define the linear objective function $J(\xi_f(x_k, \mathbf{u}^{H,k}, \mathbf{w}^{H,k})) = C_1^T \mathbf{u}^{H,k} + C_2^T \mathbf{w}^{H,k}$, where C_1 and C_2 are the vectors of coefficients. The equivalent dual reformulation of the optimization problem (9)-(10) can be then written as,

$$\min_{\lambda,\mu} \lambda^T (R(k)\mathbf{u}^{H,k} + g(k)) - \mu^T \tilde{q} \quad (11)$$

$$\text{s.t.} \quad E(k)^T \lambda + C_2 + S^T \mu = 0 \quad (12)$$

$$\lambda, \mu \geq 0 \quad (13)$$

where μ and λ are Lagrange multipliers and (11)-(13) is a mixed integer quadratic program (MIQP). We also rewrite the equivalent MILP formulation of the constraints in the optimization problem (3) as

$$P(k)\mathbf{u}^{H,k} + Q(k)\mathbf{w}^{H,k} + q(k) \leq 0 \ \forall {\mathbf{w}^*}^{H,k}(\mathbf{u}^{H,k}) \in W^H \quad (14)$$

where $P(k), Q(k)$, and $q(k)$ are appropriately defined constraint matrices and vector, respectively. Now, by using Farkas' lemma (Boyd and Vandenberghe, 2004) and equations (11)-(14), we can reformulate the optimization problem (3) as

$$\min_{\mathbf{u}^{H,k}, \Lambda, \lambda, \mu, \beta} \quad C_1^T \mathbf{u}^{H,k} + \lambda^T (R(k)\mathbf{u}^{H,k} + g(k)) - \mu^T \tilde{q} \quad (15)$$

$$\begin{aligned} \text{s.t. } & Q_i(k) = S^T \Lambda_i \quad \forall i = 1, \dots, m \quad (16) \\ & P_i(k)\mathbf{u}^{H,k} + q_i(k) = \beta_i - \Lambda_i \tilde{q} \quad \forall i = 1, \dots, m \quad (17) \\ & E(k)^T \lambda + C_2 + S^T \mu = 0 \quad (18) \\ & \Lambda, \lambda, \mu \geq 0, \quad \beta \leq 0 \quad (19) \end{aligned}$$

where Λ is a Lagrange multiplier, m denotes the number of constraints in equation (14), and β is a vector of length m. Thereby, we obtain a single MIQP minimization problem, which can be solved using available MIQP solvers. Note that if there are no constraints in the inner optimization problem that depends on $\mathbf{u}^{H,k}$ (cf. (10)), the dual problem (11)-(13) will result in a MILP, and hence (15)-(19) will also be a MILP.

5. EXAMPLE

In this section, we describe some experimental results corresponding to an implementation of each of the above approaches. We run all examples on a simple spring-damper system, where the disturbance is an additive noise to both states of the system. The continuous system dynamics are given as follows

$$\dot{x}_1(t) = x_2(t) + u(t) + w_1(t)$$
$$\dot{x}_2(t) = -x_1(t) - 2x_2(t) + w_2(t)$$

where $x_1(t)$ is the 2D position (in m) of the spring at time t, $x_2(t)$ is the velocity of the spring at time t, $u(t)$ is the input at time t, and $w(t)$ is the disturbance vector at time t. We use a sampling time of $\Delta t = 0.1$s to discretize the system. The initial state is $x(0) = [4.2; 0]$.

The STL specification used to illustrate our approach has $\varphi_e = \Box_{[0,\infty)}(|w(t)| \leq 3)$, $\varphi_s = \Diamond_{[0,3]}((5 \leq x_1(t) \leq 6) \wedge (|u(t)| \leq 20))$, i.e. the system should eventually, within 3s, bring the position and input within the specified bounds, assuming that the disturbance always stays within the bounded range.

Using this simple system allows us to focus on how the above techniques performed in comparison with results obtained using the CEGIS scheme in (Raman et al., 2015), as implemented in the MATLAB-based STL synthesis tool BluSTL [1]. We have chosen a narrow interval for the position of the spring to be maintained in order to demonstrated the effectiveness of the synthesized control input even in the presence of a high disturbance.

Figure 1 depicts the results of solving (3) over several time steps using the aforementioned approaches [2]. The

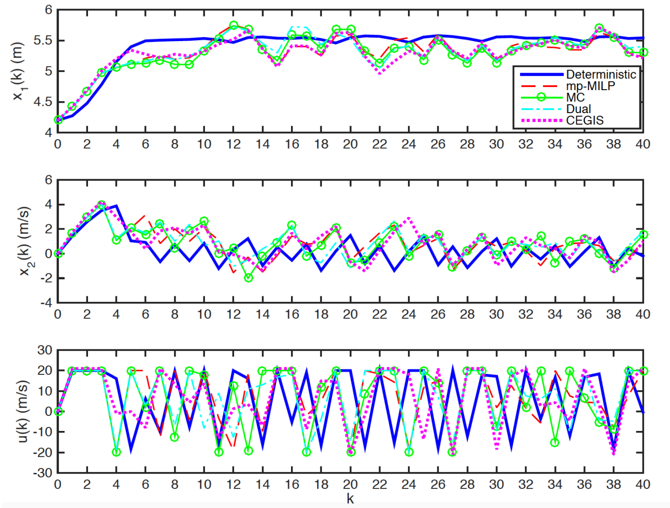

Fig. 1. Eventually maintaining the position between 5-6m within 3s, using various approaches to solve the optimization problem (3). The blue line shows performance when zero disturbance is assumed.

first plot illustrates the position of the spring at time step k, the second plot shows its velocity at time step k, and the third plot is the input that controls the position of the spring at time step k, where $k = 0, \ldots, 40$. We show results with and without randomly generated disturbances that satisfy the assumption φ_e. The prediction horizon is chosen as $H = 8$. As can be seen in Figure 1, all approaches achieve the bounds within the first 3s, i.e., before $k = 31$ time steps. Note that since the optimization problem (3) is an MILP and thus non-convex, we may only obtain locally optimal solutions (Schrijver, 1986). Minor differences in the results can also be attributed to numerical discrepancies arising from the different problem encodings used for each method.

[1] Available at https://github.com/BluSTL/BluSTL.
[2] The optimization problem is solved using MATLAB R2014b on a 2.6 GHz Intel Core i5 processor.

Approach	Time (s)
Deterministic	23.80
Uncertain (mp-MILP)	14648.00
Uncertain (Monte Carlo)	5627.00
Uncertain (Dual approach)	133.32
Uncertain (CEGIS)	127.54

Table 1. Computation time to solve (3) using the tested approaches.

The computation time for each of these approaches is given in Table 1. The computation times show that the dual approach and the CEGIS algorithm perform comparably, and both have the shortest computation time by far.

We now discuss the three alternatives in detail. We first used the multi-parametric MILP solvers in the MPT toolbox to solve the inner optimization problem (4), and the Gurobi optimizer for the outer problem (5). The computation time provided for this approach in Table 1 is for $H = 4$. We found that this approach is impractical for specifications with long horizons (> 4) or multiple temporal operators, and particularly nested temporal operators.

To solve the MILP optimization problem (7)-(8), we again used Gurobi. We chose $L = 1000$ different uncertainty vectors $\mathbf{w}^{H,k}$ to obtain a 0.95% confidence level with accuracy error of 1% (cf. (Agili et al., 2012)). As shown in Figure 1, the results are comparable to the dual approach and the result oscillates around the deterministic (no disturbance) solution. Due to the large number of noise realizations, the computation time for this approach is still very long, making it unsuitable in practice.

For the dual approach, the optimization problem (15)-(17) was solved using the MIQP solver in Gurobi. As mentioned before, the solution of this approach is very close to the Monte Carlo approach. This approach appears to be the most computationally efficient among those proposed in this paper. Moreover, since its computation time is very close to that of the CEGIS approach, it can be used as a replacement, since the dual approach guarantees termination in finite time while CEGIS does not.

6. CONCLUSION

Our goal in this work was to examine the applicability of state-of-the art robust MPC optimization techniques to the problem of reactive synthesis from STL specifications. We explored three different methods for solving the required worst-case MPC problem, namely a multi-parametric MILP solution, a Monte Carlo approach, and a dual optimization approach. Numerical results for the case study of a system with spring-damper dynamics reveals that the mp-MILP approach is not feasible for this type of problem due to a very long computation time even for a relatively short prediction horizon, and for a system with simple dynamics. On the other hand, both the Monte Carlo approach and the dual optimization approach give reliable results with reasonable computation time. Hence, these approaches can be used as an alternative to the algorithm proposed in (Raman et al., 2015), especially given that they terminate in finite time while the algorithm in (Raman et al., 2015) may not.

In future work, we will explore problem-specific extensions of the above approaches, using the insights gained during this work. We will also apply these approaches to more illustrative examples with intricate temporal logic specifications and more complex hybrid dynamics. Another research direction is to extend the problem formulation to a stochastic setting and explore efficient computational approaches; in this context, it is interesting to compare the direct methods (primal or dual) with more sophisticated Markov Chain Monte Carlo methods.

ACKNOWLEDGEMENTS

This work is supported in part by Northrop Grumman and by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

REFERENCES

Acevedo, J. and Pistikopoulos, E.N. (1999). An algorithm for multiparametric mixed-integer linear programming problems. *Operation Research Letters*, 24(3), 139–148.

Agili, S.S., Morales, A.W., Li, J., and Resso, M. (2012). Finding the probability distribution functions of s-parameters and their Monte Carlo simulation. *IEEE Transactions on Instrumentation and Measurement*, 61(11), 2993–3002.

Atamtürk, A. and Savelsbergh, M.W.P. (2005). Integer-programming software systems. *Annals of Operations Research*, 140(1), 67–124.

Bemporad, A., Heemels, W.P.M.H., and De Schutter, B. (2002). On hybrid systems and closed-loop MPC systems. *IEEE Transactions on Automatic Control*, 47(5), 863–869. doi:10.1109/TAC.2002.1000287.

Borrelli, F. (2003). *Constrained Optimal Control of Linear and Hybrid Systems*, volume 290 of *Lecture Notes in Control and Information Sciences*. Springer, Berlin, Germany.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, Cambridge, UK.

De Schutter, B. and van den Boom, T. (2001). Model predictive control for max-plus-linear discrete event systems. *Automatica*, 37(7), 1049–1056.

Dua, V. and Pistikopoulos, E.N. (2000). An algorithm for the solution of multiparametric mixed integer linear programming problems. *Annals of Operations Research*, 99(1-4), 123–139.

Fainekos, G.E., Girard, A., Kress-Gazit, H., and Pappas, G.J. (2009). Temporal logic motion planning for dynamic robots. *Automatica*, 45(2), 343–352.

Jin, X., Donzé, A., Deshmukh, J.V., and Seshia, S.A. (2013). Mining requirements from closed-loop control models. In *Hybrid Systems: Computation and Control, HSCC 2013, April 8-11, 2013, Philadelphia, PA, USA*, 43–52.

Karaman, S. and Frazzoli, E. (2008). Vehicle routing problem with metric temporal logic specifications. In *Proceedings of the 47th IEEE Conference on Decision and Control, CDC 2008, December 9-11, 2008, Cancún, México*, 3953–3958.

Karaman, S. and Frazzoli, E. (2009). Sampling-based motion planning with deterministic μ-calculus specifications. In *Proceedings of the 48th IEEE Conference on Decision and Control, CDC 2009, combined withe the 28th Chinese Control Conference, December 16-18, 2009, Shanghai, China*, 2222–2229.

Karaman, S. and Frazzoli, E. (2011). Linear temporal logic vehicle routing with applications to multi-UAV mission planning. *International Journal of Robust and Nonlinear Control*, 21(12), 1372–1395.

Kerrigan, E.C. and Mayne, D.Q. (2002). Optimal control of constrained, piecewise affine systems with bounded disturbances. In *Proceedings of the 41st IEEE Conference on Decision and Control, CDC 2002*, volume 2, 1552–1557. doi:10.1109/CDC.2002.1184740.

Kloetzer, M. and Belta, C. (2008). A fully automated framework for control of linear systems from temporal logic specifications. *IEEE Transactions on Automatic Control*, 53(1), 287–297.

Kwon, Y. and Agha, G. (2008). LTLC: linear temporal logic for control. In *Hybrid Systems: Computation and Control, 11th International Workshop, HSCC 2008, St. Louis, MO, USA, April 22-24, 2008*, 316–329.

Lazar, M., Heemels, M., Weiland, S., and Bemporad, A. (2006). Stability of hybrid model predictive control. *IEEE Transactions on Automatic Control*, 15(11), 1813–1818.

Linderoth, J.T. and Ralphs, T.K. (2005). Noncommercial software for mixed-integer linear programming. In J. Karlof (ed.), *Reinforcement Learning: State-Of-The-Art*, CRC Press Operations Research Series, 253–303.

Maciejowski, J.M. (2002). *Predictive Control with Constraints*. Prentice Hall, Harlow, England.

Maler, O. and Nickovic, D. (2004). Monitoring temporal properties of continuous signals. In *Proceedings of the Joint International Conferences on Formal Modelling and Analysis of Timed Systems, FORMATS 2004 and Formal Techniques in Real-Time and Fault-Tolerant Systems, FTRTFT 2004, Grenoble, France, September 22-24, 2004*, 152–166. doi:10.1007/978-3-540-30206-3_12.

Nuzzo, P., Xu, H., Ozay, N., Finn, J., Sangiovanni-Vincentelli, A., Murray, R., Donze, A., and Seshia, S. (2014). A contract-based methodology for aircraft electric power system design. *Access, IEEE*, 2, 1–25.

Raman, V., Donzé, A., Maasoumy, M., Murray, R.M., Sangiovanni-Vincentelli, A.L., and Seshia, S.A. (2014). Model predictive control with signal temporal logic specifications. In *53rd IEEE Conference on Decision and Control, CDC 2014, Los Angeles, CA, USA, December 15-17, 2014*, 81–87. doi:10.1109/CDC.2014.7039363.

Raman, V., Donzé, A., Sadigh, D., Murray, R.M., and Seshia, S.A. (2015). Reactive synthesis from signal temporal logic specifications. In *Hybrid Systems: Computation and Control, HSCC 2015, Seattle, WA, USA, April 14-16, 2015*, 239–248. doi:10.1145/2728606.2728628.

Schrijver, A. (1986). *Theory of Linear and Integer Programming*. John Wiley & Sons, Chichester, UK.

Wolff, E.M., Topcu, U., and Murray, R.M. (2014). Optimization-based trajectory generation with linear temporal logic specifications. In *International Conference on Robotics and Automation, ICRA 2014, Hong Kong, China, May 31 - June 7, 2014*, 5319–5325.

Wongpiromsarn, T., Topcu, U., , and Murray, R.M. (2012). Receding horizon temporal logic planning. *IEEE Transactions on Automatic Control*, 57(11), 2817–2830.