Model Predictive Control from Signal Temporal Logic Specifications: A Case Study (Work in Progress)

Vasumathi Raman
California Institute of Technology
1200 E California Blvd
Pasadena, CA 91125, USA
vasu@caltech.edu

Mehdi Maasoumy
UC Berkeley
Cory Hall
Berkeley, CA 94720, USA
mehdi@me.berkeley.edu

Alexandre Donzé
UC Berkeley
Cory Hall
Berkeley, CA 94720, USA
donze@berkeley.edu

ABSTRACT

This paper describes current work on framing the model predictive control (MPC) of cyber-physical systems as synthesis from signal temporal logic (STL) specifications. We provide a case study using a simplified power grid model with uncertain demand and generation; the model-predictive control problem here is that of the ancillary service power flow from the buildings. We show how various relevant constraints can be captured using STL formulas, and incorporated into an MPC framework. We also provide preliminary simulation results to illustrate the promise of the proposed approach.

Keywords

formal synthesis, timed logics, model predictive control

1. INTRODUCTION

Temporal logics provide a rigorous, precise and unambiguous formalism for specifying and verifying desired behaviors of systems. In particular, they have lent themselves to algorithms for verification and synthesis of discrete supervisory controllers that satisfy the specified properties. These discrete controllers have successfully be employed in the construction of hybrid controllers for cyber-physical systems in various domains, including robotics [4] and aircraft power system design [9]. However, for physical systems that require constraints not just on the order of events, but also on the temporal distance between them, simulation and testing is still the method of choice for validating properties and establishing guarantees; the exact exhaustive verification of such systems is in general undecidable [1]. Signal Temporal Logic (STL) [7] was originally developed in order to specify and monitor the expected behavior of such physical systems, including temporal constraints between events. STL allows the specification of properties of dense-time, real-valued signals, and the automatic generation of monitors for testing these properties of individual simulation traces. It has since been applied to the analysis of several types of continuous and hybrid systems, including dynamical systems and analog circuits, where the continuous variables represent quantities like currents and voltages in the circuit.

Model Predictive Control (MPC) is based on iterative, finite horizon, discrete time optimization of a model of the plant. At any given time t the current plant state is observed, and an optimal control strategy computed for some finite time horizon in the future, $[t, t+N]$. An online calculation is used to explore possible future state trajectories originating from the current state, finding an optimal control strategy until time $t+N$. Only the first step of the computed optimal control strategy is implemented; the plant state is then sampled again, and new calculations are performed on a horizon of N starting from the new current state. While the global optimality of this sort of "receding horizon" approach is not ensured, it tends to do well in practice In addition to reducing computational complexity, it improves the system robustness with respect to exogenous disturbances and modeling uncertainties [8].

In this work, we frame the MPC problem as an instance of synthesis from STL specifications. We would like to be able to specify a desired global specification, and decompose it into STL specifications over each time horizon, such that synthesizing a controller fulfilling these specifications at each horizon results in satisfaction of the global specification. For synthesis, we build on the recent success of encodings of Linear Temporal Logic (LTL) specifications as mixed integer-linear constraints [10], based on linear encodings for bounded LTL model checking [2]. We extend this encoding to STL, and incorporate the resulting constraints into the optimization problem at each finite horizon of the MPC computation.

As a case study, we consider the simplified model of a smart building-level micro-grid with uncertain demand and generation presented in [5]. We build on the hierarchical control framework introduced in [6], in which model predictive control is implemented on top of the existing state-of-the-art Automatic Generation Control (AGC), in order to exploit the demand-side flexibility of a commercial building and provide fast frequency regulation services to the power grid. We then show how the MPC scheme for controlling the ancillary service power flow from such buildings can be framed in terms of synthesis from a Signal Temporal Logic (STL) specification. Preliminary simulation results illustrate the effectiveness of the proposed methodology for grid frequency regulation.

CyPhy'14, April 14 - 17 2014, Berlin, Germany

http://dx.doi.org/10.1145/2593458.2593472.

2. PROBLEM

We consider a general model of a discrete-time continuous system of the form

$$x_{t+1} = f(x_t, u_t) \tag{1}$$

where $t = 0, 1, \ldots$ are the time indices, $x \in \mathcal{X} \subseteq (\mathbb{R}^{n_c} \times \{0,1\}^{n_l})$ are the continuous and binary/logical states, $u \in \mathcal{U} \subseteq (\mathbb{R}^{m_c} \times \{0,1\}^{m_l})$ are the (continuous and logical) control inputs, and $x_0 \in \mathcal{X}$ is the initial state. A *run* $\mathbf{x} = x_0x_1x_2...$ is an infinite sequence of its states, where $x_t \in \mathcal{X}$ is the state of the system at index t, and for each $t = 0, 1, ...,$ there exists a control input $u_t \in \mathcal{U}$ such that $x_{t+1} = f(x_t, u_t)$. Given an initial state x_0 and a control input sequence $\mathbf{u} = u_0u_1u_2...$, the resulting run $\mathbf{x} = \mathbf{x}(x_0, \mathbf{u})$ of a system modeled by (1) is unique. Restricting to finite sequences, given a control input sequence $\mathbf{u}_N = u_0u_1u_2...u_N$, we let the resulting horizon-N run be $\mathbf{x}_N = \mathbf{x}(x_0, \mathbf{u}_N) = x_0x_1x_2...x_N$. We also introduce the notion of a generic cost function $J(\mathbf{x}, \mathbf{u})$ that maps finite and infinite runs to $\mathbb{R} \cup \infty$.

For this work, we assume that STL formulas are provided in negation normal form, so all negations appear in front of predicates. An STL formula can always be rewritten as a negation normal form formula of size linear in the original size. STL formulas are thus defined recursively as:

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

where μ is a predicate of the form $\mu : \mu(x) > 0$. Additionally, we define $\Diamond_{[a,b]}\, \varphi = \top\ \mathcal{U}_{[a,b]}\ \varphi$.

The validity of a formula φ with respect to signal x at time t is defined inductively as follows:

$$\begin{array}{lll}
(x,t) \models \mu & \Leftrightarrow & \mu(x(t)) > 0 \\
(x,t) \models \neg\mu & \Leftrightarrow & \neg((x,t) \models \mu) \\
(x,t) \models \varphi \wedge \psi & \Leftrightarrow & (x,t) \models \varphi \wedge (x,t) \models \psi \\
(x,t) \models \varphi \vee \psi & \Leftrightarrow & (x,t) \models \varphi \vee (x,t) \models \psi \\
(x,t) \models \Box_{[a,b]}\, \psi & \Leftrightarrow & \forall t' \in [t+a, t+b], (x,t') \models \psi \\
(x,t) \models \varphi\ \mathcal{U}_{[a,b]}\ \psi & \Leftrightarrow & \exists t' \in [t+a, t+b] \text{ s.t. } (x,t') \models \psi \\
 & & \wedge \forall t'' \in [t, t'], (x, t'') \models \varphi.
\end{array}$$

A run $\mathbf{x} = x_0x_1x_2...$ satisfies φ, denoted by $\mathbf{x} \models \varphi$, if $(\mathbf{x}, 0) \models \varphi$.

We now formally state the STL controller synthesis problem.

Problem 1. *Given a system of the form (1), an initial state x_0 and an STL formula φ, compute a control input sequence $\mathbf{u}$ such that $\mathbf{x}(x_0, \mathbf{u}) \models \varphi$.*

We propose to solve Problem 1 using a Model Predictive Control formulation, i.e., at each step t, u_t is defined as the first input of the sequence solving

$$\underset{\mathbf{u}_N}{\text{argmin}}\, J(\mathbf{x}_N(x_t, \mathbf{u}_N), \mathbf{u}_N(t)) \text{ s.t. } \mathbf{x}_N(x_t, \mathbf{u}_N(t)) \models \varphi_t, \tag{2}$$

where φ_t is an STL property such that if $(x,t) \models \varphi_t$ for all t, then $x \models \varphi$. We add STL constraints to a traditional MPC problem formulation. To do so, first, we represent the system trajectory over the MPC prediction horizon as a finite sequence of states satisfying the model dynamics (1). Then we encode the formula φ_t with a set of Mixed Integer Linear Program (MILP) constraints, as defined in Section 3.

3. MILP FORMULATION

Given a formula φ, we introduce a variable P_t^φ, whose value is tied to a set of mixed integer-linear constraints required for the satisfaction of φ at position t in the state sequence. In other words, P_t^φ has an associated set of MILP constraints such that $P_t^\varphi = 1$ if and only if φ holds at position t. We recursively generate the MILP constraints corresponding to P_0^φ – the value of this variable determines whether to not a formula φ holds in the initial state.

3.1 Predicates

The predicates are represented by constraints on the system state variables. For each predicate $\mu \in P$, we introduce binary variables $z_t^\mu \in \{0,1\}$ for time indices $t = 0, 1, ..., N$. We enforce that $z_t^\mu = 1$ if and only if $\mu(x_t) > 0$. This is achieved with the following constraints (which employ the "big M" method from operations research):

$$\begin{array}{rcl}
\mu(x_t) & \leq & M_t z_t^\mu + \epsilon_t \\
-\mu(x_t) & \leq & M_t(1 - z_t^\mu) - \epsilon_t
\end{array}$$

where M_t are sufficiently large positive numbers, and ϵ_t are sufficiently small positive numbers that serve to bound $\mu(x_t)$ away from 0. Note that $z_t = 1$ if and only if $\mu(x_t) > 0$.

3.2 Boolean operations on MILP variables

Here we follow the example of [10] when encoding negation, conjunction and disjunction of variables using mixed integer-linear constraints. As described in Section 3.1, each predicate μ has an associated binary variable z_t^μ which equals 1 if μ holds at time t, and 0 otherwise. In fact, by the recursive definition of our MILP constraints on STL formulas, we can assume that each operand φ in a boolean operation has a corresponding (binary or continuous) variable P_t^φ which is 1 if φ holds at t and 0 if not. Here we define boolean operations on these variables; these are the building blocks of our recursive encoding.

Given a formula ψ containing a boolean operation, we add new continuous variables $P_t^\psi \in [0,1]$ to represent its truth value at each time step of the parametrized trajectory. These variables are constrained such that $P_t^\psi = 1$ if ψ holds at time t and $P_t^\psi = 0$ otherwise.

- **Negation:** $\psi = \neg\mu$ $\qquad P_t^\psi = 1 - P_t^\mu$
- **Conjunction:** $\psi = \wedge_{i=1}^m \varphi_i$ $\qquad P_t^\psi \leq P_t^{\varphi_i}, i = 1, ..., m,$ $P_t^\psi \geq 1 - m + \sum_{i=1}^m P_t^{\varphi_i}$
- **Disjunction:** $\psi = \vee_{i=1}^m \varphi_i$ $\qquad P_t^\psi \geq P_t^{\varphi_i}, i = 1, ..., m,$ $P_t^\psi \leq \sum_{i=1}^m P_t^{\varphi_i}$

3.3 Temporal constraints

Here we describe how we encode timed operators. We first present encodings for the $\Box$ and $\Diamond$ operators. We will use these encodings to define the encoding for the $\mathcal{U}_{[a,b]}$ operator.

- **Always:** $\psi = \Box_{[a,b]}\, \varphi$
  Let $a_t^N = \min(t+a, N)$ and $b_t^N = \min(t+b, N)$
  Define $P_t^\psi = \wedge_{i=a_t^N}^{b_t^N} P_i^\varphi$ The logical operation $\wedge$ on the variables P_i^φ here is as defined in Section 3.2.
- **Eventually:** $\psi = \Diamond_{[a,b]}\, \varphi$
  Define $P_t^\psi = \vee_{i=a_t^N}^{b_t^N} P_i^\varphi$

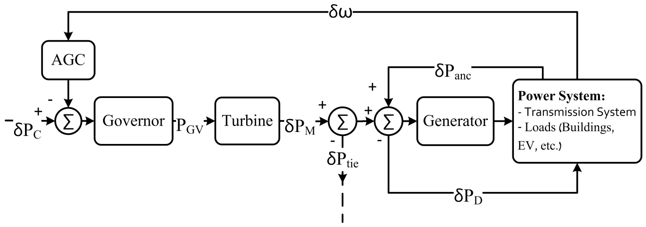

Figure 1: Block diagram of power system and its relation to governor, turbine, generator, and the AGC signal for each control area. More details on the power grid model can be found in [5].

- **Until:** $\psi = \varphi_1 \ \mathcal{U}_{[a,b]} \ \varphi_2$
  The bounded until operator $\mathcal{U}_{[a,b]}$ can be defined in terms of the unbounded $\mathcal{U}$ from LTL as follows [3]:

$$\varphi_1 \ \mathcal{U}_{[a,b]} \ \varphi_2 = \Box_{[0,a]} \varphi_1 \wedge \Diamond_{[a,b]} \varphi_2 \wedge \Diamond_{[a,a]}(\varphi_1 \ \mathcal{U} \ \varphi_2)$$

We will use the linear encoding of the unbounded $\mathcal{U}$ from [2]. We define

$$P_t^{\varphi_1 \ \mathcal{U} \ \varphi_2} = P_t^{\varphi_2} \vee (P_t^{\varphi_1} \wedge P_{t+1}^{\varphi_1 \ \mathcal{U} \ \varphi_2})$$

for $t = 1, ..., N-1$, and $P_N^{\varphi_1 \ \mathcal{U} \ \varphi_2} = P_N^{\varphi_2}$.

Given this encoding of the unbounded until and the encodings of $\Box_{[a,b]}$ and $\Diamond_{[a,b]}$ above, we can encode

$$P_t^{\varphi_1 \ \mathcal{U}_{[a,b]} \ \varphi_2} = P_t^{\Box_{[0,a]} \varphi_1} \wedge P_t^{\Diamond_{[a,b]} \varphi_2} \wedge P_t^{\Diamond_{[a,a]}(\varphi_1 \ \mathcal{U} \ \varphi_2)}.$$

By induction on the structure of STL formulas φ, $P_t^{\varphi} = 1$ if and only if φ holds on the system at time t. With this motivation, given a specification φ, we add a final constraint to the MILP:

$$P_0^{\varphi} = 0.$$

The union of the STL constraints, system constraints and loop constraints gives the MILP encoding of Problem 1; this enables checking feasibility of this set and finding a solution using an MILP solver. Other cost functions and more general dynamics can be included by using an appropriate mixed integer-nonlinear solver.

3.4 Complexity

Mixed integer-linear programs are NP-hard, but we can still describe the computational costs of our encoding and approach in terms of the number of variables and constraints in the resulting MILP. In practice, one measure of complexity is the number of binary variables required to indicate the satisfaction of the predicates μ. This depends directly on the number of predicates used in the STL formula φ. If P is the set of predicates used in the formula, then $O(N \cdot |P|)$ binary variables are introduced. In addition, continuous variables are introduced during the MILP encoding of the STL formula. The number of continuous variables used is $O(N \cdot |\varphi|)$, where $|\varphi|$ is the length (i.e. the number of operators) of the formula.

4. A SMART GRID BUILDING SYSTEM

In this paper, we use the smart grid building model of the power system presented in [5]. The interconnection of power system components, including a governor, turbine and generator, is shown in the block diagram in Figure 1. In the diagram, δP_C is a control input which acts against an increase or decrease in power demand to regulate the system frequency, and δP_D denotes fluctuations in power demand, modeled as an exogenous input (disturbance).

4.1 Two-Area System Model

We consider a two-area interconnected system consisting of two buses connected by a tie line with reactance X_{tie}. The power flow on the tie line from area 1 to area 2 is denoted by P_{tie}. A positive δP_{tie} represents an increase in power transfer from area 1 to area 2. This is equivalent in effect to increasing the load of area 1 and decreasing the load of area 2. Each area consists of the subsystems shown in Figure 1. Next, we present the mathematical model of the two-area system. Note that for states, x, the superscript refers to the control area (i.e., $i = 1, 2$), and the subscript indexes the state in each area.

$$\frac{dx_1^i}{dt} = \frac{(-D^i x_1^i + \delta P_M^i - \delta P_D^i - \delta P_{\text{tie}}^{ij} + \delta P_{\text{anc}}^i)}{M_x^i} \tag{3a}$$

$$\frac{dx_2^i}{dt} = \frac{(x_3^i - x_2^i)}{T_7^i}, \quad \frac{dx_3^i}{dt} = \frac{(x_4^i - x_3^i)}{T_6^i}, \quad \frac{dx_4^i}{dt} = \frac{(x_5^i - x_4^i)}{T_5^i} \tag{3b}$$

$$\frac{dx_5^i}{dt} = \frac{(P_{GV}^i - x_5^i)}{T_4^i}, \quad \frac{dx_6^i}{dt} = \frac{(x_7^i - x_6^i)}{T_3^i} \tag{3c}$$

$$\frac{dx_7^i}{dt} = \frac{(-x_7^i + \delta P_C^i - x_1^i/R^i)}{T_1^i}, \quad \frac{dx_8^i}{dt} = x_1^i \tag{3d}$$

where δP_M^i and P_{GV}^i are given by $\delta P_M^i = K_1^i x_5^i + K_3^i x_4^i + K_5^i x_3^i + K_7^i x_2^i$, and $P_{GV}^i = (1 - T_2/T_3)x_6^i + (T_2/T_3)x_7^i$. D is the damping coefficient, M is the machine inertia constant, R is the speed regulation constant, T_i's are time constants for power system components, and K_i's are fractions of total mechanical power outputs associated with different operating points of the turbine. In formulation (3), the first state represents the frequency increment, $x_1^i = \delta\omega_i$. All state dynamics are derived using the mathematical model of each subsystem, as presented in [5]. The state space model (3) can be discretized and written in compact form as

$$x[k+1] = Ax[k] + B_1 u_{\text{sc}}[k] + B_2 u_{\text{anc}}[k] + Ed[k]. \tag{4}$$

We use this state update equation in Section 4.3, where we present the MPC formulation. Input signals are $u_{\text{sc}} = [\delta P_C^1 \ \delta P_C^2]^T$, the ancillary inputs are $u_{\text{anc}} = [\delta P_{\text{anc}}^1 \ \delta P_{\text{anc}}^2]^T$, and the exogenous inputs (i.e. disturbances or variations in demands) are denoted by $d = [\delta P_D^1 \ \delta P_D^2]^T$.

4.2 MPC for Automatic Generation Control

In the classical AGC, a simple PI control is utilized to regulate the frequency of the grid. The Area Control Error (ACE) is defined as $ACE^i = \delta P_{tie}^i + \beta^i x_1^i$, where $\delta P_{\text{tie}}^i = P_{\text{tie}}^i - P_{\text{tie,scheduled}}^i$, and β^i is the bias coefficient of area i. The standard industry practice is to set the bias β^i at the so-called Area Frequency Response Characteristic (AFRC), which is defined as $\beta^i = D^i + 1/R^i$. The integral of ACE is used to construct the speed changer position feedback control signal (δP_C^i). In other words, the control input δP_C^i is given by $\delta P_C^i = -K^i x_9^i$, where K^i is the feedback gain and $\frac{dx_9^i}{dt} = ACE^i$. We propose a methodology for the ancillary services complementing the primary control of AGC, as described in 4.3.

4.3 MPC for Ancillary Services

We present an MPC scheme to control the ancillary service to improve on the classical AGC practice. This optimization-based control framework is utilized as a *higher-level* control in a "hierarchical" scheme on top of the *low-level* classical AGC control [5]. We require that u_{anc} satisfies $\underline{u}_{\text{anc}} \leq u_{\text{anc}}[k+j|k] \leq \overline{u}_{\text{anc}}$ for some $\underline{u}_{\text{anc}} < 0$ and $\overline{u}_{\text{anc}} > 0$, and a maximum ramp constraint:

$$|u_{\text{anc}}[k+1] - u_{\text{anc}}[k] \leq \lambda, \text{for some } \lambda > 0. \tag{5}$$

At each time step k, we thus solve the following problem:

$$\begin{aligned} \min_{U_{\text{anc}}[k]} \quad & J(\text{ACE}, U_{\text{anc}}) \\ \text{s.t.} \quad & x[k+j+1|k] = \\ & \quad Ax[k+j|k] + B_2 u_{\text{anc}}[k+j|k] + Ed[k+j|k] \\ & \underline{u}_{\text{anc}} \leq u_{\text{anc}}[k+j|k] \leq \overline{u}_{\text{anc}} \\ & |u_{\text{anc}}[k+j+1|k] - u_{\text{anc}}[k+j|k]| \leq \lambda \end{aligned} \tag{6}$$

where

$$U_{\text{anc}}[k] = (u_{\text{anc}}[k|k], u_{\text{anc}}[k+1|k], \ \ldots, u_{\text{anc}}[k+H-1|k])$$

is the vector of inputs from k to $k+H$ and H is the prediction horizon. The notation $x[k+j|k]$ denotes that predictions of x for future times $k+j$ are obtained at each time step k. All the constraints of problem (6) should hold for $j = 0, 1, \ldots, H-1$.

The cost function proposed in [5] minimizes the ℓ_2 norm of the ACE signal in areas $i = 1, 2$, by exploiting the ancillary service available in each area, while taking into account the system dynamics and constraints.. We propose to constrain the ACE signal to satisfy a specified set of STL properties, while minimizing the ancillary service used by each area. Thus we defined $J(ACE, U_{\text{anc}}) = \|U_{\text{anc}}\|_{\ell_2} = \sum_{i=1}^{2} \sum_{j=0}^{H-1} (U^i_{\text{anc}}[k+j|k])^2$, and an STL formula φ which says that whenever $|\text{ACE}^i|$ is larger than 0.01, it should become less than 0.01 in less than τ s. More precisely we used $\varphi = \square(\varphi_t)$ with

$$\begin{aligned} \varphi_t = \neg(|\text{ACE}^1| < .01)) \Rightarrow (\Diamond_{[0,\tau]}(|\text{ACE}^1| < .01) \\ \wedge (\neg(|\text{ACE}^2| < .01)) \Rightarrow (\Diamond_{[0,\tau]}(|\text{ACE}^2| < .01) \end{aligned} \tag{7}$$

We encoded this formula and added the resulting constraints to the MPC problem as described in the previous sections, and solved it for different values of τ. Results are shown in Figure 2, and demonstrate that the STL constraint is correctly enforced in the stabilization of the ACE signal.

5. DISCUSSION

This paper presented work in progress on framing STL synthesis for controllers using an MPC formulation. The idea is to encode STL formulas as MILP constraints that can be efficiently solved by the same solver used to implement the model predictive control. A preliminary implementation was applied to a power systems case study. The continuation of this work will include a better characterization of the MPC problem feasibility, proof that the resulting controller enforces the STL specification, improved soundness and efficiency of encodings (especially in the case of liveness properties), and applications to further case studies.

6. ACKNOWLEDGEMENTS

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

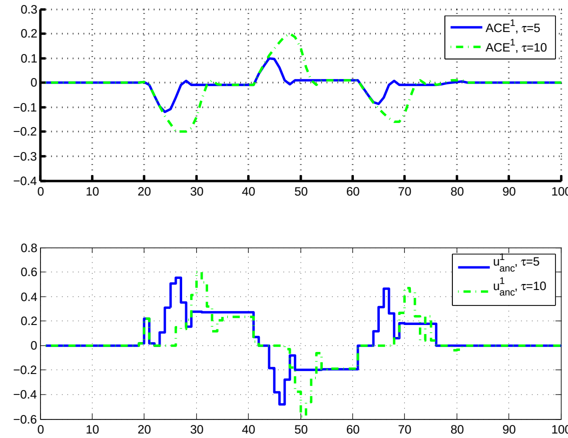

Figure 2: Comparison of ACE^1 stabilization using formula φ_t from (7) with $\tau = 5$ and $\tau = 10$. The controller correctly enforces the stabilisation delay in both cases.

7. REFERENCES

[1] R. Alur, T. A. Henzinger, G. Lafferriere, and G. J. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88(7):971–984, 2000.

[2] A. Biere, K. Heljanko, T. A. Junttila, T. Latvala, and V. Schuppan. Linear encodings of bounded LTL model checking. *Logical Methods in Comp. Sci.*, 2(5), 2006.

[3] A. Donzé, T. Ferrère, and O. Maler. Efficient robust monitoring for stl. In *CAV*, pages 264–279, 2013.

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

[5] M. Maasoumy, B. M. Sanandaji, A. Sangiovanni-Vincentelli, and K. Poolla. Model predictive control of regulation services from commercial buildings to the smart grid. In *IEEE American Control Conference (ACC)*, 2014.

[6] M. Maasoumy Haghighi. Controlling energy-efficient buildings in the context of smart grid: A cyber physical system approach. Technical Report UCB/EECS-2013-244, EECS Department, University of California, Berkeley, Dec 2013.

[7] O. Maler and D. Nickovic. Monitoring temporal properties of continuous signals. In *FORMATS/FTRTFT*, pages 152–166, 2004.

[8] R. M. Murray, J. Hauser, A. Jadbabaie, M. B. Milam, N. Petit, W. B. Dunbar, and R. Franz. Online control customization via optimization-based control. In *In Software-Enabled Control: Information Technology for Dynamical Systems*, pages 149–174. Wiley-Interscience, 2002.

[9] P. Nuzzo, H. Xu, N. Ozay, J. Finn, A. Sangiovanni - Vincentelli, R. Murray, A. Donze, and S. Seshia. A contract-based methodology for aircraft electric power system design. *Access, IEEE*, PP(99):1–1, 2013.

[10] E. M. Wolff, U. Topcu, and R. M. Murray. Optimization-based control of nonlinear systems with linear temporal logic specifications. In *ICRA*, 2014.