Belief-prefix Control for Autonomously Dodging Switching Disturbances

Kwesi Rutledge and Necmiye Ozay

***Abstract*— This paper presents a new method of controller synthesis for hidden mode switched systems, where the disturbances are the quantities that are affected by the unobserved switches. Rather than using model discrimination techniques that rely on modifying desired control actions to achieve identification, the controller uses consistency sets which map the measured external behaviors to a belief about which mode signal is being executed and a control action. This hybrid controller is a prefix-based controller, where the prefixes come from an offline constructed belief graph that incorporates prior information about switching sequences with potential reachable sets of the dynamics. While the mode signal is hidden to the controller, the system's location on the belief graph is fully observed and allows for this problem to be transformed into a design problem in which a discrete mode, in terms of beliefs, is directly observed. Finally, it is shown that affine controllers dependent on prefixes of such beliefs can be synthesized via linear programming.**

I. INTRODUCTION

In many safety-critical control applications, robustness to uncertainties is one of the major concerns. Assuming a large uniform bound on the uncertainty often leads to conservative designs or even renders the control objectives infeasible, whereas assuming a small uncertainty bound can risk safety. On the other hand, if we have some prior information on how the uncertainty sets evolve, it might be possible to mitigate conservativeness while still guaranteeing safety. Motivated by this observation, in this work we consider constrained control problems where the system is subject to additive disturbances and the disturbance sets switch among a predefined collection of sets. Moreover, we assume we are given a finite language that describes potential switching sequences. The examples of this scenario include aerial vehicles navigating gusty winds or autonomous robots interacting with moving obstacles that switch their intentions/targets. In both cases, although we might know about potential evolution of the uncertainty (the wind gust will not last too long, the moving obstacles will not switch their intentions back and forth too frequently), often times we cannot measure the switching sequence directly. This necessitates simultaneously estimating the switching mode from continuous outputs of the system while ensuring that control objectives are met. Similar to work on hidden mode hybrid systems [8], our solution approach relies on the construction of a perfect information problem on a belief space. In particular, we construct a belief graph that combines the prior information on the switching given by the language together with the potential reachable sets of the system under different control inputs. We then show how affine policies that depend on prefixes of beliefs can be computed via convex optimization.

This method can be compared with other approaches to create controllers for switching systems that guarantee that a system remains invariantly in a given set [2] or is asymptotically stable [10]. To the best of our knowledge, our approach provides an alternative to synthesize controls that solve robust reachability problems where the switching mode is not directly observed but is known to be constrained by a language.

A. Notation

Throughout this work we shall use capital letters to refer to matrices (e.g. A, B, C), lowercase letters to refer to vectors (e.g. x, v), and calligraphic capital letters to represent sets (e.g. $\mathcal{C}$, $\mathcal{L}$). The sets $\mathbb{R}$ and $\mathbb{N}$ will be notable exceptions to this rule. The symbol $\otimes$ is used to represent a Kroneker product between two matrices.

We will consider finite length sequences throughout this work and will apply the following means to represent them. Consider a vector-valued signal $s : \mathcal{N} \mapsto \mathcal{S}$ where $\mathcal{N} = \{x \in \mathbb{N} \mid x \leq \eta\}$ for some η and $\mathcal{S}$ is an arbitrary set. The value of the signal s at the discrete time $t_0 \in \mathcal{N}$ will be written as $s_{t_0} \in \mathcal{S}$. As a slight abuse of notation, we will sometimes treat the signal s as a vector (i.e. $s \in \mathcal{S}^{|\mathcal{N}|}$.)

When the codomain of the signal, $\mathcal{S}$, is a countable set we will refer to $\mathcal{S}$ as an *alphabet*. For a signal s, the length of the signal is denoted as $|s|$. For example, the signal $s = \alpha\beta\alpha\beta$ is defined on the alphabet $\mathcal{S} = \{\alpha, \beta\}$ and its length is 4.

II. PROBLEM STATEMENT

This section provides the model that we use to represent our systems and describes the problem that this work seeks to solve. First, consider linear systems with switched disturbances, represented by a triple $\Sigma = (\mathcal{D}, \mathcal{L}, \mathcal{X}_0)$, where $\mathcal{D}$ is a set of discrete-time update equations, $\mathcal{L}$ is a mode language describing how the mode of the system changes over time, and $\mathcal{X}_0$ is the set of initial states for the system. For each element $\Delta_{q_t} \in \mathcal{D}$:

$$\Delta_{q_t} : \begin{array}{rll} x_{t+1} = & Ax_t + Bu_t + w_t, & w_t \in \mathcal{W}_{q_t} \\ y_t = & Cx_t + v_t, & v_t \in \mathcal{V}_{q_t}. \end{array} \tag{1}$$

In the above equation, $x_t \in \mathbb{R}^n$ is the continuous state of the system at time t, q_t is the discrete state (or mode) of the system at time t, $u_t \in \mathcal{U} \subseteq \mathbb{R}^m$ is the input to the system

This work is supported in part by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE 1256260, NSF grant ECCS-1553873, ONR grant N00014-18-1-2501, and an Early Career Faculty grant from NASAs Space Technology Research Grants Program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies. Both authors are with the Electrical Engineering and Computer Science Department at the University of Michigan, Ann Arbor.

at time t, $w_t \in \mathcal{W}_{q_t} \subset \mathbb{R}^n$ is the process noise at time t, $y_t \in \mathbb{R}^p$ is the measured output at time t and $v_t \in \mathcal{V}_{q_t} \subset \mathbb{R}^p$ is the measurement noise at time t.

Assumption 1: It is assumed that the sets $\mathcal{W}_i$,$\mathcal{V}_i$, and $\mathcal{X}_0$ are given as polytopes.

Affine systems with switched disturbances as described above can be used to model consensus or robot manipulation tasks [4].

The mode signal itself is governed by the second part of the system S, called the language $\mathcal{L}$. If we consider the evolution of the mode signal over a finite time horizon T as a single object $q = [q_{t_0} \quad q_{t_0+1} \quad \cdots \quad q_{t_0+T-1}]^\top$, then the language $\mathcal{L}$ contains all allowed q vectors (i.e. all allowed trajectories of the mode signal) which comes from the problem specification.

Assumption 2: For simplicity, we consider languages where each word in the language has the same length (i.e. $|q| = T$ for all $q \in \mathcal{L}$).

Problem 1: Consider the system $\Sigma = (\mathcal{D}, \mathcal{L}, \mathcal{X}_0)$. Decide whether or not there exists an affine controller such that for each mode signal $q \in \mathcal{L}$ the state of the system is in the target set $\mathcal{X}_T$ at time $t = T$ (i.e. $x_T \in \mathcal{X}_T, \ \forall q \in \mathcal{L}$).

III. Solution Approach

Within this section, we first discuss how to derive hidden mode estimators using the structure of a system with switching disturbances. Second, we introduce the belief graph which captures the potential evolution of the information state in the finite time horizon problems that we are interested in. Then, we introduce the belief-prefix control law and several considerations when designing it. Finally, a sufficient condition for the existence of belief-prefix controllers that solve Problem 1 is proposed.

A. Mode Estimation

The mode signal q_t is not directly observed while the system Σ is operating. This prevents solutions such as those from [6] from being applied because those methods required the mode signal to be immediately available to the controller. To make use of the prefix-based results, we create a signal which is observable and can be interpreted as an estimate of the mode signal.

The information available for estimating the mode will be called the external behavior of the system:

Definition 1 (External Behavior): An external behavior e of the system $\Sigma = (\mathcal{D}, \mathcal{L}, \mathcal{X}_0)$ at time t is a tuple $e = (y_{[0:t]}, u_{[0:t-1]})$ where $y_{[0:t]} \in \mathbb{R}^{p(t+1)}$ is the measurement signal from time $\tau = 0$ to time $\tau = t$ and $u_{[0:t-1]} \in \mathbb{R}^{mt}$ is the input signal from time $\tau = 0$ to time $\tau = t-1$.

Because the dynamics in each mode is affine in (x, u, w, v) and the disturbance/initial sets are polytopes, whether or not external behavior is consistent with the word q is equivalent to a simple polytope inclusion test. Each of the polytopes that we use to perform mode estimation will be called consistency sets.

Definition 2 (Consistency Set for q): The consistency set $\mathcal{C}_\Sigma(q)$ of the mode signal $q \in \mathcal{L}$ for system $\Sigma = (\mathcal{D}, \mathcal{L}, \mathcal{X}_0)$ is the set of all external behaviors that are feasible for any initial condition starting somewhere in the initial state set $\mathcal{X}_0$ when the system operates under mode signal q. In math,

$$\mathcal{C}_\Sigma(q) \doteq \left\{ (y,u) \;\middle|\; \begin{array}{l} \exists (y,u,w,v,x_0,x) \in \mathbb{R}^{pT} \times \mathcal{U}^T \times \cdots \\ \quad \mathcal{W}^{(q)} \times \mathcal{V}^{(q)} \times \mathcal{X}_0 \times \mathbb{R}^{n(T+1)} : \\ x = Hw + Su + Jx_0 \\ y = \bar{C}x + v \end{array} \right\}$$

where

$$\mathcal{W}^{(q)} = \mathcal{W}_{q_0} \times \mathcal{W}_{q_1} \times \cdots \times \mathcal{W}_{q_{T-1}}$$

$$\mathcal{V}^{(q)} = \mathcal{V}_{q_0} \times \mathcal{V}_{q_1} \times \cdots \times \mathcal{V}_{q_{T-1}}$$

$$H = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ I & 0 & \cdots & 0 \\ A & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{T-1} & A^{T-2} & \cdots & I \end{bmatrix},$$

$$S = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{T-1}B & A^{T-2}B & \cdots & B \end{bmatrix}, \ J = \begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^{T-1} \end{bmatrix},$$

$$\bar{C} = \begin{bmatrix} I_T \otimes C & 0_{pT \times n} \end{bmatrix}$$

An external behavior e will be referred to as *consistent* with a word $q \in \mathcal{L}$ if $e \in \mathcal{C}_\Sigma(q)$. One can also extend this to discuss the consistency set for a sublanguage $\tilde{\mathcal{L}} \subseteq \mathcal{L}$.

Definition 3 (Consistency Set for a Language): The consistency set for the language $\tilde{\mathcal{L}} \subseteq \mathcal{L}$ of the system with switching disturbances $\Sigma = (\mathcal{D}, \mathcal{L}, \mathcal{X}_0)$ is the set of all external behaviors e that are consistent with every mode signal in $\tilde{\mathcal{L}}$. In math,

$$\mathcal{C}_\Sigma(\tilde{\mathcal{L}}) \doteq \bigcap_{q \in \tilde{\mathcal{L}}} \mathcal{C}_\Sigma(q)$$

Similarly, an external behavior e will be referred to as *consistent* with a sublanguage $\tilde{\mathcal{L}} \subseteq \mathcal{L}$ if $e \in \mathcal{C}_\Sigma(\tilde{\mathcal{L}})$. Note that, by definition, $\mathcal{C}_\Sigma(q)$ and $\mathcal{C}_\Sigma(\mathcal{L})$ are polytopes. The largest language $\tilde{\mathcal{L}} \subseteq \mathcal{L}$ such that e is consistent with $\tilde{\mathcal{L}}$ contains all words that our controller should be robust to. The following mode signal estimator identifies this maximal language $\tilde{\mathcal{L}}$ at every time step t:

$$\begin{aligned} b_t = \quad & \underset{\tilde{\mathcal{L}} \subseteq \mathcal{L}}{\arg\max} & & |\tilde{\mathcal{L}}| \\ & \text{subject to} & & e = (y_{[0:t]}, u_{[0:t-1]}) \in \mathcal{C}_S(\tilde{\mathcal{L}}_t) \end{aligned} \tag{2}$$

where $\tilde{\mathcal{L}}_t = \{ q \in \{1, ..., |\mathcal{D}|\}^t \mid \exists q' \in \tilde{\mathcal{L}}, q = q'_{[0:t-1]} \}$. The mode estimator (2) is equivalent to testing for set membership of $|\mathcal{L}|$ words and performing a union of all words that pass the test. Set membership approaches for model invalidation such as [3], [5] are performing similar tests using optimization. The sublanguage b_t is an estimate of which mode signals may be active at time t, i.e., $\mathbf{b}_t$ is a *belief* about the mode signal at time t. Moreover, by construction, this estimation procedure guarantees that the true mode signal q^* is contained in b_t for all t.

B. Belief Graphs

In this section, we discuss the belief graph, a model that will help us reason about what subset of the hidden mode signals may be feasible.

Definition 4 (Belief Graph): A *belief graph* $G_\Sigma = (\mathcal{N}, \mathcal{E})$ for the system $\Sigma = (\mathcal{D}, \mathcal{L}, \mathcal{X}_0)$ is a directed graph for which each node $n = (b, t) \in \mathcal{N}$ contains:

- a belief $b \in 2^{\mathcal{L}} \setminus \emptyset$ (referred to as $n.b$), and
- a time at which that belief is held t (referred to as $n.t$)

and each edge represents a feasible transition from one belief at a given moment to another belief at the next moment. In other words:

1) for all $e = (n_i, n_j) \in \mathcal{E}$, $n_i.t + 1 = n_j.t$
2) for all $e = (n_i, n_j) \in \mathcal{E}$, $n_i.b \supseteq n_j.b$

The construction of a belief graph can be done offline using Algorithms 1 and 2.

Algorithm 1: BeliefGraph(Σ) - Finds the belief graph associated with $\Sigma = (\mathcal{D}, \mathcal{L}, \mathcal{X}_0)$, a system with switching disturbances.

Result: $G_\Sigma = (\mathcal{N}, \mathcal{E})$ the belief graph for system Σ.

$t \leftarrow 0$;
$n_0 \leftarrow (\mathcal{L}, t)$;
$\mathcal{N} \leftarrow \{n_0\}$;
$\mathcal{N}_t \leftarrow \mathcal{N}$;
$\mathcal{E} \leftarrow \emptyset$;
while $t \leq T - 1$ **do**
  $\mathcal{N}_{t+1} \leftarrow \emptyset$;
  for $n \in \mathcal{N}_t$ **do**
    $\mathcal{B} \leftarrow$ Post(n);
    $\mathcal{N}_{t+1} \leftarrow \mathcal{N}_{t+1} \cup \mathcal{B}$;
    $\mathcal{N} \leftarrow \mathcal{N} \cup \mathcal{B}$;
    for $n' \in \mathcal{B}$ **do**
      $\mathcal{E} \leftarrow \mathcal{E} \cup \{(n, n')\}$
    end
  end
  $\mathcal{N}_t \leftarrow \mathcal{N}_{t+1}$;
  $t \leftarrow t + 1$;
end

Algorithm 1 constructs a belief graph by identifying the beliefs that can be held at time step 0, then time step 1, and so on in the following fashion. Consider a belief node $n = (b, t)$ associated with the current time step t. The algorithm considers the consistency sets for every *belief* $\tilde{b} \in 2^b$ that can follow the current belief b i.e., $\{C_\Sigma(\tilde{b})\}_{\tilde{b} \in 2^b}$. By identifying which consistency sets overlap, intersect or are empty, the algorithm determines which future beliefs $\tilde{b}$ are indistinguishable from one another or are impossible to hold (see Algorithm 2). After pruning the set of beliefs accordingly, we are left with a set of belief nodes $\mathcal{B}$ that are uniquely feasible at time $t + 1$. This process is repeated for every time $t = 0$ to $t = T - 1$ and every node $n = (b, t)$ found at each of those times to create the complete belief graph.

Algorithm 2: Post(n_i) - Finds all belief nodes that can follow $n_i = (b_i, t_i)$.

Result: $\mathcal{N}_{t+1}(n_i)$ the set of all feasible belief nodes at time instant $t_i + 1$ that can follow the current belief node n_i.

$\mathcal{P}(b_i) \leftarrow$ powerset_of(b_i);
$\mathcal{N}_{t+1}(n_i) \leftarrow \emptyset$;
for $\tilde{\mathcal{L}} \in \mathcal{P}(b_i)$ **do**
  flag_imposs $\leftarrow$ false;
  $\tilde{\mathcal{L}}_t \leftarrow$ length_t_truncation($\tilde{\mathcal{L}}$);
  if *is_empty_set*($\mathcal{C}_\Sigma(\tilde{\mathcal{L}}_t)$) **then**
    flag_imposs $\leftarrow$ true;
  end
  flag_unobs $\leftarrow$ false;
  for $\mathcal{L}' \in \mathcal{P}(b_i) \setminus \{\tilde{\mathcal{L}}\}$ **do**
    $\mathcal{L}'_t \leftarrow$ length_t_truncation($\mathcal{L}'$);
    if $|\tilde{\mathcal{L}}_t| \leq |\mathcal{L}'_t|$ *AND* $\mathcal{C}_\Sigma(\tilde{\mathcal{L}}_t) \subseteq \mathcal{C}_\Sigma(\mathcal{L}'_t)$ **then**
      flag_unobs $\leftarrow$ true;
    end
  end
  if *not(flag_imposs OR flag_unobs)* **then**
    $\tilde{n} \leftarrow (\tilde{\mathcal{L}}, t_i + 1)$;
    $\mathcal{N}_{t+1}(n_i) \leftarrow \mathcal{N}_{t+1}(n_i) \cup \{\tilde{n}\}$
  end
end

Remark 1: The main purpose of the belief graph is to identify against what subset of mode signals the control actions should be robust as will be described in section III-D. However, as a side benefit, one can improve on the estimation process in (2) by using the belief graph G_Σ by searching only over the words that appear in the belief node of the previous step, instead of searching over the entire $\mathcal{L}$ (i.e., by replacing $\mathcal{L}$ in (2) with b_{t-1}).

By finding all of the paths from the root node (the single node with $t = 0$) to any leaf node (any node that is a 'sink' in terms of edges) on this belief graph, we may create a new language, $\mathcal{L}_B$.

Definition 5 (Belief Language): The belief language $\mathcal{L}_B$ for system $\Sigma = (\mathcal{D}, \mathcal{L}, \mathcal{X}_0)$ is the language:

$$\begin{aligned}\mathcal{L}_B = \big\{ \mathbf{b} = (b_1, \ldots, b_T) \in (2^{\mathcal{L}})^T \;\; | \\ \exists (n_1, n_2, \ldots, n_T) \in \text{Paths}(G_\Sigma) \text{ s.t.} \\ b_i = n_i.b \quad \forall i \in [1, T] \big\}\end{aligned}$$

where G_Σ is the belief graph associated with system Σ and Paths(G_Σ) is all maximal paths of G_Σ (i.e. all paths of length T).

$\mathcal{L}_B$ is a language of beliefs and, as each belief is found by checking if the external behavior is a member of the polytope $\mathcal{C}_\Sigma(\mathcal{L}_B)$, we know that the evolution of beliefs is fully observed. This fact will be later used to design controller gains that are robust to multiple words simultaneously, where the controller design process will be effectively reduced to a perfect information problem on the beliefs.

C. Finite Horizon Control Laws

In this section, we introduce the form of prefix-based controller and describe how to synthesize such a controller using Q-parameterization [9], [7].

We consider a control law similar to the finite horizon affine feedback used in [7], [1]. But instead of being parameterized by the current time t, it will be parameterized by the sequence of belief states visited thus far, $\mathbf{b}_{t_0:t}$:

$$u(\mathbf{b}_{t_0:t}) = f_{(t,\mathbf{b}_{t_0:t})} + \sum_{\tau=t_0}^{t} F_{(t,\tau,\mathbf{b}_{t_0:t})} y_\tau, \tag{3}$$

where $\mathbf{b} \in \mathcal{L}_B$, and the matrices $f_{(t,\mathbf{b}_{t_0:t})}$ and $F_{(t,\tau,\mathbf{b}_{t_0:t})}$ are of appropriate dimension. Following the vector notation used in Definition 2, we can describe the output of a *prefix-based* control law over the time horizon when the sequence of beliefs is $\mathbf{b}$ with the following equation:

$$u^{(\mathbf{b})} = f^{(\mathbf{b})} + F^{(\mathbf{b})} y \tag{4}$$

where $f_{(t,\mathbf{b}_{t_0:t})} \in \mathbb{R}^m$ is the $(t+1)^{th}$ entry of the block vector $f^{(\mathbf{b})}$ and $F_{(t,\tau,\mathbf{b}_{t_0:t})} \in \mathbb{R}^{m \times p}$ is the $(t+1,\tau)^{th}$ block entry in $F^{(\mathbf{b})}$. In math, $f^{(\mathbf{b})}_{t+1} \doteq f_{(t,\mathbf{b}_{t_0:t})}$ and $F^{(\mathbf{b})}_{t+1,\tau} \doteq F_{(t,\tau,\mathbf{b}_{t_0:t})}$.

To make the closed loop expressions for x and u easier to optimize, we introduce the following two variables $r^{(\mathbf{b})}$ and $Q^{(\mathbf{b})}$ defined as follows:

$$Q^{(\mathbf{b})} \doteq F^{(\mathbf{b})}(I - \bar{C}SF^{(\mathbf{b})})^{-1}, \quad r^{(\mathbf{b})} \doteq (I + Q^{(\mathbf{b})}\bar{C}S)f^{(\mathbf{b})} \tag{5}$$

Note that the mapping between $(f^{(\mathbf{b})}, F^{(\mathbf{b})})$ and $(r^{(\mathbf{b})}, Q^{(\mathbf{b})})$ is bijective [7], so one pair uniquely determines the other. Furthermore, the closed loop state and input sequences can now be written as follows:

$$\begin{bmatrix} x^{(\mathbf{b})} \\ u^{(\mathbf{b})} \end{bmatrix} = P^{(\mathbf{b})} \begin{bmatrix} w \\ v \end{bmatrix} + \begin{bmatrix} \tilde{x}^{(\mathbf{b})} \\ \tilde{u}^{(\mathbf{b})} \end{bmatrix}, \tag{6}$$

where

$$P^{(\mathbf{b})} = \begin{bmatrix} P^{(\mathbf{b})}_{xw} & P^{(\mathbf{b})}_{xv} \\ P^{(\mathbf{b})}_{uw} & P^{(\mathbf{b})}_{uv} \end{bmatrix},$$

$$\begin{aligned} P^{(\mathbf{b})}_{xw} &= H + SQ^{(\mathbf{b})}\bar{C}H, & P^{(\mathbf{b})}_{xv} &= SQ^{(\mathbf{b})}, \\ P^{(\mathbf{b})}_{uw} &= Q^{(\mathbf{b})}\bar{C}H, & P^{(\mathbf{b})}_{uv} &= Q^{(\mathbf{b})}, \end{aligned}$$

and

$$\begin{aligned} \tilde{x}^{(\mathbf{b})} &= (I + SQ^{(\mathbf{b})}\bar{C})Jx_0 + Sr^{(\mathbf{b})}, \\ \tilde{u}^{(\mathbf{b})} &= Q^{(\mathbf{b})}\bar{C}Jx_0 + r^{(\mathbf{b})}. \end{aligned}$$

Now, $x^{(j)}$ and $u^{(j)}$ both depend linearly on $Q^{(j)}$ and $r^{(j)}$, as well as the variables w and v. The disturbances w and v belong to the sets $\mathcal{W}^{(i)}$ and $\mathcal{V}^{(i)}$, respectively, and both depend on the mode signal q which is not known.

This follows from the way that beliefs evolve over time. $\mathbf{b}_T$ is the belief after receiving all of the available external behavior information from a word $q \in \mathcal{L}$. Thus, a controller for this belief sequence need only to be robust towards all possible mode signals $q \in \mathbf{b}_T$.

D. Control Synthesis

The components outlined in the previous subsections can be combined to propose a solution for Problem 1. The solution will be a hybrid controller which uses the belief graph to control the switching of its gains.

The belief language (i.e., all paths of the belief graph) is incorporated into the controller design using a prefix-based, finite-horizon affine control law on the *observed sequence of beliefs*. Any controller (3) for belief language $\mathcal{L}_B = \mathcal{L}(G_\Sigma)$ that uses prefixes of beliefs is parameterized by a unique set of matrices $\{f^{(i)}, F^{(i)}\}_{i=1}^{|\mathcal{L}_B|}$ where the blocks of each matrix are determined as they were in (3).

The set of matrices must also satisfy the following constraints. For every word ϕ that is a prefix of two or more elements in the belief language (e.g., $\phi \in \text{Pref}(\mathbf{b}^{(i)} \cap \text{Pref}(\mathbf{b}^{(h)})$ for $\mathbf{b}^{(i)}, \mathbf{b}^{(j)} \in \mathcal{L}_B$), the gains associated with each word must satisfy:

$$\begin{aligned} F^{(i)}_{jk} &= F^{(h)}_{jk} \quad \forall 1 \leq j \leq |p|, k \leq j, \\ f^{(i)}_{j} &= f^{(h)}_{j} \quad \forall 1 \leq j \leq |p|. \end{aligned}$$

Because (3) is a function and because our belief detection process is causal, an additional constraint is placed on the control law: The controller must output the same value when the observed sequence of beliefs is the same even if that same sequence belongs to two different belief words $\mathbf{b}^{(i)}$ and $\mathbf{b}^{(h)}$ i.e. if $\mathbf{b}^{(i)}_{t_0:t} = \mathbf{b}^{(j)}_{t_0:t}$, then $f^{(i)}_j = f^{(h)}_j$ and $F^{(i)}_{jk} = F^{(h)}_{jk}$. This requires carefully modifying the Theorem 1 in [6] to account for beliefs as we do next.

Theorem 1: Given a system with switching disturbances $\Sigma = (\mathcal{D}, \mathcal{L}, \mathcal{X}_0)$, consider the belief language $\mathcal{L}_B$ constructed by applying Algorithms 1 and 2. The set of all feasible prefix-based, belief space controllers $\left\{f^{(i)}, F^{(i)}\right\}_{i=1}^{|\mathcal{L}_B|}$ as in (3) is bijective to the following polyhedral set:

$$\mathcal{Q}(\mathcal{L}_B) \doteq \left\{ \begin{array}{c|l} \{(Q^{(i)}, & Q^{(i)} \text{ is block lower diagonal,} \\ r^{(i)})\}_{i=1}^{|\mathcal{L}_B|} & (p \in Pref(\mathbf{b}^{(i)}) \cap Pref(\mathbf{b}^{(j)})) \Longrightarrow \\ & \quad (\mathcal{BM}_{|p|}(Q^{(i)}) = \mathcal{BM}_{|p|}(Q^{(j)})) \\ & \quad\quad \wedge \left((r^{(i)})_{1:|p|m} = (r^{(j)})_{1:|p|m}\right), \\ & \quad \forall \mathbf{b}^{(i)}, \mathbf{b}^{(j)} \in \mathcal{L}_B \end{array} \right\} \tag{7}$$

where $Q^{(j)}$ is used as the shorthand for $Q^{b^{(j)}}$, $\mathcal{BM}_a(Q^{(i)})$ is the $a \times a$ leading block submatrix of the matrix $Q^{(i)}$ (as in [6]).

The proof follows from the relation (5) and the proof of Theorem 1 in [6]. A prefix-based belief space control law that solves Problem 1 can then be found using the following proposition:

Proposition 1: Suppose that the state x_t of system $\Sigma = (\mathcal{D}, \mathcal{L}, \mathcal{X}_0)$ is given. Construct the belief graph $G_\Sigma = (\mathcal{N}, \mathcal{E})$ according to Algorithms 1 and 2. The belief graph also defines the language of beliefs $\mathcal{L}_B$. If the following robust linear programming problem is feasible

$$\text{Find} \quad \left\{Q^{(i)}, r^{(i)}\right\}_{i=1}^{|\mathcal{L}_B|} \in \mathcal{Q}(\mathcal{L}_B) \tag{8a}$$

$$\text{s.t.} \quad \forall \mathbf{b}^{(j)} \in \mathcal{L}_B : \forall q^{(i)} \in b_T^{(j)} : \tag{8b}$$

$$\forall (w^{(i)}, v^{(i)}, x_0) \in \mathcal{W}^{(i)} \times \mathcal{V}^{(i)} \times \mathcal{X}_0 : \tag{8c}$$

$$R_T(P_{xw}^{(j)} w^{(i)} + P_{xv}^{(j)} v^{(i)} + \tilde{x}^{(j)}) \in \mathcal{X}_T, \tag{8d}$$

where $R_T = \begin{bmatrix} 0_{n \times nT} & I_n \end{bmatrix}$ and $P_{xw}^{(j)}$, $P_{xv}^{(j)}$ and $\tilde{x}^{(j)}$ are defined as they are in (6) for $\mathbf{b}^{(j)}$, then its solution defines a controller that solves Problem 1.

Proof: The proof is nearly identical to that of Theorem 2 of [6] with the main difference being that the belief $\mathbf{b}^{(j)}$ demands that the controller gains $(Q^{(j)}, r^{(j)})$ be robust to any word $q \in b_T^{(j)}$. By construction of the belief graph, a controller using (2) to generate the belief sequence $\mathbf{b}^*$ guarantees that the true mode signal of the system q^* satisfies $q^* \in b_T^*$. ∎

Note that the answer to (8) defines the gains that constitute the finite horizon control law on the prefixes of the belief language $\mathcal{L}_B$ for system Σ and the mapping of external behavior to the proper gain is done using the estimation process (2) or its simplified version that uses the consistency sets which make up the belief graph G_Σ as described in Remark 1.

The feasibility problem in Proposition 1 may be simplified into a problem with fewer constraints using duality:

Theorem 2: There exists a prefix-based, finite horizon controller on the beliefs with the form (3) for system with switched disturbances $\Sigma = (\mathcal{D}, \mathcal{L}, \mathcal{X}_0)$ that solves Problem 1 if the following linear program is feasible:

$$\begin{array}{ll} \text{Find} & \left\{Q^{(i)}, r^{(i)}\right\}_{i=1}^{|\mathcal{L}_B|} \in \mathcal{Q}(\mathcal{L}_B), \\ & \left\{\{\Pi_1^{(i,j)}, \Pi_u^{(i,j)},\}_{j=1}^{|\mathbf{b}^{(i)}|}\right\}_{i=1}^{|\mathcal{L}_B|} \\ \text{s.t.} & \forall \mathbf{b}^{(i)} \in \mathcal{L}_B : \\ & \quad \forall q^{(j)} \in b_T^{(i)} : \\ & \qquad \Pi_1^{(i,j)} \geq 0, \Pi_u^{(i,j)} \geq 0 \\ & \qquad \Pi_1^{(i,j)} \Gamma_\eta^{(j)} = \Gamma_T R_T G^{(i)} \\ & \qquad \Pi_1^{(i,j)} \gamma_\eta^{(j)} \leq \gamma_T - \Gamma_T R_T S r^{(i)} \\ & \qquad \Pi_u^{(i,j)} \Gamma_\eta^{(j)} = \Gamma_u G_u^{(i)} \\ & \qquad \Pi_u^{(i,j)} \gamma_\eta^{(j)} \leq \gamma_u - \Gamma_u r^{(i)} \end{array} \tag{9}$$

where $\{\Gamma_\eta^{(i)} \eta \leq \gamma_\eta^{(i)}\}$ is the hyperplane representation of the polytope $N^{(i)} = \mathcal{W}^{(i)} \times \mathcal{V}^{(i)} \times \mathcal{X}_0$, $\{\Gamma_T x \leq \gamma_T\}$ is the hyperplane representation of the polytope $\mathcal{X}_T$, $\{\Gamma_u x \leq \gamma_u\}$ is the hyperplane representation of the polytope $\mathcal{U}$,

$$G^{(i)} = \begin{bmatrix} P_{xw}^{(i)} & P_{xv}^{(i)} & (I + SQ^{(i)} \bar{C}) J \end{bmatrix},$$

$$G_u^{(i)} = \begin{bmatrix} P_{uw}^{(i)} & P_{uv}^{(i)} & Q^{(i)} \bar{C} J \end{bmatrix}.$$

IV. Examples

This section discusses two examples that illustrate the proposed method. The first example describes graphically how a belief graph is constructed using a simple two-dimensional system. In the second example, a pair of agents attempts to reach a consensus region during a time window in which agents receive a large gust of wind. The controller obtained for this example allows for the pair of agents to reach consensus despite the windy circumstances and the unknown time at which the wind occurs.

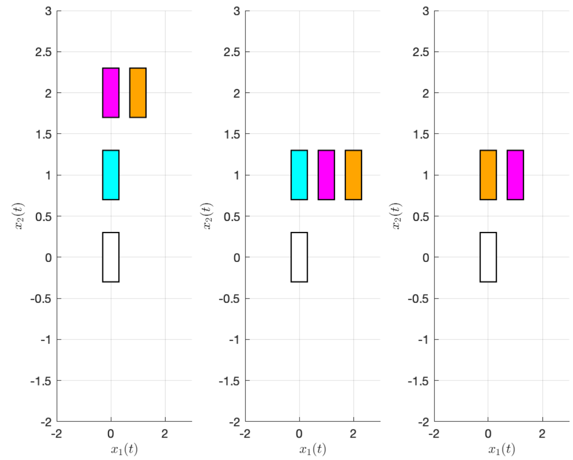

Fig. 1. The external behavior of the system $\Sigma^{(1)}$ at time t lies within one of the rectangles shown in one of the plots above. The rectangles are color coded for the time at which the state of the system would be contained in that rectangle (white for $t = 0$, cyan for $t = 1$, magenta for $t = 2$, and orange for $t = 3$) and each plot corresponds to which mode signal is occuring for the system $\Sigma^{(1)}$ (left plot is for mode signal 1 in $\mathcal{L}^{(1)}$, center for the second mode signal in $\mathcal{L}^{(1)}$ and right for the third signal in $\mathcal{L}^{(1)}$). Therefore, if $\Sigma^{(1)}$ is operating under mode signal 2, then at time $t = 2$ the state of the system will be located in the magenta rectangle in the center plot.

A. Belief Graph Examples

First, we present a simple dynamical system which illustrates how the values of a system with switching disturbances determines the structure of a Belief Graph. Consider the system $\Sigma^{(1)} = (\mathcal{D}^{(1)}, \mathcal{L}^{(1)}, \mathcal{X}_0^{(1)})$ where $\Delta_{q_t} \in \mathcal{D}^{(1)}$ is defined as:

$$\Delta_{q_t} : \begin{array}{ll} x_{t+1} & = x_t + w_t, \quad w_t \in \mathcal{W}_i \\ y_t & = x_t \end{array}$$

where $x_t \in \mathbb{R}^2$, $|\mathcal{D}^{(1)}| = 4$ and

$$\begin{array}{ll} \mathcal{W}_1 = \left\{\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right\}, & \mathcal{W}_2 = \left\{\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right\}, \\ \mathcal{W}_3 = \left\{\begin{bmatrix} 0 \\ -1 \end{bmatrix}\right\}, & \mathcal{W}_4 = \left\{\begin{bmatrix} -1 \\ 0 \end{bmatrix}\right\}. \end{array} \tag{10}$$

The language $\mathcal{L}^{(1)}$ contains three words defined as follows:

$$\mathcal{L}^{(1)} = \{\{1,1,2,3\}, \{1,2,2,4\}, \{1,2,4,3\}\}$$

the evolution of the state according to each of these words is illustrated by Figure 1 when the initial state of the system is known to belong to the set $\mathcal{X}_0^{(1)} = \{x \in \mathbb{R}^2 \mid \|x\|_\infty \leq 0.3\}$.

Note that the first mode in all words of $\mathcal{L}^{(1)}$ is the same (i.e. the second component of x_0 is increased by 1). Thus in the belief graph shown in 2 the initial node (node 1) has

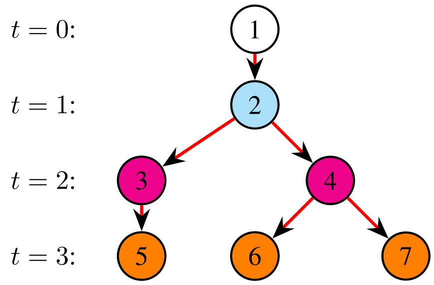

Fig. 2. The belief graph for the system with switching disturbances $\Sigma_1 = (\mathcal{D}^{(1)}, \mathcal{L}^{(1)}, \mathcal{X}_0^{(1)})$. The nodes with the same value of t are displayed with the same y-value (they lie on a horizontal line together as indicated by the text on the left).

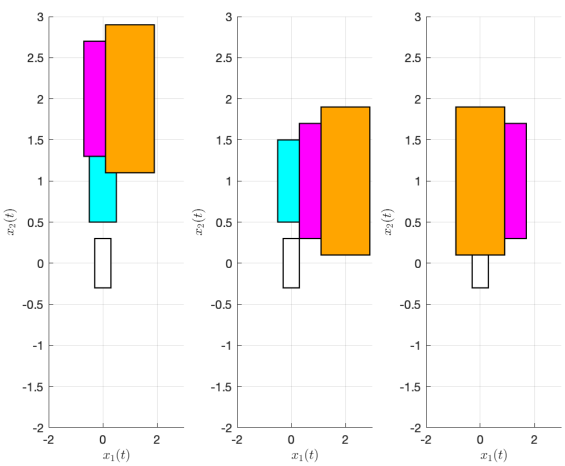

Fig. 3. The external behavior of the system $\Sigma^{(2)}$ at time t lies within one of the rectangles shown in one of the plots above. The rectangles are color coded for the time at which the state of the system would be contained in that rectangle (white for $t = 0$, cyan for $t = 1$, magenta for $t = 2$, and orange for $t = 3$) and each plot corresponds to which mode signal is occuring for the system $\Sigma^{(2)}$ (left plot is for mode signal 1 in $\mathcal{L}^{(1)}$, center for the second mode signal in $\mathcal{L}^{(1)}$ and right for the third signal in $\mathcal{L}^{(1)}$). Therefore, if $\Sigma^{(2)}$ is operating under mode signal 2, then at time $t = 2$ the state of the system will be located in the magenta rectangle in the center plot.

only 1 predecessor because it is impossible to differentiate any of the words based on the trajectory up to time $t = 1$. To put this in terms of our belief graph, the initial node n_1 has $b_1 = \mathcal{L}^{(1)}$ by definition and there is only one node with value $\tau = 1$, n_2. Because the first mode in every word from $\mathcal{L}^{(1)}$ is the same, we cannot remove any words from the first belief of b_1 and thus node 2's belief b_2 satisfies $b_2 = b_1 = \mathcal{L}^{(1)}$.

The prefix $\{1, 2\}$ is shared by the second and third words in $\mathcal{L}^{(1)}$. This manifests itself in Figure 1 as the magenta box of the second and third word are in the same location and it manifests itself in the Belief Graph (Figure 2) as the node n_4 in the belief graph which is a predecessor of n_2 where $b_4 = \{q^{(2)}, q^{(3)}\}$.

Now, consider the slightly modified system $\Sigma^{(2)} = (\mathcal{D}^{(2)}, \mathcal{L}^{(1)}, \mathcal{X}_0^{(2)})$ where $\Delta_{q_t} \in \mathcal{D}^{(2)}$ is defined as:

$$\Delta_{q_t} : \begin{array}{rll} x_{t+1} & = x_t + w_t, & w_t \in \mathcal{W}_i' \\ y_t & = x_t + v_t & v_t \in \mathcal{V} \end{array}$$

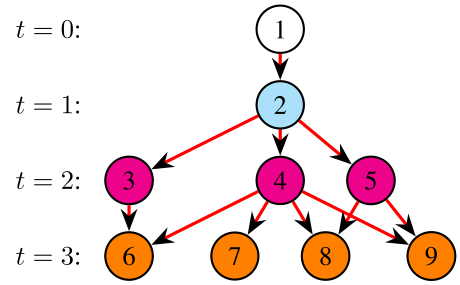

Fig. 4. The belief graph for system with switching disturbances $\Sigma^{(2)} = (\mathcal{D}^{(2)}, \mathcal{L}^{(1)}, \mathcal{X}_0^{(2)})$.

with $|\mathcal{D}^{(2)}| = 4$, $\mathcal{W}_i' = \mathcal{W}_i + \{x \in \mathbb{R}^2 \mid \|x\|_\infty \leq 0.2\}$, $\mathcal{W}_i$ is defined as in (10), and $\mathcal{V} = \{v \in \mathbb{R}^2 \mid \|v\|_\infty \leq 0.2\}$. In this case, the system begins from within the initial state set $\mathcal{X}_0^{(2)} = \{x \in \mathbb{R}^2 \mid \|x\|_\infty \leq 0.1\}$.

All of the edges and belief nodes in the graph for S_1 (shown in Figure 2) are contained within the graph for $\Sigma^{(2)}$ (shown in Figure 4), as expected. To give a flavor for why there are more nodes in Figure 4 than in Figure 2, look at the magenta boxes in Figure 3. In fact, all of the magenta boxes overlap (one of the corners of the magenta rectangle in the left plot overlaps with a corner of the magenta rectangles in the center and right-most plots). But note that there are parts of the magenta square in the left plot that do not overlap with the other two magenta rectangles. This means that some external behaviors of the first word (corresponding to the left plot) will be consistent with only the first word, but if the trajectories end up in the bottom corner that overlaps with other magenta rectangles then that means that the external behavior could have been generated by any three of the words in $\mathcal{L}^{(1)}$. Thus, at time $t = 2$, there should be three belief nodes, where an additional node with belief $b_i = \mathcal{L}^{(1)}$.

B. Tracking Under Changing Disturbance Sets

Consider the system that represents a pair of drones occupying a two-dimensional state space $\Sigma^{(3)}(T) = (\mathcal{D}^{(3)}, \mathcal{L}^{(3)}(T), \mathcal{X}_0^{(3)})$. While accomplishing a surveillance task, they must maintain a grid formation in the 2d plane, but at an upcoming but unknown time there will be a strong, sustained gust of wind. The dynamics for this pair of drones is governed by $\Delta_{q_t} \in \mathcal{D}^{(3)}$:

$$\Delta_{q_t} : \begin{array}{rll} x_{t+1} = & x_t + \delta t \cdot (u_t + B_w w_t), & w_t \in \bar{\mathcal{W}}_i \\ y_t = & x_t + v_t, & v_t \in \bar{\mathcal{V}} \end{array}$$

where $x_t = [e_x^{(1)} \;\; e_y^{(1)} \;\; e_x^{(2)} \;\; e_y^{(2)}]^\top$ is the state, $u_t = [u_x^{(1)} \;\; u_y^{(1)} \;\; u_x^{(2)} \;\; u_y^{(2)}]^\top$ is the input provided in the x- and y-directions for each drone, and δt is the sampling time of the discrete-time system. The process noise $w = [w_x \; w_y]^\top$ represent the velocity that the wind is moving in the x- and y-directions, and affects each drone identically $B_w = [I_2 \; I_2]^\top$. The two different modes representing wind speeds are defined as follows:

$$\bar{\mathcal{W}}_1 \doteq \{w \in \mathbb{R}^2 \mid \|w\|_\infty \leq \eta_w\}, \quad \bar{\mathcal{W}}_2 \doteq \mathcal{W}_1 + \eta_w \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

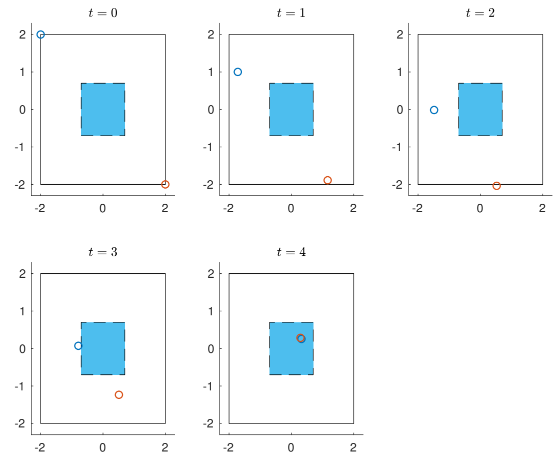

Fig. 5. In this scenario, the consensus system is trying to reach a smaller hypercube within its initial state set. To emphasize the effectiveness of this approach, we place the initial conditions on the boundary of the set $\mathcal{X}_0$, and provide a very strong 'wind' from the word $q \in \mathcal{L}^{(3)}(4)$ to the system, yet the agents are still able to reach the desired consensus region.

System $\Sigma^{(3)}(T)$	Belief Graph Construction Time (s)	Belief Graph Size $(\lvert\mathcal{N}\rvert, \lvert\mathcal{E}\rvert)$	LP Solve Time (s)
$T = 4$	216	(12,15)	1.32
$T = 5$	2101	(19,34)	6.57
$T = 6$	15007	(26,53)	22.1

TABLE I

COMPUTATION TIME FOR BELIEF GRAPH CONSTRUCTION AND LINEAR PROGRAM SOLUTION FOR SYSTEM $\Sigma^{(3)}(T)$.

where $\eta_w = 0.5$. The measurement noise $v \in \mathbb{R}^4$ affects each sensor independently and is not affected by the mode. This will be encapsulated in the set $\bar{\mathcal{V}} = \{v \in \mathbb{R}^4 \mid \|v\| \leq \eta_v\}$ where $\eta_v = 0.2$.

Suppose that at an unknown time t^* the dynamical system switches to a high-wind environment (i.e. mode 2). This is modeled by the following potential mode signals:

$$\mathcal{L}^{(3)}(T) = \left\{ \begin{array}{l} \{1,2,2\} \cdot \{1\}^{T-3}, \\ \{1,1,2\} \cdot \{1\}^{T-3}, \\ \{1,2,1\} \cdot \{1\}^{T-3} \end{array} \right\}$$

We would like to guarantee that the system can reach a tighter form of consensus from the initial state when it doesn't know when the strong wind will hit, but has the model $\mathcal{L}^{(3)}$ for the arrival. As shown in Figure 5, the controller correctly forces the system to enter the desired consensus state when the unknown mode signal is applied.

V. CONCLUSIONS

The problem of achieving a task or specification while a hidden system mode is switching is a difficult task, with wide applicability from spacecraft flight control to autonomous vehicle cruise control. Unlike other approaches to this problem, which typically design a mode estimator or discriminator that operates separately from a robust controller, we provide a unified controller which internally keeps an estimate of the unknown mode. When the set of allowable switching sequences is represented by a language, the approach that we provide in this paper constructs a belief graph and develops a prefix-based controller for the paths along that belief graph. The control problem on the belief graph is a perfect information problem on the belief space and thus we are able to apply the insights of earlier prefix-based control problems to formulate the robust reachability problem as a robust linear programming problem. The controllers created from this method are applied to a consensus problem where a dispersed set of agents is able to return to a tighter configuration in the presence of wind.

Future work will focus on extending this method so that it can be applied to a larger class of switched systems. Systems with switching input matrices $\{B_i\}_{i=1}^{|\mathcal{D}|}$ or measurement matrices $\{C_i\}_{i=1}^{|\mathcal{D}|}$ appear often in contexts such as fault-tolerant control and thus this method would find immediate use. We also would like to consider the inclusion of priorities into reachability problems, so that if a specific branch of the belief graph cannot reach the target then it can switch to trying to achieve a simpler task with less priority and the controller synthesis problem does not have to fail because of a single wayward branch. We are also interested in developing belief graph abstractions to improve the scalability of this method.

REFERENCES

[1] S. Amin, A. A. Cárdenas, and S. S. Sastry. Safe and secure networked control systems under denial-of-service attacks. In R. Majumdar and P. Tabuada, editors, *Hybrid Systems: Computation and Control*, pages 31–45, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[2] M. Dehghan and C.-J. Ong. Characterization and computation of disturbance invariant sets for constrained switched linear systems with dwell time restriction. *Automatica*, 48(9):2175 – 2181, 2012.

[3] F. Harirchi and N. Ozay. Model invalidation for switched affine systems with applications to fault and anomaly detection**this work is supported in part by darpa grant n66001-14-1-4045. *IFAC-PapersOnLine*, 48(27):260 – 266, 2015. Analysis and Design of Hybrid Systems ADHS.

[4] D. Meng, Y. Jia, and J. Du. Robust consensus tracking control for multiagent systems with initial state shifts, disturbances, and switching topologies. *IEEE Transactions on Neural Networks and Learning Systems*, 26(4):809–824, April 2015.

[5] N. Ozay, M. Sznaier, and C. Lagoa. Convex certificates for model (in) validation of switched affine systems with unknown switches. *IEEE Transactions on Automatic Control*, 59(11):2921–2932, 2014.

[6] K. Rutledge, S. Z. Yong, and N. Ozay. Finite horizon constrained control and bounded-error estimation in the presence of missing data. *Nonlinear Analysis: Hybrid Systems*, 36:100854, 2020.

[7] J. Skaf and S. P. Boyd. Design of affine controllers via convex optimization. *IEEE Trans. on Automatic Control*, 55(11):2476–2487, Nov 2010.

[8] R. Verma and D. Del Vecchio. Control of hybrid automata with hidden modes: Translation to a perfect state information problem. In *49th IEEE Conference on Decision and Control (CDC)*, pages 5768–5774. IEEE, 2010.

[9] D. Youla, H. Jabr, and J. Bongiorno. Modern wiener-hopf design of optimal controllers–part ii: The multivariable case. *IEEE Transactions on Automatic Control*, 21(3):319–338, June 1976.

[10] L. Zhang, S. Zhuang, and R. D. Braatz. Switched model predictive control of switched linear systems: Feasibility, stability and robustness. *Automatica*, 67:8 – 21, 2016.