Optimal planning on register automata

Jie Fu and Herbert G. Tanner

***Abstract*— This paper addresses the optimal planning problem on register automata, a special class of finite state machines which can process continuous inputs. Register automata emerge as abstractions of a class of switched dynamical systems with convergent continuous component dynamics, and dwell times sufficiently large to allow convergence to neighborhoods of parameterized limit sets in finite time. A word in a register automaton corresponds to a specific switching sequence in the switched system. The goal is to construct switching input sequences with the appropriate limit set parameterization so that the switched system is steered to a given region of its state space in minimum time.**

***Index Terms*— Hybrid systems, register automata, dynamic programming.**

I. INTRODUCTION

This paper investigates the optimal planning and control design for a class of hybrid systems with convergent continuous dynamics. A hybrid system in this class consists a set of low-level continuous controllers, each giving rise to a vector field which has a parameterized positive limit set. We assume that the control objectives can be achieved by *temporally sequencing* a number of different controllers. Controller sequences are subject to constraints relating to how the different parameterized component dynamics can switch between each other. To plan this temporal sequencing of controllers in order to achieve a certain reachability task in a time-optimal fashion, we propose a new purely discrete abstract computational model, and develop a way to apply dynamic programming (DP) on this model. The reachability specification is expressed in terms of first order logic.

The optimal control problem in hybrid systems has been studied under a diverse set of problem formulations and approaches. In [2], a hybrid system with a a discretized state space is considered, a cost function is defined, and a value function which lower bounds the optimal cost is introduced. A two stage decomposition is used to parameterize the switching instants in the switched system in [3], [4]. Reference [5] addresses optimal timing in hybrid systems when the scheduling sequence of the control modes is predetermined and the instantaneous cost function is continuously varying. Instead of considering the switching times as the design parameters, the optimal parameterization of a switching surface is analyzed in [6], A two-layer optimization approach is adopted in [7], where the lower level optimally selects the parameters of the switching surface, while the higher level schedules the transition sequence.

Jie Fu and Herbert Tanner are with the Mechanical Engineering Department at the University of Delaware, Newark DE 19716. {jiefu,btanner}@udel.edu. This work is supported by NSF under award # 0907003.

Similarly to [6], [7], this paper focuses on hybrid systems in which the switching is controlled through the selection of a set of parameters. However, here the component continuous dynamics of this hybrid system have some convergence properties which support particular discrete abstractions in the form of register automata [8], [9]. Without including continuous dynamics—contrary to timed automata—these models manipulate continuous data. We show that under certain conditions, the register automata abstractions can weakly simulate the concrete hybrid system, and preserve some of its continuous information in a resolution-free way. We utilize regular expressions [1] to generate the sequence of control modes and develop a DP variant yielding the optimal parameterizations of the controllers sequenced.

Section II introduces the particular class of hybrid system and some notions from automata theory. Section III presents the finite model as an abstraction of the hybrid system and some relevant results to be used in subsequent optimization method. In Section IV we present a solution to the planning problem on the register automata. A case study in Section V serves as an illustration of the abstraction method and time-optimal control design. Section VI concludes the paper.

II. PRELIMINARIES

Let Σ be a finite set of symbols σ_i, and $D \subseteq \mathbf{R}^k$. Pairs of the form $w_i = (\sigma_i, d_i) \in \Sigma \times D$ are called *data atoms*, and form finite sequences over $\Sigma \times D$, $w = w_1 \cdots w_n$, called *data words*. The *length* of w is denoted $|w|$, and $\mathrm{dom}(w)$ is the index set $\{1, \dots, |w|\}$ of the positions of the atoms $w_i = (\sigma_i, d_i)$ in w. For $i \in \mathrm{dom}(w)$, $\mathrm{val}_w(i) = d_i$ is the *data projection* map that gives the data value associated with the symbol σ_i. Automata operate on strings $w \in (\Sigma \times D)^*$.

Definition 1 (Register Automaton [8]–[10]): A non-deterministic two-way register automaton is a tuple $\mathbf{R} = \langle \mathcal{Q}, q_0, F, \Sigma, D, k, \tau, \Delta \rangle$, in which

- $\mathcal{Q}$ is a finite set of states;
- $q_0 \in \mathcal{Q}$ is the initial state;
- $F \subseteq \mathcal{Q}$ is the set of final states;
- Σ is a finite alphabet;
- D is an infinite set of data;
- $k \in \mathbb{N}$ indicates the number of registers;
- $\tau : \{1, \dots, k\} \to D \cup \{\varnothing\}$ is the register assignment, where $\varnothing$ means that the register is empty. Given input data atom (σ, d), define the set $\mathrm{Test}(\tau)$ of register tests (first order logic formulae) of the form: $\varphi \in \mathrm{Test}(\tau) \triangleq \quad d \leq \tau(i) \mid d < \tau(i) \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2,\ i \in \{1, \dots, k\}$.
- Δ is a finite set of transitions of the form $(i, q, \varphi_r(i)) \xrightarrow{\sigma} (q', \delta)$ and $(q, \varphi_w) \xrightarrow{\sigma} (q', i, \delta)$, where $i \in \{1, \dots, k\}$, q and $q' \in Q$, $\sigma \in \Sigma$ and $\delta \in \{\text{stay}, \text{left}, \text{right}\}$, and

$\text{Test}(\tau) = \{\cup_{i=1}^{k}\varphi_r(i), \varphi_w\}$: (1) $\varphi_r(i) : d = \tau(i)$; (2) $\varphi_w : \wedge_{h=1,\dots,k}\neg(d = \tau(h))$.

Given a data word w, a *configuration* γ of $\mathbf{R}$ is a tuple $[j, q, \tau]$, where $q \in \mathcal{Q}$, τ is a register assignment, and $j \in \text{dom}^+(w)$ with $w(j) = (\sigma_j, d_j)$. A configuration $[j, q, \tau]$ with $q \in F$ is accepting, while a configuration of the form $[1, q_0, \tau_0]$, with $q_0 \in \mathcal{Q}$ is an initial one. The transition $(i, q, \varphi_r(i)) \overset{\sigma_j}{\to} (q', \delta)$ (respectively $(q, \varphi_w) \overset{\sigma_j}{\to} (q', i, \delta)$) *applies* to γ iff $\varphi_r(i) : d_j = \tau(i)$ is true (respectively, $\varphi_w : d_j \neq \tau(h)$ for all $h \in \{1, \dots, k\}$, is true). For the case of the read, upon reading w_j, $\mathbf{R}$ enters states q' and the read head moves in the direction of δ, i.e., $j' = j+1, j' = j, j' = j-1$ for $\delta = \text{right}, \text{stay}$, and left, and the configuration is updated to $[j', q', \tau]$. For the case of the write, $\mathbf{R}$ enters state q', d_j is copied to register i, and the read head on w moves in the direction δ (in this order). The configuration is now $[j', q', \tau']$, where τ' denotes the contents of the updated register. If there are no left-transitions ($\delta \in \{\text{stay}, \text{right}\}$) in $\mathbf{R}$, it is called *one-way* register automaton.

The special class of hybrid systems considered in this paper is defined as follows.

Definition 2 (Hybrid Agent): The hybrid agent is a tuple: $\mathbf{H} = \langle \mathcal{Z}, \Sigma, \iota, \mathcal{P}, \pi_i, \mathcal{AP}, f, \overleftarrow{\cdot}, \overrightarrow{\cdot}, s, T\rangle$.

- $\mathcal{Z} = \mathcal{X} \times \mathcal{L}$ is a set of *composite* (continuous and boolean) states, where $\mathcal{X} \subset \mathbb{R}^n$ is a compact set, and $\mathcal{L} \subseteq \{\mathbf{0}, \mathbf{1}\}^r$.
- Σ is a set of finite discrete locations (*control modes*).
- $\iota : \Sigma \to \{1, \dots, k\}$ is a bijection, indexing Σ.
- $\mathcal{P} \subseteq \mathbb{R}^m$ is a vector of continuous parameters.
- $\pi_i : \mathbb{R}^m \to \mathbb{R}^{m_i}$, for $i = 1, \dots, k$ is a finite set of canonical projections, such that $p = [\pi_1(p), \dots, \pi_k(p)]^T$.
- $\mathcal{AP}$: $\{\alpha_h\}_{h=1}^{\mathcal{AP}}$ is a set of (logical) propositions and $\neg\mathcal{AP}$ is used to denote $\{\neg\alpha_h\}_{h=1}^{\mathcal{AP}}$.
- f_σ: $\mathcal{X} \times \mathcal{L} \times \mathcal{P} \longrightarrow T\mathcal{X}$ is a finite set of vector fields parameterized by $p \in \mathcal{P}$ and $\ell \in \mathcal{L}$, with $\sigma \in \Sigma$, with respect to which $\mathcal{X}$ is positively invariant. These vector fields have limit sets which are parameterized by $p \in \mathcal{P}$.
- $\overleftarrow{\cdot}$: $\Sigma \to 2^{\mathcal{AP} \cup \neg\mathcal{AP}}$ maps a mode σ to a set of propositions denoted $\text{PRE}(\sigma)$ that need to be satisfied for the system to switch to mode σ. When composite state z and parameter p satisfy $\text{PRE}(\sigma)$ we write $(z, p) \models \overleftarrow{\sigma}$.
- $\overrightarrow{\cdot}$: $\Sigma \to 2^{\mathcal{AP} \cup \neg\mathcal{AP}}$ maps a mode σ to a set of propositions denoted $\text{POST}(\sigma)$ that are satisfied when the vector field reaches steady state. When composite state z and parameter p satisfy $\text{POST}(\sigma)$ we write $(z, p) \models \overrightarrow{\sigma}$.
- s: $\mathcal{Z} \times \mathcal{P} \to \mathcal{P}$ is the reset map for the parameters. It assigns to each (z, p) a subset of $\mathcal{P}$ from which the current value of $p \in \mathcal{P}$ can be chosen.
- T: $\mathcal{Z} \times \mathcal{P} \times \Sigma \to \mathcal{Z} \times \mathcal{P} \times \Sigma$ is the location transition map, according to which $(z, p, \sigma) \mapsto (z, p', \sigma')$ iff $(z, p) \models \overrightarrow{\sigma}$ and $(z, p') \models \overleftarrow{\sigma'}$ with $p' \in s(z, p)$.

The *configuration* of $\mathbf{H}$ is denoted $[z, p, \sigma]$.

This model is intended to describe a continuous dynamical system that switches between different control laws based on some discrete logic. In $\mathbf{H}$, the continuous vector field f_σ of control mode $\sigma \in \Sigma$ evolves in the same compact set and always converges to some limit set, the location and shape of which is defined by its parameterization $p \in \mathcal{P}$.

Let $\phi_\sigma(t, x_0; p)$ denote the flow of vector field $f_\sigma(x; \ell, p)$ passing from x_0 at time $t = 0$. The positive limit set in control mode σ, when parameterized by p is denoted $L^+(p, \sigma)$ and its existence is ensured by the compactness and invariance of $\mathcal{X}$. We will assume that $L^+(p, \sigma)$, for a given σ and for all $p \in \mathcal{P}$ is path connected. If it is not, and there are $B(\sigma)$ isolated components $L_i^+(p, \sigma)$ for $i = 1, \dots, B(\sigma)$, one can refine control mode σ into $\sigma^1, \dots \sigma^{B(\sigma)}$, one for each $L_i^+(p, \sigma)$. Let $\Omega_i(p, \sigma) \subset \mathcal{X}$ be the attraction region of each σ^i. The POSTs and PREs are related to the limit sets and their attraction regions as follows:

$$\{(z, p) \mid (z, p) \models \overrightarrow{\sigma}\} \subseteq \left\{(x, \ell, p) \mid x \in L_i^+(p, \sigma^i) \oplus \mathcal{B}_\varepsilon\right\},$$
$$\{(z, p) \mid (z, p) \models \overleftarrow{\sigma}\} \subseteq \left\{(x, \ell, p) \mid x \in \Omega_i(p, \sigma^i)\right\} \quad .$$

where $\oplus$ the Minkovski (set) sum, $\mathcal{B}_\varepsilon$ is the open ball of radius ε. For simplicity we assume that for $\mathbf{H}$ of Definition 2, Σ does not afford any further refinement.

As a result of the continuous flow of f_σ parameterized by p, we say that the composite state z *evolves* to some other composite state z', written $z \overset{\sigma[p]}{\hookrightarrow} z'$. A sequence of the form $(\sigma_1, p_1) \cdots (\sigma_N, p_N)$ is an *input* to $\mathbf{H}$, specifying how parameterized control modes are to be concatenated in $\mathbf{H}$. At configuration $[z, p, \sigma]$ in $\mathbf{H}$, (σ', p') is *admissible* if there is a $[z, p', \sigma'] \in \mathcal{Z} \times \Sigma \times \mathcal{P}$ such that $T([z, p, \sigma]) = [z, p', \sigma']$. A pair of data atoms $(\sigma_j, p_j)(\sigma_{j+1}, p_{j+1})$ is admissible at $[z, \sigma, p]$ if $\sigma = \sigma_j$, $p = p_j$, and there is a $z' \in \mathcal{Z}$ for which $z \overset{\sigma_j[p_j]}{\hookrightarrow} z'$ with (σ_{j+1}, p_{j+1}) being admissible at $[z', p_j, \sigma_j]$. A data word $w = w_1 \cdots w_n = (\sigma_1, p_1) \cdots (\sigma_n, p_n)$ is admissible if every prefix of w is admissible.

The planning problem addressed is the following:

Problem 1: Given a hybrid agent $\mathbf{H}$ with initial state z_0 and parameter p_0, find an admissible input $(\sigma_1, p_1) \cdots (\sigma_N, p_N)$ (N to be determined) so that the configuration of $\mathbf{H}$ after the Nth transition satisfies a set of propositions denoted $\text{SPEC} \subset \mathcal{AP} \cup \{\neg\mathcal{AP}\}$ with p_N in the set $P_f = \{p \mid \exists\, z \in \mathcal{Z} : (z, p) \models \text{SPEC}\}$.

III. Symbolic abstraction

The relation between the composite states and parameters of $\mathbf{H}$ and the states of the register semiautomaton[1] is formalized by means of a discrete map as follows:

Definition 3 (Valuation map): The valuation map V_M: $\mathcal{X} \times \mathcal{L} \times \mathcal{P} \to \mathcal{V} \subseteq \{\mathbf{1}, \mathbf{0}\}^{|\mathcal{AP}|}$ is a function that maps a pair of composite state and parameter, to a binary vector $v \in \mathcal{V}$ of length $|\mathcal{AP}|$. The element at position i in v, denoted $v[i]$, is $\mathbf{1}$ or $\mathbf{0}$ if $\alpha_i, \alpha_i \in \mathcal{AP}$ is true or false, respectively, for a particular pair (z, p). We write $\alpha_i(z, p) = v[i]$, for $v \in \mathcal{V}$.

Let us relate SPEC with a set of binary vectors $\mathcal{V}_{\text{spec}} \subseteq \mathcal{V}$ according to: $\forall\ v \in \mathcal{V}_{\text{spec}}$, $v[i] = \mathbf{1}$ if $\alpha_i \in \text{SPEC} \cap \mathcal{AP}$, $v[i] = \mathbf{0}$ if $\alpha_i \in \text{SPEC} \cap \neg\mathcal{AP}$ and $v[i] \in \{\mathbf{0}, \mathbf{1}\}$ if $\alpha_i \notin \text{SPEC}$. To represent $\mathcal{V}_{\text{spec}}$ compactly we will denote any $v[i] \in \{\mathbf{0}, \mathbf{1}\}$ using the symbol $*$.

The register automaton $\mathbf{R}(\mathbf{H})$ which serves as an abstraction of $\mathbf{H}$ is now defined as follows:

[1]A semiautomaton is an automaton without defining initial and final states.

Definition 4 (Induced register semiautomaton): The deterministic finite one-way register semiautomaton induced by hybrid agent $\mathbf{H}$ (with reference to Definition 2), is a tuple $\mathbf{R}(\mathbf{H}) = (\mathcal{Q}, \Sigma, \mathcal{P}, 1, \tau, \Delta)$.

- $\mathcal{Q} = \left\{q \in \{\mathbf{0}, \mathbf{1}\}^{|\mathcal{AP}|} \mid \exists\, z \in \mathcal{Z}, p \in \mathcal{P} : V_M(z,p) = q\right\}$ is a finite set of states.
- Σ is the alphabet shared with $\mathbf{H}$.
- $\mathcal{P}$ is an infinite set of data coinciding with $\mathcal{P}$ of $\mathbf{H}$.
- 1 denotes a single k-dimensional *array* register.
- $\tau : 1 \mapsto \mathcal{P} \cup \{\varnothing\}$ is the register assignment map (we denote $\tau(1) \equiv \tau$). Given input data atom $(\sigma, p) \in \Sigma \times \mathcal{P}$, the set $\mathrm{Test}(\tau)$ consists of formulae defined by $\varphi \triangleq p = \tau \mid \pi_j(p) = \pi_j(\tau) \mid p \in \varphi(\tau) \mid \neg\varphi \mid \varphi \wedge \varphi$, where $j \in \{1, \ldots, k\}$, $\sigma \in \Sigma$ and $q \in \mathcal{Q}$.
- Δ is a finite set of transitions of two types, (a) a *read transition* $(q, \top) \xrightarrow{\sigma_j} (q', \mathrm{right})$ is defined iff *for all* z such that $V_M(z,\tau) = q$, it holds that $(z,\tau) \models \overleftarrow{\sigma_j}$ and $\exists\, z' \in \mathcal{Z}$ such that $V_M(z',\tau) = q'$ with $(z',\tau) \models \overrightarrow{\sigma_j}$. $\top \in \mathrm{Test}(\tau)$ is true iff $p_j = \tau$ given input $w_j = (\sigma_j, p_j)$. (b) a *write transition* $(q, \varphi_i) \xrightarrow{\sigma_j} \big(q', \iota(\sigma_j), \mathrm{stay}\big)$ is defined iff *for all* z such that $V_M(z,\tau) = q$, there exists $p_j \in s(z,\tau)$, $V_M(z,p_j) = q'$ and $(z,p_j) \models \overleftarrow{\sigma_j}$. Associated with this transition, a set valued map is defined as $\varphi_i(\tau) = \{p' \mid \forall z \in \mathcal{Z} : V_M(z,\tau) = q, p' \in s(z,\tau)$ that satisfies $(z,p') \models \overleftarrow{\sigma_j}$ and $V_M(z,p') = q'$ and $\pi_{\iota(\sigma_j)}(p') \neq \pi_{\iota(\sigma_j)}(\tau)\}$. Each set-valued map is indexed with $i \in \{1, \ldots, W\}$ where W is the total number of write transitions. At configuration $[j,q,\tau]$, upon receiving input $w_j = (\sigma_j, p_j)$, a read transition $(q,\top) \xrightarrow{\sigma_j} (q', \mathrm{right})$ *applies* iff $p_j = \tau$ and the machine moves to state q' and the input read head advances one position. On $w_j = (\sigma_j, p_j)$, a write transition $(q, \varphi_i) \xrightarrow{\sigma_j} \big(q', \iota(\sigma_j), \mathrm{stay}\big)$ *applies* iff $p \in \varphi_i(\tau)$. The machine reaches q' without moving the input read head and updates τ to $\tau' = p_j$.

Without any ambiguity, $\mathbf{\Phi} \triangleq \bigcup_i \varphi_i(\cdot)$, the set of all set-valued maps associated with write transitions.

A data atom (σ_j, p_j) is *admissible* at configuration $[j,q,\tau]$ if there is a transition in Δ applies to $[j,q,\tau]$ on input (σ_j, p_j). A pair of data atoms $(\sigma_j, p_j)(\sigma_{j+1}, p_{j+1})$ is admissible if there is a transition in Δ that applies to $[j,q,\tau]$ on input (σ_j, p_j), taking $\mathbf{R}(\mathbf{H})$ to $[j+1, q', \tau']$, where a subsequent transition in Δ applies to, on input (σ_{j+1}, p_{j+1}). A data word $w = w_1 \ldots w_n = (\sigma_1, p_1) \cdots (\sigma_n, p_n)$ is admissible if every prefix of w is admissible.

A concatenation of any number of write transitions with a single read transition triggered by input atom w_j, taking the machine from state q to state q' is denoted $q \overset{w_j}{\rightsquigarrow} q'$, and we refer to this transition sequence as a *composite* transition.

Proposition 1: In $\mathbf{R}(\mathbf{H})$, any composite transition contains either only one read transition or a pair - a write transition followed by a read one.

Proof: Let $\mathbf{R}(\mathbf{H})$ be at configuration $[j,q,\tau]$. Suppose $\mathbf{R}(\mathbf{H})$ takes a composite transition, $q \overset{w_j}{\rightsquigarrow} q'$, where $w_j = (\sigma_j, p_j)$. If $p_j = \tau$ then the machine jumps from q to q' and advances the read head by one position. If $p_j \neq \tau$, no read transition can follow and we have to assume w_j triggers a write transition first. After the write transition, τ takes the value of p_j. The machine still reads $w_j = (\sigma_j, p_j)$ on the input tape since the read head stays. Upon reading w_j again, the machine now finds $\tau = p_j$, a read transition is triggered and the read head advances one step. In this case configuration $[j,q,\tau]$ evolves as follows: $q \overset{w_j}{\rightsquigarrow} q' \Leftrightarrow [j,q,\tau] \xrightarrow{w_j} [j, q^t, \tau'] \xrightarrow{w_j} [j+1, q', \tau]$, where q^t is some intermediate state where the machine lands right after taking the write transition. We thus see that a composite transition either includes a single read transition or a write transition followed by a read transition—the latter referred to as a write-read transition pair. ■

To ensure that plans devised on the abstraction are implementable on the concrete hybrid system, we need to establish an equivalence relation between $\mathbf{R}(\mathbf{H})$ and $\mathbf{H}$ such that any admissible data word generated in $\mathbf{R}(\mathbf{H})$ is guaranteed to be implementable in $\mathbf{H}$. To this end, let us recall the concept of a transition system.

Definition 5: A *labeled transition system* is a tuple $\mathbf{T} = (Q, \Sigma, \rightarrow)$ where: 1) Q is a set of states, 2) Σ is a set of labels, 3) $\rightarrow \subseteq Q \times \Sigma \times Q$ is a transition relation. For $(q_1, \sigma, q_2) \in \rightarrow$ we write $q_1 \xrightarrow{\sigma} q_2$.

In $\mathbf{T}$, we distinguish an element $\lambda \in \Sigma$ and we call a transition labeled λ silent. We write $q \rightsquigarrow q'$ to denote that q' is reachable from q with an arbitrary number of λ transitions, and $q \overset{\sigma}{\rightsquigarrow} q'$ if q' is reachable from q with an arbitrary number of λ transitions interleaved with an $\sigma \in \Sigma \setminus \{\lambda\}$; that is, $q \rightsquigarrow q' \Leftrightarrow q \xrightarrow{\lambda^*} q'$, while $q \overset{\sigma}{\rightsquigarrow} q' \Leftrightarrow q \xrightarrow{\lambda^*\sigma\lambda^*} q'$. [2]

Definition 6 (Weak (observable) simulation [11]): Consider two (labeled) transition systems over the same input alphabet Σ_T, $\mathbf{T}_1 = (Q_1, \Sigma_T, \rightarrow_1)$ and $\mathbf{T}_2 = (Q_2, \Sigma_T, \rightarrow_2)$, and let $\Sigma_\epsilon \subset \Sigma_T$ be a set of labels associated with silent transitions. An ordered binary relation $\mathfrak{R}$ on $Q_1 \times Q_2$ is a *weak (observable) simulation* if: (i) $\mathfrak{R}$ is total, i.e., for any $q_1 \in Q_1$ there exists $q_2 \in Q_2$ such that $(q_1, q_2) \in \mathfrak{R}$, and (ii) for every ordered pair $(q_1, q_2) \in \mathfrak{R}$ for which there exists q_1' such that $q_1 \overset{\sigma}{\rightsquigarrow}_1 q_1'$, then $\exists\, (q_1', q_2') \in \mathfrak{R} : q_2 \overset{\sigma}{\rightsquigarrow}_2 q_2'$. Then $\mathbf{T}_2$ weakly simulates $\mathbf{T}_1$ and we write $\mathbf{T}_2 \gtrsim \mathbf{T}_1$.

Theorem 1: The hybrid agent $\mathbf{H}$ weakly simulates its induced register semiautomaton $\mathbf{R}(\mathbf{H})$ in the sense that there exists an ordered total binary relation $\mathfrak{R}$ such that $(q,z) \in \mathfrak{R} \Leftrightarrow \exists\ p \in \mathcal{P}, V_M(z,p) = q$, which satisfies $(q,z) \in \mathfrak{R} \subset \mathcal{Q} \times \mathcal{Z}$ and $q \overset{w_j}{\rightsquigarrow} q'$ with $w_j = (\sigma_j, p_j) \Rightarrow \exists\, z' \in \mathcal{Z} : z \overset{\sigma_j[p_j]}{\hookrightarrow} z'$ with $(q', z') \in \mathfrak{R}$.

Proof: First we indicate that $\mathfrak{R}$ is total by construction, since any $q \in \mathcal{Q}$ such that for all $(z,p) \in \mathcal{Z} \times \mathcal{P}$, $V_M(z,p) \neq q$ would violate the definition of $\mathcal{Q}$. To establish that $\mathfrak{R}$ is a weak simulation, let $\mathbf{R}(\mathbf{H})$ be at configuration $[j,q,\tau]$, where $(q,z) \in \mathfrak{R}$ for some $z \in \mathcal{Z}$. Suppose now that $\mathbf{R}(\mathbf{H})$ takes a (composite) transition w denoted $q \overset{w_j}{\rightsquigarrow} q'$, then according to Proposition 1, $q \overset{w_j}{\rightsquigarrow} q'$ is either a single read transition $[j,q,\tau] \xrightarrow{w_j} [j+1, q', \tau]$, or a composite

[2]$*$ is Kleene star. Σ^* is the collection of all possible finite-length strings generated from the strings in Σ.

$[j,q,\tau] \xrightarrow{w_j} [j,q^t,\tau'] \xrightarrow{w_j} [j+1,q',\tau']$, or a write-read pair. The mere existence of a transition originating from q on input (σ_j,p_j) ensures that for *any* z that satisfies $V_M(z,p)=q$, it holds that $p_j \in s(z,p)$ and either $(z,p) \models \overleftarrow{\sigma_j}$ (in the case of a single read) or $(z,p_j) \models \overleftarrow{\sigma_j}$ with $V_M(z,p_j)=q^t$ (in the case of a write-read pair). For the first case, there must exist z' such that $V_M(z',p_j)=q'$ and $(z',p_j) \models \overrightarrow{\sigma_j}$. By definition, there exists a (continuous) evolution $z \overset{\sigma_j[p_j]}{\hookrightarrow} z'$ in $\mathbf{H}$. Since $V_M(z',p_j)=q'$ it follows that $(q',z') \in \mathfrak{R}$. For the second case, after updating its register with $\tau'=p_j$, $\mathbf{R}(\mathbf{H})$ still reads (σ_j,p_j) on its input tape. Since $(z,p_j) \models \overleftarrow{\sigma_j}$ and $\tau=p_j$, it follows that $\mathbf{R}(\mathbf{H})$ takes a read transition to q'. The argument of the previous case applies again. ∎

Given $\mathbf{H} \gtrsim \mathbf{T}(\mathbf{H})$, it follows that $\mathbf{H}$ might be able to evolve in ways that $\mathbf{R}(\mathbf{H})$ cannot, and therefore the behavior[3] of the former is a superset of the behavior of the latter. Weak *bisimulation* can only be established in special cases when 1) reset maps are constant in z for all $(z,p): V_M(z,p)=q$, or 2) $\forall \sigma_i,\sigma_j \in \Sigma$ either $\textsc{Post}(\sigma_i) \Rightarrow \textsc{Pre}(\sigma_j) \vee \neg[\textsc{Pre}(\sigma_i) \wedge \textsc{Post}(\sigma_j)]$.

Definition 7: The transformation semiautomaton of $\mathbf{R}(\mathbf{H})$ is a triple $\mathbf{TR}(\mathbf{H}) = \{\bar{\mathcal{Q}},\bar{\Sigma},\bar{\Delta}\}$ consisting of:

- $\bar{\mathcal{Q}} \subseteq \mathcal{Q} \times \{\mathfrak{p},\mathfrak{p}'\}$, where $\mathfrak{p}$ and $\mathfrak{p}'$ are two symbols. The set $\bar{\mathcal{Q}}$ is constructed by first including all pairs $(q,\mathfrak{p})$ with $q \in \mathcal{Q}$. Then, for each state $q \in \mathcal{Q}$ that $\mathbf{R}(\mathbf{H})$ can arrive at with a write transition, we create a copy $(q,\mathfrak{p}')$ to mark the change in the register contents.
- $\bar{\Sigma} = \Sigma \cup \Lambda \cup \{\mathrm{id}_P\}$, where Λ is a set of labels for write transitions in $\mathbf{R}(\mathbf{H})$, and id_P is a label for linking $(q,\mathfrak{p})$ with their copies $(q,\mathfrak{p}')$. (This transition has no effect on parameters and is denoted id_P.)
- $\bar{\Delta}$, the set of transitions of the following types: (i) $(q,\mathfrak{p}) \overset{\lambda_i}{\dashrightarrow} (q',\mathfrak{p}')$, defined if q' is accessible from q via a write transition $(q,\varphi_i) \xrightarrow{\sigma} \big(q',\iota(\sigma),\mathrm{stay}\big)$. As each transition labeled $\lambda \in \Lambda$ defined in $\mathbf{TR}(\mathbf{H})$ is in one-to-one correspondence with a write transition, the index i is the same number as the index of set-valued map $\varphi_i(\cdot)$ associated with that write. (ii) $(q,\mathfrak{p}) \overset{\mathrm{id}_P}{\dashrightarrow} (q,\mathfrak{p}')$, defined if q is accessible from $q' \in \mathcal{Q}$ via a write transition; (iii) $(q,\mathfrak{p}) \xrightarrow{\sigma} (q',\mathfrak{p})$, defined if there exists a read transition $(q,\top) \xrightarrow{\sigma} (q',\mathrm{right})$, and q is not accessible via a write transition from any $q'' \in \mathcal{Q}$; (iv) $(q,\mathfrak{p}') \xrightarrow{\sigma} (q',\mathfrak{p})$, defined if there exists a read transition $(q,\top) \xrightarrow{\sigma} (q',\mathrm{right})$, and q is accessible via at least one write transition from $q'' \in \mathcal{Q}$.

Define $\mathrm{id}_P(\cdot)$ as the identity map on $\mathcal{P}$. For $M_1(\cdot),M_2(\cdot) \in \mathbf{\Phi} \cup \{\mathrm{id_P}(\cdot)\}$, $M_2(\cdot)$ composed with $M_1(\cdot)$ defines a set-valued map: $M_2 \circ M_1(p) = \cup_{p' \in M_1(p)} M_2(p')$. The composition can be defined recursively. Besides, $M_1^{-1} \circ M_2^{-1}(\cdot)$ can be obtained in the same manner since the inverse of a set-valued map is also a set-valued map.

Each walk $\mathfrak{w}$ in $\mathbf{TR}(\mathbf{H})$ can be interpreted as a data word $w = (\sigma_1,p_1)(\sigma_2,p_2)\dots(\sigma_N,p_N)$ with $p_i, i=1,\dots,N$ as unknown parameters and a set-valued map $M(\cdot) = M_N \circ M_{N-1} \circ \dots \circ M_1(\cdot), M_i(\cdot) \in \mathbf{\Phi} \cup \{\mathrm{id}_P(\cdot)\}$ by Algorithm 1.

[3]Behavior is hereby understood as the set of feasible input data words.

Algorithm 1 `dataword(𝔴)`: for the walk $\mathfrak{w} = u_1 \dots u_m$, compute a data word w and the set-valued map $M(\cdot): \mathcal{P} \to 2^{\mathcal{P}}$.

```
$i := 1$, $j := 1$;
while $i \leq m$ do
    if $u_i \in \Sigma$ then
        $\sigma_j = u_i$, $w_j = (\sigma_j, p_j)$, $M_j = \mathrm{id}_P(\cdot)$, $j = j+1$, $i = i+1$;
    else if $u_i \in \Lambda$ then
        $M_j(\cdot) = \varphi_h(\cdot)$ where $h$ is the index such that $\lambda_h = u_i, \lambda_h \in \Lambda$,
         and $\sigma_j = u_{i+1}$, $w_j = (\sigma_j, p_j)$, $j = j+1$, $i = i+2$;
    else
        $M_j(\cdot) = \mathrm{id}_P(\cdot)$, $\sigma_j = u_{i+1}$, $w_j = (\sigma_j, p_j)$, $j = j+1$, $i = i+2$;
    end if
end while
return $M$, $w$.
```

IV. Time-optimal control design

Assume that the write transitions (updates in parameters) in $\mathbf{R}(\mathbf{H})$ incur no cost. The cost of a read transition in $\mathbf{R}(\mathbf{H})$ (evolution in $\mathbf{H}$) is determined by the component continuous dynamics active at that time, the initial condition for the continuous state and the assignment of parameter. The component dynamics when $\mathbf{H}$ is at control mode σ is expressed in the form $\dot{x} = f_\sigma(x;\ell,p)$, with $\sigma \in \Sigma$, $p \in \mathcal{P}$, $\ell \in \mathcal{L}$, and $x \in \mathcal{X}$. An incremental (or running) cost function $R: \mathcal{X} \times \mathbb{R}_+ \to \mathbb{R}_+$ can generally be used to define the *atomic cost* $g_\sigma(x_0,p)$ of $\mathbf{H}$ evolving in control mode σ along flow $\phi_\sigma(t;x_0,p)$ for $t \in [t_0,t_f]$, $g_\sigma(x_0,p) = \int_{t_0}^{t_f} R\big(\phi_\sigma(t;x_0,p),t\big)\,\mathrm{d}t$.

In the context of this paper, we take $R(x,t) \triangleq \mathbf{1}_{\{L^+(p,\sigma)\oplus\mathcal{B}_\varepsilon\}^{\mathrm{c}}}(x)$, where $\mathbf{1}_A$ denotes the indicator function of set A, and $\{\cdot\}^{\mathrm{c}}$ denotes set complement. Thus, g_σ simply measures the length of the interval $[t_0,t_f]$ during which mode σ is active. Given $p \in \mathcal{P}$, when evaluated over an infinite time horizon over all $(x_0,\ell) \in S \subseteq \{(x,\ell) \mid (x,\ell,p) \models \overleftarrow{\sigma}.\}$, we have $g_\sigma(S,p) = \max_{(x,p)\in S} \int_0^\infty \mathbf{1}_{\{\overrightarrow{\sigma}[p]\}^{\mathrm{c}}}(\phi_\sigma(t;x,p))\,\mathrm{d}t$, with the understanding that $\overrightarrow{\sigma}[p] \subseteq L^+(p,\sigma) \oplus \mathcal{B}_\varepsilon$. The time required for a continuous state $x \in \mathcal{X}$ to converge under controller σ to an ε neighborhood of $\overrightarrow{\sigma}$ can be over-approximated using Lyapunov-based techniques which cannot be elaborated on here due to space constraints.

We define the *accumulated cost* J_w over the execution of an admissible data word $w = (\sigma_1,p_1)\dots(\sigma_N,p_N)$ from configuration $[z,p,\sigma]$ with $z=(x,\ell)$ as $J_w(z,\{p_j\}_{j=1}^N) \triangleq g_{\sigma_1}(z,p_1) + \sum_{i=2}^N g_{\sigma_i}\big(\overrightarrow{\sigma}_{i-1}[p_{i-1}],p_i\big)$.

The optimization problem can then be stated as follows:

Problem 2: Out of all admissible sequences $w = (\sigma_1,p_1)\cdots(\sigma_N,p_N)$ that solve Problem 1, find one that

$$\min_{\{p_i\}} J_w(z_0,\{p_i\}_{i=1}^N),\ \text{s.t.} p_i \in s(z,p_{i-1}), (z,p_{i-1}) \models \overrightarrow{\sigma}_{i-1} \tag{1}$$

We want to find a solution for Problem 2 on $\mathbf{H}$ without involving the continuous dynamics; rather, we want get a (sub)optimal solution to Problem 2 by solving the following optimization problem on a discrete system:

Problem 3: For a given z_0,p_0 and $\textsc{Spec}$, and with $q_0 = V_M(z_0,p_0)$, $F = \mathcal{V}_{\mathrm{spec}} \cap \mathcal{Q}$, find a walk from $(q_0,\mathfrak{p})$ to some $(q_f,\mathfrak{p})$ for $q_f \in F$ that satisfy the following: 1) The map $M(\cdot)$ associated with $\mathfrak{w}$ satisfies $M(p_0) \cap P_f \neq \emptyset$; 2) among all walks satisfying $M(p_0) \cap P_f \neq \emptyset$, $\mathfrak{w}$ is the shortest;

3) The data word $w = (\sigma_1, p_1) \dots (\sigma_N, p_N)$ associated to $\mathfrak{w}$ (Algorithm 1) is such that $\{p_j\}$ solves (1). (If q_f is not reachable via a read transition, the final state is $(q_f, \mathfrak{p}')$.)

After specifying initial state $(q_0, \mathfrak{p})$ and final state $(q_f, \mathfrak{p})$(or $(q_f, \mathfrak{p}')$), $\mathbf{TR}(\mathbf{H})$ becomes a deterministic finite state automaton (DFA) and the problem of finding a set of walks from the initial to the final state can be solved by generating the regular expression (RE) of this DFA.

Definition 8 ([12]): A regular expression is defined recursively as follows: 1) $\emptyset$ is an (empty) RE, ϵ is a RE denoting the set including empty string $\{\epsilon\}$ and σ is a RE denoting the set $\{\sigma\}$, for all $\sigma \in \Sigma$; 2) If r and s are REs, then rs(concatenation), $(r + s)$ (union), r^*, s^* (Kleene-closure) are REs; 3) There are no RE other than those constructed by applying rules 1 and 2 finite number of times.

One way of converting a DFA into a RE is Brzozowski's method [13]. By replacing Kleene star $*$ in $\mathbf{RE}(\mathbf{H})$ with natural numbers (including 0), we can generate all walks of length $m < \infty$ and express their set as $\mathfrak{W}(m) \triangleq \{\mathfrak{w} | \mathfrak{w} \in \mathbf{RE}(\mathbf{H}), |\mathfrak{w}| = m\}$. To find the shortest walks that solve Problem 3, we start with $m = 1$ and increase m until a walk $\mathfrak{w}$ is found, with a set-valued map $M(\cdot)$ which satisfies $M(p_0) \cap P_f \neq \emptyset$. Then there exists a sequence of n parameter values $\{p_j\} = p_1, \dots, p_n$ with $n = |\mathfrak{w}/_\Sigma| \leq m$ such that $w = (\sigma_1, p_1) \cdots (\sigma_n, p_n)$ is an admissible input for $\mathbf{R}(\mathbf{H})$ at configuration $[1, q_0, p_0]$, and can take it to configuration $[n+1, q_f, p_f]$ with $q_f \in F, p_f \in P_f$. An upper bound $U < \infty$ on the length of walk can be computed if one places a limit on the maximum allowable cost, say $\bar{J}$, and has an estimate of the minimum cost of executing any controller $\sigma \in \Sigma$, say $J_{\min} > 0$. As any read transition incurs a nonzero cost, $U = \frac{\bar{J}}{J_{\min}}$. In the current implementation, the walk search is terminated once a walk satisfying $M(p_0) \cap P_f \neq \emptyset$ is found. There might exist walks yielding better plans, which is why the proposed solutions are considered sub-optimal.

The following DP algorithm is adapted from [14] for the specific requirements of the problem considered in this paper. Given the input word w with set-valued map M from Algorithm 1, i.e. $w = (\sigma_1, p_1) \dots (\sigma_N, p_N)$, $M(\cdot) = M_N \circ \dots \circ M_1(\cdot)$, the optimal cost $J^*_w(z_0)$ equals $J_1(z_0)$ given by the following two steps: **(1)** compute the set of feasible value of parameter p_i: $P_i = M_i \circ \cdots \circ M_1(p_0) \cap \{\cup_{p_N \in P_f} (M_N \circ \cdots \circ M_{i+1})^{-1}(p_N)\}$. **(2)** the optimal cost $J^*_w(z_0)$ is equal to $J_1(z_0)$ given by the last step of the following algorithm, which proceeds from the stage governed by σ_N to the one by σ_1: $J_N(p_{N-1}) = \min_{p_N \in P_N \cap M_N(p_{N-1})} g_{\sigma_N}(\overrightarrow{\sigma}_{N-1}[p_{N-1}], p_N)$, $J_i(p_{i-1}) = \min_{p_i \in P_i \cap M_i(p_{i-1})} (g_{\sigma_i}(\overrightarrow{\sigma}_{i-1}[p_{i-1}], p_i) + J_{i+1}(p_i))$, for $i = 2, \dots, N-1$, and $J_1(z_0) = \min_{p_1 \in P_1} (g_{\sigma_1}(z_0, p_1) + J_2(p_1))$.

Executing this DP algorithm in the general case would typically require a discretization of the parameter space, and backtracking from the final parameter values to the initial ones. Due to space limitations we cannot provide the details of such an implementation. The process however, once the DP equations have been stated, is straightforward.

V. Case study: Goldilocks' thermostat

The temperature in an indoor space is regulated through a heating system, which is controlled by a digital thermostat. We refer to this thermostat as "Goldilocks' thermostat," because it reports the temperature conditions within the space as one of three states: "too cold," "too hot," and "just right." The physics of the temperature variation is described by a first order dynamical system with two modes, H for heating, and O for (natural) cooling:

$$\dot{x} = f_H(x, p) = -a_h (x - p), \tag{2a}$$

$$\dot{x} = f_o(x, p) = -a_o (x - p_e), \tag{2b}$$

where x denotes the inside temperature, a_h and a_o are positive constants, and p is the temperature at which the thermostat is set, and p_e is a constant expressing the exterior ambient temperature. We take $a \triangleq a_h = a_o = 0.5$.

The system can be modeled as a hybrid agent $\mathbf{H} = \left\{\mathcal{Z}, \mathcal{P}, \Sigma, \mathcal{AP}, f, \overleftarrow{\cdot}, \overrightarrow{\cdot}, s, T\right\}$, where:

- $\mathcal{Z} \subseteq \mathbb{R}$; $\mathcal{P} \subseteq \mathbb{R}$, $p \in \mathcal{P}$ is the set temperature;
- $\Sigma = \{H, O_1, O_2\}$, where H denotes heating, and O_1, O_2 denote cooling modes with different PRE and POST;
- $\mathcal{AP} = \Big\{\big[\alpha_1 : p_e \leq x \leq (1 - \rho)p\big], \big[\alpha_2 : x \geq p - \epsilon\big], \big[\alpha_3 : x \geq (1 + \rho)p\big], \big[\alpha_4 : x \leq p + \epsilon\big]\Big\}$, where $\rho \in (0, 1)$ is a parameter related to the sensitivity of the thermostat and ϵ incorporates the settling tolerance for control modes;
- f_H (see (2a)) and $f_{o_1} = f_{o_2} = f_o$ (see (2b));
- $\overleftarrow{O_1} = \{\neg\alpha_1, \alpha_2, \alpha_3, \neg\alpha_4\}$, $\overrightarrow{H} = \overleftarrow{O_2} = \overrightarrow{O_1} = \{\neg\alpha_1, \alpha_2, \neg\alpha_3, \alpha_4\}$, $\overleftarrow{H} = \overrightarrow{O_2} = \{\alpha_1, \neg\alpha_2, \neg\alpha_3, \alpha_4\}$;
- $s(z, p) = p' \in \mathcal{P}, p \neq p', \forall (z, p) \in \mathcal{Z} \times \mathcal{P}$;
- the transition function T, which follows Definition 2.

Note as all modes share the same parameter p, there is no need to use indexing ι and the canonical projection π.

Problem 4: Given an initial temperature $x_0 = 10$ °C and an initial thermostat setting $p_0 = 16$ °C, find the sequence of parameterized modes that bring the system to a state that satisfies SPEC $= \{\alpha_3\}$ with the thermostat set to $p_f = 32$ °C in minimum time.

The induced register semiautomaton $\mathbf{R}(\mathbf{H}) = (\mathcal{Q}, \Sigma, \mathcal{P} \cap \{\varnothing\}, 1, \tau, \Delta)$, where:

- $\mathcal{Q} = \{q_1, q_2, q_3\} = \{\mathbf{1001}, \mathbf{0101}, \mathbf{0110}\}$ is the set of states: "too cold," "just right," and "too hot."
- Σ and $\mathcal{P}$ are as in $\mathbf{H}$;
- $\tau : 1 \rightarrow \mathcal{P} \cup \{\emptyset\}$ and is associated with $\text{Test}(\tau) = \{\top : p' = \tau\} \cup \{\varphi : p' \in \varphi(\tau), \varphi \in \mathbf{\Phi}\}$, where $\mathbf{\Phi} = \{\cup_i \varphi_i\}$, $\varphi_1(p) = \{p' \mid p' \in (p, 100]\}$, $\varphi_2(p) = \{p' \mid p' \in (\frac{p+\epsilon}{1-\rho}, 100]\}$, $\varphi_3(p) = \{p' \mid p' \in [p_e, \frac{p-\epsilon}{1+\rho})\}$, $\varphi_4(p) = \{p' \mid p' \in (p(1-\rho) - \epsilon, p(1-\rho) + \epsilon)\}$, $\varphi_5(p) = \{p' \mid p' \in [p_e, \frac{(1-\rho)p}{1+\rho})\}$, $\varphi_6(p) = \{p' \mid p' \in (\frac{p+\epsilon}{1+\rho}, p)\}$, $\varphi_7(p) = \{p' \mid p' \in (\frac{(1+\rho)p}{1-\rho}, 100]\}$, $\varphi_8(p) = \{p' \mid p' \in (p(1+\rho) - \epsilon, p(1+\rho) + \epsilon)\}$;
- Δ is a set of (a) *Read* transitions: $(q_1, \top) \xrightarrow{H} (q_2, \text{right})$, $(q_3, \top) \xrightarrow{O_1} (q_2, \text{right})$, $(q_2, \top) \xrightarrow{O_2} (q_1, \text{right})$; (b) *Write* transitions: $(q_1, \varphi_1) \xrightarrow{H} \big(q_1, \iota(H), \text{stay}\big)$;

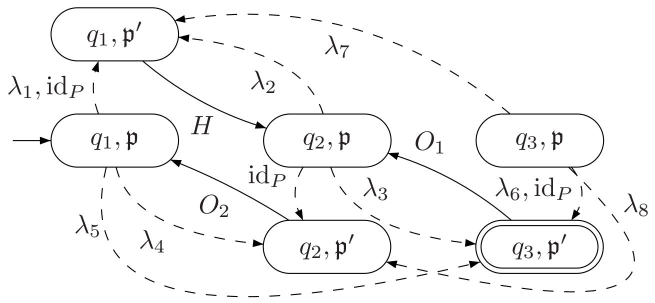

Fig. 1. The transformation semiautomaton $\mathbf{TR}(\mathbf{H})$.

$(q_2, \varphi_2) \xrightarrow{H} \big(q_1, \iota(H), \text{stay}\big)$; $(q_2, \varphi_3) \xrightarrow{O_1} \big(q_3, \iota(O_1), \text{stay}\big)$; $(q_1, \varphi_4) \xrightarrow{O_2} \big(q_2, \iota(O_2), \text{stay}\big)$; $(q_1, \varphi_5) \xrightarrow{O_1} \big(q_3, \iota(O_1), \text{stay}\big)$; $(q_3, \varphi_6) \xrightarrow{O_1} \big(q_3, \iota(O_1), \text{stay}\big)$; $(q_3, \varphi_7) \xrightarrow{H} \big(q_1, \iota(H), \text{stay}\big)$; $(q_3, \varphi_8) \xrightarrow{O_2} \big(q_2, \iota(O_2), \text{stay}\big)$.

The transformation semiautomaton that can be constructed from $\mathbf{R}(\mathbf{H})$ is the triple $\mathbf{TR}(\mathbf{H}) = (\bar{\mathcal{Q}}, \bar{\Sigma}, \bar{\Delta})$ in which:

- $\bar{\mathcal{Q}} = \{(q_1, \mathfrak{p}), (q_1, \mathfrak{p}'), (q_2, \mathfrak{p}), (q_2, \mathfrak{p}'), (q_3, \mathfrak{p}), (q_3, \mathfrak{p}')\}$;
- $\bar{\Sigma} = \{H, O_1, O_2, \lambda_1, \ldots, \lambda_8, \mathrm{id}_P\}$;
- $\bar{\Delta}$ as represented in Fig. 1.

The range for the thermostat settings is $\mathcal{P} = [p_e, 100]\,°\mathrm{C}$. The variables in (2) are $p_e = 6$ °C, $\rho = 0.05$, and $\epsilon = 0.1$.

The settling times Δt of each of the modes of $\mathbf{H}$ governed by (2a) or (2b) are given, (with t measured in minutes) $\Delta t(H) = \frac{1}{a_h} \log\left[\frac{p - x(t_0)}{\epsilon}\right], \Delta t(O_1) = \frac{1}{a_o} \log\left[\frac{p_e - x(t_0)}{p_e - (p + \epsilon)}\right], \Delta t(O_2) = \frac{1}{a_o} \log\left[\frac{p_e - x(t_0)}{p_e - p(1-\rho)}\right]$. With $x_0 < (1-\rho)p_0 = 15.2$ °C and thus $q_0 = V_M(x_0, p_0) = \mathbf{1001} = q_1$. For the final state as SPEC $\cap Q = \{{*}{*}\mathbf{1}{*}\} \cap Q = \{q_3\}$ we have $q_f = q_3$.Intuitively, the solution would be to set the thermostat desired temperature *above* the final thermostat setting, after which the thermostat is reset to p_f and the system finds itself in the "too hot" (q_3) state. Simply searching the graph of Fig. 1 does not yield that solution. This example shows that the proposed methodology does.

First, given initial and final states $(q_1, \mathfrak{p})$ and $(q_3, \mathfrak{p}')$ (cannot choose $(q_3, \mathfrak{p})$ because q_3 can not be reached by a read transition), generate the regular expression $\mathbf{RE}(\mathbf{H}) = \big[((\lambda_1 + \mathrm{id}_P)H + \lambda_5 O_1)(\lambda_2 H + \lambda_3 O_1)^* \mathrm{id}_P O_2 + \lambda_4 O_2\big]^* [((\lambda_1 + \mathrm{id}_P)H + \lambda_5 O_1)(\lambda_2 H + \lambda_3 O_1)^* \lambda_3 + \lambda_5]$.

We first test for walks of length $m = 1$: $\mathfrak{w}_1 = \lambda_5$, but $p_f = 32 \notin \varphi_5(p_0) = [p_e, \frac{(1-\rho)p_0}{(1+\rho)}) = [6, 14.47)$. Thus w_1 is excluded. There is no walk of length $m = 2$ and for $m = 3$, we have $\mathfrak{w}_2 = \lambda_4 O_2 \lambda_5$, $\mathfrak{w}_3 = \lambda_1 H \lambda_3$, $w_4 = \lambda_5 O_1 \lambda_3$, $w_5 = \mathrm{id}_P H \lambda_3$. Checking the associated set-valued maps, we find that only $M_{\mathfrak{w}_3}(p_0) = \varphi_3 \circ \varphi_1(p_0) = (p_e, \sup_{p \in (p_0, 100]} \frac{p+\epsilon}{1+\rho}) = (6, 95.33)$ has p_f in its range. Since each write transition has to be followed by a read one, the associated input data word is $w = (H, p_1)(O_1, p_2)$, $p_2 = p_f = 32\ °C$.[4]

A (rather trivial) DP implementation can be used to select p_1; however the same principle applies to walks of any finite length. First we compute $P_1 = M_1(p_0) \cap M_2^{-1}(p_f) = ((1+\rho)p_f - \epsilon, 100) = (33.7, 100)$ and $P_2 = \{32\}$. Then $J_2(p_1) = \frac{1}{a_o} \log\left[\frac{p_e - p_1(1+\rho)}{p_e - (p_1 + \epsilon)}\right]$, $J_2(p_1) = \min_{p_1 \in P_1} \frac{1}{a_h} \log\left[\frac{p_1 - x(t_0)}{\epsilon}\right] + \frac{1}{a_o} \log\left[\frac{p_e - p_1(1+\rho)}{p_e - (p_1 + \epsilon)}\right]$, and since the cost function happens to be a strictly increasing function of p, the solution is $p^* = 33.71$°C, $\quad J^* = 11.05\,\text{min}$. The optimal plan is $(H, 33.71)\,(O_1, 32)$.

[4]The presence of O_1 at the end of each candidate walk has no significance other than reminding us that the end state for candidate solutions should be q_3 and thus satisfy PRE(O_1).

VI. CONCLUSION

The class of hybrid systems studied in this paper have convergent continuous dynamics, lending themselves to a type of predicate abstraction that partitions the continuous state space based on the convergence properties of the set of component vector fields. A weak simulation relation between concrete and abstract system allows the control plan devised using the abstraction to be implemented on the concrete one. In this way, one can design time-optimal controller sequencing working at a purely discrete domain. We propose a combination of a DP algorithm with a graph search algorithm to generate these switching sequences and their associated control parameterizations. Derived solutions may be sub-optimal, depending on the discretization resolution on the solution (parameter) space.

REFERENCES

[1] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2006.

[2] S. Hedlund and A. Rantzer, "Optimal control of hybrid systems," in *IEEE Conference on Decision and Control, 38th, Phoenix, AZ*, 1999, pp. 3972–3977.

[3] X. Xu and P. Antsaklis, "Optimal control of switched systems based on parameterization of the switching instants," *Automatic Control, IEEE Transactions on*, vol. 49, no. 1, pp. 2 – 16, jan. 2004.

[4] X. Xu and P. J. Antsaklis, "Results and perspectives on computational methods for optimal control of switched systems," in *Proceedings of the 6th international conference on Hybrid systems: computation and control*, ser. HSCC'03. Berlin, Heidelberg: Springer-Verlag, 2003, pp. 540–555.

[5] X. C. Ding, Y. Wardi, and M. Egerstedt, "On-line adaptive optimal timing control of switched systems," in *Conference on Decision and Control*, 2009, pp. 5305–5310.

[6] Y. Boccadoro, M. Egerstedt, and E. Verriest, "Optimal control of switching surfaces in hybrid dynamical systems," *Discrete Event Dynamic Systems*, vol. 15, pp. 433–448, December 2005.

[7] H. Axelsson, Y. Wardi, M. Egerstedt, and E. I. Verriest, "Gradient descent approach to optimal mode scheduling in hybrid dynamical systems," *Journal of Optimization Theory and Applications*, vol. 136, pp. 167–186, 2008.

[8] M. Kaminski and N. Francez, "Finite-memory automata," *Theoretical Computer Science*, vol. 134, no. 2, pp. 329–363, 1994.

[9] F. Neven, T. Schwentick, and V. Vianu, "Finite state machines for strings over infinite alphabets," *ACM Transactions on Computational Logic*, vol. 5, no. 3, pp. 403–435, 2004.

[10] Y. Cohen-Sygal and S. Wintner, "Finite-state registered automata for non-concatenative morphology," *Computational Linguistics*, vol. 32, pp. 49–82, March 2006.

[11] F. Moller, "Logics for concurrency: structure versus automata," *ACM Computing Surveys*, p. 50, 1996.

[12] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Kluwer Academic, 2001.

[13] J. A. Brzozowski, "Derivatives of regular expressions," *J. ACM*, vol. 11, pp. 481–494, October 1964.

[14] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Athena Scientific, Nov. 2000, vol. I.