2017 American Control Conference
Sheraton Seattle Hotel
May 24–26, 2017, Seattle, USA

Fuel Cell Thermal Management: Modeling, Specifications and Correct-by-Construction Control Synthesis

Liren Yang, Amey Karnik, Benjamin Pence, Md Tawhid Bin Waez, Necmiye Ozay

Abstract— **Thermal management is crucial for safe and efficient operation of fuel cells. The goal of this paper is to algorithmically synthesize a provably-correct controller for a fuel cell thermal management system. For this purpose, we start with developing a control-oriented model for the fuel cell thermal management system and list the associated requirements. Then, we identify some structural properties of the system dynamics that can be leveraged for making the abstraction-based synthesis algorithm computationally efficient. Finally, we synthesize a controller for this system and demonstrate the closed-loop system behavior via simulations.**

I. Introduction

Fuel cells are electrochemical devices that convert chemical energy of gaseous fuel (i.e., hydrogen) into electricity [10]. In a fuel cell stack, electrochemical reaction of oxygen and hydrogen generates electrical power, while heat and water are produced as by-products. The fuel cell control system needs to supply the reactant and remove the by-product. In this work, we focus on developing the thermal management portion of the controller, which guarantees the fuel cell operates in a proper temperature range (340K to 350K), for a safety and efficiency consideration [5].

A simplified schematic of the fuel cell thermal management system is shown in Fig. 1. The stack coolant inlet temperature and the coolant flow-rate are the two main factors that affect the heat supplied or removed from the fuel cell stack, and hence the stack temperature. The coolant flow rate is controlled by an electric pump, while the coolant inlet temperature is regulated by appropriately flowing the coolant through a radiator or a heater, where the flow path is selected by a 2-position 3-way valve. Thus the system dynamics is hybrid in its nature.

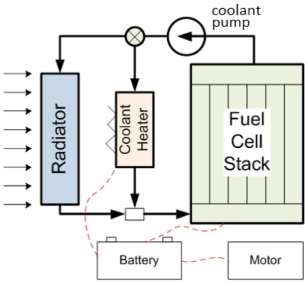


Fig. 1: Layout of fuel cell thermal management system.

LY and NO are with the Dept. of Electrical Engineering and Computer Science, Univ. of Michigan, Ann Arbor, MI 48109, USA yliren,necmiye@umich.edu. AK, BP and MTBW are with the Ford Research & Advanced Engineering, Dearborn, MI 48121, USA akarnik,bpence1,mwaez@ford.com. This work is supported by Ford Motor Co.

The electrical power requirements have a direct influence on thermal management, and some of the aspects are studied in [3]. The vehicle driving requirements at the motor are met through appropriate power management between the battery and the fuel cell stack. Heat generated in the stack increases with increasing fuel cell power request. Additional power requirements from the heater when used for warm-up under cold conditions also affect power management. In this paper, we provide some of the key requirements for fuel-cell thermal management in presence of battery state of charge energy constraints. These requirements are evaluated under the case where the ambient temperature is near 283K, where the fuel cell stack loses significant heat to the ambient due to large temperature gradient.

In this paper we propose to synthesize a controller for fuel cell thermal management system using abstraction-based formal synthesis techniques. Such techniques allow us to algorithmically generate a controller that is correct-by-construction, meaning the controller can be proved to satisfy a given specification, typically expressed in temporal logic [11], rather than tested by Monte-Carlo simulations. In order to apply abstraction-based synthesis to the complex fuel cell dynamics, we extend the existing techniques in two directions by (i) deriving a novel sufficient condition for mixed monotonicity, a structural property of dynamics that eases abstraction process; (ii) proposing multi-action state-dependent progress groups to capture a rich set of transience properties of underlying dynamics under different controls.

Fig. 2 summarizes the methodology used in this paper. For modeling, we adopt the fuel cell stack thermal model developed in [9] and enhance it using radiator and heater model components. For specifications, we include requirements regarding to temperature targets, energy management as well as requirements for battery state of charge (SOC), and formally summarize them in linear temporal logic (LTL). The model and specifications are further analyzed to develop formally-correct-switching controller using abstraction-based synthesis.

II. Fuel Cell Model

A block diagram of the fuel cell thermal management system is shown in Fig. 3. The solid lines (red) are temperature signals, the dotted lines (purple) are power signals, the dashed lines (blue) are battery state of charge signals, and thinner solid lines (black) are control/reference input signals. The physical meanings of variables in Fig. 3 can be found in Appendix A. Other operating conditions (such as hydrogen and oxygen partial pressure, ambient temperature,

978-1-5090-5992-8/$31.00 ©2017 AACC

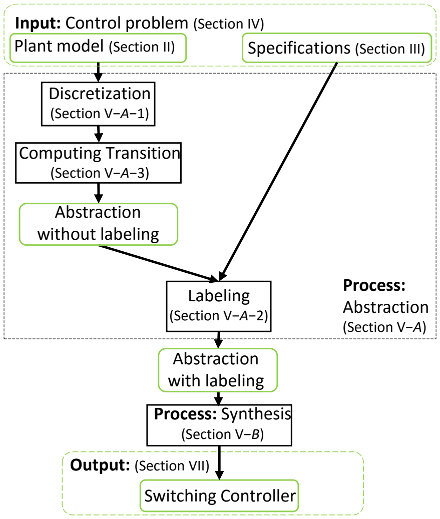

Input: Control problem (Section IV)
Plant model (Section II)
Specifications (Section III)
Discretization (Section V–A–1)
Computing Transition (Section V–A–3)
Abstraction without labeling
Labeling (Section V–A–2)
Process: Abstraction (Section V–A)
Abstraction with labeling
Process: Synthesis (Section V–B)
Output: (Section VII)
Switching Controller

Fig. 2: Methodology and paper organization.

vehicle speed) that affect system dynamics are not explicitly included in the block diagram for simplicity.

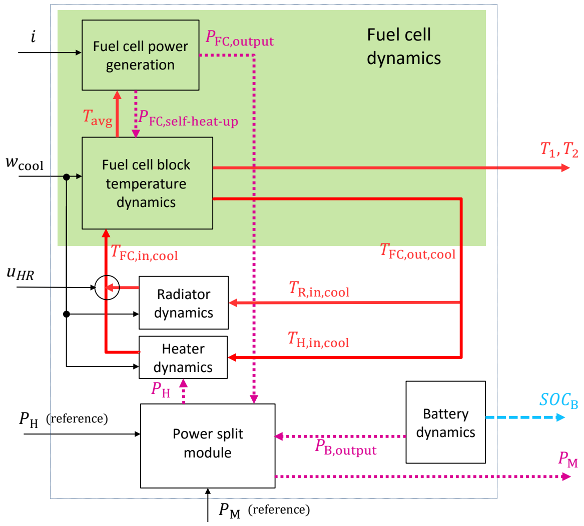


Fig. 3: Block diagram of fuel cell thermal management system.

In what follows, we give the formulas describing each block in Fig. 3.

A. Fuel Cell Power Generation

The fuel cell stack output power and generated heat are computed using the formulas developed in [9],

$$P_{\mathrm{FC,output}} = iA_{\mathrm{G}}E_{\mathrm{FC,stack}}, \tag{1}$$

$$P_{\mathrm{FC,self-heat-up}} = iA_{\mathrm{G}}\frac{\Delta h_{\mathrm{rxn}}}{2F}n_{\mathrm{FC,cell}} - P_{\mathrm{FC,output}}, \tag{2}$$

and

$$\begin{aligned}E_{\mathrm{FC,stack}} = n_{\mathrm{FC,cell}}\Bigg(&\frac{\Delta h_{\mathrm{rxn}}}{2F} - T_{\mathrm{avg}}\frac{\Delta s_{\mathrm{rxn}}}{2F}\\ &+\frac{RT_{\mathrm{avg}}}{2F}\ln\left(\frac{p_{\mathrm{H_2}}}{P_{\mathrm{ref}}}\left(\frac{p_{\mathrm{O_2}}}{P_{\mathrm{ref}}}\right)^{\frac{1}{2}}\right) - \frac{RT_{\mathrm{avg}}}{\alpha F}\ln\left(\frac{i+i_{\mathrm{x}}}{i_0}\right)\\ &-iR_{\Omega} - a_{\mathrm{MT}}\left(\frac{i}{i_{\mathrm{MT}}}\right)^{b_{\mathrm{MT}}}\Bigg),\end{aligned} \tag{3}$$

where $\frac{\Delta h_{\mathrm{rxn}}}{2F}$ and $T_{\mathrm{avg}}\frac{\Delta s_{\mathrm{rxn}}}{2F}$ correspond to the effect of enthalpies and entropies, iR_{Ω} describes Ohmic loss due to cell resistivity, and $a_{\mathrm{MT}}(\frac{i}{i_{\mathrm{MT}}})^{b_{\mathrm{MT}}}$ describes potential loss caused by mass transport limitations. The variables h_{rxn}, s_{rxn}, i_0, R_{Ω} depend on fuel cell average temperature T_{avg} and operating conditions [9].

B. Fuel Cell Temperature Dynamics

The fuel cell stack is divided into two control volumes to capture its temperature gradient. One control volume is at the coolant inlet side and the other is at the coolant outlet side. The fuel cell temperature dynamics is described in terms of the temperature of the two control volumes, i.e., T_1, T_2. The temperature dynamics are governed by the following differential equation [9]:

$$\begin{aligned}\frac{dT_1}{dt} = \frac{1}{c_{\mathrm{FC}}\rho_{\mathrm{FC}}}\Bigg(&\frac{c_{\mathrm{cool}}w_{\mathrm{cool}}(T_{\mathrm{FC,in,cool}} - T_1)}{n_{\mathrm{FC,cell}}A_{\mathrm{FC}}\delta_{\mathrm{FC}}/2}\\ &+\frac{\kappa_{\mathrm{T}}(T_2 - T_1)}{(\delta_{\mathrm{FC}}/2)^2} + k_{\mathrm{amb}\rightarrow\mathrm{FC}}(T_{\mathrm{amb}} - T_1)\\ &+\frac{P_{\mathrm{FC,self-heat-up}}}{V_{\mathrm{FC}}n_{\mathrm{FC,cell}}} - r_{\mathrm{v}}\Delta h_{\mathrm{v}}\Bigg),\end{aligned} \tag{4}$$

$$\begin{aligned}\frac{dT_2}{dt} = \frac{1}{c_{\mathrm{FC}}\rho_{\mathrm{FC}}}\Bigg(&\frac{c_{\mathrm{cool}}w_{\mathrm{cool}}(T_1 - T_2)}{n_{\mathrm{FC,cell}}A_{\mathrm{FC}}\delta_{\mathrm{FC}}/2}\\ &+\frac{\kappa_{\mathrm{T}}(T_1 - T_2)}{(\delta_{\mathrm{FC}}/2)^2} + k_{\mathrm{amb}\rightarrow\mathrm{FC}}(T_{\mathrm{amb}} - T_2)\\ &+\frac{P_{\mathrm{FC,self-heat-up}}}{V_{\mathrm{FC}}n_{\mathrm{FC,cell}}} - r_{\mathrm{v}}\Delta h_{\mathrm{v}}\Bigg),\end{aligned} \tag{5}$$

where the inlet coolant temperature $T_{\mathrm{FC,in,cool}}$ in Eq. (4) is defined as

$$T_{\mathrm{FC,in,cool}} = u_{\mathrm{HR}}T_{\mathrm{H}} + (1 - u_{\mathrm{HR}})T_{\mathrm{R}}, \tag{6}$$

where u_{HR} is the binary variable controlling the 2-position 3-way valve. The average fuel cell temperature used in Eq. (3) is defined as $T_{\mathrm{avg}} = (T_1 + T_2)/2$, while $T_{\mathrm{FC,out,cool}}$, the outlet coolant temperature from fuel cell stack, is assumed to be equal to T_2.

C. Radiator and Heater Temperature Dynamics

The radiator and heater dynamics are given by

$$\begin{aligned}\frac{dT_{\mathrm{R}}}{dt} = \frac{1}{C_{\mathrm{R}}}\big(&(1 - u_{\mathrm{HR}})c_{\mathrm{cool}}w_{\mathrm{cool}}(T_{\mathrm{FC,out,cool}} - T_{\mathrm{R}})\\ &+ c_{\mathrm{air}}\varepsilon(v)v(T_{\mathrm{amb}} - T_{\mathrm{R}})\big),\end{aligned} \tag{7}$$

$$\frac{dT_{\mathrm{H}}}{dt} = \frac{1}{C_{\mathrm{H}}}\left(u_{\mathrm{HR}}c_{\mathrm{cool}}w_{\mathrm{cool}}(T_{\mathrm{FC,out,cool}} - T_{\mathrm{H}}) + P_{\mathrm{H}}\right). \tag{8}$$

Note that when binary control $u_{\mathrm{HR}} = 1$ (or 0), the coolant is fed to heater (or radiator). The term $\varepsilon(v)$ in radiator dynamics is the vehicle-speed-dependent effectiveness of radiator, which is modeled as an affine function of vehicle speed v. The outlet coolant temperature from radiator (heater, respectively) is assumed to be T_{R} (T_{H}, respectively).

D. Battery SOC Dynamics

The battery SOC dynamics is adopted from the one given in [6],

$$\frac{dSOC_{\mathrm{B}}}{dt} = -n_{\mathrm{s}}n_{\mathrm{p}}E_{\mathrm{B,cell}}\frac{E_{\mathrm{B,cell}} - \sqrt{E_{\mathrm{B,cell}}^2 - \frac{4P_{\mathrm{B,output}}r_{\mathrm{B,cell}}}{n_{\mathrm{s}}n_{\mathrm{p}}}}}{2r_{\mathrm{B,cell}}G_{\mathrm{B,stack,total}}}. \tag{9}$$

Note that in Eq. (9), $P_{\mathrm{B,output}}$ can be negative, meaning charging the battery.

E. Power Split Module

The power split module combines the output power from the fuel cell and the battery, and passes part of the combined power to the heater, and the remaining portion to the motor. To deliver the required power to the motor, we assume the battery always provide right amount of power to compensate what is generated by the fuel cell, i.e.,

$$P_{\mathrm{B,output}} = P_{\mathrm{M}} + P_{\mathrm{H}} - P_{\mathrm{FC,output}}. \tag{10}$$

III. Specifications

In this section we give the specifications (or requirements) of fuel cell thermal management in both plain English and linear temporal logic (TABLE I). The listed specification are classified into 3 types, i.e., (i) "reach-stay" type, (ii) "avoid" type, (iii)" liveness" type.

A. Limitations of Fuel Cell Output Power

Fig. 4 gives the fuel cell output power predicted by model in section II-A. This part impose some requirements regarding to fuel cell output power.

Spec1: (avoid) The fuel cell output power $P_{\mathrm{FC,output}}$ should not drop below zero.

As show in Fig. 4, the model predicted fuel cell output power becomes negative when the current density is too high, which makes the model invalid at that value of current density. We hence need this requirement to avoid operating in the region where the model is invalid.

Spec2: (avoid) The fuel cell current density should not exceed the one giving maximum output power $P_{\mathrm{FC,max}}$ because operating above $P_{\mathrm{FC,max}}$ is inefficient and could lead to irreversible degradation [4]. where i^* is the current density that gives the maximum fuel cell output power, graphically illustrated by Fig. 4. Note that i^* is a function of state and operating conditions.

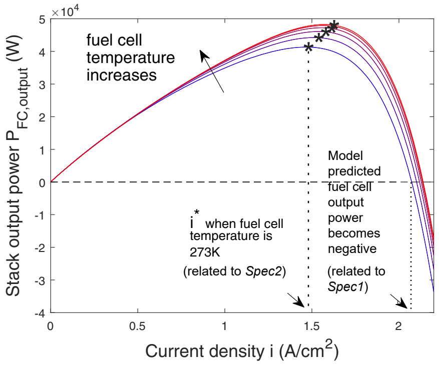


Fig. 4: Fuel cell power vs current density, fuel cell average temperature varies in [273, 360]K, membrane water content $\lambda = 6$.

By Fig. 4, it is obvious that *Spec2* actually implies *Spec1*. In this work, however, we have not included requirement *Spec2* because of the difficulty computing i^*. Thus we consider requirement *Spec1* instead.

B. Battery Energy & Power Limitations

Spec3: (avoid) The battery stack energy should not drop below 10%, or exceed 90%.

Spec4: (liveness) Battery energy should always recover to $SOC_{\mathrm{B,target}}$ (with at most an error δ) in finite time, where $SOC_{\mathrm{B,target}}$ is a set point given by energy management module. We omit this specification for now in this paper because our current synthesis technique is specialized for reach-stay-avoid specifications.

Spec5: (avoid) The battery power should not exceed peak power requirements.

Spec6: (avoid) Power for battery charge should not exceed maximum allowable charging power. Note that by our convention, charging power (both $P_{\mathrm{B,output}}$ and $P_{\mathrm{B,charge,max}}$) is negative.

C. Regular Operation Requirements

Spec7: (reach-stay) Fuel cell block temperatures will reach and then stay at target temperature range $[340, 350]$K.

By this requirement, when the fuel cell is temporarily shut down and motor power is completely delivered by the battery, we still want the fuel cell temperature to stay in the range.

Spec8: (avoid) Fuel cell block temperatures never exceed maximum allowable temperature 353K.

TABLE I: Specifications in LTL

Specification	LTL formula	type	
Spec1	$\varphi_1 = \Box(P_{\mathrm{FC,output}} \geq 0)$	avoid	✓
Spec2	$\varphi_2 = \Box(i \leq i^*)$	avoid	x
Spec3	$\varphi_3 = \Box(0.1 \leq SOC_{\mathrm{B}} \leq 0.9)$	avoid	✓
Spec4	$\varphi_4 = \Box\Diamond(SOC_{\mathrm{B}} = SOC_{\mathrm{B,target}})$	liveness	x
Spec5	$\varphi_5 = \Box(P_{\mathrm{B,output}} \leq P_{\mathrm{B,output,max}})$	avoid	✓
Spec6	$\varphi_6 = \Box(P_{\mathrm{B,output}} \geq P_{\mathrm{B,charge,max}})$	avoid	✓
Spec7	$\varphi_7 = \Diamond\Box(\wedge_{j=1,2}(T_j \in [340, 350]))$	reach-stay	✓
Spec8	$\varphi_8 = \Box(\wedge_{j=1,2}(T_j \leq 353))$	avoid	✓

"✓": the specification is considered in the synthesis.
"x": the specification is omitted for now.

IV. Problem Statement

In this section we formally state the control problem, by summarizing the plant model given in section II, and selecting a set of requirements defined in section III.

As a short notation, denote the system dynamics described in section II by

$$\dot{x} = f(x, u, d) \tag{11}$$

where $x = [T_1, T_2, T_{\mathrm{R}}, T_{\mathrm{H}}, SOC_{\mathrm{B}}]^T$ denotes the state, $u = [i, w_{\mathrm{cool}}, u_{\mathrm{HR}}, P_{\mathrm{H}}]^T$ denotes the control, $d = [P_{\mathrm{M}}, p_{\mathrm{O_2}}, p_{\mathrm{H_2}}, T_{\mathrm{amb}}, v, \lambda, r_{\mathrm{v}}]^T$ denotes the operating condition. In particular, vector field f are defined by Eq. (4) to (9). Let X, U, D denote the domains for x, u, d. Set X, U, D are rectangular sets defined in Appendix A.

In this work we consider all requirements listed in section III except *Spec2* and *Spec4*. In addition to the selected requirements, define LTL formula

$$\Box(x \in X), \qquad (\varphi_9)$$

to constrain that the system states never leave the considered domain X, and define assumptions on the environment

$$\Box(d \in D), \qquad (\varphi_{\mathrm{env}})$$

to constrain that the operating conditions always stay in allowable range. Then the overall specification for consideration in LTL is given by

$$\Phi := \varphi_{\text{env}} \rightarrow \bigwedge_{\substack{i=1 \\ i \neq 2,4}}^{9} \varphi_i, \tag{12}$$

that is, if the environment variables (operating conditions) always stay in their allowable range, all the selected specifications are satisfied.

Problem 1: Given the plant model defined in (11), and desired closed-loop behavior specified by LTL formula Φ, defined by (12), synthesize a state feedback controller $K : X \rightarrow U$, under which all closed-loop trajectories governed by $\dot{x} = f(x, K(x), d)$, satisfy the LTL specification Φ.

V. Solution Approach

We formulate the problem as a reach-stay-avoid game [7] for a switched system and solve the game for a switching protocol using abstraction-based synthesis technique. The idea is: instead of solving the control problem directly on the given model (concrete system), we create a finite transition system (abstraction) with discretized control that captures the properties of interest for the continuous dynamics of the concrete system, and solve the control problem on the abstraction [11]. The obtained controller is provably correct for the concrete system because by construction the behaviors of the concrete system is a subset of that of the abstraction[1]. If one can find a controller, under which all the closed-loop behaviors of the abstraction satisfy specified requirement, the closed-loop behaviors of the concrete system also satisfy the requirement with the same controller, by the behavior inclusion relation.

A. Abstraction

The abstraction process returns a finite transition system for given plant model and specifications. The transitions capture the flow of the continuous plant dynamics, and the (discrete) states of the finite transition system are properly labeled according to the given specification. For a formal definition and algorithms generating abstractions, we refer the reader to [8].

As shown in Fig. 2, abstraction process is decomposed into three steps, i.e., discretization, labeling and transition computation. For general nonlinear system, the computations involved in the above process are hard. In the rest of this section, we identify several system properties of the considered fuel cell thermal management system that make the abstraction computations relatively efficient.

1) Discretization: We first partition the state space of given concrete system into finitely many regions. Each region is mapped to a discrete state in the finite transition system, called the symbol of that region. We use a manually constructed non-uniform rectangular grid partition in the state space and control space. Rectangular partition reduces the computation effort required for abstraction significantly under certain conditions, as will be shown in section V-A.2 and V-A.3.

2) Labeling: After state space partition, each region in the partition needs to be labeled as "target", "safe" and "unsafe" according to the specification.

The selected requirements are either "reach-stay" or "avoid" type. Consider "reach-stay" specification *Spec7*. The regions contained by set $\{x \in X \mid T_1 \text{ and } T_2 \in [340, 350]\}$ are labeled as "target". For "avoid" specifications, a region is labeled as "safe" if the specification is satisfied everywhere in that region for all operating conditions, or labeled as "unsafe" if the specification is violated somewhere in the region for some operating conditions.

The challenge is that some "avoid" specifications are implicitly related to states and operating conditions. For example, requirement *Spec1* requires fuel cell output power $P_{\text{FC,output}} \geq 0$ (or equivalently $E_{\text{FC,stack}} \geq 0$ by Eq. (1)), $P_{\text{FC,output}}$ is a function of both system state (fuel cell temperature T_{avg}) and operating condition (membrane water content λ, hydrogen-oxygen partial pressure $p_{\text{H}_2}, p_{\text{O}_2}$). Therefore, to label a region to be safe or unsafe in terms of *Spec1*, we need to check the worst case in that region. That is, if the minimum fuel cell output power $P_{\text{FC,output}}$ (or equivalently $E_{\text{FC,stack}}$) in the region is negative under some operating conditions (which violate specification *Spec1*), the region is labeled as unsafe. *Spec1.*

As described in section II-A, $E_{\text{FC,stack}}$ is a nonlinear function in state x and operating condition d. Therefore, finding the exact minimum value of $E_{\text{FC,stack}}$ requires to solve a nonlinear optimization problem over x and d, which might be intractable. However, function $E_{\text{FC,stack}}(x, d)$ allows an efficient and reasonable approximation by Theorem 1 in Appendix B if the considered regions are rectangles. Theorem 1 applies to function $E_{\text{FC,stack}}(x, d)$, because $E_{\text{FC,stack}}(x, d)$ is continuously differentiable w.r.t. x and d on compact set $X \times D$. This means all the continuous partial derivatives $\frac{\partial E_{\text{FC,stack}}}{\partial x}$ $\frac{\partial E_{\text{FC,stack}}}{\partial d}$ are bounded on $X \times D$, thus $E_{\text{FC,stack}}(x, d)$ satisfies the hypothesis of Theorem 1.

By Theorem 1, approximating the minimum (or maximum) value of $E_{\text{FC,stack}}$ reduces to evaluating $E_{\text{FC,stack}}$ at two extreme points of the considered rectangular region. The result of the approximation is illustrated using Fig. 5, the dashed line is the maximum and minimum value when x_1, x_2, or T_1, T_2 varies in $[273, 360]$K. As Fig. 5 shows, there is a gap between approximated minimum value of $E_{\text{FC,cell}}$ and real values. This gap indicates that the approximation is conservative. However, when size of the region to be labeled is smaller, the approximation is tighter.

3) Computing Transitions: We compute transitions in the abstraction by arguing about the vector field directions of the concrete system, over a region in state space, and also over all operating conditions. In this part we give an efficient way to compute these transitions using Theorem 1 and Theorem 2 in Appendix B.

As shown in the left part of the Fig. 6, Y_1 and Y_2 are

[1] An abstraction may contain more behavior than the underlying concrete system. The "bad" behaviors that exist in abstraction but not in concrete system are called spurious.

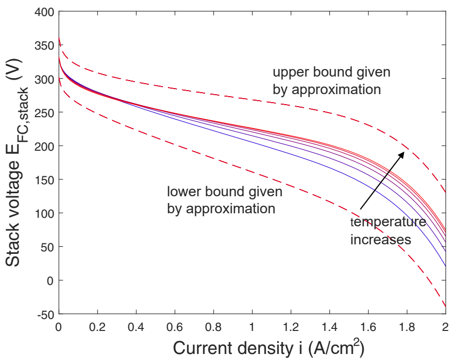


Fig. 5: Approximation of polarization, fuel cell average temperature varies in [273, 360]K.

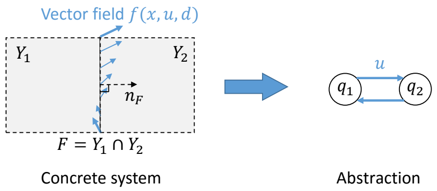


Fig. 6: Computing transitions by arguing direction of vector field.

two adjacent regions in state space of concrete system, $F = Y_1 \cap Y_2$ is the adjacent facet between two regions, dashed line arrow n_F is the normal vector of facet F (pointing from Y_1 to Y_2), and the solid arrows on F are the vector field $f(x, u, d)$ under some given u and operating conditions d. The right part of the figure shows the symbols in abstraction, in particular, symbol q_1 (q_2) corresponds to region Y_1 (Y_2), and the transitions between q_1 and q_2 are defined as follows

$$q_1 \xrightarrow{u} q_2 \text{ iff } \max_{\substack{x \,\in\, F, \\ d \,\in\, D}} n_F^T f(x, u, d) > 0. \tag{13}$$

Assume a rectangular partition of the state space, the adjacent facets are all rectangular facets, i.e., $F = \{x \in X \mid x_j \in [\underline{x}_j, \overline{x}_j]\}$, and their normal vectors are natural basis vectors e_i (a vector whose i^{th} entry is one, and the other entries are zeros). Also note that allowable operating condition set D is also a rectangular set by definition, i.e., $D = \{d \mid d_k \in [\underline{d}_k, \overline{d}_k]\}$. Eq. (13) thus becomes

$$q_1 \xrightarrow{u} q_2 \text{ iff } \max_{\substack{x_j \,\in\, [\underline{x}_j,\, \overline{x}_j] \\ d \,\in\, [\underline{d}_k,\, \overline{d}_k]}} f_i(x, u, d) > 0. \tag{14}$$

The optimum values in (14) can be over approximated using Theorem 1. Fix control u, and let ϕ^u be the decomposition function of $f(\cdot, u, \cdot)$ defined by Eq. (21), we have

$$\max_{\substack{x_j \,\in\, [\underline{x}_j,\, \overline{x}_j] \\ d \,\in\, [\underline{d}_k,\, \overline{d}_k]}} f_i(x, u, d) \leq \phi_i^u([\overline{x}, \overline{d}], [\underline{x}, \underline{d}]), \tag{15}$$

where $\underline{x} = [\underline{x}_1, \ldots, \underline{x}_n]^T$ and $\overline{x} = [\overline{x}_1, \ldots, \overline{x}_n]^T$ (similar for $\underline{d}$, $\overline{d}$), and

$$q_1 \xrightarrow{u} q_2 \text{ if } \phi_i^u([\overline{x}, \overline{d}], [\underline{x}, \underline{d}]) > 0, \tag{16}$$

By Remark 1 in Appendix B, if the partial derivative $\frac{\partial f_i}{\partial x_j}$ is not sign-stable, the approximations in Eq. (15) are not tight. In that case we may create more transitions than necessary by Eq. (16), and hence more spurious behavior in the abstraction. This leads to a more conservative solution but does not harm the correctness.

Note that the partial derivative $\frac{\partial f_3}{\partial v}$ is not sign-stable, where $f_3 = \dot{T}_{\mathrm{R}}$ is defined by Eq. (7). The sign of $\frac{\partial f_3}{\partial v}$ depends on which one of T_{R} and T_{amb} is larger. In this case the conservatism can be reduced using Theorem 2 in Appendix B. We show f_3 is affine in state x and multi-affine in $[T_{\mathrm{amb}}, w]$ where $w := \varepsilon(v)v$. By Theorem 2, to maximize (minimize, respectively) vector field component f_3, one only needs to evaluate f_3 at both upper and lower bounds of w, and pick the maximum (minimum, respectively) f_3 value. This is equivalent to evaluating f_3 at upper and lower bounds of vehicle speed v, because $w = \varepsilon(v)v$ is monotone increasing in v. With this modification, Eq. (15), become

$$\max_{\substack{x_j \,\in\, [\underline{x}_j,\, \overline{x}_j] \\ d \,\in\, [\underline{d}_k,\, \overline{d}_k]}} f_i(x, u, d) \leq$$
$$\max\left\{\phi_i^u([\overline{x}, \overline{\overline{d}}], [\underline{x}, \underline{d}]),\ \phi_i^u([\overline{x}, \overline{d}], [\underline{x}, \underline{\underline{d}}])\right\}, \tag{17}$$

where $\underline{\underline{d}}$ is the same as $\underline{d}$ except that its fifth entry (represent vehicle speed v) takes upper bound value; and $\overline{\overline{d}}$ is the same as $\overline{d}$ except that its fifth entry takes lower bound value.

Note that to compute transition from q_2 to q_1, one only need to pick the normal vector in Eq. (13) to be $n_F = -e_i$, and the above approximation process still applies to this case.

B. Synthesis

We synthesize a controller for the given abstraction by solving a reach-stay-avoid game using graph search algorithms. By such algorithms we will assign each symbol with a set of control actions (the set could be empty). The symbols assigned with nonempty control actions are called "winning". Under assigned actions, the closed-loop path starting from "winning" symbols will reach and stay in "target" symbols in finite time, meanwhile never entering "unsafe" symbols. Finally the obtained actions for each symbol are assigned to corresponding regions in concrete system's state space.

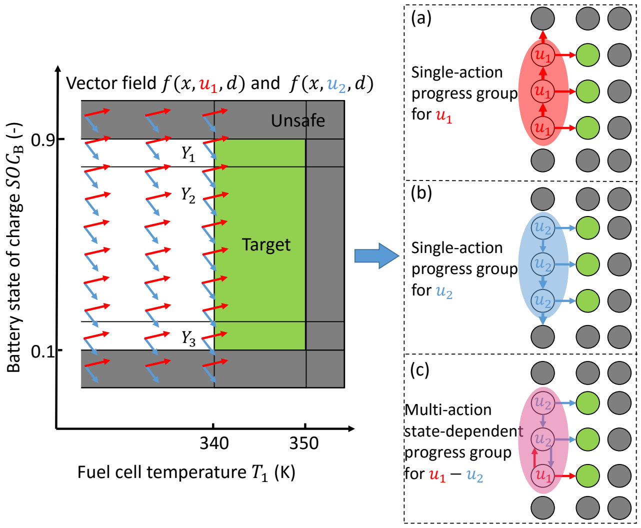
The synthesis algorithms used in this work are modified from the ones given in [7]. Starting with "stay" requirements, the algorithm first searches for a controlled invariant set within the "target" symbols. Then the algorithm solves for "reach" and "avoid" requirements by backwards expanding starting from obtained controlled invariant set, meanwhile avoiding "unsafe" symbols. In this process, control actions will be assigned to the expanded symbols. In particular, to solve the reachability part, we need to eliminate some spurious loops in abstraction that prevent target being reached. To this end we encode in abstraction some liveness properties of the underlining continuous system by progress groups. A set of symbols (each symbol assigned a set of control actions) form a progress group if these symbols correspond to a transient[2] region in original concrete system, under the

[2] A region is transient under some control actions if all trajectories starting from that region eventually leave the region in finite time under assigned control actions.

assigned actions.

C. Multi-action State-dependent Progress Group

In a previous work [8], the progress groups are defined for single action and can be encoded in abstraction. For this application, however, single action progress group is insufficient for accommodating battery SOC requirement and reachability requirement at the same time. We, hence, need multi-action state-dependent progress groups.



Vector field $f(x, u_1, d)$ and $f(x, u_2, d)$
Unsafe
Y_1
Y_2
Y_3
Target
0.9
0.1
340
350
Battery state of charge SOC_B (-)
Fuel cell temperature T_1 (K)
(a)
Single-action progress group for u_1
u_1
(b)
Single-action progress group for u_2
u_2
(c)
Multi-action state-dependent progress group for $u_1 - u_2$
u_2
u_1

Fig. 7: System flow on SOC_{B}-T_1 subspace (left), abstraction with single-action progress group (right (a), (b)) and abstraction with multi-action state-dependent progress group (right (c)).

Fig. 7 shows why the battery SOC requirement *Spec3* and reachability requirement cannot be satisfied at the same time by single-action progress groups. In the left part of the figure, we plot the rectangular partition and vector field projected on SOC_{B}-T_1 space. The green color marks the regions labeled as target and grey color marks unsafe regions. To reach the target region, we can either (i) let fuel cell do self-heat-up, meanwhile generating excess power and charging the battery (corresponding to action u_1), or (ii) use heater to warm up the fuel cell, and thus draw power from battery (corresponds to action u_2). Note that no action can keep battery SOC to be a constant because of the uncertainty in motor requested power P_{M}—P_{M} is an operating condition whose range is defined in Appendix A. The right part of the Fig. 7 shows the abstraction. By choosing a single action (i.e., case (a) (b)) there is always a path leading to unsafe symbols, therefore the battery SOC requirement is violated on the abstraction. Note that such path is spurious because it does not represent any real trajectories in concrete system (e.g., choosing u_1 at low battery SOC actually leads the trajectory into target region before saturating the battery). Such spurious behaviour exists in abstraction due to the conservatism introduce by rectangular partition.

However the battery SOC requirement can be satisfied by applying multiple actions, as shown in Fig. 7 (c), but the reachability requirement is violated by the infinite loop caused by alternatively choosing u_1 and u_2. We thus need multi-action state-dependent progress groups to eliminate such loops when they are spurious.

Since the number of multi-action state-dependent progress groups grows exponentially in the number of available control actions, it easily exhausts time and memory to pre-compute these progress groups and encode them in the abstraction before synthesis. Therefore, instead of doing pre-computation and storage before synthesis, we compute multi-action state-dependent progress group in synthesis process, and we will restrict the control actions on-the-fly based on the synthesis. Specifically, we do the following:

1. starting from a controlled invariant set C, as initial winning set, compute a set P of "safe" one-step-predecessors[3] of set C, with each symbol in P assigned with a set of actions, so that $C \cup P$ is controlled invariant under the assigned actions;
2. if the symbols in P correspond to a transient region under some actions assigned to them, these predecessors form a multi-action state-dependent progress group, and are added to winning set;
3. repeat step 1, 2, 3 until winning set has satisfactory size.

The rest of this part shows how to check transience (step 2 in above process). Take Fig. 7 as an example, shaded symbols form a multi-action state-dependent progress group when u_2 is assigned to the two symbols on the top and u_1 is assigned to the bottom symbol. This is because the region represented by these symbols is transient under corresponding actions. The transience can be checked efficiently by arguing the direction of vector field of underlying concrete system. As shown in left part of Fig. 7, the union of three regions $Y_1 \cup Y_2 \cup Y_3$ is transient because the horizontal component of vector field is always positive (i.e., pointing rightwards) when control u_2 is applied in Y_1 Y_2, and u_1 is applied in Y_3. More generally, given a set of regions $\{Y_k\}_{k=1}^m$ in n dimensional state space, each region equipped with one control action u_k, $Y = \bigcup_{k=1}^m Y_k$ is transient under assigned actions if there exists $v \in \mathbb{R}^n$,

$$\forall k = 1 \ldots m : \max_{x \in Y_k, d \in D} v^T f(x, u_k, d) > 0. \tag{18}$$

If vector v is $\pm e_i$ (natural bases), and Y_k are rectangles, the optimization problem in (18), can be solved efficiently, by the approach developed in section V-A.3.

VI. Results and Discussion

By the solution approach described in section V, a switching protocol is synthesized. The controller is able to achieve the specifications on the entire state domain X. The closed-loop behaviors are illustrated by simulation in Fig. 8, from which we make the following observations:

1. By Fig. 8 (1-1) (1-2) (5-2), all states stay in the domain, fuel cell temperature reaches and stays in target range, and battery SOC never exceeds upper or lower bounds.
2. By Fig. 8 (1-2), the heater temperature switch between 340K to 400K from about $t = 20$s to 90s. This is

[3] A symbol p is called a one-step-predecessor of a set Q, if there is a transition (under some actions assigned to p) leading p to some symbols $q \in Q$.

because heater needs to stay above 340K to be able to warm up the fuel cell, while it also needs to stay below its temperature upper bound (400K).

3. By Fig. 8 (1-1), the switching pattern of fuel cell temperature changes at about $t = 360$s, this is caused by low level of battery SOC at that time. To be specific, after fuel cell temperature reaches target range at about $t = 90$s, the controller starts to feed coolant through heater so that the fuel cell stays warm by drawing heat from coolant. Since heater draws power from battery, battery SOC keeps decreasing until it reaches allowable lower bound at $t = 360$s. In order to protect battery from over-discharging, the controller starts to alternatively use heater and radiator starting at $t = 360$s (and hence alternatively discharging and charging the battery), so that the battery SOC stops dropping, while fuel cell temperature still stay in target range.

APPENDIX

A. Variables and Constants

Control u		
u_{HR}	u_{HR} = 1	indicating that the coolant flow goes through the heater,
	u_{HR} = 0	indicating that the coolant flow goes through the radiator
i	[0,1.5]	(A cm^{-2}) Cell current density
P_{H}	[0, 35000]	(W) Power requested by heater
w_{cool}	[0,800]	(g s^{-1}) Coolant mass flow rate

State x		
SOC_{B}	[0,1]	(-) Battery energy
T_1	[273, 360]	(K) Temperature of first control volumes
T_2	[273, 360]	(K) Temperature of second control volumes
T_{H}	[250, 400]	(K) Heater temperature
T_{R}	[250, 340]	(K) Radiator temperature

Operating Condition d		
P_{M}	[2, 17]	(kW) Power requested by motor
p_{O_2}	5×10^4	(Pa) Oxygen partial pressure
p_{H_2}	1.5×10^5	(Pa) Hydrogen partial pressure
r_{v}	$[0, 10^{-7}]$	(mol cm^{-3} s^{-1}) Volumetric evaporating rate
v	[10,20]	(ms^{-1}) Vehicle speed
T_{amb}	[273,290]	(K) Ambient temperature
λ	[4,22]	(-) Membrane water content

Other Variables	
$E_{\mathrm{FC,stack}}$	(V) Fuel cell stack electrical potential
i_0	(A cm^{-2}) Exchange current density
$P_{\mathrm{B,output}}$	(W) Battery output power
$P_{\mathrm{FC,output}}$	(W) Fuel cell output power
$P_{\mathrm{FC,self-heat-up}}$	(W) Power for fuel cell self-heat-up
R_{Ω}	(Ω cm^2) Cell resistivity
T_{avg}	(J mol^{-1} K^{-1}) Average fuel cell temperature
$T_{\mathrm{FC,in,cool}}$	(K) Inlet coolant temperature (into fuel cell)
$T_{\mathrm{FC,out,cool}}$	(K) Outlet coolant temperature (from fuel cell)
Δh_{rxn}	(J mol^{-1}) Reaction enthalpy
Δh_{v}	(J mol^{-1}) Evaporation enthalpy
Δs_{rxn}	(J mol^{-1} K^{-1}) Reaction entropy

Constants	
c_{air}	(1.0 J g^{-1}K^{-1}) Air specific heat capacity
F	(96485 C mol^{-1}) Faraday constant
P_{ref}	(101325 Pa) Reference pressure
R	(8.314 J mol^{-1} K^{-1}) Universal gas constant

Parameters	
a_{MT}	(V) Mass transfer potential loss coefficient
A_{FC}	(cm^2) Fuel cell cross section area
A_{G}	(cm^2) Fuel cell geometric area
b_{MT}	(-) Mass transfer potential loss exponent
c_{cool}	(J g^{-1}K^{-1}) Coolant specific heat capacity
c_{FC}	(J g^{-1}K^{-1}) Fuel cell specific heat capacity
C_{H}	(J K^{-1}) Heater heat capacity
C_{R}	(J K^{-1}) Radiator heat capacity
$E_{\mathrm{B,cell}}$	(V) Battery cell open-circuit potential
$G_{\mathrm{B,stack,total}}$	(Ws) Battery stack energy capacity
$i_{0,\mathrm{ref}}$	(A cm^{-2}) Reference exchange current density
i_{MT}	(A cm^{-2}) Mass transfer current density
i_{x}	(A cm^{-1}) Crossover current density
$k_{\mathrm{amb}\to\mathrm{FC}}$	(W cm^{-3} K^{-1}) Heat transfer coefficient: ambient to stack
$k_{\mathrm{amb}\to\mathrm{H}}$	(W cm^{-3} K^{-1}) Heat transfer coefficient: ambient to heater
n_{p}	(-) Number of battery cells in parallel
n_{s}	(-) Number of battery cells in series
$n_{\mathrm{FC,cell}}$	(-) Number of fuel cells in stack
$r_{\mathrm{B,cell}}$	(Ω) Battery cell internal resistance
V_{FC}	(cm^3) Fuel cell volume
α	(-) Charge transfer coefficient
δ_{FC}	(cm) Channel or cell length
κ_T	(W cm^{-1} K^{-1}) thermal conductivity
ρ_{FC}	(g cm^{-3}) Fuel cell density

B. Preliminary Results of System Properties

This part gives some useful results that make abstraction computation efficient. The proofs can be found in [12].

Theorem 1: Assume $f : \mathbb{R}^n \to \mathbb{R}^m$ is differentiable, and

$$\frac{\partial f_i}{\partial x_j}(x) \in [a_{ij}, b_{ij}], \forall x \in X \subseteq \mathbb{R}^n, \tag{19}$$

where a_{ij} and b_{ij} are finite real numbers, set $X = \{x \in \mathbb{R}^n \mid x_j \in [\underline{x}_j, \overline{x}_j]\}$ is a rectangle, then the following inequality holds in element-wise sense:

$$\phi(\underline{x}, \overline{x}) \le f(x) \le \phi(\overline{x}, \underline{x}), \forall x \in X, \tag{20}$$

where $\underline{x} = [\underline{x}_1, \ldots, \underline{x}_n]^T$ and $\overline{x} = [\overline{x}_1, \ldots, \overline{x}_n]^T$, and function $\phi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^m$ is defined to be:

$$\begin{aligned} &\forall i \in 1 \ldots, m : \\ &\phi_i(x, y) = f_i(z) + (\alpha_i - \beta_i)^T (x - y). \end{aligned} \tag{21}$$

In Eq. (21), $z = [z_1, \ldots, z_n]^T$, $\alpha_i = [\alpha_{i1}, \ldots, \alpha_{in}]^T$, $\beta_i = [\beta_{i1}, \ldots, \beta_{in}]^T$ are n vectors defined as follows

$$z_j = \begin{cases} x_j & \text{if } b_{ij} \ge |a_{ij}| \\ y_j & \text{otherwise} \end{cases} \tag{22}$$

$$\alpha_{ij} = \begin{cases} |a_{ij}| + \epsilon & \text{if } a_{ij} \le 0, b_{ij} \ge |a_{ij}| \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

$$\beta_{ij} = \begin{cases} -|b_{ij}| - \epsilon & \text{if } b_{ij} \ge 0, a_{ij} \le -|b_{ij}| \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

where ϵ is a small positive number.

Remark 1: Theorem 1 is related to mixed monotonicity of function f, and the function ϕ is called a decomposition function of f. The decomposition function constructed above is a natural extension of the one given in [2], which only handles f with sign-stable partial derivatives. The idea here is to use linear terms to create additional offset to overcome the sign-unstable partial derivatives. In the case where all

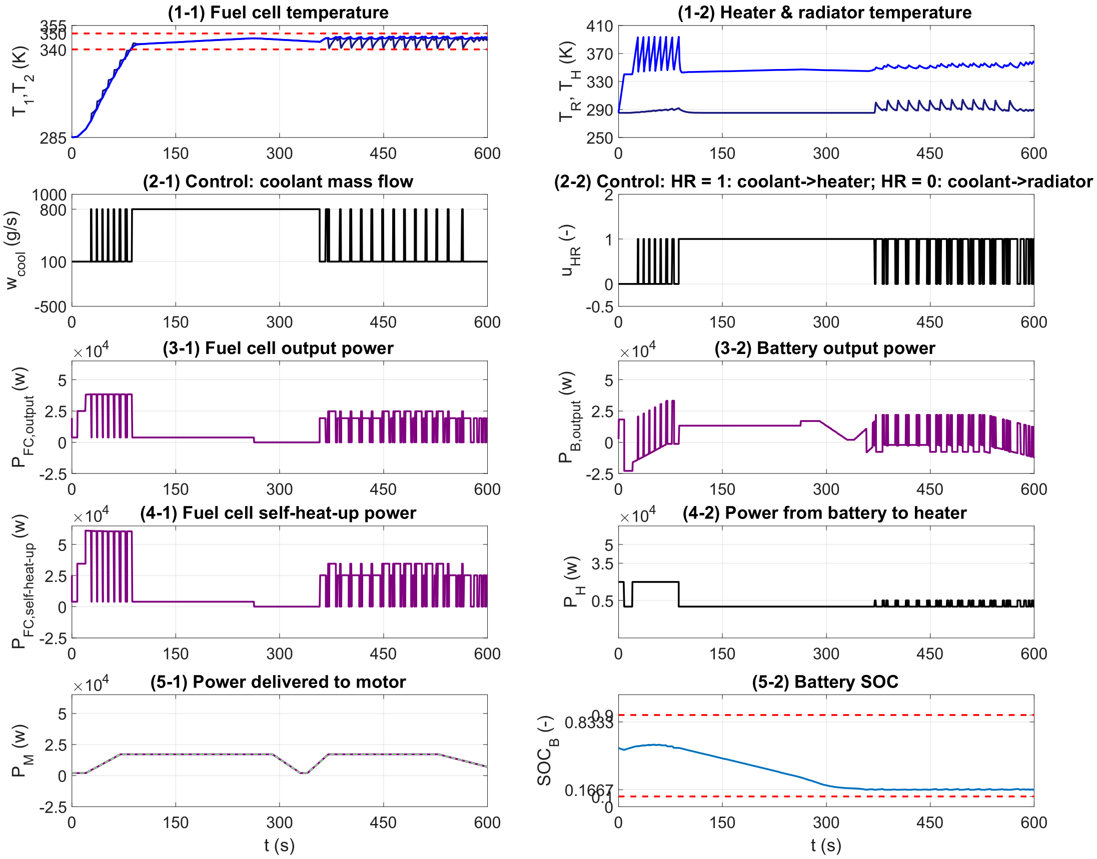


Fig. 8: Simulations results: states, powers and selected controls. Temperature states start from 285K and battery SOC starts from 0.58.

the partial derivatives $\frac{\partial f_i}{\partial x_j}$ are sign-stable, the decomposition function constructed by Theorem 1 gives a tight approximation in Eq. (20), that is, the inequality in Eq. (20) reduces to equality at some $x \in X$ [2]. However this is not true when there are sign-unstable partial derivatives. Thus in general the approximation given by (20) might be conservative.

Theorem 2: (Theorem 1 in [13]) Let $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ be a function affine in the first argument $x \in \mathbb{R}^n$ and multi-affine [1] in the second argument $d \in \mathbb{R}^m$, i.e., $f(x, d) = A(d)x+K(d)$, where $A(d)$, $K(d)$ are matrixes whose entries are in form of

$$\sum_{p_1,\ldots,p_m \in \{0,1\}} c_{p_1,\ldots,p_m} \prod_{j}^{m} (d_j)^{p_j}. \qquad (25)$$

Let $X \subseteq \mathbb{R}^n$ be a polytope, and $D \subseteq \mathbb{R}^m$ be a rectangle. Define V_X and V_D to be the set of vertices of set X and D. Then the maximum and minimum values of f on $X \times D$ are obtained at vertices set $V_X \times V_D$, i.e.,

$$\max_{\substack{x \in X \\ d \in D}} f(x, d) = \max_{\substack{x \in V_X \\ d \in V_D}} f(x, d), \qquad (26)$$

and similarly (26) holds for minimization of f.

References

[1] C. Belta and L. Habets. Controlling a class of nonlinear systems on rectangles. *IEEE Trans. Autom. Control*, 51(11):1749–1759, 2006.

[2] S. Coogan and M. Arcak. Efficient finite abstraction of mixed monotone systems. In *Proc. of HSCC*, pages 58–67. ACM, 2015.

[3] N. Henao, S. Kelouwani, K. Agbossou, and Y. Dubé. Proton exchange membrane fuel cells cold startup global strategy for fuel cell plug-in hybrid electric vehicle. *Journal of Power Sources*, 220:31–41, 2012.

[4] K. Jiao and X. Li. Water transport in polymer electrolyte membrane fuel cells. *Progress in Energy and Combustion Science*, 37(3):221–291, 2011.

[5] S. G. Kandlikar, Z. Lu, and T. A. Trabold. Current status and fundamental research needs in thermal management within a pemfc stack. *in ASME Journal of Fuel Cells Science and Technology*, 2008.

[6] S. J. Moura, D. S. Callaway, H. K. Fathy, and J. L. Stein. Impact of battery sizing on stochastic optimal power management in plug-in hybrid electric vehicles. In *Proc. of IEEE ICVES*, pages 96–102. IEEE, 2008.

[7] P. Nilsson and N. Ozay. Incremental synthesis of switching protocols via abstraction refinement. In *Proc. of IEEE CDC*, pages 6246–6253, 2014.

[8] N. Ozay, J. Liu, P. Prabhakar, and R. Murray. Computing augmented finite transition systems to synthesize switching protocols for polynomial switched systems. In *Proc. of ACC*, pages 6237–6244, 2013.

[9] B. L. Pence and J. Chen. A framework for control oriented modeling of pem fuel cells. In *ASME 2015 Dynamic Systems and Control Conference*, pages V002T26A002–V002T26A002. American Society of Mechanical Engineers, 2015.

[10] J. T. Pukrushpan, A. G. Stefanopoulou, and H. Peng. *Control of fuel cell power systems: principles, modeling, analysis and feedback design*. Springer Science & Business Media, 2004.

[11] P. Tabuada. *Verification and control of hybrid systems: a symbolic approach*. Springer, 2009.

[12] L. Yang and N. Ozay. A note on some sufficient conditions for mixed monotone systems. Technical report, University of Michigan, Department of EECS, 2017. Available at `http://hdl.handle.net/2027.42/136122`.

[13] L. Yang, N. Ozay, and A. Karnik. Synthesis of fault tolerant switching protocols for vehicle engine thermal management. In *Proc. of ACC*, pages 4213–4220, 2016.