2016 American Control Conference (ACC)
Boston Marriott Copley Place
July 6-8, 2016. Boston, MA, USA

Synthesis of Fault Tolerant Switching Protocols for Vehicle Engine Thermal Management

Liren Yang, Necmiye Ozay, and Amey Karnik

***Abstract*— Thermal management is very important to guarantee ideal performance of compact vehicle engines. One challenge in the vehicle engine thermal management is to control the engine temperature in a small interval while tolerating component failures and the uncertainties in complex environment and different operating conditions. We formulate this control problem as a temporal logic game for a switched affine system and solve it by synthesizing a switching protocol based on an abstraction. The existing algorithms for computing abstractions either cannot handle parametric uncertainties in the dynamics or can be computationally expensive. Besides, they usually do not deal with possible component failures. The main contribution of this work is to show: (i) how to compute an abstraction more efficiently under the assumption that the vector fields are multiaffine in constant uncertainties and affine in state variables, (ii) how to result in a graceful degradation in case of component failures.**

I. INTRODUCTION

Thermal management is crucial to guarantee reliable performance of compact automotive engines. The benefits include improvements in fuel efficiency, reductions in tailpipe emissions and component failures due to excessive heat[13]. In conventional passive engine cooling systems, the coolant pump is linked to the crankshaft and the coolant flow rate is controlled by a wax element thermostat [13]. Recently it has been shown that the performance of the engine can be improved by introducing more actuators like a radiator fan [18], or an electric coolant pump independent of the crankshaft [4]. Introduction of new actuators has necessitated the development of advanced active control strategies for thermal management.

Many different control methods have been proposed for engine thermal management including PID control [17] and model predictive control [16], just to mention a few. These methods primarily focus on continuous actuators and nominal system operation. Continuous actuators are more expensive, and usually discrete-valued (0-1) actuators are found in automotive applications. This requires the synthesis of switching control strategies (i.e., switching protocols). Moreover, these strategies need to be robust to uncertainties arising out of component variabilities or environment factors. And finally, component failures are common and they must be mitigated properly. Motivated by these challenges, we propose algorithms to synthesize correct-by-construction switching protocols that are (i) robust against parameter uncertainty, (ii) fault tolerant in the sense that they result in graceful degradation in case of component failures. We build on recent results on abstraction-based switching protocol synthesis [8], [12], [11], and in particular, extend them to incorporate these two important aspects.

Abstraction-based correct-by-construction synthesis techniques have attracted considerable attention in recent years, with applications in robotic motion planning [6], autonomous driving and cruise control [19], [3], [10], control of aircraft subsystems [9], and building thermal management [5], [11]. By correct-by-construction, we mean controllers that are guaranteed to satisfy a given specification, typically expressed using temporal logics, under explicitly spelled out assumptions on the system model and the environment the system operates in [15].

Our main contributions are to present (i) a framework for modeling an uncertain system with failure modes and specifying its desired behavior with possibly different requirements for each degraded mode of operation, and (ii) algorithms for synthesizing robust switching protocols for systems specified in this framework to guarantee the satisfaction of the "most stringent" requirements associated with the current health status of the system at run-time. The engine thermal management problem is specified in the proposed framework and the closed-loop behavior with the synthesized switching protocols is illustrated with simulations.

II. NOTATION AND PRELIMINARIES

Let $\mathbb{R}^n$ be the n dimensional Euclidean space. A half space is a subset of $\mathbb{R}^n$ defined as $\{\boldsymbol{x} \in \mathbb{R}^n \mid \alpha^T \boldsymbol{x} \leq \beta, \alpha \in \mathbb{R}^n, \beta \in \mathbb{R}\}$. A polyhedron is the intersection of finitely many half spaces, and a polytope is a bounded polyhedron. Hyper rectangles are special types of polytopes, which can be defined as $\{\boldsymbol{x} \in \mathbb{R}^n \mid x_i \in [a_i, b_i], \forall i = 1, \ldots, n\}$ where x_i is the i^{th} component of vector $\boldsymbol{x}$. We say two polytopes P_1 P_2 are adjacent if their intersection is a nonempty set. In particular, if $P_1 \cap P_2$ is an $n-1$ dimensional polytope, we define $F_{P_1,P_2} := P_1 \cap P_2$ to be the adjacent facet of polytopes P_1 and P_2. The normal vector of the adjacent facet F_{P_1,P_2}, denoted as $\boldsymbol{n}_{P_1,P_2}$, is a unit vector such that for all $\boldsymbol{x} \in F_{P_1,P_2} : \boldsymbol{n}_{P_1,P_2}^T \boldsymbol{x} = 0$, and by convention $\boldsymbol{n}_{P_1,P_2}^T$ points from P_1 to P_2 (i.e., for all $\boldsymbol{x} \in P_1, \boldsymbol{x}_F \in F_{P_1,P_2} : \boldsymbol{n}_{P_1,P_2}^T (\boldsymbol{x} - \boldsymbol{x}_F) \leq 0$). A convex combination of $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}$ is $\sum_i^m \theta_i \boldsymbol{x}_i$ with $\theta_i \geq 0$ and $\sum_i^m \theta_i = 1$. The convex hull of a finite set $X \subseteq \mathbb{R}^n$, denoted as $Conv(X)$, is the set of all possible convex combination of X. A polytope P can always be written as a convex hull of a finite set, the smallest such set is defined to be the vertices of polytope P, denoted as V_P .

LY and NO are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA yliren,necmiye@umich.edu. AK is with the Ford Research Center, Dearborn, MI 48121, USA akarnik@ford.com. This work is supported in part by Ford Motor Co. and NSF grant CNS-1446298.

A. Multiaffine Functions

Definition 1: A function $f : X \subseteq \mathbb{R}^n \to \mathbb{R}^m$ is said to be *multiaffine* in $\boldsymbol{x} = [x_1, \ldots, x_n]^T \in X$, if for all $j \in \{1, \ldots, m\}$, f_j, the j^{th} component of f, is in the form of

$$f_j(\boldsymbol{x}) = \sum_{p_j^1, \ldots, p_j^n \in \{0,1\}} c_{p_j^1, \ldots, p_j^n} \prod_i^n (x_i)^{p_j^i}, \tag{1}$$

where $c_{p_j^1, \ldots, p_j^n} \in \mathbb{R}$ are some constants.

Lemma 1: (Proposition 2 in [2]) Given a multiaffine function $f : X \to \mathbb{R}^m$, where $X \subseteq \mathbb{R}^n$ is a hyper rectangle, the value of f at arbitrary point in X can be written as a convex combination of its values at the vertices of X, i.e.,

$$\forall \boldsymbol{x} \in X : f(\boldsymbol{x}) = \sum_{\boldsymbol{x}_{\text{v}}^i \in V_X} \theta_i f(\boldsymbol{x}_{\text{v}}^i), \tag{2}$$

where $\theta_i \geq 0$, $\sum_i^{|V_X|} \theta_i = 1$.

Lemma 2: Given a multiaffine function $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$ defined on a hyper rectangle X, f attains its maximum and minimum at V_X.

Lemma 3: Given a function $h : X \subseteq \mathbb{R}^n \to \mathbb{R}$ defined on hyper rectangle X, if h is in the form of f/g, where f and g are multiaffine in $\boldsymbol{x}$ and g is non-zero in X, then h attains its maximum and minimum at V_X.

B. Linear Temporal Logic

We use a fragment linear temporal logic (LTL) for specifying the correct behaviors of a system. Given a set AP of atomic propositions, the fragment used is defined by the following grammar: $\varphi ::= \pi \mid \neg\varphi \mid \varphi \vee \varphi \mid \Box\varphi$, for $\pi \in AP$. We write $\varphi \wedge \psi$ and $\varphi \to \psi$ as abbreviations for the formulas $\neg(\neg\varphi \vee \neg\psi)$ and $\neg\varphi \vee \psi$, respectively. We also use the short hand notation $\Diamond\varphi$, for $\neg\Box\neg\varphi$. We refer the reader to [12] for continuous semantics of LTL but just informally introduce the meaning of formulas that are used subsequently. For a propositional formula ϕ and a signal $\sigma : [0, \infty) \to 2^{AP}$, σ satisfies $\Box\phi$ if $\sigma(t)$ satisfies ϕ for all t; σ satisfies $\Diamond\phi$ if there exists a t^* such that $\sigma(t^*)$ satisfies ϕ; and σ satisfies $\Diamond\Box\phi$ if there exists a t^* such that $\sigma(t)$ satisfies ϕ for all $t \geq t^*$.

C. System with Failure Modes

We represent component failures with a finite set $AP^{\text{F}} = \{\pi_1, \ldots, \pi_N\}$, whose element π_i is an atomic proposition indicating that component i has failed. A *fault configuration* F is a subset of AP^{F}. A fault configuration $F = \emptyset$ corresponds to the case when no failure happens and the system is called healthy when in this configuration. Let $\mathcal{F} = \{F_1, \ldots, F_M\}$ be a nonempty collection of such fault configurations. $\mathcal{F}$ is a partially ordered set under set inclusion, i.e.,

$$\forall F_i, F_j \in \mathcal{F} : F_i \preceq (\prec) F_j \text{ if } F_i \subseteq (\subsetneq) F_j. \tag{3}$$

Under this partial order, for all $\mathcal{E} \subseteq \mathcal{F}$, let minimal and maximal elements of $\mathcal{E}$ be $\min(\mathcal{E}) = \{F \in \mathcal{E} \mid \forall F_i \in \mathcal{E} : F_i \not\prec F\}$, $\max(\mathcal{E}) = \{F \in \mathcal{E} \mid \forall F_i \in \mathcal{E} : F \not\prec F_i\}$. For a fault configuration $F_j \in \mathcal{F}$, the set of its *strict successors* is defined as $Succ(F_j) = \min(\{F \in \mathcal{F} \mid F_j \prec F\})$.

For each fault configuration, the system dynamics are governed by a (potentially different) continuous-time system. In particular, in this paper we consider the following systems.

Definition 2: A *continuous-time switched system with parameter uncertainties* is a tuple $\Sigma = (X, \mathcal{U}, Q, \{f_{\boldsymbol{u}}\}_{\boldsymbol{u} \in \mathcal{U}})$, where $X \in \mathbb{R}^n$ is the domain, $\mathcal{U} \in \mathbb{R}^p$ is a finite set of control inputs, $Q \in \mathbb{R}^m$ is an uncertainty set, $f_{\boldsymbol{u}} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the vector field with parameter uncertainties under control $\boldsymbol{u} \in \mathcal{U}$. Let $\boldsymbol{x} \in X$ denote the state and $\boldsymbol{q} \in Q$ denote the uncertainties, the dynamics of the system under control $\boldsymbol{u} \in \mathcal{U}$ is defined by $\dot{\boldsymbol{x}} = f_{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{q})$.

In what follows continuous-time switched systems with parameter uncertainties are called switched system for short.

Definition 3: Given a finite collection $\mathcal{F} = \{F_i\}_{i=1}^M$ of fault configurations $F_i \subseteq AP^{\text{F}}$ with a unique minimum $F_{i^*} = \min(\mathcal{F})$, and the corresponding collection of switched systems $\{\Sigma_i\}_{i=1}^M$, a *system with failure modes* is defined as a tuple $\Sigma^{\mathcal{F}} = (\mathcal{X}, Init, \to_{\Sigma_{\mathcal{F}}}, AP^{\text{F}}, h_{\mathcal{X}})$, where $\mathcal{X} = \{\Sigma_1, \ldots, \Sigma_M\}$ is the domain, $Init = \Sigma_{i^*}$ is the initial state, AP^{F} is the set of atomic propositions, $h_{\mathcal{X}} : \mathcal{X} \to 2^{AP^{\text{F}}}$, $h_{\mathcal{X}} : \Sigma_i \mapsto F_i$ for $i = 1, \ldots M$ is the observation map, and the transition relation $\to_{\Sigma_{\mathcal{F}}} \subseteq \mathcal{S} \times \mathcal{S}$ is defined as:

$$\forall \Sigma_i, \Sigma_j \in \mathcal{X}, (\Sigma_i, \Sigma_j) \in \to_{\Sigma_{\mathcal{F}}} \text{ iff } h_{\mathcal{X}}(\Sigma_j) \in Succ(h_{\mathcal{X}}(\Sigma_i)). \tag{4}$$

A system with failure modes as defined above is a type of hierarchical hybrid system [7] since each state in $\mathcal{X}$ is itself a hybrid (switched) system. The transition relation as defined in (4) captures the assumption that the failures are permanent, that is, if a component fails, it does not recover.

D. Abstraction Refinement

Augmented finite transition systems (AFTS) will be used to abstract the switched system dynamics. The definitions in this section are adapted from [12], [11].

Definition 4: An *augmented finite transition system* is a tuple $\mathcal{T} = (S, \mathcal{U}, \to_{\mathcal{T}}, \mathcal{G})$ where S is the finite set of states, $\mathcal{U}$ is the finite set of inputs, $\to_{\mathcal{T}} \subseteq S \times \mathcal{U} \times S$ is a transition relation, and $\mathcal{G} : \mathcal{U} \to 2^{2^S}$ is a progress group map.

To compare the behaviors of an AFTS $\mathcal{T}$ and a switched system Σ over a set AP^{S} of atomic propositions, $\mathcal{T}$ and Σ are decorated with observation maps, $h_S : S \to 2^{AP^{\text{S}}}$ and $h_X : X \to 2^{AP^{\text{S}}}$, respectively.

Definition 5: An augmented finite transition system $\mathcal{T} = (S, \mathcal{U}, \to_{\mathcal{T}}, \mathcal{G}, AP^{\text{S}}, h_S)$ is said to be an *over-approximation* for the switched system $\Sigma = (X, \mathcal{U}, Q, \{f_{\boldsymbol{u}}\}_{\boldsymbol{u} \in \mathcal{U}}, AP^{\text{S}}, h_X)$, denoted by $\mathcal{T} \succeq \Sigma$, if there exists a function $\alpha : X \to S$ such that the following statements hold.

1) For all $\xi \in X$, $h_X(\xi) = h_S(\alpha(\xi))$.
2) Given states $s, s' \in S$, there is a transition $(s, u, s') \in \to_{\mathcal{T}}$, if there exist $\xi_0 \in \alpha^{-1}(s)$, $\tau > 0$, and some parameter $q \in Q$ such that the corresponding trajectory x of the subsystem $f_{\boldsymbol{u}}$ starting from ξ_0, i.e., $\boldsymbol{x} : [0, \tau] \to \mathbb{R}^n$ with $\boldsymbol{x}(0) = \xi_0$, $\dot{\boldsymbol{x}}(t) = f_{\boldsymbol{u}}(\boldsymbol{x}(t), \boldsymbol{q})$ for all $t \in (0, \tau)$, satisfies $\boldsymbol{x}(\tau) \in \alpha^{-1}(s')$, $\boldsymbol{x}(t) \in \alpha^{-1}(s) \cup \alpha^{-1}(s')$, $t \in [0, \tau]$.

3) The progress group map $\mathcal{G}$ is such that given an action $\boldsymbol{u} \in \mathcal{U}$, for all $G \in \mathcal{G}(\boldsymbol{u})$, the set $\bigcup_{s\in G} \alpha^{-1}(s)$ is transient[1] on mode $\boldsymbol{u}$ of Σ.

A proposition preserving partition of the domain X induces a function α where each $s \in S$ corresponds to a cell in the partition (i.e., if item 1 is satisfied). Transitions of an over-approximation $\mathcal{T}$ capture how the trajectories of Σ behave across cells, and progress groups capture transient cells and eliminate the effect of spurious cycles the transitions of $\mathcal{T}$ might form. It is shown in [12] that if one can find a switching protocol for an over approximation $\mathcal{T}$ to satisfy an LTL property, a switching protocol for the underlying switched system Σ that guarantees the satisfaction of the same property exists; and there exist ways to search for switching protocols for AFTSs. If a given over-approximation $\mathcal{T}$ is not enough for finding a switching protocol, it can be *refined* to form a less conservative over approximation $\hat{\mathcal{T}}$. A formal definition of the refinement relation, denoted as $\mathcal{T} \succeq \hat{\mathcal{T}}$, can be found in [12], [11].

III. PROBLEM DESCRIPTION

A. Engine Thermal Model

This section describes a simplified model of the dynamics of an engine thermal management system, shown in Fig. 1. The heat is generated by combustion of fuel. Part of the heat generated is transferred to the engine combustion chamber walls. The engine block rejects the heat to the coolant which increases the coolant temperature. Coolant rejects heat to ambient at the radiator. These heat exchange processes are affected by temperature of ambient air, actual vehicle speed and the coolant pump flow rate. Our goal is to maintain the engine temperature in a proper interval with limited control on coolant flow valve position and the radiator grill shutter opening.

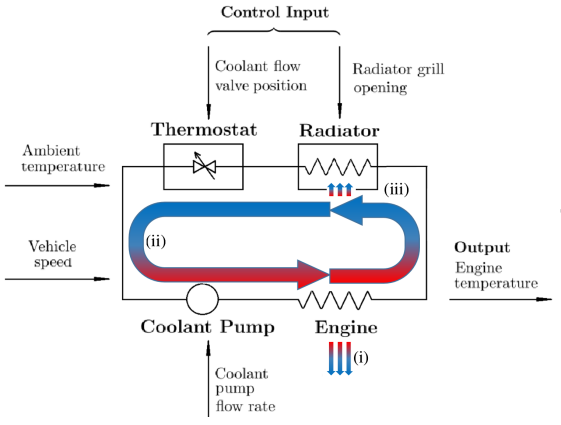


Fig. 1: Schematic of thermal dynamics of an engine, showing the heat exchange between (i) engine and ambient air, (ii) engine and coolant, (iii) coolant and radiator

Let $\boldsymbol{x} = [T_e, T_r]^T \in \mathbb{R}^2$ be state variables, where T_e denotes the engine temperature and T_r denotes the radiator temperature. The temperature dynamics are modeled as follows:

$$\begin{cases} \dot{T}_{\mathrm{e}} &= c_{\mathrm{ea}}(T_{\mathrm{e}} - T_{\mathrm{a}}) + c_{\mathrm{er}}(v,w)(T_{\mathrm{e}} - T_{\mathrm{r}}) + c_{\mathrm{e}}(h) \\ \dot{T}_{\mathrm{r}} &= c_{\mathrm{ra}}(s,g)(T_{\mathrm{r}} - T_{\mathrm{a}}) + c_{\mathrm{re}}(v,w)(T_{\mathrm{r}} - T_{\mathrm{e}}), \end{cases} \tag{5}$$

where the coefficients c_{ea}, c_{er}, $c_{\mathrm{ra}}(v,w)$, $c_{\mathrm{re}}(s,g)$ depend on engine heat h, vehicle speed s, coolant pump flow rate w and ambient temperature T_{a}, coolant flow valve position v and radiator grill shutter opening g. We assume h, s, w and T_{a} are external inputs that can be measured, and g and v are control inputs. The ranges of these variables are given in Table I. To be specific, the coefficients are defined as

$$\begin{gathered} c_{\mathrm{ea}} = -U_{\mathrm{ea}}/C_{\mathrm{e}}, \quad c_{\mathrm{er}}(v,w) = -C_{\mathrm{c}} v w / C_{\mathrm{e}}, \\ c_{\mathrm{e}}(h) = h/C_{\mathrm{e}}, \quad c_{\mathrm{re}}(v,w) = -C_{\mathrm{c}} v w / C_{\mathrm{r}}, \\ c_{\mathrm{ra}}(s,g) = (-U_{\mathrm{ra}} - C_{\mathrm{a}} A_{\mathrm{a}} D_{\mathrm{a}} s g)/C_{\mathrm{r}}, \end{gathered} \tag{6}$$

where the constants $U_{\mathrm{ea}}, U_{\mathrm{ra}}, C_{\mathrm{e}}, C_{\mathrm{r}}, C_{\mathrm{a}}, C_{\mathrm{c}}, A_{\mathrm{a}}, D_{\mathrm{a}}$ are heat exchange speed from engine block to ambient air, heat exchange speed from radiator to ambient air, heat capacity of engine, heat capacity of radiator, specific heat of air, specific heat of coolant, radiator frontal area, and density of air, respectively.[2] The functions defining coefficients in (6) are multiaffine in h, s, and w. Note that the actual range of external inputs h, s, w and T_{a} in reality can be larger than the ones presented in Table I. But since these inputs can be measured, we can synthesize different robust controllers for different ranges of external inputs and switch among these robust controllers according to the measurements.

TABLE I: Measured Inputs & Control Inputs

Symbol	Physical Meaning	Unit	Range Used
h	Heat from engine combustion	W	[15000, 19000]
s	Vehicle speed	m/s	[10,20]
w	Coolant pump flow rate	kg/s	[0.03,0.045]
T_{a}	Ambient temperature	K	[282, 288]
v	Flow valve position	-	{0.25,1}
g	Radiator grill shutter opening	-	{0.25,1}

The operating region, i.e., the domain D is given by:

$$\begin{aligned} D := \{\boldsymbol{x} = [T_{\mathrm{e}}, T_{\mathrm{r}}]^T \in \mathbb{R}^2 \mid 260 \le T_{\mathrm{e}} \le 500, \\ 200 \le T_{\mathrm{r}} \le 400\}. \end{aligned} \tag{7}$$

Let the nominal uncertain parameter set be:

$$\begin{aligned} Q := \{\boldsymbol{q} = [h, s, w, T_{\mathrm{a}}]^T \in \mathbb{R}^4 \mid h \in [1.5, 1.9] \times 10^4, \\ s \in [10, 20], w \in [0.03, 0.045], T_{\mathrm{a}} \in [282, 288]\}, \end{aligned} \tag{8}$$

and nominal set of inputs be

$$\mathcal{U} := \{\boldsymbol{u} = [v, g]^T \in \mathbb{R}^2 \mid v \in \{0.25, 1\} \text{ and } g \in \{0.25, 1\}\}. \tag{9}$$

We consider three fault configurations corresponding to healthy operation (F_1), radiator grill shutter stuck (F_2), and flow valve stuck (F_3). A switched system $\Sigma_i = (D, \mathcal{U}_i, Q_i, \{f_{\boldsymbol{u}_i}(x, q_i)\}_{\boldsymbol{u}_i \in \mathcal{U}_i})$ for $i \in \{1, 2, 3\}$ can be constructed to represent the dynamics in each configuration.

[1] A set $X_o \subseteq X$ is called transient on mode $\boldsymbol{u}$ of Σ, if for all $\xi \in X_o$ and for all $\boldsymbol{q} \in Q$, the trajectory that starts in ξ, eventually leaves X_o with the flow of $f_{\boldsymbol{u}}(x, q)$.

[2] We take C_{e}=750 J/K, C_{r}=200 J/K, C_{a}=1005 J/kg/K, C_{c}=3400 J/kg/K, U_{ea}=100 W/K, U_{ra}=100 W/K, A_{a}=0.2m^2, and D_{a}=1 kg/m^3.

For healthy system dynamics Σ_1, we have $\mathcal{U}_1 := \mathcal{U}$, and $Q_1 := Q$; for the dynamics Σ_2 when the radiator grill shutter is stuck, we have $\mathcal{U}_2 := \{\boldsymbol{u} = v \in \mathbb{R} \mid v \in \{0.25, 1\}\}$ and $Q_2 := Q \times [0.25, 1]$, where $g \in [0.25, 1]$ is added to the uncertain parameters; and for dynamics Σ_3 when the flow valve is stuck in the middle, we have $\mathcal{U}_3 := \{\boldsymbol{u} = g \in \mathbb{R} \mid g \in \{0.25, 1\}\}$ and $Q_3 := Q \times [0.62, 0.63]$, where $v \in [0.62, 0.63]$ is added to the uncertain parameters. The vector fields $f_{\boldsymbol{u}_i}(\boldsymbol{x}, \boldsymbol{q}_i)$ are defined according to (5) with the corresponding input and uncertainty sets for each $i \in \{1, 2, 3\}$ so that $\dot{\boldsymbol{x}} = f_{\boldsymbol{u}_i}(\boldsymbol{x}, \boldsymbol{q}_i)$ for fixed values of $\boldsymbol{u}_i \in \mathcal{U}_i$ and $\boldsymbol{q}_i \in Q_i$.

Letting π^g_{fail} be an atomic proposition for radiator grill shutter being stuck and π^v_{fail} be an atomic proposition for the flow valve being stuck, we get $AP^{\text{F}} = \{\pi^g_{\text{fail}}, \pi^v_{\text{fail}}\}$ with $F_1 = \emptyset$, $F_2 = \{\pi^g_{\text{fail}}\}$ and $F_3 = \{\pi^v_{\text{fail}}\}$, and $\mathcal{F} = \{F_1, F_2, F_3\}$. Putting everything together, we obtain a representation of the engine thermal dynamics with different fault configurations as a system with failure modes:

$$\Sigma^{\mathcal{F}} = (\mathcal{X}, Init, \rightarrow_{\Sigma_{\mathcal{F}}}, AP^{\text{F}}, h_{\mathcal{X}}) \tag{10}$$

where $\mathcal{X} = \{\Sigma_1, \Sigma_2, \Sigma_3\}$, $Init = \Sigma_1$, $h_{\mathcal{X}} : \Sigma_i \mapsto F_i$ and $\rightarrow_{\Sigma_{\mathcal{F}}}$ is defined by (4).

B. Specifications

In this section we specify the desired behavior of the system $\Sigma^{\mathcal{F}}$ in (10) using LTL. The main specification in thermal management is to avoid high temperatures that can lead to cracking of surfaces and to steer the temperature and maintain it in a range where the engine operates efficiently. Typically as the number of failed components increase, the requirements on the system get more relaxed. In order to capture this graceful degradation in performance, we specify the required behavior for each fault configuration F_i separately.

For each fault configuration F_i, we want the engine temperature T_{e} to reach and stay in a goal set G_i, while avoiding an unsafe set U_i and remaining in the domain D. For the goal sets, we have:

$$G_1 := \{\boldsymbol{x} = [T_{\text{e}}, T_{\text{r}}]^T \in D \mid 385 \leq T_{\text{e}} \leq 395\}, \tag{11}$$

$$G_2 := \{\boldsymbol{x} = [T_{\text{e}}, T_{\text{r}}]^T \in D \mid 390 \leq T_{\text{e}} \leq 400\}, \tag{12}$$

$$G_3 := \{\boldsymbol{x} = [T_{\text{e}}, T_{\text{r}}]^T \in D \mid 360 \leq T_{\text{e}} \leq 410\}, \tag{13}$$

and, for the unsafe sets, we have

$$U_1 = U_2 := \{\boldsymbol{x} = [T_{\text{e}}, T_{\text{r}}]^T \in D \mid T_{\text{e}} \geq 400\}.$$
$$U_3 := \{\boldsymbol{x} = [T_{\text{e}}, T_{\text{r}}]^T \in D \mid T_{\text{e}} \geq 410\} \tag{14}$$

Let $\rho^i_{\text{goal}} : \boldsymbol{x} \in G_i$ and $\rho^i_{\text{safe}} : \boldsymbol{x} \in D \setminus U_i$, then, the desired behavior for $i = 1, 2, 3$ is given by:

$$\Phi_i = \Diamond\Box\rho^i_{\text{goal}} \wedge \Box\rho^i_{\text{safe}}, \tag{15}$$

denoted by $\Phi(D, G_i, U_i)$ in parametrized form when necessary. Since the specifications in (15) are conditioned on the fault configuration, the desired behavior of the overall system $\Sigma^{\mathcal{F}}$ becomes:

$$\Psi = \bigwedge_{F_i \in \mathcal{F}} (\Diamond\Box F_i \rightarrow \Phi_i), \tag{16}$$

where, with slight abuse of notion we use F_i to mean $\bigwedge_{a_j \in F_i} a_j$.

C. Problem statement

We now state the engine thermal management problem.

Problem 1: Given engine thermal system model $\Sigma^{\mathcal{F}}$ defined in (10), and specification Ψ defined in (16), find a set of initial states $\mathcal{I} \in D$ (failure tolerant winning set) and a controller (failure tolerant switching protocol) $\mathcal{K} : \mathcal{I} \times \mathcal{F} \rightarrow \bigcup 2^{\mathcal{U}_i}$, with $(\boldsymbol{x}, F_i) \mapsto \boldsymbol{u}_i \in \mathcal{U}_i$, such that while using $\mathcal{K}$, all the closed loop trajectories starting from $\mathcal{I}$ satisfy Ψ.

We propose an approach for the problem that seeks a set $\mathcal{I}$ of initial states that is as large as possible but in general there is no guarantee that it will be maximal.

IV. SOLUTION APPROACH

Problem 1 can be seen as a switching protocol synthesis problem for a polynomial system with mode-target objectives. Therefore, it can essentially be solved by combining techniques from abstractions of polynomial systems [12] and mode-target games [1]. However, it also admits some structure that can be utilized to develop more efficient solutions. In this section we propose a solution approach that utilizes this structure and that is applicable for a class of systems with failure modes as long as the following assumptions hold.

Assumption 1: Let $\Sigma_i = (D, \mathcal{U}_i, Q_i, \{f_{\boldsymbol{u}_i}\}_{\boldsymbol{u}_i \in \mathcal{U}_i})$ be a switched system that corresponds to the i^{th} fault configuration of a system with failure modes. Then, for all control $\boldsymbol{u}_i \in \mathcal{U}_i$, $f_{\boldsymbol{u}_i}(\boldsymbol{x}, \boldsymbol{q}_i) = A_{\boldsymbol{u}_i}(\boldsymbol{q}_i)x + K_{\boldsymbol{u}_i}(\boldsymbol{q}_i)$ where $A_{\boldsymbol{u}_i} : \mathbb{R}^m \rightarrow \mathbb{R}^{n \times n}$ and $K_{\boldsymbol{u}_i} : \mathbb{R}^m \rightarrow \mathbb{R}^n$ are multiaffine functions of $\boldsymbol{q}_i$. Furthermore, for all $\boldsymbol{q}_i \in Q_i$ and all $\boldsymbol{u}_i \in \mathcal{U}_i$, $A_{\boldsymbol{u}_i}(\boldsymbol{q}_i)$ is full rank and has no pure imaginary eigenvalues.

Assumption 2: Under all fault configurations $F_i \in \mathcal{F}$, uncertain parameters $\boldsymbol{q}_i$ are constant (but unknown) and the allowable parameter set $Q_i \subseteq \mathbb{R}^m$ is a hyper rectangle.

Assumption 3: The failures are permanent, that is, if a component fails, it never recovers later.

For the engine thermal system with failure modes defined by (10), the system Σ_i satisfies Assumption 1 in each fault configuration F_i. One can show this by substituting (6) into (5) and checking $A_{\boldsymbol{u}_i}(\boldsymbol{q}_i)$, $K_{\boldsymbol{u}_i}(\boldsymbol{q}_i)$ satisfies Definition 1 for $i = 1, 2, 3$ and all $\boldsymbol{u}_i \in \mathcal{U}_i$. All Q_is are by definition hyper rectangles in the engine model and the parameters $\boldsymbol{q}_i$ are close to being constant, therefore Assumption 2 is also reasonable. Finally, Assumption 3 holds by definition of the transitions of the system with failure modes in (4) and again reasonable for the engine model.

Our approach to solve Problem 1 consists of two steps. First, we show that by Assumption 3, the solution of Problem 1 can be reduced to solving a set of simpler subproblems (of the form Problem 2 below) recursively one for each fault configuration. Then, we present an abstraction based

approach to solve these subproblems. We first state this simpler subproblem.

Problem 2: Given a continuous-time switched system with parameter uncertainties $\Sigma = (D, \mathcal{U}, Q, \{f_{\boldsymbol{u}}\}_{\boldsymbol{u}\in\mathcal{U}})$, a polytopic goal set $G \subseteq D$ and an unsafe set U, assume Assumption 1 is true for Σ, and Assumption 2 is true for q and Q. Find a set of initial states (wining set) $I \subseteq D$ and a controller (robust switching protocol) $K : I \to 2^{\mathcal{U}}$ such that under K, all trajectories starting from I satisfy $\Phi(D, G, U)$ defined as in (15).

A. Solution of Problem 1

In this section, we assume a solution approach for Problem 2 exists and we propose a solution for Problem 1 based on it. Let $\mathbf{Prob}^2(\Phi, \Sigma)$ denote an instance of Problem 2 defined by specification $\Phi(D, G, U)$ and system $\Sigma = (D, \mathcal{U}, Q, \{f_{\boldsymbol{u}}\}_{\boldsymbol{u}\in\mathcal{U}})$ and let $\mathrm{Win}(\cdot)$ denote the function solving Problem 2. Also, let $\mathbf{Prob}^1(\Psi, \Sigma^{\mathcal{F}}, F_j)$ denote an instance of Problem 1 where $\Sigma^{\mathcal{F}}$ starts operating from the initial configuration F_j. With this notation, Problem 1 becomes $\mathbf{Prob}^1(\Psi, \Sigma^{\mathcal{F}}, F_1)$.

Algorithm 1 $[\mathcal{I}_j, \mathcal{K}_j] = \mathrm{Win}^{\mathrm{F}}(\mathbf{Prob}^1(\Psi, \Sigma^{\mathcal{F}}, F_j))$
Compute the failure tolerant winning set $\mathcal{I}_j$ and controller $\mathcal{K}_j$ for $\mathbf{Prob}^1(\Psi, \Sigma^{\mathcal{F}}, F_j)$.

Input: $\mathbf{Prob}^1(\Psi, \Sigma^{\mathcal{F}}, F_j)$.
Output: Failure tolerant winning set $\mathcal{I}_j$ and controller $\mathcal{K}_j$ so that $\Sigma^{\mathcal{F}}$ satisfies Ψ when starting from F_j.

Initialize $\mathcal{I}_j = \emptyset$, $\mathcal{K}_j = \emptyset$
if $F_j \in \max(\mathcal{F})$ **then**
  $[I_j, K_j] = \mathrm{Win}(\mathbf{Prob}^2(\Phi_j, \Sigma_j))$
  $\mathcal{I}_j = I_j$
  $\mathcal{K}_j = \{(\boldsymbol{x}, F_j) \mapsto K_j(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathcal{I}_j\}$
else
  $\overline{U_j} = U_j$
  for $F_i \in \mathrm{Succ}(F_j)$ **do**
    $[\mathcal{I}_i, \mathcal{K}_i] = \mathrm{Win}^{\mathrm{F}}(\mathbf{Prob}^1(\Psi, \Sigma^{\mathcal{F}}, F_i))$
    $\overline{U_j} = \overline{U_j} \cup (D \setminus \mathcal{I}_i)$
    $\mathcal{K}_j = \mathcal{K}_j \cup \mathcal{K}_i$
  $\overline{\Phi_j} = \Phi(D, G_j, \overline{U_j})$
  $[I_j, K_j] = \mathrm{Win}(\mathbf{Prob}^2(\overline{\Phi_j}, \Sigma_j))$
  $\mathcal{K}_j = \mathcal{K}_j \cup \{(\boldsymbol{x}, F_j) \mapsto K_j(\boldsymbol{x}), \forall \boldsymbol{x} \in I_j\}$
  $\mathcal{I}_j = I_j$
return $\mathcal{I}_j$, $\mathcal{K}_j$

Proposition 1: If $\mathrm{Win}(\cdot)$ computes the maximal (in set inclusion sense) winning set for $\mathbf{Prob}^2(\Phi, \Sigma)$, then Algorithm 1 computes the maximal winning set of Problem 1.

Proof: To solve problem $\mathbf{Prob}^1(\Psi, \Sigma^{\mathcal{F}}, F_j)$, Algorithm 1 first solves a set of similar problems $\mathbf{Prob}^1(\Psi, \Sigma^{\mathcal{F}}, F_i)$ for all configurations F_is that are strict successors of F_j. By doing this we get the failure tolerant set $\mathcal{I}_i$ under configuration F_i. Then we expand the unsafe set U_j to $\overline{U_j}$ by adding the complement of the winning sets $\mathcal{I}_i$s. Then the failure tolerant winning set for problem $\mathbf{Prob}^1(\Psi, \Sigma^{\mathcal{F}}, F_j)$ can be found by solving Problem 2 with the enlarged unsafe set $\overline{U_j}$. By doing this we are guaranteed that starting from fault configuration F_j, if some failure happens and the system ends up in one of F_j's strict successors F_i, the states are still in the failure tolerant winning set $\mathcal{I}_i$ under configuration F_i by construction. Thus the system under configuration F_i still satisfies corresponding specification Φ_i. By induction, it can be shown that $\Sigma^{\mathcal{F}}(F_j)$ satisfies Ψ from $\mathcal{I}_1$. Conversely, from any state $\boldsymbol{x}$ not in $\mathcal{I}_1$, it is not possible to satisfy one of Φ_i's. Since, there is no restriction on how fast the system can move between failure configurations, $\boldsymbol{x}$ is not contained in the winning set of Problem 1. ■

B. Solution of Problem 2

To solve Problem 2, we follow the procedure proposed in [11]. In short, we start with the coarsest possible partition of the domain induced by atomic propositions. Then, we construct an AFTS $\mathcal{T}$, consistent with this partition, over-approximating the switched system Σ, and synthesize a controller by doing graph search on $\mathcal{T}$. If a controller is found, we stop; if not, we construct a refinement $\hat{\mathcal{T}}$ of $\mathcal{T}$ and go to the previous step. Construction of the AFTS and refinement involves three tasks: (i) given a polytopic partition of the domain, how to effectively compute transitions between two cells under all possible parameter uncertainties; (ii) how to compute progress group under uncertainties; (iii) how to split cells properly in abstraction refinement. In what follows, we show how these tasks can be completed in a computationally efficient way (compared to [12]) under Assumptions 1–2.

1) Computation of Transitions: To compute the transitions from polytopic cell C_1 to one of its adjacent polytopic cells C_2 under $\boldsymbol{u}$, we need to check if $f_{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{q})$ is pointing from cell C_1 to cell C_2 or vice versa at some points on their intersection $C_1 \cap C_2$ for some possible parameter uncertainties $\boldsymbol{q} \in Q$. We say there exists a *positive (negative) flow* from cell C_1 to cell C_2 if there exists $\boldsymbol{x} \in F_{C_1,C_2} = C_1 \cap C_2$ and $\boldsymbol{q} \in Q$ such that $\boldsymbol{n}_{C_1,C_2}^T f_{\boldsymbol{u}}(\boldsymbol{x}; \boldsymbol{q}) > 0$ (< 0).[3] For general nonlinear systems, computing such transitions demands solving non-convex optimization problems [12], but under Assumption 1, it is sufficient to compute all $\boldsymbol{n}_{C_1,C_2}^T f_{\boldsymbol{u}}(\boldsymbol{x}, \boldsymbol{q})$ on $V_{F_{C_1,C_2}} \times V_Q$, which is a finite set.

Theorem 1: Given $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n, (\boldsymbol{x}, \boldsymbol{q}) \mapsto A(\boldsymbol{q})\boldsymbol{x} + K(\boldsymbol{q})$, where $A : \mathbb{R}^m \to \mathbb{R}^{n\times n}$ and $K : \mathbb{R}^m \to \mathbb{R}^n$ are multiaffine functions in $\boldsymbol{q}$, also given $\boldsymbol{x} \in F$ and $\boldsymbol{q} \in Q$, where F is an adjacent facet in $\mathbb{R}^n$ with normal vector $\boldsymbol{n}_F$ and Q is a hyper rectangle in $\mathbb{R}^m$, then

$$\begin{aligned} &\forall \boldsymbol{x} \in F, \forall \boldsymbol{q} \in Q : \boldsymbol{n}_F^T f(\boldsymbol{x}, \boldsymbol{q}) \lesseqgtr 0 \Leftrightarrow \\ &\forall \boldsymbol{x} \in V_F, \forall \boldsymbol{q} \in V_Q : \boldsymbol{n}_F^T f(\boldsymbol{x}, \boldsymbol{q}) \lesseqgtr 0, \end{aligned} \tag{17}$$

where V_F, V_Q are vertices of F and Q.

This result follows from (i) the fact that $f(\boldsymbol{x}, \boldsymbol{q})$ being affine in $\boldsymbol{x}$ implies that the value of $f(\boldsymbol{x}, \boldsymbol{q})$ at arbitrary point on polytope F can be written as a convex combination of

[3] When $C_1 \cap C_2$ has dimension less than $n-1$, a positive (negative) flow exists if there exists $\boldsymbol{x} \in C_1 \cap C_2, \boldsymbol{q} \in Q$ such that $\boldsymbol{n}^T f_{\boldsymbol{u}}(\boldsymbol{x}; \boldsymbol{q}) > 0$ (< 0) for all $\boldsymbol{n}$ in the normal cone of C_2 at $\boldsymbol{x}$. Here, for simplicity, we only describe the case where $C_1 \cap C_2$ has dimension $n-1$, but our implementation takes into account lower dimensional intersections, which follows similar lines.

its values on V_F, and (ii) Lemma 1. Detailed proof can be found in Appendix A.

By Theorem 1 (or more precisely, the contrapositive of Theorem 1), we immediately know that if there exists a positive (negative) flow from cell C_1 to cell C_2 somewhere on their adjacent facet F_{C_1,C_2} under some $\boldsymbol{q} \in Q$, there must be a positive (negative) flow from cell C_1 to cell C_2 at some vertex of facet F_{C_1,C_2}, under some extreme values of uncertainties $\boldsymbol{q}$. In other words, to compute all possible transitions from cell C_1 to cell C_2, it is sufficient to compute the transitions on $V_{F_{C_1,C_2}} \times V_Q$. Algorithm 2 gives the pseudo code to compute the transitions from C_1 to C_2 under dynamics f_u.

Algorithm 2 $[\rightarrow_{\boldsymbol{u}}^{C_1,C_2}] = \text{ComputeTrans}(C_1, C_2, f_{\boldsymbol{u}} = (A_{\boldsymbol{u}}, K_{\boldsymbol{u}}), Q)$

Compute all possible transitions between cells C_1 and C_2, under control u and arbitrary parameter uncertainty $\boldsymbol{q} \in Q$.

Input: Adjacent polytopic cells C_1 and C_2, vector field $f_{\boldsymbol{u}}$, allowable set of parameter uncertainties Q.

Output: Transitions $\rightarrow_{\boldsymbol{u}}^{C_1,C_2}$.

Initialize $\rightarrow_{\boldsymbol{u}}^{C_1,C_2} = \emptyset$
$F = \text{getAdjacnetFacet}(C_1, C_2)$
$\boldsymbol{n}_F = \text{getNormal}(F)$
Assume $\boldsymbol{n}_F$ points from C_1 to C_2
$V_F = \text{getVertices}(F)$
$V_Q = \text{getVertices}(Q)$
for $\boldsymbol{x} \in V_F$ **do**
  for $\boldsymbol{q} \in V_Q$ **do**
    if $\boldsymbol{n}_F^T(A_{\boldsymbol{u}}(\boldsymbol{q})\boldsymbol{x} + K_{\boldsymbol{u}}(\boldsymbol{q})) > 0$ **then**
      $\rightarrow_{\boldsymbol{u}}^{C_1,C_2} = \rightarrow_{\boldsymbol{u}}^{C_1,C_2} \cup \{(C_1, \boldsymbol{u}, C_2)\}$
    if $\boldsymbol{n}_F^T(A_{\boldsymbol{u}}(\boldsymbol{q})x + K_{\boldsymbol{u}}(\boldsymbol{q})) < 0$ **then**
      $\rightarrow_{\boldsymbol{u}}^{C_1,C_2} = \rightarrow_{\boldsymbol{u}}^{C_1,C_2} \cup \{(C_2, \boldsymbol{u}, C_1)\}$
return $\rightarrow_{\boldsymbol{u}}^{C_1,C_2}$

2) Computation of Progress Groups: To compute the progress group, it is important to be able to justify whether a cell is transient or not. Under Assumption 1 and 2, the set of all transient cells form a progress group, and a cell is transient if it contains no equilibrium [14] under all possible values of the uncertainty. Define the set of all possible equilibria under parameter uncertainties $\boldsymbol{q} \in Q$ to be

$$E_{\boldsymbol{u}} := \{\boldsymbol{x} \in \mathbb{R}^n \mid \exists \boldsymbol{q} \in Q : \boldsymbol{x} = A_{\boldsymbol{u}}^{-1}(\boldsymbol{q}) K_{\boldsymbol{u}}(\boldsymbol{q})\}. \tag{18}$$

By arguments above, a cell is transient under control $\boldsymbol{u}$ if it has empty intersection with $E_{\boldsymbol{u}}$. For an arbitrary affine system with uncertainties, it is usually difficult to compute $E_{\boldsymbol{u}}$ precisely. We can, however, compute a hyper rectangle containing $E_{\boldsymbol{u}}$ to approximate it under Assumption 1 and 2. To show this we need Theorem 2.

Theorem 2: Given dynamic system $\dot{\boldsymbol{x}} = A(\delta)\boldsymbol{x} + K(\delta)$, where $A = [\delta_{ij}] \in \mathbb{R}^{n \times n}$, $K = [\delta_l] \in \mathbb{R}^n$ and $\delta = [\delta_{11}, \ldots, \delta_{nn}, \delta_1, \ldots, \delta_n]^T \in \mathbb{R}^{n^2+n}$. Assume $\delta_{ij} \in [m_{ij}, M_{ij}]$, $\delta_l \in [m_l, M_l]$, also assume for all δ, $A(\delta)$ is always full rank, and define rectangular uncertainty set

$$\Delta = \prod_{i,j=1}^{n,n} [m_{ij}, M_{ij}] \prod_{l=1}^{n} [m_l, M_l], \tag{19}$$

and the set of all possible equilibria

$$E = \{\boldsymbol{x} \in \mathbb{R}^n \mid \exists \delta \in \Delta : \boldsymbol{x} = A^{-1}(\delta)K(\delta)\}, \tag{20}$$

then

$$E \subseteq R_E := \prod_{i=1}^{n} [\min_{\boldsymbol{x} \in \widetilde{E}} e_i^T \boldsymbol{x}, \max_{\boldsymbol{x} \in \widetilde{E}} e_i^T \boldsymbol{x}], \tag{21}$$

where e_i is the i^{th} natural basis of $\mathbb{R}^n$ and $\widetilde{E} := \{\boldsymbol{x} \in \mathbb{R}^n \mid \exists \delta \in V_\Delta : \boldsymbol{x} = A^{-1}(\delta)K(\delta)\}$

In short, Theorem 2 says the maximum (or minimum) coordinates of possible equilibra must be attained when δ is at the vertices of the hyper rectangular uncertainty set Δ. This is true because the equilibrium $A^{-1}(\delta)K(\delta)$ can be written in the form of $f(\delta)/g(\delta)$ where f and g are both multiaffine functions in δ. Then by Lemma 3 we can prove Theorem 2.

By Theorem 2, we can overestimate E by R_E, which can be found by computing the equilibria of the system at finitely many δs. Furthermore if given $\dot{\boldsymbol{x}} = A(\boldsymbol{q})\boldsymbol{x} + K(\boldsymbol{q})$ with $\boldsymbol{q} \in Q$ satisfies Assumption 1 and 2, then we know $\delta_{ij} = a_{ij}(\boldsymbol{q})$ is multiaffine in $\boldsymbol{q}$, and by Lemma 2, m_{ij}, M_{ij}, m_l, M_l can be found at the vertices of Q, which is also a finite set.

3) Abstraction Refinement: In order to include all dynamic behavior of a switched system in its AFTS approximation $\mathcal{T}$, it is usually inevitable to introduce some additional nonexisting behaviors into $\mathcal{T}$, which makes it harder to synthesize a controller. One typical source of conservatism is existence of a two-way flow between two adjacent cells C_1 and C_2. Such conservatism can be reduced to some extent by splitting the cell C_1 into two and constructing a refined AFTS $\hat{\mathcal{T}}$. Fig. 2 shows a sketch of the idea for splitting.

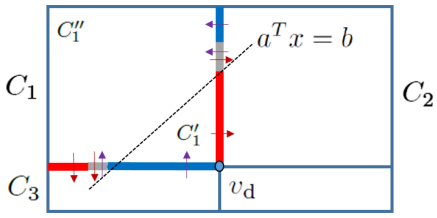


Fig. 2: Split cell C_1 into C_1', C_1'' by hyperplane $a^T x = b$, so that only one transition exists between C_1' and C_2, or C_1' and C_3.

The following theorem shows, under Assumption 1 and 2, how some of the two-way flows can be eliminated.

Theorem 3: Given an adjacent facet F in the state space with normal vector $\boldsymbol{n}_F$, we can partition F into three parts according to the direction of a given vector field f on F. To be specific, $F = D^+ \cup D^? \cup D^-$, where

$$D^+ := \{\boldsymbol{x} \in F \mid \forall \boldsymbol{q} \in Q : \boldsymbol{n}_F^T f(\boldsymbol{x}, \boldsymbol{q}) > 0\}$$
$$D^- := \{\boldsymbol{x} \in F \mid \forall \boldsymbol{q} \in Q : \boldsymbol{n}_F^T f(\boldsymbol{x}, \boldsymbol{q}) < 0\}$$
$$D^? := \{\boldsymbol{x} \in F \mid \exists \boldsymbol{q}_1, \boldsymbol{q}_2 \in Q : \boldsymbol{n}_F^T f(\boldsymbol{x}, \boldsymbol{q}_1) \geq 0 \text{ and } \boldsymbol{n}_F^T f(\boldsymbol{x}, \boldsymbol{q}_2) \leq 0\}.$$

If f and Q satisfy Assumption 1 and 2, D^+ and D^- are polytopes.

The proof of Theorem 3 can be found in Appendix B. By Theorem 3, we can compute $D^+(D^-)$ as polytopes and do the splitting in Fig. 2 according to $D^+(D^-)$. Note that simply taking a set difference (e.g., $C \setminus D^+$) can lead to non-convex cells. Therefore, we compute a splitting hyperplane $\boldsymbol{a}^T\boldsymbol{x} = b$ using Algorithm 3. The following functions are called in Algorithm 3:

1) $V_{\text{det}} = \text{GetDeterminedVertex}(C, \boldsymbol{f_u}, Q)$: Given polytopic cell C, and vector field $\boldsymbol{f_u}$ under allowable uncertainty set Q, return the set of vertices V_{det} of polytope C, at which the vector field has determined direction in terms of all the facets incident to this vertex. That is, $\boldsymbol{v} \in V_{\text{det}}$ iff for all F incident to $\boldsymbol{v}$, we either have $\boldsymbol{n}_F^T \boldsymbol{f_u}(\boldsymbol{v}, \boldsymbol{q}) > 0$, for all $\boldsymbol{q} \in Q$ or have $\boldsymbol{n}_F^T \boldsymbol{f_u}(\boldsymbol{v}, \boldsymbol{q}) < 0$, for all $\boldsymbol{q} \in Q$.
2) $[\boldsymbol{a}, b] = \text{SplittingHyperplane}(\boldsymbol{v}_{\text{d}}, V)$: Find splitting hyperplane $\boldsymbol{a}^T\boldsymbol{x} = b$ by solving convex optimization problem:

$$\begin{array}{ll} \text{minimize}_{\boldsymbol{x}} & ||\boldsymbol{x} - \boldsymbol{v}_{\text{d}}||_2 \\ \text{subject to} & \boldsymbol{x} \in Conv(V) \end{array} \tag{22}$$

Let $\boldsymbol{x}^*$ be the minimal to (22), then the splitting hyperplane $\boldsymbol{a}^T\boldsymbol{x} = b$ is given by $\boldsymbol{a} = \boldsymbol{x}^* - \boldsymbol{v}_{\text{d}}$, $b = -\boldsymbol{a}^T\boldsymbol{x}^*$.

Algorithm 3 $[C', C''] = \text{SplittingCell}(C, \boldsymbol{f_u}, Q)$
Split cell C to reduce conservatism

Input: Polytopic cell C to split, vector field $\boldsymbol{f_u}$, uncertainty set Q, assume there are two-way flows between C and at least one of its adjacent cells.
Output: two children cells C' and C''.

$\widehat{V} = \emptyset$
for C_i adjacent to C, with two-way flows between C and C_i **do**
  $\widehat{V} = \widehat{V} \cup V_{F_{C,C_i}}$
$V_{\text{det}} = \text{GetDeterminedVertex}(C, \boldsymbol{f_u}, Q)$
$\widehat{V} = \widehat{V} \cap V_{\text{det}}$
if $\widehat{V} \neq \emptyset$ **then**
  Let $\boldsymbol{v}_{\text{d}} \in \widehat{V}$
  $V = \emptyset$
  for F incident $\boldsymbol{v}_{\text{d}}$ **do**
    if $\boldsymbol{n}_F^T \boldsymbol{f}(\boldsymbol{v}_{\text{d}}, \boldsymbol{q}) > 0$ **then**
      $V = V \cup (V_{D^+} \setminus V_C)$
    else
      $V = V \cup (V_{D^-} \setminus V_C)$
  $[\boldsymbol{a}, b] = \text{SplittingHyperplane}(\boldsymbol{v}_{\text{d}}, V)$
  $C' = \{x \in C \mid \boldsymbol{a}^T\boldsymbol{x} \leq b\}$
  $C'' = \{x \in C \mid \boldsymbol{a}^T\boldsymbol{x} \geq b\}$
return

V. RESULTS

In this section, the proposed approach is used to solve the engine thermal management problem (i.e., Problem 1). Figs. (3a), (3b) show the winning sets found under fault configurations F_2 and F_3. Fig. (3c) shows the failure tolerant winning set for $\Sigma^{\mathcal{F}}$ defined by (10). The closed loop system behavior is simulated under randomly picked allowable constant uncertainties and possible component failures. Fig. 4 shows the plots of engine temperature obtained by simulation.

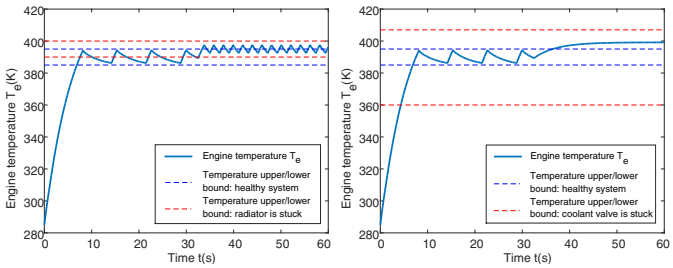


Fig. 4: Plots of engine temperature. Left: Radiator grill shutter gets stuck at 32.5s. Right: Coolant valve gets stuck at 32.5s.

VI. CONCLUSIONS

In this paper we formulated the engine thermal system as a system with failure modes, specified the requirements in thermal management problem using linear temporal logic and proposed algorithms necessary to synthesize a failure tolerant switching protocol. Simulation results showing the closed loop trajectories of the system, controlled by a switching protocol synthesized using the proposed framework, under uncertainties and component failures were presented.

APPENDIX

A. Proof of Theorem 1

Given $f(\boldsymbol{x}, \boldsymbol{q}) = A(\boldsymbol{q})\boldsymbol{x} + K(\boldsymbol{q})$, where A and K are multiaffine in $\boldsymbol{q} \in Q$, and Q is a hyper rectangle in $\mathbb{R}^m$, let $F \subset \mathbb{R}^n$ be an adjacent facet and $\boldsymbol{n}_F$ be its normal vector. Denote $g(\boldsymbol{x}, \boldsymbol{q}) = \boldsymbol{n}_F^T f(\boldsymbol{x}, \boldsymbol{q}) = \boldsymbol{n}_F^T (A(\boldsymbol{q})\boldsymbol{x} + K(\boldsymbol{q}))$.

Since F is a polytope, we know

$$\forall \boldsymbol{x} \in F: \ \exists \{\theta_i\} \text{ s.t. } \boldsymbol{x} = \sum_{\boldsymbol{x}_{\text{v}}^i \in V_F} \theta_i \boldsymbol{x}_{\text{v}}^i, \tag{23}$$

where $\sum \theta_i = 1$ and $\theta_i \geq 0$.

Substituting (23) into $g(\boldsymbol{x}, \boldsymbol{q})$, we have $\forall \boldsymbol{x} \in F : \forall \boldsymbol{q} \in Q$:

$$\begin{aligned} g(\boldsymbol{x}, \boldsymbol{q}) &= \boldsymbol{n}_F^T (A(\boldsymbol{q}) \sum_{\boldsymbol{x}_{\text{v}}^i \in V_F} \theta_i \boldsymbol{x}_{\text{v}}^i + K(\boldsymbol{q})) \\ &= \sum_{\boldsymbol{x}_{\text{v}}^i \in V_F} \theta_i g(\boldsymbol{x}_{\text{v}}^i, \boldsymbol{q}). \end{aligned} \tag{24}$$

Obviously, $g(\boldsymbol{x}, \boldsymbol{q})$ is multiaffine in $\boldsymbol{q}$. Then by Lemma 1,

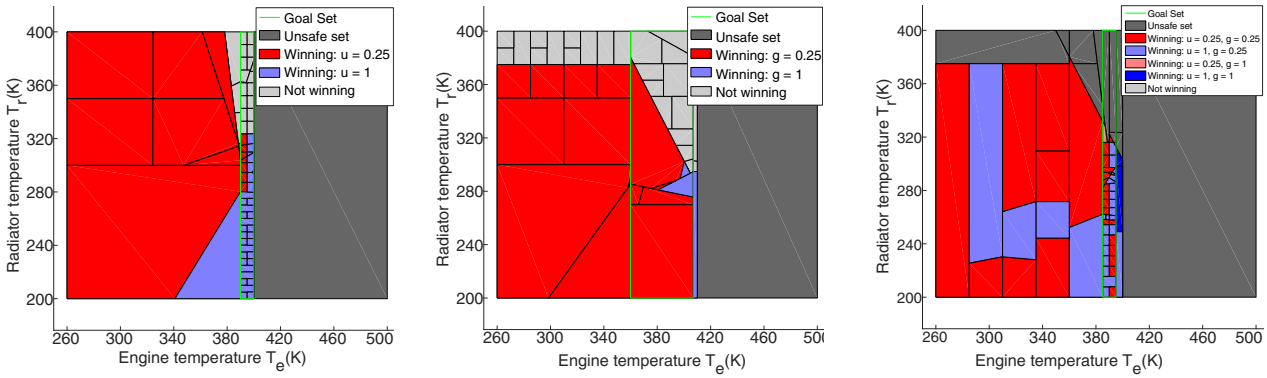
$$\forall i : \forall \boldsymbol{q} \in Q : \exists \{\lambda_j^i\} \text{ s.t. } g(\boldsymbol{x}_{\text{v}}^i, \boldsymbol{q}) = \sum_{\boldsymbol{q}_{\text{v}}^j \in V_Q} \lambda_j^i g(\boldsymbol{x}_{\text{v}}^i, \boldsymbol{q}_{\text{v}}^j), \tag{25}$$

where $\sum \lambda_j^i = 1$ and $\lambda_j^i \geq 0$.

Substituting (25) into (24), we have $\forall \boldsymbol{x} \in F : \forall \boldsymbol{q} \in Q$:

$$g(\boldsymbol{x}, \boldsymbol{q}) = \sum_{\boldsymbol{x}_{\text{v}}^i \in V_F} \sum_{\boldsymbol{q}_{\text{v}}^j \in V_Q} \theta_i \lambda_j^i g(\boldsymbol{x}_{\text{v}}^i, \boldsymbol{q}_{\text{v}}^j) \tag{26}$$

and $\sum_i \sum_j \theta_i \lambda_j^i = 1$ and $\theta_i \lambda_j^i \geq 0$.



(a) winning set I_2 for system Σ_2 (b) winning set I_3 for system Σ_3 (c) failure tolerant winning set $\mathcal{I}_1$ for system $\Sigma^{\mathcal{F}}$

Fig. 3: Winning sets (initial conditions from where the specification can be satisfied) under different fault configurations: (a) for configuration F_2, (b) for F_3, (c) for F_1 (computed recursively).

In other words, $\forall(\boldsymbol{x},\boldsymbol{q}) \in F \times Q$, $g(\boldsymbol{x},\boldsymbol{q})$ can be written as a convex combination of $g(\boldsymbol{x}_{\mathrm{v}}^i, \boldsymbol{q}_{\mathrm{v}}^j)$ with $(\boldsymbol{x}_{\mathrm{v}}^i, \boldsymbol{q}_{\mathrm{v}}^j) \in V_F \times V_Q$. Then it is obvious that

$$\forall \boldsymbol{x} \in F, \forall \boldsymbol{q} \in Q : \boldsymbol{n}_F^T f(\boldsymbol{x},\boldsymbol{q}) \lesseqgtr 0 \Leftrightarrow$$

$$\forall \boldsymbol{x} \in V_F, \forall \boldsymbol{q} \in V_Q : \boldsymbol{n}_F^T f(\boldsymbol{x},\boldsymbol{q}) \lesseqgtr 0.$$

B. Proof of Theorem 3

Here we only show D^+ is a polytope (the same holds for D^-). Let $F \subseteq \mathbb{R}^n$ be an adjacent facet and $\boldsymbol{n}_F$ be its normal vector. Denote $g(\boldsymbol{x},\boldsymbol{q}) = \boldsymbol{n}_F^T f(\boldsymbol{x},\boldsymbol{q}) = \boldsymbol{n}_F^T(A(\boldsymbol{q})\boldsymbol{x} + K(\boldsymbol{q}))$. Then D^+ is given by

$$D^+ = \{\boldsymbol{x} \in F \mid \forall \boldsymbol{q} \in Q : g(\boldsymbol{x},\boldsymbol{q}) > 0\}. \tag{27}$$

Define $\overline{D^+} := \{\boldsymbol{x} \in F \mid \forall \boldsymbol{q} \in V_Q : g(\boldsymbol{x},\boldsymbol{q}) > 0\}$. Note that $\overline{D^+}$ is a polytope. This is because for any specific $\boldsymbol{q} \in Q$, $g(\boldsymbol{x},\boldsymbol{q})$ is affine in $\boldsymbol{x}$ and $g(\boldsymbol{x},\boldsymbol{q}) > 0$ defines a half space. Since V_Q is a finite set, $\overline{D^+}$ is just the intersection of F with finite number of half spaces. Next we show $D^+ = \overline{D^+}$. To show this, we show either set contains the other.

1) $D^+ \subseteq \overline{D^+}$. Obviously because $V_Q \subseteq Q$.

2) $D^+ \supseteq \overline{D^+}$. $\forall \boldsymbol{x} \in \overline{D^+}$, by definition of $\overline{D^+}$ we have

$$\forall \boldsymbol{q}_{\mathrm{v}} \in V_Q : g(\boldsymbol{x},\boldsymbol{q}_{\mathrm{v}}) > 0. \tag{28}$$

Since $g(\boldsymbol{x},\boldsymbol{q}) = \boldsymbol{n}_F^T f(\boldsymbol{x},\boldsymbol{q})$ is multilaffine in $\boldsymbol{q}$, and Q is a hyper rectangle, by Lemma 1, we can write

$$\forall \boldsymbol{q} \in Q : g(\boldsymbol{x},\boldsymbol{q}) = \sum_{\boldsymbol{q}_{\mathrm{v}}^i \in V_Q} \theta_i g(\boldsymbol{x},\boldsymbol{q}_{\mathrm{v}}^i), \tag{29}$$

where $\sum \theta_i = 1$, and $\theta_i \geq 0$. It follows from (28) that $\forall \boldsymbol{q} \in Q : g(\boldsymbol{x},\boldsymbol{q}) > 0$, i.e., $\boldsymbol{x} \in \overline{D^+} \Rightarrow \boldsymbol{x} \in D^+$.

REFERENCES

[1] A. Balkan, M. Vardi, and P. Tabuada. Controller Synthesis for Mode-Target Games. In *Proc. of IFAC ADHS*, 2015.

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

[3] E. Dallal, A. Colombo, D. D. Vecchio, and S. Lafortune. Supervisory control for collision avoidance in vehicular networks with imperfect measurements. In *Proc. of IEEE CDC*, pages 6298–6303, 2013.

[4] P. Geels, B. Gessier, M. Chanfreau, and M. Tarquis. Advance control strategy for modern engine cooling thermal systems, and effect on co2 and pollutant reduction. *Proc. Veh. Therm. Manag. Syst., VTMS*, 6:631–641, 2003.

[5] A. Girard, G. Goessler, and S. Mouelhi. Safety controller synthesis for incrementally stable switched systems using multiscale symbolic models. *IEEE Trans. Autom. Control*, preprint, 2015.

[6] H. Kress-Gazit, G. Fainekos, and G. Pappas. Temporal-logic-based reactive mission and motion planning. *IEEE Trans. on Robotics*, 25:1370–1381, 2009.

[7] J. Liu, X. Liu, T.-K. J. Koo, B. Sinopoli, S. Sastry, and E. Lee. A hierarchical hybrid system model and its simulation. In *Proc. of IEEE CDC*, volume 4, pages 3508–3513, 1999.

[8] J. Liu, N. Ozay, U. Topcu, and R. Murray. Synthesis of reactive switching protocols from temporal logic specifications. *IEEE Trans. Autom. Control*, 58(7):1771 – 1785, 2013.

[9] O. Mickelin, N. Ozay, and R. M. Murray. Synthesis of correct-by-construction control protocols for hybrid systems using partial state information. In *Proc. of ACC*, pages 2305–2311, 2014.

[10] P. Nilsson, O. Hussien, Y. Chen, A. Balkan, M. Rungger, A. Ames, J. Grizzle, N. Ozay, H. Peng, and P. Tabuada. Preliminary results on correct-by-construction control software synthesis for adaptive cruise control. In *Proc. of IEEE CDC*, pages 816–823, 2014.

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

[12] 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*, 2013.

[13] H. Pang and C. Brace. Review of engine cooling technologies for modern engines. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 218(11):1209–1215, 2004.

[14] F. Sun, N. Ozay, E. M. Wolff, J. Liu, and R. M. Murray. Efficient control synthesis for augmented finite transition systems with an application to switching protocols. In *Proc. of ACC*, pages 3273–3280, 2014.

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

[16] C. Vermillion, J. Sun, and K. Butts. Predictive control allocation for a thermal management system based on an inner loop reference model: design, analysis, and experimental results. *Control Systems Technology, IEEE Transactions on*, 19(4):772–781, 2011.

[17] C. Vermillion, J. Sun, K. Butts, and A. Hall. Modeling and analysis of a thermal management system for engine calibration. In *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*, pages 2048–2053. IEEE, 2006.

[18] J. R. Wagner, V. Srinivasan, D. M. Dawson, and E. E. Marotta. Smart thermostat and coolant pump control for engine thermal management systems. Technical report, SAE Technical Paper, 2003.

[19] T. Wongpiromsarn, U. Topcu, and R. Murray. Receding horizon temporal logic planning. *IEEE Trans. Autom. Control*, 57(11):2817–2830, 2012.