Submitted, 2014 American Control Conference (ACC)
http://www.cds.caltech.edu/~murray/papers/mom14-acc.html

Synthesis of Correct-by-construction Control Protocols for Hybrid Systems Using Partial State Information

Oscar Mickelin[1], Necmiye Ozay[2] and Richard M. Murray[3]

Abstract

This paper considers the problem of synthesizing output-feedback control laws for a class of discrete-time hybrid systems in order for the trajectories of the system to satisfy certain high-level specifications expressed in linear temporal logic. By leveraging ideas from robust interpretation of temporal logic formulas and bounded-error estimation, we identify a subclass of systems for which it is possible to reduce the problem to a state-feedback form. In particular, we use locally superstable hybrid observers to resolve the partial information at the continuous level. This allows us to use recent results in temporal logic planning to synthesize the desired controllers based on two-player perfect-information games. The overall control architecture consists of a hybrid observer, a high-level switching protocol and a low-level continuous controller. We demonstrate the proposed framework in a case study on designing control protocols for an aircraft air management system.

I. Introduction

Correct-by-construction controller synthesis for hybrid systems from temporal logic specifications has attracted considerable attention in the past decade. At present, hybrid systems are put to use both in industrial settings and in products such as cars and airplanes [14]. Safety-criticality of such systems creates a need to synthesize controllers that enforce hybrid systems to satisfy certain high-level specifications, e.g., on safety, reliability and performance.

A typical solution for such control synthesis problems is a hierarchical control architecture with several discrete and continuous layers (see, for instance, [11], [7], [12], [23], [10] and references therein). One of the limitations of these approaches is that they rely on availability of the full system state for feedback. However, in many applications of interest, it is not possible to equip the system with a multitude of sensors both for reasons of economy and physical space. Motivated by these limitations, in this paper we propose a framework that can guarantee correctness with limited measurements through output feedback.

[1]O. Mickelin is with the Royal Institute of Technology (KTH), Sweden, oscarmi@kth.se.

[2]N. Ozay is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, necmiye@umich.edu.

[3]R. M. Murray is with the Department of Control and Dynamical Systems, California Institute of Technology, murray@cds.caltech.edu.

Previous work on synthesis with partial state information has mostly focused on the discrete level [3], [15]. Except for some special cases [8], handling the imperfect state information at the discrete level requires a power set construction (i.e., construction of a belief space) that has prohibitive computational complexity. In this paper, we consider partial observability of continuous state. In order not to experience a complexity blow-up at the discrete level, we deal with the partial state information at the continuous level. We leverage ideas from robust interpretation of temporal logic formulas [6] and bounded-error estimation [13] to develop a framework for synthesizing provably-correct output-feedback control laws for discrete-time piecewise-affine systems. In particular, for a class of systems admitting locally superstable hybrid observers [4], [2], we show that the problem can be reduced to a state-feedback form, which can then be solved using available tools [24].

The rest of the paper is organized as follows. Section II presents some background results. The problem is formally stated and an overview of the solution strategy is given in section III. Main results are presented in sections IV-V. Section VI demonstrates an application of the proposed framework to a case study on aircraft air management systems. Finally, section VII concludes the paper with some remarks.

II. Preliminaries

A. Notation

All matrix norms considered in this article will be the infinity norms, denoted without subscripts. Therefore, given a matrix $A = (A)_{ij}$ and a vector $x = [x_1, \ldots, x_n]^T$, we define $\|A\| = \max_i \sum_j |A_{ij}|, \|x\| = \max_i |x_i|$ and $\|x\|_1 = \sum_{i=1}^{n} |x_i|$. The i^{th} row vector of A is written as $[A]_i$. The row and column spaces of A is denoted by row(A) and col(A), respectively, with dimensions dim(row(A)) and dim(col(A)); and the dimension of the null space is written as nullity(A). The diameter of a set $X \subseteq R^n$ is denoted by $\mathrm{diam}(X)$, its closure as $\overline{X}$ and its distance from a point $p \in \mathbb{R}^n$ by $d(p, X) = \inf_{x \in X} \|p - x\|$.

With a point $p \in \mathbb{R}^n$ and $r \in \mathbb{R}$, we denote the ball centered at p with radius r as $B(p, r) = \{x \in \mathbb{R}^n : \|x - p\| \leq r\}$. Lastly, given a matrix $H \in \mathbb{R}^{m \times n}$ and a vector $k \in \mathbb{R}^m$, a *polytope* is a set $P = \{x : Hx \leq k\} \subseteq \mathbb{R}^n$, where the inequality is interpreted element-wise, i.e., $P = \{x : [Hx]_i \leq [k]_i, i = 1, \ldots, m\}$.

Given a set Q, Q^ω (Q^+) denotes the set of infinite (non-empty finite) sequences of elements in Q.

B. System and environment models

We consider discrete-time piecewise affine systems formally defined as follows.

Definition 1: A discrete-time piecewise-affine system is a tuple $S = (X, \{R_k\}_{k=1}^{k_{max}}, \{\mathfrak{D}_k\}_{k=1}^{k_{max}})$ where:

- $X \subseteq \mathbb{R}^n$ is a compact set called the state space.
- The regions $R_k \subseteq X$, $R_i \cap R_j = \emptyset$ for $i \neq j$, form a partition of X.
- $\mathfrak{D}_k$ is the dynamics in region R_k, that is the state x evolves with

$$\begin{aligned} x(t+1) &= A_k x(t) + B_k u(t) + F_k + E_k \delta(t) \\ y(t) &= C_k x(t) \end{aligned} \tag{1}$$

when $x(t) \in R_k \subseteq X$. In Eq. (1), $y(t) \in \mathbb{R}^l$ is the measured output, $u(t) \in U \subseteq \mathbb{R}^m$ is the control input, $x(t) \in \mathbb{R}^n$ is the state variable, and $\delta(t) \in W \subseteq \mathbb{R}^d$ is the disturbance.

We let Y denote the set of outputs, that is, $Y \doteq \{Cx : x \in X\}$. Given a system S, a subset $X' \subseteq X$ is said to *respect the dynamics* if $X' \cap R_k \neq \emptyset$ only for a unique k.

In addition to the disturbances in the system model, we consider an external *environment* to refer to the "discrete" factors that are relevant to the operation of the system, but do not impact its dynamics directly, i.e., not explicitly appear in Eq. (1). Since such factors are not necessarily controlled by the system, e.g., traffic lights, weather conditions, user inputs; they are treated as adversaries. We use a simple transition system to model environment evolution.

Definition 2: An environment model is a tuple $\mathcal{T}_e = (\mathcal{E}, \mathcal{E}_0, \rightarrow)$ where:

- $\mathcal{E}$ is a finite set of states.
- $\mathcal{E}_0 \subseteq \mathcal{E}$ is a set of initial states, i.e., $e(0) \in \mathcal{E}_0$.
- $\rightarrow \subseteq \mathcal{E} \times \mathcal{E}$ is a transition relation that governs the evolution of the environment. That is, $(e(t), e(t+1)) \in \rightarrow$ for all $t \geq 0$.

The discrete environment is assumed to be fully observable by the system.

Remark 1: For the clarity of the presentation, we restrict the system dynamics to the form in Eq. (1). Our framework can be easily extended to cases where there is measurement noise or where the dynamics include controllable and uncontrollable switches (e.g., using ideas from [16], [10]). Also, our framework allows general U, W and $\{R_k\}_{k=1}^{k_{max}}$, but, in what follows, we assume these sets are bounded convex polytopes to facilitate computation.

C. Linear temporal logic and protocols

Linear temporal logic (LTL) is a formal language that extends the standard propositional logic with temporal operators to express complex, temporal tasks [1]. LTL has proven useful in e.g., software and hardware verification, robotics and other applications of control synthesis, allowing for succinct and expressive specification of system behavior.

1) Syntax and semantics: Before defining the syntax and semantics of LTL, we need a few definitions. A combined state of the system and the environment is a tuple $s(t) \doteq (e(t), x(t)) \in \mathcal{E} \times X$ and a *trajectory* is an infinite sequence of states of the form $s = s(0)s(1)\ldots \in (\mathcal{E} \times X)^{\omega}$. An *atomic proposition* is a function from the set of states to boolean $true$ and $false$. We denote the set of all atomic propositions by Π. In our context each $\pi_i \in \Pi$ is an indicator function of a set $[\![\pi_i]\!] = \{e\} \times X_i$, with $e \in \mathcal{E}$ and $X_i \subseteq X$ is a convex polytope, wherein π evaluates to $true$. A set $X' \in X$ is said to *respect the propositions* if for all $x \in X'$ the same set of propositions hold.

The syntax of an LTL formula over a set of atomic propositions Π is given by the following grammar:

$$\varphi ::= true | \pi | \varphi_1 \wedge \varphi_2 | \neg\varphi | \bigcirc \varphi | \varphi_1 \mathcal{U} \varphi_2,$$

where $\pi \in \Pi$, and φ_1 and φ_2 are LTL formulas. The symbols $\wedge$, $\neg$, $\bigcirc$ and $\mathcal{U}$ stand for the logical operators conjunction, negation and temporal operators next and until, respectively. These operators can be used to define additional operators such as disjunction ($\vee$), implication ($\rightarrow$), always ($\Box$) and eventually ($\Diamond$). We will consider specifications in assume/guarantee form

$$\varphi \doteq \varphi_e \rightarrow \varphi_s, \tag{2}$$

where φ_e encodes assumptions on the environment and φ_s specifications of the desired system behavior.

Satisfaction of a formula φ at a state $s(t)$ in a trajectory s is denoted by $s(t) \models \varphi$ and is defined by letting

1) $s(t) \models true$;
2) For any atomic proposition π, $s(t) \models \pi \quad \text{if } s(t) \in [\![\pi]\!]$;
3) $s(t) \models \varphi_1 \wedge \varphi_2 \quad \text{if } s(t) \models \varphi_1$ and $s(t) \models \varphi_2$;
4) $s(t) \models \neg\varphi \quad \text{if } s(t) \not\models \varphi$;
5) $s(t) \models \bigcirc\varphi$ if $s(t+1) \models \varphi$;
6) $s(t) \models \varphi_1 \mathcal{U} \varphi_2$ if $\exists j \in \mathbb{N}$ s.t. $s(i) \models \varphi_1$, $s(j) \models \varphi_2$, $\forall i \in [t, j)$.

The trajectory s satisfies the formula φ if $s(0) \models \varphi$. We say that a formula φ on the form (2) is *satisfied by the system* if it is satisfied by all possible trajectories of the system which are consistent with the dynamics in (1) and for all environment behaviors captured by the environment model.

A *state-feedback control protocol* $\mathcal{C}$ is a partial function on non-empty sequences of states of the system with

$$\begin{aligned} \mathcal{C} : (\mathcal{E} \times X)^+ \times \mathcal{E} \rightarrow U \\ (s(0), s(1) \ldots, s(t-1), e(t)) \mapsto u(t) \end{aligned} \tag{3}$$

where $u(t)$ is the input signal to be used in the subsequent time-step. Lastly, by letting $r(t) \doteq (y(t), e(t))$, we define an *output-feedback control protocol* as

$$\begin{aligned} \mathcal{C} : (\mathcal{E} \times Y)^+ \times \mathcal{E} \rightarrow U \\ (r(0), r(1) \ldots, r(t-1), e(t)) \mapsto u(t) \end{aligned} \tag{4}$$

where $u(t)$ is decided upon by only using the measured output.

2) Robust satisfaction of LTL formulas: Following [6], this section describes a robust interpretation of LTL formulas. Given $\pi \in \Pi$, define π^ε, by $[\![\pi^\varepsilon]\!] = \{(e, x) \in [\![\pi]\!] : (e, x') \in [\![\pi]\!], \forall x' \in B(x, \varepsilon)\}$. $[\![\pi]\!]$ denotes a robust version of the atomic proposition $[\![\pi]\!]$, which will be used in connection with estimation errors below. We can extend the robustness properties to general LTL formulas as follows. Take φ as a formula written on Negation Normal Form [5]. Form $\neg\Pi = \{\neg\pi : \pi \in \Pi\}$ and let $\hat{\Pi} = \Pi \cup \neg\Pi$. Now, interpreting φ as a formula over $\hat{\Pi}$, replace all atomic propositions π by π^ε and denote the resulting formula by φ^ε. We say that a system *satisfies a formula φ ε-robustly* if it satisfies φ^ε.

III. Problem formulation and solution strategy

Next, we formally state the problem and give an overview of the proposed solution.

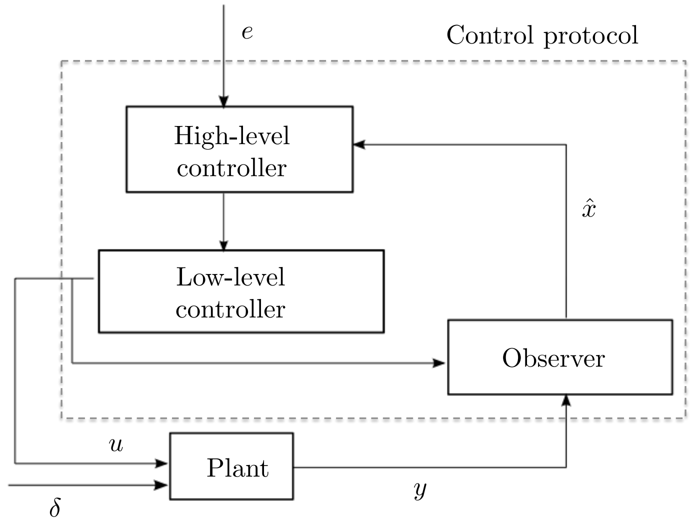

Figure 1. The proposed control architecture.

Problem 1: Given a system $S = (X, \{R_k\}_{k=1}^{k_{max}}, \{\mathfrak{D}_k\}_{k=1}^{k_{max}})$, an environment $\mathcal{T}_e = (\mathcal{E}, \mathcal{E}_0, \rightarrow)$, a set Π of propositions together with an LTL formula φ as in (2), and a set $X_0 \subseteq X$ that respects the propositions and dynamics, construct an output feedback control protocol that satisfies φ for all initial conditions $x(0)$ in X_0 and for all possible environment behaviors in $\mathcal{T}_e$ using only the measured output y.

Starting from a system model given in the form of Def. 1 and an LTL specification in the form of (2), we use an approach centered on observer estimations of the state space in order to solve Problem 1. The proposed framework exploits the hierarchical approach considered in earlier work [23], [16] and consists of the following steps:

1) Find a locally superstable observer with an appropriate equalized performance level and redefine the system dynamics and LTL specifications based on the estimated state.
2) Produce a discrete abstraction based on the redefined dynamics.
3) Use existing techniques in automata theory to design a control protocol guaranteeing correctness of the system.
4) Implement the automaton for continuous execution by combining the observer with low-level controllers.

In section IV, we discuss certain types of observers that are suitable for steps 1 and 4. In section V, we prove that given such an observer, one can still guarantee correctness when using the redefined dynamics in steps 2 and 3 and treating the problem as a state-feedback problem as in [23], [16]. We also briefly overview the results from [23], [16] necessary to complete these design steps.

The overall control architecture shown in Fig. 1 consists of a hybrid observer, a high-level switching protocol and a low-level continuous controller.

IV. OBSERVER DESIGN

In order to utilize possible partially known state information, we use observers

$$\begin{aligned} \mathcal{O} : (Y \times U)^* &\to X \\ (p(0), p(1) \ldots, p(t-1)) &\mapsto \hat{x}(t) \end{aligned} \tag{5}$$

where $p(t) \doteq (y(t), u(t))$ and $\hat{x}(t)$ is an estimate of the state. The estimation error at time t is denoted by $\xi(t) \doteq x(t) - \hat{x}(t)$. The design of an observer is made more difficult for piecewise-affine systems as an error in the estimate ambiguities the underlying dynamics. Moreover, any atomic proposition $\pi \in \Pi$ holds true in a limited region $[\![\pi]\!] \subseteq \mathcal{E} \times X$; and $(e, \hat{x}) \in [\![\pi]\!]$ does not imply $(e, x) \in [\![\pi]\!]$. Therefore, upper bounds on the estimation errors are needed.

Typically, optimal observers that minimize the estimation error when there are persistent disturbances can be arbitrarily complex even for linear systems [18], [13]. Instead of seeking optimal bounds, we adopt the notion of equalized performance from [2] to characterize observers.

Definition 3: [Equalized performance] An observer is said to achieve an *equalized performance level* μ if, whenever $\|\xi(t)\| \leq \mu$, we have $\|\xi(t+1)\| \leq \mu$.

For a piecewise-affine system as in Def. 1, we consider fixed-complexity locally-affine observers of the form

$$\hat{x}(t+1) = (A_k - L_k C_k)\hat{x}(t) + B_k u(t) + F_k + L_k y(t), \tag{6}$$

with a collection of linear filter gains $L_k \in \mathbb{R}^{n \times l}$, one for each R_k.

Proposition 1: Consider an observer of the form (6). Assume, for the time being, that the observer has perfect knowledge of k (i.e., the region R_k the true state $x(t)$ is in) at all times.[1] Then, choosing the filter gains L_k in Eq. (6) such that

$$\|A_k - L_k C_k\| \leq 1 - \frac{\max_{\delta \in W} \|E_k \delta\|}{\varepsilon} \tag{7}$$

for all k leads to an equalized performance level ε.

Proof: Since k is known by assumption, the estimation error evolves as

$$\xi(t+1) = (A_k - L_k C_k)\xi(t) + E_k \delta(t). \tag{8}$$

By equating the norms of both sides of Eq. (8) and with simple manipulation, one can see that if $\|\xi(t)\| \leq \epsilon$ and Eq. (7) holds, we have $\|\xi(t+1)\| \leq \epsilon$. ■

Definition 4: [Locally detectable system] An observer of the form (6) that satisfies Eq. (7) is called a *locally superstable observer with equalized performance* ε. A piecewise affine system is called *locally detectable with performance level* ε if it admits a locally superstable observer with equalized performance ε.

[1]Note that this is true at $t = 0$ because X_0 respects the dynamics. In section V, we show how to synthesize the control protocol so that it chooses the consequent control inputs u to ensure this at later time steps.

Note that this condition is more restrictive than the notion of detectability as, firstly, detectability only concerns the behavior of a trajectory as time goes to infinity, and secondly, the existence of a matrix L such that $\|A-LC\|<1$ as in Eq. (7) is a sufficient condition for detectability of (A, C).

Since, by assumption, X_0 respects propositions, we can pick $\hat{x}(0) \in X_0$, which yields the bound $\|\xi(0)\| \leq \text{diam}(X_0)$. To achieve sufficient control of the estimation error, we require the system in Problem 1 to be locally detectable with performance level diam(X_0).

A. Conditions on system matrices for superstability

In this section, we provide a characterization of a subclass of systems for which there exist locally superstable observers. In what follows we suppress the subscript k. The last section outlines a strategy for synthesizing correct-by-construction control protocols for systems with partial state information, under the condition that there exists a matrix L with $\|A - LC\| \leq \varepsilon'$, for some choice of vector and induced matrix norm. Choosing the infinity norm reduces the construction of such an L into a linear programming problem, accompanied by simple theoretical conditions for the existence of a filter as outlined below.

1) Reformulation into linear programming: Take $A = (a_{ij})_{i,j=1}^n$, let l_i be the i^{th} row vector of L and c_j be the j^{th} column vector of C. Then $(A - LC)_{ij} = a_{ij} - l_i \cdot c_j$, so

$$\|A - LC\| = \max_{i=1,\dots,n} \sum_{j=1}^{n} |a_{ij} - l_i \cdot c_j|. \tag{9}$$

Minimizing the above for a fixed value of i gives the problem

$$\min_x \|\overline{a}_i - Bx\|_1, \tag{10}$$

where we let $x = l_i$, $\overline{a}_i = (a_{i1}, a_{i2}, \dots, a_{in})$, $B = C^T$. This can be phrased as a linear programming problem with

$$\begin{aligned} &\min_{x,y} \sum_{i=1}^{n} y_i \\ &\text{s.t.} - y \leq \overline{a}_i - Bx \leq y, \end{aligned} \tag{11}$$

where $y = (y_1, \dots, y_n)$ and the inequality in the constraints is interpreted elementwise. This is easily solved, but it is also of interest to characterize when it is at all possible to obtain $\|A - LC\| \leq \varepsilon'$.

2) Conditions for existence of locally superstable linear observer: By the above, for all $i = 1, \dots, n$, we consider the problem of finding l_i with

$$\|\overline{a}_i - C^T l_i\|_1 < \varepsilon'. \tag{12}$$

Evidently, this is possible exactly when there is a $\overline{d}_i \in \mathbb{R}^n$ with $\|d_i\|_1 < \varepsilon'$, such that $\overline{a}_i + \overline{d}_i \in \text{col}(C^T)$. Stacking the d_i as row vectors of the matrix D, we obtain the following result.

Proposition 2: There exists a linear filter L for a system on the form (6) with $\|A - LC\| \leq \varepsilon'$ if and only if there exists a $D \in \mathbb{R}^{n \times n}$ with

1) $\text{row}(A + D) \subseteq \text{row}(C)$

2) $\|D\| < \varepsilon'$

Proof: By the above, existence of a filter is equivalent for each row to satisfy $[A+D]_i \in \text{col}(C^T) = \text{row}(C)$. Next, $\sum_{j=1}^{n} |d_{ij}| < \varepsilon'$ for all $i = 1, \ldots, n$, so certainly $\max_{i=1,\ldots,n} \sum_{j=1}^{n} |d_{ij}| = \|D\| < \varepsilon'$ and one direction is done. As all steps are reversible, the converse is also proved. ■

Note that as $\dim(\text{col}(C^T)) \leq l$, a filter achieving the required bound is more easily constructed when $A+D$ has low rank or when $l \to n$, which intuitively corresponds to the two cases where either $A+D$ does not hold much information or when C gives practically full state information, respectively.

Next, the rank-nullity matrix for a matrix $A \in \mathbb{R}^{n_1 \times n_2}$ states that $\dim(\text{col}(A)) + \text{null}(A) = n_2$, which for the two matrices above becomes

$$\dim(\text{col}(A^T + D^T)) + \text{null}(A^T + D^T) = n \tag{13}$$

$$\dim(\text{col}(C^T)) + \text{null}(C^T) = l. \tag{14}$$

Now, $\dim(\text{col}(A^T + D^T)) = \dim(\text{row}(A + D))$ and likewise for C and, further, $\text{row}(A + D) \subseteq \text{row}(C) \Rightarrow \dim(\text{row}(A + D)) \leq \dim(\text{row}(C))$. Subtracting the equations above gives

$$\begin{aligned} n - l = \dim(\text{col}(A^T + D^T)) - \dim(\text{col}(C^T)) + \\ + \text{null}(A^T + D^T) - \text{null}(C^T), \end{aligned} \tag{15}$$

i.e.,

$$\begin{aligned} \text{null}(A^T + D^T) &= n - l + \text{null}(C^T) + \dim(\text{col}(C^T)) - \\ &\quad - \dim(\text{col}(A^T + D^T)) \\ &\geq n - l. \end{aligned} \tag{16}$$

Summarizing the calculations above, we obtain

Proposition 3: If there exists a linear filter L for a system on the form (6) with $\|A - LC\| \leq \varepsilon'$, then there exists a $D \in \mathbb{R}^{n \times n}$ with

1) $\text{nullity}(A^T + D^T) \geq n - l$
2) $\|D\| < \varepsilon'$

Note that this sufficiently characterizes the systems for which the developed framework is possible to use, for systems with zero measurement error and that violations of these conditions mean that the strategy typically does not work when there is measurement error present, as this, intuitively speaking, makes the problem more difficult.

Note that this sufficiently characterizes the systems for which the developed framework is possible to use.

V. Reduction to state feedback

Obviously, if we can ensure that the estimated state trajectory robustly satisfies an LTL formula and the estimation error can be kept globally bounded, then the true trajectory satisfies the LTL formula. This fact is stated formally next.

Theorem 1: Let $s = s(0)s(1)...$ be an infinite sequence where $s(t) = (e(t), x(t))$ for all t. Similarly, define $\hat{s} = \hat{s}(0)\hat{s}(1)...$ where $\hat{s}(t) = (e(t), \hat{x}(t))$ for all t. Given an LTL formula φ, if there exists a bound $\varepsilon \geq 0$ such that $\hat{s} \models \varphi^{\varepsilon}$ and $||\hat{x}(t) - x(t)|| \leq \varepsilon$ for all t, then $s \models \varphi$.

Proof: Follows directly from the definition of robust satisfaction of LTL formulas. ■

In order to be able to employ this result, we need to establish a global bound ε on the estimation error using locally superstable observers. We first consider an intermediate result.

Lemma 1: Let $S = (X, \{R_k\}_{k=1}^{k_{max}}, \{\mathfrak{D}_k\}_{k=1}^{k_{max}})$ be a system. If S is locally detectable with performance level $\text{diam}(X_0)$ and $\hat{x}(t) \in \bigcup_{k=1}^{k_{max}} \hat{R}_k$ for all $t \geq 0$, where $\hat{R}_k \doteq \{x \in R_k : B(x, \text{diam}(X_0)) \subseteq R_k\}$ then $\|\xi(t)\| \leq \text{diam}(X_0)$ for all $t \geq 0$.[2]

Proof: Since S is locally detectable with performance level $\text{diam}(X_0)$, there exist L_k such that Eq. (7) holds. By Proposition 1, the estimation error can be bounded if the unique region R_k with $x \in R_k$ is known. By the assumptions in problem formulation 1, this is known at $t = 0$ and we proceed by induction on t. Given $x(t) \in R_k$, we know $\|\xi(t)\| \leq \text{diam}(X_0)$. By the definition of the shrunk regions, $d(x(t), \hat{R}_j) > \text{diam}(X_0)$, for $j \neq k$ so if $\hat{x}(t) \notin \hat{R}_k$, then $\|\xi(t)\| = \|x(t) - \hat{x}(t)\| \geq d(x(t), \hat{R}_j) > \text{diam}(X_0)$, which contradicts the induction hypothesis. Therefore, $\hat{x}(t) \in \hat{R}_k$ and, in the next time step, we can measure $\hat{x}(t+1) \in \hat{R}_j$, for some $1 \leq j \leq k_{max}$ and obtain $\|\xi(t+1)\| \leq \text{diam}(X_0)$, by Proposition 1. This concludes the induction step and the proof is done. ■

We associate each locally detectable system with another system whose outputs are equal to its states.

Definition 5: Given a locally detectable system $S = (X, \{R_k\}_{k=1}^{k_{max}}, \{\mathfrak{D}_k\}_{k=1}^{k_{max}})$ that admits a locally superstable observer with performance level ε and corresponding gains L_k, the *ε-robust observer system* $\hat{S} = (\hat{X}, \{\hat{R}_k\}_{k=1}^{k_{max}}, \{\hat{\mathfrak{D}}_k\}_{k=1}^{k_{max}})$ is given by the following parameters:

- $\hat{X} = \bigcup_{k=1}^{k_{max}} \hat{R}_k$,
- $\hat{R}_k = \{x \in R_k : B(x, \varepsilon) \subseteq R_k\}$,
- $\hat{\mathfrak{D}}_k$ is the dynamics in region $\hat{R}_k$, with

$$\begin{aligned} \hat{x}(t+1) &= \hat{A}_k \hat{x}(t) + \hat{B}_k \hat{u}(t) + \hat{F}_k + \hat{E}_k \hat{\delta}(t) \\ \hat{y}(t) &= \hat{x}(t), \end{aligned} \tag{17}$$

where $\hat{A}_k = A_k$, $\hat{B}_k = B_k$, $\hat{F}_k = F_k$, $\hat{E}_k = L_k C_k$, $\hat{u}(t) \in \hat{U} = U$ and $\hat{\delta}(t) \in \hat{W} = B(0, \varepsilon)$.

An observer for S as in Eq. (6) is said to be *consistent* with an ε-robust observer system $\hat{S}$ if they use the same gains L_k.

Now we state the main result of this section where ε-robust observer systems are used to pose an alternative perfect-information problem, the solution of which provides a solution to Problem 1.

Theorem 2: Define $\varepsilon' \doteq \text{diam}(X_0)$. Given an instance $(S, X_0, \mathcal{T}_e, \varphi)$ of Problem 1, assume S is locally detectable with performance level ε'. Let $\hat{S} = (\hat{X}, \{\hat{R}_k\}_{k=1}^{k_{max}}, \{\hat{\mathfrak{D}}_k\}_{k=1}^{k_{max}})$ be a ε'-robust observer system for S. Then, if there exists a state-feedback control protocol for $\hat{S}$ that makes the system satisfy $\varphi^{\varepsilon'}$ for some the initial condition

[2] To be precise, a performance level strictly less than $\text{diam}(X_0)$ is required to accommodate trajectories with $x(t)$ on a border between two regions. However, such cases are negligible from a practical standpoint and will be disregarded.

$\hat{x}(0) \in X_0 \cap \hat{X}$ and for all possible environment behaviors in $\mathcal{T}_e$, $(S, X_0, \mathcal{T}_e, \varphi)$ is realizable. Moreover, an output-feedback protocol for S can be constructed by using an observer consistent with the dynamics of $\hat{S}$ and by driving the system S with the same control signals as applied to $\hat{S}$.

Proof: By definition, the state-feedback control protocol for $\hat{S}$ ensures that $\hat{x}(t) \in \hat{X}$ for all t as the range of a protocol is the state space of the system. Therefore, by Lemma 1, the estimation error can be globally bounded by ε' for any initial condition $\hat{x}(0) \in X_0 \cap \hat{X}$ while running the system S and the observer with the input signal from this protocol. Finally, invoking Theorem 1 concludes the proof. ■

A. Overview of full-information synthesis

This section briefly describes the process of obtaining state-feedback control protocols. The full details can be found in e.g., [16], [23].

Based on previous work [16], [23], a discrete synthesis procedure can be phrased as a two-player perfect information game, where the environment is treated as an adversary; i.e., it is assumed to make the worst-case transitions consistent with its transition relation in Def. 2 and the assumption φ_e part of the specification. In order to incorporate the piecewise affine system in this game, constructing a finite transition system representing the dynamics is required [23]. This construction relies on partitioning the continuous state-space to create discrete-states and solving short-horizon constrained reachability problems between regions to establish the transition relations.

In order to solve the state-feedback synthesis problem stated in Theorem 2, we create a discrete-transition system for the ε-robust observer system, where the reachability computations are performed on shrunk regions. The first step in doing so produces a proposition preserving partition $X = \bigcup_{i=1}^{n} P_i$ respecting the system dynamics. Assuming P_i to be a convex polytope defined by $H_i x \leq k_i$, a shrunk polytope can then be defined as

$$\hat{H}_i x \leq \hat{k}_i, \tag{18}$$

where $\hat{H}_i = H_i$ and $[\hat{k}_i]_j = [k_i]_j - \varepsilon \|[H_i]_j\|$. This gives the following result.

Proposition 4: If $\|\xi(t)\| \leq \varepsilon$ and $\hat{x}(t) \in \hat{P}_i$, then $x(t) \in P_i$.

Proof: We have $H_i x(t) = H_i\,(x(t) - \hat{x}(t) + \hat{x}(t)) = H_i\,(x(t) - \hat{x}(t)) + \hat{H}_i \hat{x}(t)$. In the last equality, $\|x(t) - \hat{x(t)}\| \leq \varepsilon$, and by construction, $[\hat{H}_i \hat{x}(t)]_j \leq [k_i]_j - \varepsilon \|[H_i]_j\|$, so we obtain $[H_i x(t)]_j \leq [\hat{H}_i \hat{x}(t)]_j + \varepsilon \|[H_i]_j\| \leq [k_i]_i - \varepsilon \|[H_i]_i\| + \varepsilon \|[H_i]_j\| = [k_i]_j$. ■

Now, appealing to Theorem 2, transitions for the estimated state $\hat{x}$ between regions $\hat{P}_i$ and $\hat{P}_j$ lead to transitions of the actual state between regions P_i and P_j, meaning that algorithms for control synthesis [24], [17] applied directly to the estimated system in Problem **??** lead to control protocols of the original system in Problem 1.

Lastly, note that if a region P_i in Proposition 4 is given by a union of several convex polytopes, shrinking each of these will yield a valid, although conservative, region for control synthesis.

We summarize the ideas above in the following algorithm:

1) Establish a proposition preserving partition $X = \bigcup_{i=1}^{n} P_i$ of the state space domain X, respecting the system dynamics.

2) Shrink each region P_i. Establish transition relations for each of the new polytopes $\hat{P}_i$.
3) Force $\hat{x}$ to transition between the shrunk polytopes.

B. Simple example

This section uses a simple example to demonstrate the procedure outlined in sections II-V above. We consider the following problem in robot mission planning. Let the state space be $\{x = [x_1, x_2]^T \in \mathbb{R}^2 : 0 \leq x_1 \leq 3,\, 0 \leq x_2 \leq 2\}$, divided into six identical squares which we enumerate as X_i, $i = 0, \ldots, 5$. A robot is to patrol the X_i, always eventually reaching X_0 and X_2 and always eventually reaching X_3 when receiving a signal to do so. We introduce a boolean environment variable $park$ which, when $true$, orders the robot to proceed to X_3. In LTL, the specifications can be written as

$$(\Box\Diamond\neg park) \rightarrow \Box\Diamond(X_0) \wedge \Box\Diamond(X_2) \wedge \Box\,(park \rightarrow \Diamond X_3)\,. \tag{19}$$

We assume dynamics defined throughout the whole state space of the form (1), where $k = 1$:

$$\begin{aligned} \begin{bmatrix} x_1(t+1) \\ x_2(t+1) \end{bmatrix} &= \begin{bmatrix} 0.75 & 0 \\ 0 & 0.65 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} + \begin{bmatrix} \delta_1(t) \\ \delta_2(t) \end{bmatrix} \\ y(t) &= x_1(t) \end{aligned} \tag{20}$$

i.e. only the x-coordinate is measurable. Here, $\|[u_1(t), u_2(t)]^T\| \leq 1.25$, $\|[\delta_1(t), \delta_2(t)]^T\| \leq 0.05$. The initial errors are taken as $\|x(0) - \hat{x}(0)\| = 0.25$, i.e., half of the square is removed in the x-direction. Note that the specification is not realizable when posing requirements in terms of the measured states, as it requires control in the unmeasurable y-direction. We use a filter $L = [0.375, 0]^T$, which produces Figure 2. Here, the red trajectory corresponds to the estimated system and the blue trajectory to the actual system.

The estimation errors are not shown due to space constraints, but these never exceed their initial values, due to using a locally superstable observer and instead decline to a level determined by the disturbance.

VI. Case study: Air management system of aircraft

This section uses a simplified and linearized model of an air management system (AMS) of an aircraft as a test-case for the theory developed in sections II-V above.

A conventional AMS operates by admitting ambient air into the engines of an aircraft and forwarding this to a so called pressurization and air conditioning kit, where pressure is controlled by electrical compressors, temperature by a heat exchanger and possibly expansion cooling in a turbine, and finally humidity by a high pressure water extraction loop [14]. The AMS needs to be designed so as to supply sufficient pressure to the cabin at bearable temperature and humidity, preferably under comfortable conditions. It is also responsible for providing the cabin with its supply of fresh oxygen. Restrictions in the amount of power that can be supplied to the electronics and sensitivity of sensors to e.g, high temperature, exist; also, freezing of different parts of the craft pose operational problems. Lastly, the AMS should be fault tolerant as it is a critical part of the craft.

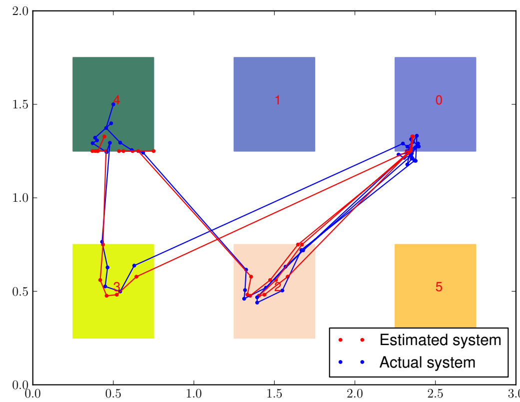

Figure 2. A trajectory for the simple example showing the trajectories for the estimated and actual systems as well as the domains wherein $\pi^{\varepsilon} \in \Pi$ evaluate to true.

Table I

THE SYMBOLS USED IN THE SIMPLIFIED AMS MODEL.

Symbol	Unit	Description
Measurable states		
T_x	° C	Temperature of metal in heat exchanger
T_c	° C	Temperature of cabin
Non-measurable state		
p_v	kPa	Outlet air pressure of valve 1
Controllable variables		
C_1		Valve coefficient for valve 1
C_2		Valve coefficient for valve 2
W_a	kg/s	Mass flow rate of cold inflow in HX
Switched variables		
T_a	K	Temperature of cold inflow in HX (ambient air)
T_e	K	Temperature of the air from the engine
Other derived variables		
W_i	kg/s	Incoming mass flow rate of the air from the engine
W_v	kg/s	Mass flow rate of the air that goes through valve 2
W_h	kg/s	Mass flow rate of the air that goes through the HX
T_h	K	Outlet air temperature of the HX
Constant variables		
p_e	kPa	Pressure of the air from the engine
p_c	kPa	Pressure of the cabin
W_f	kg/s	Mass flow rate passing through the fan

A simplified schematic of an AMS is included in Figure VI and the details of the model can be seen in the Appendix. The symbols used in this section are given in Table III with numerical values listed in Table II. The units of the parameters are suppressed in the text below. We introduce switching to the system by assuming the engine temperature to toggle uncontrollably between $T_e = 207$ and $T_e = 25$. Also, the airplane can switch its dynamics by having ambient air at either $T_a = -39$ or heated to $T_a = 161$. The uncontrollable switches are assumed to have a time scale larger than the sampling time of the controller, and is here set to $0.5sec$.

Table II
NUMERICAL VALUES USED IN THE SIMPLIFIED AMS MODEL

Symbol	Value	Description
Measurable states		
T_x	297.2 K	Equilibrium value
T_c	268 K	Equilibrium value
Non-measurable state		
p_v	136791 Pa	Equilibrium value
Controllable variables		
C_1	0.155	Equilibrium value
C_2	0.18	Equilibrium value
W_a	2.49 kg/s	Equilibrium value
Switched variables		
T_a	-39, 161 ° C	Arbitrary value
T_e	207, 25 ° C	Arbitrary value
Constant variables		
p_e	275.790 kPa	Arbitrary value
p_c	101.325 kPa	Arbitrary value

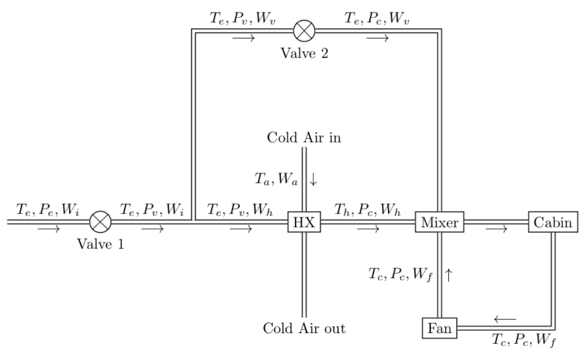

Figure 3. A simplified AMS.

We consider the state space $X = \{[T_c, T_x, p_v]^T \in \mathbb{R}^3 : 13 \leq T_c \leq 33, -25 \leq T_x \leq 15, 101.325 \leq p_v \leq 275.790\}$. Due to the non-linear dynamics of the system, the model results in a set of piecewise affine and linearized dynamics, with three different regions of definition, for every choice of the two environmental and controllable switching modes. These are determined by $R_1 = \{[T_c, T_x, p_v]^T \in X : 101.325 \leq p_v \leq 137.895\}$, $R_2 = \{[T_c, T_x, p_v]^T \in X : 137.895 \leq p_v \leq 202.65\}$ and $R_3 = \{[T_c, T_x, p_v]^T \in X : 202.65 \leq p_v \leq 275.790\}$, respectively. The control inputs are given by $U = \{[C_1, C_2, W_a]^T \subseteq \mathbb{R}^3 : 0 \leq C_1, C_2 \leq 1, 0 \leq W_a \leq 8.316\}$. Lastly, in order to obtain interesting results with limited hardware, the B-matrices obtained are amplified by a factor of 7.5. In all, this gives a discrete-time switched piecewise affine system with dynamics of the form

$$\begin{aligned} \begin{bmatrix} T_c(t+\Delta t) \\ T_x(t+\Delta t) \\ p_v(t+\Delta t) \end{bmatrix} &= A_k \begin{bmatrix} T_c(t) \\ T_x(t) \\ p_v(t) \end{bmatrix} + B_k \begin{bmatrix} C_1(t) \\ C_2(t) \\ W_a(t) \end{bmatrix} + F_k + E_k \delta(t) \\ y(t) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} T_c(t) \\ T_x(t) \\ p_v(t) \end{bmatrix}, \end{aligned} \tag{21}$$

for $k = 1, 2, 3$. We use a sampling time $\Delta t = 0.1 sec$.

A. Specifications

We assume the cabin crew to be able to set the reference values of T_c to hot ($T_c \in I_1 = [23.5, 25]$), cold ($T_c \in I_2 = [21, 22.5]$) or intermediate ($T_c \in I_3 = [22.5, 23.5]$). Cabin crew input is treated as the environment $\mathcal{E}$. The system should eventually reach the reference levels and stay within these levels until told otherwise and we require the environment to not change the reference value until the reference interval has been reached. Also, the cabin temperature should always stay within the temperature range $T_c \in [21, 25]$. Lastly, we require to always have non-freezing heat exchanger temperature in order to prevent freezing. In order to phrase this in LTL, we represent the cabin crew reference value by a level variable $l \in \{1, 2, 3\}$ which corresponds to when the reference value is hot, cold and intermediate, respectively. We also introduce a timer $t \in \{0, 1, \ldots, 5\}$ and require the reference values to be constant when $t \neq 5$ in order to increase the time scale of the reference value change. The specifications then become:

$$\begin{aligned} &\varphi_e \to \varphi_s, \\ &\varphi_e = \left(\bigwedge_{i=1}^{3} \Box \left((l = i \wedge T_c \in I_i) \to \bigcirc(l) = i \right) \right) \wedge \Box \left((t \neq 5) \to (\bigcirc(t) = t + 1) \right) \wedge \\ &\quad \Box \left((t = 5) \to (\bigcirc(t) = 0) \right) \wedge \Box \left((t \neq 5) \to (\bigcirc(l) = l) \right), \\ &\varphi_s = \left(\bigwedge_{i=1}^{3} \Box (l = i \to \Diamond T_c \in I_i) \right) \wedge \left(\bigwedge_{i=1}^{3} \Box ((l = i \wedge T_c \in I_i) \to \bigcirc(T_c) \in I_i) \right) \wedge \Box\Diamond(T_x \geq 0). \end{aligned} \tag{22}$$

The controllers were synthesized using the Temporal Logic Planning (TuLiP) Toolbox [24], which is a software package designed for temporal logic motion planning interfacing with JTLV [17].

B. Simulation

A sample simulation is included in Figures 4-6 below, where the initial error in T_c and T_x were 0.0175 and the initial error in p_v was 1.75. The disturbance term was bounded by 0.016 and 1.6 for T_c, T_x and p_v, respectively. For these values and the numerical values of the system matrices, Propositions 2 and 3 can be seen to guarantee existence of a locally superstable observer. In the figures, note the reference following of the cabin temperature and that T_c and p_v always remain within the state space. The error magnitudes never exceed their initial values,

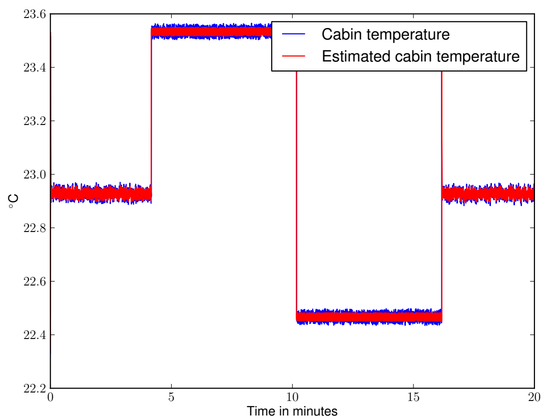

Figure 4. Cabin temperature for the sample AMS simulation

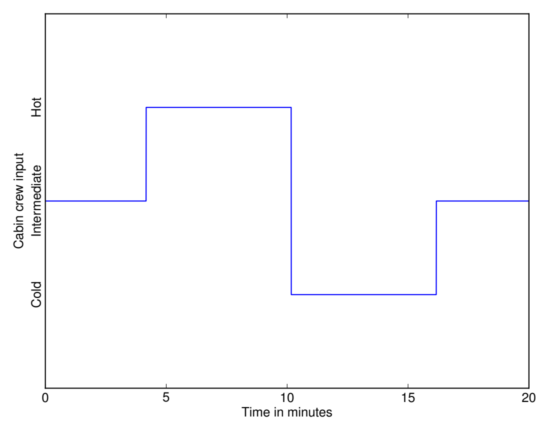

Figure 5. Cabin temperature reference intervals for the sample AMS simulation

due to local superstability and reduce to the magnitude of the disturbance term during the simulation. Numerically, $\|T_c(t) - \hat{T}_c(t)\| \leq 0.016$, $\|T_x(t) - \hat{T}_x(t)\| \leq 0.016$, $\|p_v(t) - \hat{p}_v(t)\| \leq 1.6$ for all times. Note that T_c has an error term due to the effect of the disturbance for which the observer cannot compensate between a time step and the next.

VII. Conclusions and future research

In this paper, we described a framework for synthesizing correct-by-construction control protocols for discrete-time piecewise-affine systems using only partial state information. The main insight of the proposed approach is to resolve the uncertainty in the continuous state at the continuous level of a hierarchical controller so that it is

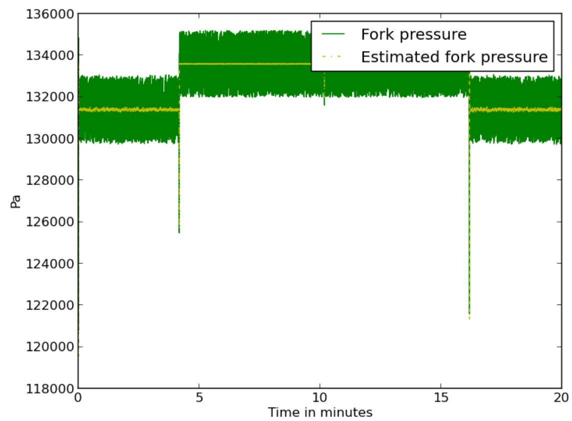

Figure 6. Pipe fork pressure for the sample AMS simulation

possible to solve a full information problem at discrete level. Ideas from robust estimation were used to design appropriate local observers that are synergistically integrated with the controller stack to achieve global bounds on the estimation errors. The approach was demonstrated on a case-study in the form of an air management system of aircraft.

Future research will consider employing nonlinear or higher-order observers within the proposed framework. Another interesting direction is to investigate whether the relation between the dynamics of a system and the corresponding bounded-error observer can be characterized in terms of alternating approximate simulation relations [20].

ACKNOWLEDGMENT

This work was supported in part by IBM and UTC through the iCyPhy consortium. The authors wish to thank the iCyPhy team, in particular Yilin Mo from Caltech and Jeff Ernst from UTAS, for helpful discussions on the AMS example and modelling.

REFERENCES

[1] C. Baier, J.-P. Katoen, et al. *Principles of model checking*, volume 26202649. MIT press Cambridge, 2008.

[2] F. Blanchini and M. Sznaier. A convex optimization approach to synthesizing bounded complexity ℓ^∞ filters. *Automatic Control, IEEE Trans. on*, 57(1):216–221, 2012.

[3] K. Chatterjee, L. Doyen, T.A. Henzinger, and J.-F. Raskin. Algorithms for omega-regular games with imperfect information. *Logical Methods in Computer Science*, 3(4):1–23, 2007.

[4] J. Chen and C.M. Lagoa. Observer design for a class of switched systems. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 2945–2950. IEEE, 2005.

[5] E.M. Clarke, O. Grumberg, and D.A. Peled. *Model checking*. MIT press, 1999.

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

[7] A. Girard and G.J. Pappas. Hierarchical control system design using approximate simulation. *Automatica*, 45(2):566–571, 2009.

[8] M.R. Hafner and D. Del Vecchio. Computational tools for the safety control of a class of piecewise continuous systems with imperfect information on a partial order. *SIAM Journal on Control and Optimization*, 49(6):2463–2493, 2011.

[9] Frank P Incropera, Adrienne S Lavine, and David P DeWitt. *Fundamentals of heat and mass transfer*, pages 676–680. John Wiley & Sons Incorporated, 2011.

[10] L. Jun, N. Ozay, U. Topcu, and R.M. Murray. Synthesis of reactive switching protocols from temporal logic specifications. *Automatic Control, IEEE Trans. on*, 58(7):1771–1785, 2013.

[11] M. Kloetzer and C. Belta. A fully automated framework for control of linear systems from temporal logic specifications. *Automatic Control, IEEE Trans. on*, 53(1):287–297, 2008.

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

[13] M. Milanese and A. Vicino. Optimal estimation theory for dynamic systems with set membership uncertainty: an overview. *Automatica*, 27(6):997–1009, 1991.

[14] I. Moir and A. Seabridge. *Aircraft systems: mechanical, electrical and avionics subsystems integration*, volume 21, pages 1–51,259–296. Wiley. com, 2008.

[15] G.E. Monahan. State of the art–a survey of partially observable markov decision processes: Theory, models, and algorithms. *Management Science*, 28(1):1–16, 1982.

[16] P. Nilsson, N. Ozay, U. Topcu, and R.M. Murray. Temporal logic control of switched affine systems with an application in fuel balancing. In *American Control Conference (ACC), 2012*, pages 5302–5309. IEEE, 2012.

[17] A. Pnueli, Y. Sa'ar, and L.D. Zuck. Jtlv: A framework for developing verification algorithms. In *Computer Aided Verification*, pages 171–174. Springer, 2010.

[18] J.S. Shamma and K.-Y. Tu. Set-valued observers and optimal disturbance rejection. *Automatic Control, IEEE Trans. on*, 44(2):253–264, 1999.

[19] Lan Shang, Guangjun Liu, and Peter Hodal. Development of high performance aircraft bleed air temperature control system with reduced ram air usage. *Control Systems Technology, IEEE Transactions on*, 18(2):438–445, 2010.

[20] P. Tabuada. *Verification and control of hybrid systems: a symbolic approach*. Springer-Verlag New York Inc, 2009.

[21] Rev. 0-2007 The international society of Automation Std. 75.01. Flow equations for sizing control valves. Accessed: 2013-08-20.

[22] Y Tu and GP Lin. Dynamic simulation of aircraft environmental control system based on flowmaster. *Journal of Aircraft*, 48(6):2031–2041, 2011.

[23] T. Wongpiromsarn, U. Topcu, and R.M. Murray. Synthesis of control protocols for autonomous systems. *Unmanned Systems*, pages 1–19, 2013.

[24] T. Wongpiromsarn, U. Topcu, N. Ozay, H. Xu, and R.M. Murray. TuLiP: a software toolbox for receding horizon temporal logic planning. In *Proceedings of the 14th international conference on Hybrid systems: computation and control*, pages 313–314. ACM, 2011.

Appendix A

Derivation of a Piecewise Linear Model of a Simplified Air Management System

A. Introduction

This appendix aims at constructing a simple yet realistic model of a complicated physical process, usable as a test case in control synthesis and formal methods. We define firstly a non-linear model defined on a set of polytopes and then linearize this. Lastly, we provide sample parameters to yield a numerical, piecewise affine system model.

A conventional AMS operates by admitting ambient air into the engines of the aircraft and forwarding this to a so called pressurization and air conditioning kit, where pressure is controlled by electrical compressors, temperature by a heat exchanger and possibly expansion cooling in a turbine, and finally humidity by a high pressure water extraction loop [14]. The AMS needs to be designed so as to supply sufficient pressure to the cabin at bearable temperature and humidity, preferably under comfortable conditions. It is also responsible for providing the cabin with its supply of fresh oxygen. Restrictions in the amount of power that can be supplied to the electronics and sensitivity of sensors to e.g, high temperature, exist; also, the AMS should be fault tolerant as it is a critical part of the craft.

A simplified schematic of an AMS is included in Figure VI. Engine bleed air enters the system and proceeds through a valve where it is divided into two flows, with one passing through a bypass valve and the other flow passing through a heat exchanger (HX). The leftmost valve has the function of controlling the flow-rate into the system, whereas the bypass valve directly influences the temperature by controlling the fraction of inflow that is heated in the HX. Lastly, the flows are recombined and enter the cabin.

B. Modeling of heat exchanger

A critical component of the AMS system in Figure VI is the heat exchanger, denoted by HX. In this section, we develop a simple dynamic model for this part.

A heat exchanger operates by admitting cold and hot flows and passing heat between these. To model this, we let the inputs be a hot flow with temperature T_e, pressure p_v, mass flow W_h and a cold flow with temperature T_a and mass flow W_a, and the output hot flow be T_h, pressure p_c, mass flow W_h. Assume a metal with temperature T_x dividing the two flows, with flux Q_h and Q_a entering the metal from the cold and hot sides, respectively. Newton's cooling law [19] gives

$$Q_h = h_h A_h (T_e - T_x) \tag{23}$$

$$Q_a = h_a A_a (T_x - T_a), \tag{24}$$

where the h are heat transfer coefficients and A areas. With C_{air} as the specific heat capacity of air, this results in

$$\dot{T}_x = \frac{1}{M_x C_{metal}} \left(h_h A_h (T_e - T_x) - h_a A_a (T_x - T_a) \right) \tag{25}$$

$$W_h T_e - W_h T_h = \frac{h_h A_h}{C_{air}} (T_e - T_x), \tag{26}$$

where M_x is the mass of the heat exchanger and C_{metal} is the specific capacity of the metal in the heat exchanger.

Note that heat transfer coefficients are functions of the corresponding flow rate, i.e. $h_a = h_a(T_a, W_a), h_h = h_h(T_e, W_h)$. A polynomial of the form $h_i = a_i W_i + b_i + c_i T_i$ for $i \in \{a, h\}$ is extrapolated from existing data.

1) Non-linear model: Next, the dynamical and algebraic equations governing the behaviour of the simplified AMS in Figure VI are summarized. The symbols used in this section are given in Table III. The following equations govern the simplified AMS:

1) Mass flow rate equation [21] for valve 1:

$$W_i = \begin{cases} 4.72 \times 10^{-4} \times C_1 \left(p_e + 2p_v\right) \sqrt{\frac{1}{T_e}\left(1 - \frac{p_v}{p_e}\right)} & p_v > 0.5 p_e \\ 6.67 \times 10^{-4} \times C_1 p_e \sqrt{\frac{1}{T_e}} & p_v \leq 0.5 p_e \end{cases} \tag{27}$$

Table III

THE SYMBOLS USED IN THE AMS MODEL.

Symbol	Unit	Description
Measurable states		
T_x	K	Temperature of metal in heat exchanger
T_c	K	Temperature of cabin
Non-measurable state		
p_v	kPa	Outlet air pressure of valve 1
Controllable variables		
C_1		Valve coefficient for valve 1
C_2		Valve coefficient for valve 2
W_a	kg/s	Mass flow rate of cold inflow in HX
Switched variables		
T_a	K	Temperature of cold inflow in HX (ambient air)
T_e	K	Temperature of the air from the engine
Other derived variables		
W_i	kg/s	Incoming mass flow rate of the air from the engine
W_v	kg/s	Mass flow rate of the air that goes through valve 2
W_h	kg/s	Mass flow rate of the air that goes through the HX
T_h	K	Outlet air temperature of the HX
$h_h(W_h)$	W/m^{2}K	Heat transfer coefficient of hot side of HX
$h_a(W_a)$	W/m^{2}K	Heat transfer coefficient of cold side of HX
Constant variables		
p_e	kPa	Pressure of the air from the engine
p_c	kPa	Pressure of the cabin
W_f	kg/s	Mass flow rate passing through the fan
$Q_{passenger}$	W	Heat flux generated by passengers in the cabin
ΔQ	W	Heat flux transferred from the environment to the cabin
M	g/mol	Molar mass of air (28.97)
R	J/(mol · K)	Ideal gas constant(8.31)
C_{air}	J/kgK	Specific heat capacity of air
C_{metal}	J/kgK	Specific heat capacity of metal in HX
M_x	kg	Mass of metal in the heat exchanger
V_{fork}	m^3	volume of the fork
V_c	m^3	Volume of the cabin
A_{HX}	m^2	Cross-sectional area of HX
A_h	m^2	Surface area of air/metal interface on hot side of HX
A_a	m^2	Surface area of air/metal interface on cold side of HX

2) Mass flow rate equation for valve 2:

$$W_v = \begin{cases} 4.72 \times 10^{-4} \times C_2 \left(p_v + 2p_c\right) \sqrt{\frac{1}{T_e}\left(1 - \frac{p_c}{p_v}\right)} & p_c > 0.5p_v \\ 6.67 \times 10^{-4} \times C_2 p_v \sqrt{\frac{1}{T_e}} & p_c \leq 0.5p_v \end{cases}. \tag{28}$$

3) Equations for the fork [22] (i.e., the point where the pipe splits into two):

$$p_v - p_c = \frac{K}{2\rho_{air} A_{HX}^2} W_h^2 \tag{29}$$

$$\dot{p}_v = \frac{RT_e}{MV_{fork}} (W_i - W_v - W_h), \tag{30}$$

where A_{HX} is the cross-sectional area of the heat exchanger, K a constant and ρ_{air} the density of air, which is given by

$$\rho_{air} = \frac{M(p_v + p_c)}{R(T_e + T_h)}.$$

M is the molar mass of air and V_{fork} the volume of the air in the fork.

4) Equations for the heat exchanger:

$$\dot{T}_x = \frac{1}{M_x C_{metal}} \left(h_h A_h (T_e - T_x) - h_a A_a (T_x - T_a) \right) \tag{31}$$

$$W_h T_e - W_h T_h = \frac{h_h A_h}{C_{air}} (T_e - T_x). \tag{32}$$

5) Equation for the cabin:

$$\frac{M p_c V_c}{R} \frac{\dot{T}_c}{T_c} = (T_e - T_c) W_v + (T_h - T_c) W_h + \frac{Q_{passenger}}{C_{air}} + \frac{\Delta Q}{C_{air}}, \tag{33}$$

with $Q_{passenger}$ as the heat flux from passengers, ΔQ the heat flux from sunlight et.c. in the cabin.

Combining all of these equations results in a piecewise affine system, with dynamics dependent on p_v. These can be listed as

$$\dot{p}_v = \frac{RT_e}{MV_{fork}} \left[W_i - W_v - W_h \right] \tag{34}$$

$$\dot{T}_c = \frac{RT_c}{Mp_c} \left[T_e \left(W_v + W_h - \frac{h_h A_h}{C_{air}} \right. - (W_h + W_v) T_c + \frac{h_h A_h T_x + Q_{pass} + \Delta Q}{C_{air}} \right] \tag{35}$$

$$\dot{T}_x = \frac{1}{M_x C_{metal}} \left[h_h A_h (T_e - T_x) - h_a A_a (T_x - T_a) \right] \tag{36}$$

2) Linearized model: This section computes a linearized version of the non-linear equations above, which is usable for demonstrating techniques for systems on the standard state-space form. The equations given in (34), (35), (36) in the last section are piecewise smooth in p_v, with four regions given by

$$R_1 : \begin{cases} p_v \leq 0.5 p_e \\ p_c \leq 0.5 p_v \end{cases}, \quad R_2 : \begin{cases} p_v > 0.5 p_e \\ p_c \leq 0.5 p_v \end{cases}, \quad R_3 : \begin{cases} p_v \leq 0.5 p_e \\ p_c > 0.5 p_v \end{cases}, \quad R_4 : \begin{cases} p_v > 0.5 p_e \\ p_c > 0.5 p_v \end{cases}$$

In order to transform these into a manageable linear model on the standard state-space form, we make the following assumptions.

- The density of air ρ_{air} is constant in the heat exchanger. See for instance [9] for density values in the relevant ranges.
- The heat transfer coefficients are affine in their respective flow rates and temperatures, i.e., $h_h = a_h W_i + b_h + c_h T_e$ and $h_a = a_a W_a + b_a + c_a T_a$.

Using this, we linearize the system (34)-(36) around equilibrium points in each region. Denoting equilibrium values with stars, this gives four different sets of dynamics valid on different polytopes in the phase space.

$$
\begin{aligned}
&R_1 : \\
\dot{p}_v &= \frac{RT_e}{MV_{fork}} k_1 c_1 - \frac{k_2 p_v^* RT_e}{MV_{fork}} c_2 - p_v \frac{RT_e}{MV_{fork}} \left(k_2 c_2^* + \frac{\sqrt{k_3}}{2\sqrt{p_v^* - p_c}} \right) \\
\dot{T}_c \frac{Mp_cV_c}{R} &= T_c \left[(T_e - 2T_c^*) \left(\sqrt{k_3(p_v^* - p_c)} + k_2 c_2^* p_v^* \right) + \frac{h_h A_h (T_x^* - T_e) + Q_{pass} + \Delta Q}{C_{air}} \right] + \\
&+ \frac{a_h A_h T_c^* (T_x^* - T_e) k_1 c_1}{C_{air}} + c_2 k_2 p_v^* T_c^* (T_e - T_c^*) + T_x \frac{h_h A_h T_c^*}{c_{air}} + \\
&+ p_v (T_e - T_c^*) \left(\frac{T_c^* \sqrt{k_3}}{2\sqrt{p_v^* - p_c}} + k_2 c_2^* \right) \\
\dot{T}_x &= \frac{1}{M_x C_{metal}} \left(-T_x (h_h^* A_h + h_a^* A_a) + T_e a_h A_h k_1 c_1 + T_a A_a a_a W_a \right)
\end{aligned}
$$

$$
\begin{aligned}
&R_2 : \\
\dot{p}_v &= \frac{RT_e}{MV_{fork}} k_4 c_1 (p_e + 2p_v^*) \sqrt{1 - \frac{p_v^*}{p_e}} - \frac{RT_e}{MV_{fork}} k_2 c_2 p_v^* \\
&+ \frac{RT_e}{MV_{fork}} \left(k_4 c_1^* \left[2\sqrt{1 - \frac{p_v^*}{p_e}} - \frac{p_e + 2p_v^*}{p_e} \frac{1}{2\sqrt{1 - \frac{p_v^*}{p_e}}} \right] - \frac{\sqrt{k_3}}{2\sqrt{(p_v^* - p_c)}} - k_2 c_2^* \right) p_v \\
\dot{T}_c \frac{Mp_cV_c}{R} &= T_c \left[(T_e - 2T_c^*) \left(\sqrt{k_3(p_v^* - p_c)} + k_2 c_2^* p_v^* \right) + \frac{h_h A_h (T_x^* - T_e) + Q_{pass} + \Delta Q}{C_{air}} \right] + \\
&+ \frac{a_h A_h T_c^* (T_x^* - T_e)}{C_{air}} k_4 c_1 (p_e + 2p_v^*) \sqrt{1 - \frac{p_v^*}{p_e}} + c_2 k_2 p_v^* T_c^* (T_e - T_c^*) + \\
&+ T_x \frac{h_h A_h T_c^*}{c_{air}} + p_v ((T_e - T_c^*) \left(\frac{T_c^* \sqrt{k_3}}{2\sqrt{p_v^* - p_c}} + k_2 c_2^* \right) + \\
&+ \frac{k_4 c_1^* a_h A_h T_c^* (T_x^* - T_e)}{C_{air}} \left[2\sqrt{1 - \frac{p_v^*}{p_e}} - \frac{p_e + 2p_v^*}{p_e} \frac{1}{2\sqrt{1 - \frac{p_v^*}{p_e}}} \right]) \\
\dot{T}_x &= \frac{1}{M_x C_{metal}} (-T_x (h_h^* A_h + h_a^* A_a) + T_e a_h A_h [k_4 c_1^* \left(2\sqrt{1 - \frac{p_v^*}{p_e}} - \frac{p_e + 2p_v^*}{2p_e \sqrt{1 - \frac{p_v^*}{p_e}}} \right) p_v + \\
&+ k_4 (p_e + 2p_v^*) \sqrt{1 - \frac{p_v^*}{p_e}} c_1] + T_a A_a a_a W_a)
\end{aligned}
$$

$R_3:$

$$\dot{p}_v = \frac{RT_e}{MV_{fork}}k_1c_1 - \frac{RT_e}{MV_{fork}}k_5(p_v^* + 2p_c)\sqrt{1-\frac{p_c}{p_v^*}}c_2 +$$
$$+\frac{RT_e}{MV_{fork}}\left(-k_5c_2^*\sqrt{1-\frac{p_c}{p_v^*}} - k_5c_2^*\frac{p_c(p_v^*+2p_c)}{p_v^{*2}2\sqrt{1-\frac{p_c}{p_v^*}}} - \frac{\sqrt{k_3}}{2\sqrt{p_v^*-p_c}}\right)p_v$$
$$\dot{T}_c\frac{Mp_cV_c}{R} = T_c\left[(T_e-2T_c^*)\left(k_5c_2^*(p_v^*+2p_c)\sqrt{1-\frac{p_c}{p_v^*}} + \sqrt{k_3(p_v^*-p_c)}\right) + \frac{h_hA_h(T_x^*-T_e)+Q_{pass}+\Delta Q}{C_{air}}\right] +$$
$$+\frac{a_hA_hT_c^*(T_x^*-T_e)k_1c_1}{C_{air}} + c_2\left[k_5T_c^*(p_v^*+2p_c)\sqrt{1-\frac{p_c}{p_v^*}}(T_e-T_c^*)\right] + T_x\frac{h_hA_hT_c^*}{c_{air}} +$$
$$+p_v\left[T_c^*(T_e-T_c^*)\left(k_5c_2^*\left[\sqrt{1-\frac{p_c}{p_v^*}} + \frac{(p_v^*+2p_c)p_c}{2p_v^{*2}\sqrt{1-\frac{p_c}{p_v^*}}}\right] + \frac{\sqrt{k_3}}{2\sqrt{p_v^*-p_c}}\right)\right]$$

$$\dot{T}_x = \frac{1}{M_xC_{metal}}\left(-T_x(h_h^*A_h + h_a^*A_a) + T_ea_hA_hk_1c_1 + T_aA_aa_aW_a\right)$$

$R_4:$

$$\dot{p}_v = \frac{RT_e}{MV_{fork}}k_4(p_e+2p_v^*)\sqrt{1-\frac{p_v^*}{p_e}}c_1 - \frac{RT_e}{MV_{fork}}k_5(p_v^*+2p_c)\sqrt{1-\frac{p_c}{p_v^*}}c_2 +$$
$$+\left(k_4c_1^*\left[2\sqrt{1-\frac{p_v^*}{p_e}} - \frac{p_e+2p_v^*}{2p_e\sqrt{1-\frac{p_v^*}{p_e}}}\right] - k_5c_2^*\sqrt{1-\frac{p_c}{p_v^*}} - k_5c_2^*p_c\frac{p_v^*+2p_c}{2p_v^{*2}\sqrt{1-\frac{p_c}{p_v^*}}} - \frac{\sqrt{k_3}}{2\sqrt{p_v^*-p_c}}\right)p_v$$
$$\dot{T}_c\frac{Mp_cV_c}{R} = T_c\left[(T_e-2T_c^*)\left(k_5c_2^*(p_v^*+2p_c)\sqrt{1-\frac{p_c}{p_v^*}} + \sqrt{k_3(p_v^*-p_c)}\right) + \frac{h_hA_h(T_x^*-T_e)+Q_{pass}+\Delta Q}{C_{air}}\right] +$$
$$+\frac{a_hA_hT_c^*(T_x^*-T_e)k_4(p_e+2p_v^*)}{C_{air}}\sqrt{1-\frac{p_v^*}{p_e}}c_1 + c_2\left[k_5T_c^*(p_v^*+2p_c)\sqrt{1-\frac{p_c}{p_v^*}}(T_e-T_c^*)\right] +$$
$$+p_v(T_c^*(T_e-T_c^*)\left(k_5c_2^*\left[\sqrt{1-\frac{p_c}{p_v^*}} + \frac{(p_v^*+2p_c)p_c}{2p_v^{*2}\sqrt{1-\frac{p_c}{p_v^*}}}\right] + \frac{\sqrt{k_3}}{2\sqrt{p_v^*-p_c}}\right) +$$
$$+\frac{k_4c_1^*a_hA_hT_c^*(T_x^*-T_e)}{C_{air}}\left[2\sqrt{1-\frac{p_v^*}{p_e}} - \frac{p_e+2p_v^*}{2p_e\sqrt{1-\frac{p_v^*}{p_e}}}\right]) + T_x\frac{h_hA_hT_c^*}{c_{air}}$$
$$\dot{T}_x = \frac{1}{M_xC_{metal}}(-T_x(h_h^*A_h+h_a^*A_a) + T_ea_hA_h[k_4c_1^*\left(2\sqrt{1-\frac{p_v^*}{p_e}} - \frac{p_e+2p_v^*}{2p_e\sqrt{1-\frac{p_v^*}{p_e}}}\right)p_v +$$
$$+k_4(p_e+2p_v^*)\sqrt{1-\frac{p_v^*}{p_e}}c_1] + T_aA_aa_aW_a)$$

where

$$k_1 = \frac{6.67 \cdot 10^{-4}}{\sqrt{T_e}}$$
$$k_2 = k_1$$
$$k_3 = \frac{2\rho A_{HX}^2}{K} \tag{37}$$
$$k_4 = \frac{4.72 \cdot 10^{-4}}{\sqrt{T_e}}$$
$$k_5 = k_4$$

C. *Numerical model*

In this section, we provide numerical values for all the parameters listed in Table III and list the resulting piecewise affine dynamics. The values are given in Table IV and should only be seen as example values giving interesting and non-trivial system behaviour. Tuning of the model can be performed by changing the parameter values and inserting into the piecewise affine equations in the last section.

If we let

$$x = \begin{pmatrix} T_c \\ T_x \\ p_v \end{pmatrix}, \quad u = \begin{pmatrix} c_1 \\ c_2 \\ W_a \end{pmatrix} \tag{38}$$

and $\Delta x = x - x^*$, $\Delta u = u - u^*$, equations (34)-(36) can be summarized as $\dot{x} = f(x, u)$ which linearizes to

$$\dot{x} \approx \frac{\partial f(x^*, u^*)}{\partial x}\Delta x + \frac{\partial f(x^*, u^*)}{\partial u}\Delta u + f(x^*, u^*) = Ax + Bu + F. \tag{39}$$

Because of the numerical values of p_e and p_c, R_1 is empty, giving a total of three regions with different dynamics. These are described by $R_2 = \{[T_c, T_x, p_v]^T \in \mathbb{R}^3 : 202.65 \le p_v \le 275.79\}, R_3 = \{[T_c, T_x, p_v]^T \in \mathbb{R}^3 : 101325 \le p_v \le 137895\}, R_4 = \{[T_c, T_x, p_v]^T \in \mathbb{R}^3 : 137895 \le p_v \le 202650\}$. Evaluating equation (39) in each of R_2, R_3, R_4 yields the data given in Table V.

Table IV

NUMERICAL VALUES FOR THE SYMBOLS USED IN THE LINEARIZED MODEL.

Symbol	Value	Description
Measurable states		
T_x	297.2 K	Equilibrium value
T_c	268 K	Equilibrium value
Non-measurable state		
p_v	136791 Pa	Equilibrium value
Controllable variables		
C_1	0.155	Calculated value
C_2	0.18	Calculated value
W_a	2.49 kg/s	Calculated value
Switched variables		
T_a	234 K	Arbitrary value
T_e	480 K	Arbitrary value
Constant variables		
p_e	275790 Pa	Arbitrary value
p_c	101325 Pa	Arbitrary value
$Q_{passenger}$	$90 \cdot 200$ W	Data provided by iCyPhy team
ΔQ	8792 W	Data provided by iCyPhy team
M	28.97 g/mol	Physical constant
R	8.31 J/(mol $\cdot$ K)	Physical constant
C_{air}	1003.5 J/kgK	Physical constant
C_{metal}	837 J/kgK	Data provided by iCyPhy team
M_x	13.61 kg	Data provided by iCyPhy team
V_{fork}	0.004916 m^3	Data provided by iCyPhy team
V_c	141.6 m^3	Data provided by iCyPhy team
A_{HX}	0.00161 m^2	Data provided by iCyPhy team
A_h	5 m^2	Arbitrary value
A_a	5 m^2	Arbitrary value
a_h	174.026 J/m^2kg K	Interpolated from lookup tables
b_h	8.8312 W/m^2 K	Interpolated from lookup tables
c_h	0.3109 W/m^2 K^2	Interpolated from lookup tables
a_a	52.7525 J/m^2kg K	Interpolated from lookup tables
b_a	46.7809 W/m^2 K	Interpolated from lookup tables
c_a	0.3677 W/m^2 K^2	Interpolated from lookup tables

Table V

TABLE SHOWING THE SYSTEM MATRICES A, B, F IN EACH OF THE REGIONS R_2, R_3, R_4 FOR THE PWA SYSTEM.

Region	A	B	F
R_2	$\begin{bmatrix} 1.005 & 0.0007144 & 0.002339 \\ 0 & 0.979 & 0.0005619 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 10.13 & 0.1892 & 0.000194 \\ 30.61 & -0.06435 & 0.5361 \\ 6.738 & -3.34 & 0 \end{bmatrix}$	$\begin{bmatrix} 0.000058 \\ -0.1591 \\ -2.483662 \end{bmatrix}$
R_3	$\begin{bmatrix} 1.005 & 0.0007144 & 0.002231 \\ 0 & 0.979 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 10.1 & 0.1787 & 0.000194 \\ 30.46 & 0 & 0.5361 \\ 5.21 & -2.311 & 0 \end{bmatrix}$	$\begin{bmatrix} -0.00237196 \\ -0.159101 \\ -1.530935 \end{bmatrix}$
R_4	$\begin{bmatrix} 1.005 & 0.0007144 & 0.002239 \\ 0 & 0.979 & 0.0004362 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 10.11 & 0.1779 & 0.000194 \\ 30.58 & -0.04497 & 0.5361 \\ 5.232 & -2.318 & 0 \end{bmatrix}$	$\begin{bmatrix} -0.00237196 \\ -0.159101 \\ -1.85701 \end{bmatrix}$