Approximate Verification of a Class of Adaptive Control Systems

Ravi Prasanth, Lingji Chen, Jovan Bošković and Raman Mehra

***Abstract*— We present a procedure for the approximate verification of a class of adaptive control systems. In the systems considered, the continuum state evolves according to a differential equation that is jointly polynomial in states and exogenous inputs. Our computational procedure for approximate verification consists of outer approximating reach sets and checking behavioral inclusion. The main tool for computing outer approximations is a result due to Handelman on the representation of polynomials that are positive in a polytope. It is a special case of Schmüdgen's theorem which is a central result in sum of squares (SOS) programming. Using the result, a linear program for outer approximation is derived. An extension to adaptive systems with state-dependent switching is described. Some difficulties with our approach are also discussed.**

I. Introduction

In the past several years it has been widely recognized that a crucial element in achieving the desired levels of autonomy in Unmanned Aerial Vehicles (UAV) is on-line adaptive identification and control. Real-time parameter adjustment is used for a variety of tasks, including on-line Failure Detection, Identification and Reconfiguration (FDIR), adaptive disturbance rejection, real-time trajectory generation, and on-line learning to improve the accuracy of the vehicle's models, particularly after failures, battle damage or other sudden changes in the flight regime. It has also been found that finding adaptive laws that will improve a vehicle's performance under a variety of external and internal perturbations is a highly challenging objective, and that the resulting adaptive controllers are highly complex, which makes their verification and validation a formidable task. It is believed that many features of adaptive control systems will necessitate the development of new formal modeling and verification tools. Affordable verification and validation (V&V) of intelligent and adaptive control systems is by far the most important challenge in the development and certification of unmanned air vehicles (UAVs) faced by both commercial and military aerospace industry [1].

This paper presents an approach to the verification of a class of adaptive systems. Almost all interesting properties of adaptive systems are not exactly verifiable in view of the results of [2], [3], [4]. Our interest is in approximate verification by which we mean an iterative process with the following guarantee: *If the iterations on the approximating system terminate, then the original adaptive system satisfies the requirements.* It follows that the verifier will never produce a false "error-free" certificate when the adaptive system contains errors, which is the most important requirement in V&Vof safety critical systems.

The authors are with Scientific Systems Company, Inc., 500 West Cummings Park, Suite 3000, Woburn, MA 01801. Email: prasanth,chen,jovan,rkm@ssci.com

The verification process consists of:

1) *Construction of an approximating automaton:* We extend the results of [5] to the class of adaptive systems considered in this paper, using sum of squares (SOS) programming techniques, to construct a finite state automaton whose behaviors contain those of the hybrid adaptive system and specification. The number of automaton states depends on the refinement level of the approximation of behaviors.
2) *Emptiness-checking:* The approximate verification problem is to check if the language accepted by the automaton is empty. This is done by depth-first search. The major twist is that we need to solve a SOS programming problem in order to determine if there is an edge between two states.
3) *Convergence and refinement:* If there is no path from initial states to unwanted states of the approximating automaton, then specifications are guaranteed to be met by the original hybrid system. Otherwise, we conclude that the approximation is too coarse and return to Step 1 for refinement.

Figure 1 shows the iterative process.

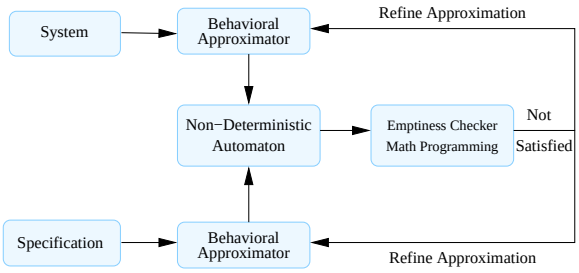

Fig. 1. Overall structure of verifier

The states of the approximating automaton are subsets of the state space of the closed loop adaptive system that are potential initial condition sets. The transition function of the approximating automaton computes an outer approximation of reach set originating from an automaton state and checks if it intersects with other automaton states. So, the underlying computational problem is that of finding good outer approximations of reach sets originating from a given set of initial conditions and evolving under the influence of exogenous inputs. This problem, for the class of systems considered, is equivalent to the problem of finding a polynomial that is positive in a polytope. We show that recent developments in sum of squares (SOS) programming [6], [7], [8], [9], [10], [11], [12], [13] can therefore be used to

derive good computational procedures. Positive polynomials have received great attention recently [7], [8], [9], [10], [13] due to the fact that they provide a general scheme to obtain convex approximations of intractable problems. The basic idea is to seek representations of positive polynomials as SOS of polynomials. We use a result due to Handelman [6] to derive a linear program for outer approximation. Handelman's theorem is a special case of Schmüdgen's theorem [11], which is a central result in SOS programming, obtained by considering polytopes.

Our approach to verification can be termed as behavior-based explicit model checking [14]. Many well-known model checkers for finite state machines, e.g. SPIN [15] adopt a similar approach. In fact, by incorporating SOS programming techniques into SPIN's repertoire of capabilities, we can use SPIN for implementing the verifier for adaptive systems shown in Figure 1. Thus, at some level, we are showing that existing tools can be modified to obtain approximate verifiers for certain classes of adaptive systems. Our use of SOS programming techniques is also motivated by its close connection to algebraic geometry and the possibility of symbolic computation [16], [17], [13]. So, a symbolic model checker of adaptive systems may be feasible bringing in the advantages observed in finite state machines [14], [18].

The paper is organized as follows. The next section describes the class of adaptive control systems and specifications considered in this paper. For simplicity, switching is not considered until Section V. Section III gives a linear program for outer approximation of reach sets using SOS programming techniques. An iterative procedure for approximate verification is given in Section IV and its extension to switching adaptive systems is discussed in Section V. Finally, conclusions and some deficiencies in our approach are presented in Section VI.

II. Statement of the Verification Problem

A verification problem is to check if a system model satisfies a specification. This section defines the class of systems and specifications considered.

A. Class of adaptive control systems

The adaptive system is composed of a plant, a reference model, a state observer, a feedback control law and an adaptive law. We describe these sub-systems and then write down a general form for the closed loop system. The plant is a linear time-invariant (LTI) system given by:

$$\dot{x} = Ax + B_u u + B_d d \tag{1}$$

where d is an external disturbance, u is the control input, $A \in \mathbb{R}^{n\times n}$ is unknown (but constant) and $B_u \in \mathbb{R}^{n\times n}$ is known. We *assume that B_u is invertible and that the disturbance input belongs to a set $\mathcal{D}$ of bounded $\mathcal{L}_2$ functions with norm less than* 1. The reference model is an LTI system:

$$\dot{x}_m = A_m x_m + B_m r \tag{2}$$

where $A_m \in \mathbb{R}^{n\times n}$ has all its eigenvalues in the open left half plane and r is a reference input. The reference input belongs to the set of bounded continuous $\mathcal{L}_2$ functions with norm less than 1. An observer for the plant state is chosen as:

$$\dot{\hat{x}} = \hat{A}x + B_u u - \lambda(\hat{x} - x) \tag{3}$$

where $\hat{x}$ is an estimate of x, $\hat{A}$ is an estimate of A and $\lambda > 0$. The adaptive law for adjusting $\hat{A}$ is given by:

$$\dot{\Phi}_A = -\Gamma_A \hat{e} x' \tag{4}$$

where $\Phi_A = \hat{A} - A$ and $\hat{e} = \hat{x} - x$ are the estimation errors in A and x respectively and, $\Gamma_A \in \mathbb{R}^{n\times n}$ is a strictly positive adaptive gain matrix.

The adaptive control problem is to find a state feedback law so that:

(a) the closed loop system is asymptotically stable in a neighborhood of the origin (when $d = 0$ and $r = 0$),
(b) the tracking error tends to zero asymptotically (with $d = 0$), i.e, $\lim_{t\to\infty}[x(t) - x_m(t)] = 0$ for all reference inputs and all initial conditions, and
(c) the plant states and control inputs are within specified bounds (that may depend on the reference input) for all disturbance inputs in $\mathcal{D}$ and all initial conditions.

To this end, we choose the control law:

$$u = B_u^{-1}(-\hat{A}x + A_m x + B_m r), \tag{5}$$

which has the form of an inverse dynamics law. We can combine the sub-system equations into the closed loop equations:

$$\begin{aligned} \dot{x} &= \Phi_A x + A_m x + B_m r + B_d d \\ \dot{\hat{e}} &= -\lambda e + \Phi_A x - B_d d \\ \dot{\Phi}_A &= -\Gamma_A \hat{e} x' - \delta \Phi_A \end{aligned} \tag{6}$$

whose right hand side is a vector-valued polynomial in $(x, \hat{e}, \Phi_A, r, d)$. It can be shown [19] that the closed loop system meets the control objectives (a) and (b) for any choice of the design variables $\lambda > 0$ and Γ_A. The adaptive controller design problem thus becomes a problem of finding λ and Γ_A so that the remaining specification is also met.

The adaptive control systems considered in this paper are such that their resulting closed loop systems have the form:

$$\dot{x} = f(x, w), \;\; x(0) = x_0 \in \mathcal{X}_0 \subset \mathbb{R}^{n_x} \tag{7}$$

where $x(t)$ is the state vector, $w(t)$ is the exogenous input vector consisting of disturbance and reference inputs, f is a vector-valued polynomial in $z = (x', w')'$, $\mathcal{X}_0$ is a polytope of initial conditions. We assume that w belongs to the set $\mathcal{BL}_2$ of functions with $\mathcal{L}_2$-norm less than 1. It is reasonable in practical applications to further suppose that the exogenous inputs are (essentially) bounded in addition to being in $\mathcal{BL}_2$.

B. Specification and the verification problem

We are primarily interested in the transient behavior of the adaptive control system. The size of transients, in both tracking error and parameter error, depends on the initial mis-match between the true and estimated parameters as well as the reference input. Further, in practical applications, there are external disturbances that enter the plant (1) additively or multiplicatively. These disturbances may also cause unacceptable transients in the closed loop system. Though transient behavior is a control objective, it is usually neglected in the design stage which considers asymptotic stability and tracking as done above. The design variables are adjusted through simulation-based analysis in the hope of bounding transient behavior within specified limits. This approach cannot guarantee that transient requirements are indeed satisfied.

Let $T > 0$ and $\{Q_k\}_{k=1}^K$ be polytopes in the closed loop state space. We say that the *kth transient specification is satisfied* if and only if all trajectories of the closed loop system emanating from initial conditions in $\mathcal{X}_0$ subject to exogenous inputs stays within Q_k during the time interval $[0, T]$. We refer to Q_k as the kth specification satisfaction set and its complement as the kth specification violation set. The verification problem is to check if the K transient specifications are satisfied.

III. Outer approximation of reach set

Fix $x_0 \in \mathcal{X}_0$ and $w \in \mathcal{BL}_2$. Let $\phi(\cdot; x_0, w)$ denote the solution of (7). For each $t \geq 0$, define:

$$\mathcal{R}_t = \left\{ \begin{array}{l} x \in \mathbb{R}^{n_x} : x = \phi(t; x_0, w) \text{ for some} \\ x_0 \in \mathcal{X}_0 \text{ and } w \in \mathcal{BL}_2 \end{array} \right\} \quad (8)$$

which is the set of all states that can be reached at time t starting from some initial condition in $\mathcal{X}_0$ and evolving under some exogenous input w in $\mathcal{BL}_2$. The set of all states that can be reached within time $T \geq 0$ (reachable set) is:

$$\mathcal{R} = \bigcup_{0 \leq t \leq T} \mathcal{R}_t \quad (9)$$

where, since $T \geq 0$ is fixed throughout this section, we do not explicitly show the dependence of reachable set on T. Our aim in this section is to derive a computational procedure to outer approximate $\mathcal{R}$.

For bounded inputs and finite time T, we can show that $\mathcal{R}$ is bounded using the fact that f is a polynomial and $\mathcal{X}_0$ is bounded. Furthermore, the specifications in V&V problems generally constrain the admissible set of states to some bounded set (so-called "safe set") containing the initial condition set. If $\mathcal{R}$ is unbounded, then the specifications are violated. So, the interesting V&V problems involve reach sets that are bounded. This is stated below as an assumption.

Assumption 3.1: Let $n = n_x + n_w$. There are known strictly positive real numbers $z_k^{\max}, k = 1, \cdots, n$ such that the polytope:

$$\mathcal{X} = \{x \in \mathbb{R}^{n_x} : -z_i^{\max} \leq x_i \leq z_i^{\max},\ i = 1, \cdots, n_x\}$$

contains the reach set and the safe set, and the polytope:

$$\mathcal{W} = \{w \in \mathbb{R}^{n_w} : -z_i^{\max} \leq w_i \leq z_i^{\max},\ i = n_x + 1, \cdots, n\}$$

is the set of values taken by exogenous inputs.

With later developments in mind, define the bounding linear polynomials:

$$\begin{aligned} \lambda_1(z) = z_1^{\max} - z_1,\ \lambda_2(z) = z_1^{\max} + z_1, \quad (10) \\ \lambda_3(z) = z_2^{\max} - z_2,\ \cdots,\ \lambda_{2n}(z) = z_n^{\max} + z_n, \end{aligned}$$

and the polytope:

$$\mathcal{Z} = \{z \in \mathbb{R}^n : \lambda_i(z) \geq 0 \text{ for } i = 1, \cdots, 2n\} \quad (11)$$

where $n = n_x + n_w$ is as in Assumption 3.1. It can be checked that $\mathcal{Z} = \mathcal{X} \times \mathcal{W}$.

We will also assume that:

Assumption 3.2: $f(x, w) = 0$ in $\mathcal{Z}$ if and only if $x = 0$ and $w = 0$

which guarantees that $x = 0$ is the only equilibrium point of the unforced $(w = 0)$ closed loop system (7) in $\mathcal{Z}$. For computational purposes, it is enough to assume that f has a finite number of zeros in $\mathcal{Z}$.

A. Background material and notations

We denote the set of scalar-valued polynomials in n variables by $\mathcal{P}_n$, and the set of scalar-valued polynomials in n variables of degree at most m by $\mathcal{P}_n^m$. The latter is a finite dimensional vector space, and we denote its dimension by m_n. .

Bases: We will use two different *ordered* collections to represent polynomials. The standard basis $\mathcal{B}_n^m$ for $\mathcal{P}_n^m$ consists of monomials with lex order:

$$\mathcal{B}_n^m = \{b_1, b_2, b_3, \cdots, b_{m_n}\} \quad (12)$$

where b_k's are monomials and $b_1 = 1$. The second collection of interest $\mathcal{H}_n^m$ consists of Handelman functions [6] which are defined relative to a polytope. Consider the bounding linear polynomials in (10) and the polytope $\mathcal{Z}$ defined in (11). The Handelman functions in n variables are of the form:

$$\lambda^\alpha = \lambda_1^{\alpha_1} \lambda_2^{\alpha_2} \cdots \lambda_k^{\alpha_k}$$

and may be ordered lexicographically as before with λ replacing x. The degree of a Handelman function is $|\alpha|$. Let

$$\mathcal{H}_n^m = \left\{h_1, h_2, \cdots, h_{\widehat{m}_n}\right\} \quad (13)$$

be lex ordered Handelman functions of degree at most m with $h_1 = 1$. Here, $\widehat{m}_n$ is the number of Handel functions in n variables of degree at most m. It is easy to see that any polynomial in $\mathcal{P}_n^m$ can be written as a linear combination of Handelman functions in $\mathcal{H}_n^m$, but $\mathcal{H}_n^m$ is not a basis.

Given a polynomial p in $\mathcal{P}_n^m$, we write:

$$\begin{aligned} p &= \sum_{i=1}^{m_n} b_i c_i^p = [\, b_1 \quad b_2 \quad \cdots \quad b_{m_n} \,] \, [\, c_1^p \quad c_2^p \quad \cdots \quad c_{m_n}^p \,]' \\ &\triangleq B_n^m p_b \quad (14) \end{aligned}$$

where $p_b \in \mathbb{R}^{m_n}$ is the coordinate of p in $\mathcal{B}_n^m$. Similarly, write:

$$p = [\, h_1 \quad h_2 \quad \cdots \quad h_{\widehat{m}_n} \,] \, [\, d_1^p \quad d_2^p \quad \cdots \quad d_{\widehat{m}_n}^p \,]' \triangleq H_n^m p_h \tag{15}$$

for the collection $\mathcal{H}_n^m$. This representation, however, is not unique. Define:

$$\mathcal{N}_h = \left\{ x \in \mathbb{R}^{\widehat{m}_n} : H_n^m x = 0 \right\} \tag{16}$$

which is the subspace of $\mathbb{R}^{\widehat{m}_n}$ that corresponds to the zero polynomial. The orthogonal complement (with respect to the standard inner product on $\mathbb{R}^{m_n}$) of $\mathcal{N}_h$ is denoted by $\mathcal{N}_h^\perp$. We have:

Proposition 3.1: Let $p \in \mathcal{P}_n^m$ be given. There exist unique elements $p_b \in \mathbb{R}^{m_n}$ and $p_h \in \mathcal{N}_h^\perp$ such that $p = B_n^m p_b$ and $p = H_n^m p_h$.

Throughout this paper, the linear invertible transformation that takes $p_b \in \mathbb{R}^{m_n}$ into $p_h \in \mathcal{N}_h^\perp$ is denoted by $\mathcal{T}_{h \leftarrow b}^{nm}$. Thus, p_b is the coordinate of p in the standard basis if and only if $\mathcal{T}_{h \leftarrow b}^{nm} p_b$ is such that $p = H_n^m \mathcal{T}_{h \leftarrow b}^{nm} p_b$.

Embedding: Let m and M be positive integers with $m < M$. Then, $\mathcal{P}_n^m$ can be seen as a subspace of $\mathcal{P}_n^M$. There is a permutation matrix $A_m^M \in \mathbb{R}^{M_n \times M_n}$ with the following property:

$$B_n^m p_b = B_n^M A_m^M \begin{bmatrix} p_b \\ 0 \end{bmatrix}$$

for all $p_b \in \mathbb{R}^{m_n}$. That is, p_b is the coordinate of p in the standard basis on $\mathcal{P}_n^m$ if and only if $A_m^M \begin{bmatrix} p_b \\ 0 \end{bmatrix}$ is the coordinate of p in the standard basis on $\mathcal{P}_n^M$.

The following is a slightly modified version of a result due to Handelman [6] which is a special case of a more recent theorem of Schmüdgen [11].

Theorem 3.1 (Handelman [6]): Let $p \in \mathcal{P}_n$ and $\mathcal{Z}$ be a polytope as defined in (10-11). p is strictly positive in $\mathcal{Z}$ if and only if there exist a positive integer $M \geq m$ and $p_h \in \mathbb{R}^{\widehat{M}_n}$ such that $p = H_n^M p_h$, the first entry of p_h is strictly positive and all the remaining entries are positive real numbers.

Handelman's theorem in its original form states that any polynomial which is strictly positive in $\mathcal{Z}$ can be expressed as a positive linear combination of Handelman functions. If p is strictly positive in $\mathcal{Z}$, then $\gamma_{\min} = \min_{z \in \mathcal{Z}} p(z)$ is strictly positive due to compactness of $\mathcal{Z}$, and the function $q = p - \frac{1}{2}\gamma_{\min}$ is also strictly positive in $\mathcal{Z}$. So, by the original form of Handelman's theorem, q can be expressed as $q = H_n^M q_h$, where $q_h \geq 0$. Define

$$p_h = q_h + \begin{bmatrix} \gamma_{\min}/2 \\ 0 \end{bmatrix}$$

and note that p_h satisfies all the requirements of Theorem 3.1. The reverse implication given in Theorem 3.1 follows from the fact that non-constant Handelman functions are strictly positive in the interior of $\mathcal{Z}$.

Handling positivity: Handelman functions are strictly positive in the interior of $\mathcal{Z}$. As a consequence, a positive linear combination of Handelman functions is zero at some point in the interior of $\mathcal{Z}$ if and only if it is zero in $\mathcal{Z}$. Since we are interested in non-constant positive functions in the next section, the span of Handelman functions is not sufficiently large for our purpose. Therefore, we augment $\mathcal{H}_n^m$ with the positive elements of $\mathcal{B}_n^m$. Let

$$\mathcal{B}_n^{m+} = \left\{ p_1, p_2, \cdots, p_{m_n^+} : p_k \in \mathcal{B}_n^m \text{ is positive } \right\} \tag{17}$$

be the lex ordered positive polynomials in $\mathcal{B}_n^m$ and

$$B_n^{m+} = [\, p_1 \quad p_2 \quad \cdots \quad p_{m_n^+} \,] \tag{18}$$

be the matrix obtained by stacking the elements. There is a full rank matrix E_m^+ whose entries are all zeros and ones, no row or column has more than one non-zero entry and, for any x in $\mathbb{R}^{m_n^+}$, we have:

$$B_n^{m+} x = B_n^m E_m^+ x \tag{19}$$

i.e, if x is the coordinate of a polynomial in the basis B_n^{m+}, then $E_m^+ x$ is the coordinate in B_n^m.

B. Reduction to a linear program

Suppose that $v \in \mathcal{P}_{n_x}$ satisfies:

$$w'w - \left(\frac{\partial v}{\partial x}(x) \right)' f(x,w) \geq 0 \tag{20}$$

for all $x \in \mathbb{R}^{n_x}$ and $w \in \mathbb{R}^{n_w}$. It is known that if $x \in \mathcal{R}$, then there exists $x_0 \in \mathcal{X}_0$ such that $v(x) - v(x_0) \leq 1$. Hence, the set

$$\mathcal{R}(v) = \{ x : v(x) - v(x_0) \leq 1 \ \text{ for some } x_0 \in \mathcal{X}_0 \} \tag{21}$$

is an outer approximation of $\mathcal{R}$. So, the problem of finding an outer approximation of $\mathcal{R}$ is equivalent to the problem of finding a polynomial that satisfies the positivity condition (20). We can improve upon the estimate given by $\mathcal{R}(v)$ as shown in the proposition below. Note that, in (20), we required the inequality to hold for all x and w. However, in order for the inclusion $\mathcal{R} \subset \mathcal{R}(v)$ to hold, we only need to enforce the inequality of all x in $\mathcal{R}$ and all $w \in \mathcal{W}$ where $\mathcal{W}$ is the polytope defined in Assumption 3.1.

Proposition 3.2: Let $V = \{v_i\}_{i=1}^N$ be a collection of polynomials in $\mathcal{P}_{n_x}$ with the property:

$$w'w - \left(\frac{\partial v_i}{\partial x}(x) \right)' f(x,w) \geq 0 \quad \text{for all } (x', w')' \in \mathcal{Z} \tag{22}$$

for $i = 1, 2, \cdots, N$. Define

$$\mathcal{R}_V = \bigcap_{i=1}^N \mathcal{R}(v_i) = \bigcap_{i=1}^N \left\{ \begin{array}{c} x : v_i(x) - v_i(x_0) \leq 1 \\ \text{for some } x_0 \in \mathcal{X}_0 \end{array} \right\} \tag{23}$$

Then, $\mathcal{R} \subset \mathcal{R}_V$.

That is, better outer approximations of the reach set can be obtained by taking intersections of sets of the form $\mathcal{R}(v)$. The problem lies in finding non-zero polynomials that satisfy the positivity constraint in (22). We consider this

problem in detail below by first examining a strict positivity case and then generalizing to the positivity case.

Let l be a positive integer. Then, for each $v \in \mathcal{P}^l_{n_x}$, the quantity on the left hand side of (20) is a polynomial in x and w whose degree is smaller than some positive integer m. Let $z = (x', w')$ and $v \in \mathcal{P}^l_{n_x}$ with representation $v = B^l_{n_x} v_b$ where v_b is the coordinate of v in the standard basis $\mathcal{B}^l_{n_x}$. After carrying out the partial differentiation and multiplications in (20), the left hand side of the inequality has the form:

$$\sum_{i=1}^{m_n} \mathcal{L}_i(v_b) b_i(z) = B^m_n(z) \mathcal{L}(v_b)$$

where $\mathcal{L}_i$ is a linear functional and $\mathcal{L} : \mathbb{R}^{l_{n_x}} \to \mathbb{R}^{m_n}$ is a linear operator. We are interested in the following:

Strict Positivity problem: *Check if*

$$\mathcal{S} = \left\{ v_b \in \mathbb{R}^{l_{n_x}} : \sum_{i=1}^{m_n} \mathcal{L}_i(v_b) b_i(z) > 0 \ \textit{for all } z \in \mathcal{Z} \right\}$$
$$= \left\{ v_b \in \mathbb{R}^{l_{n_x}} : B^m_n \mathcal{L}(v_b) \textit{ is strictly positive in } \mathcal{Z} \right\} \quad (24)$$

is empty. If not empty, then find an element $v_b \in \mathcal{S}$.

The main result of this section is the following:

Theorem 3.2 (Linear feasibility program for strict positivity): Recall the definitions of $\mathcal{N}_h$, $\mathcal{T}^{nm}_{h \leftarrow b}$ and A^M_m from Section III-A. The following statements are equivalent.

1) $\mathcal{S}$ is not empty.
2) There exist a positive integer $M \geq m$, a strictly positive real number ϵ, $v_b \in \mathbb{R}^{l_{n_x}}$ and $u \in \mathcal{N}_h$ such that

$$u + \mathcal{T}^{nM}_{h \leftarrow b} A^M_m \begin{bmatrix} \mathcal{L}(v_b) \\ 0 \end{bmatrix} \geq \begin{bmatrix} \epsilon \\ 0 \end{bmatrix}$$

The proof is omitted due to space limitations.

The definition of $\mathcal{S}$ given in (24) requires strict positivity in $\mathcal{Z}$. This is overly restrictive in many control problems. For example, candidate quadratic Lyapunov functions for linear systems lead to the vanishing of the right hand side of (20) at $z = (x, w) = 0$. In fact, Assumption 3.2 implies that the left hand side of (20) at $z = 0$ is zero for any v. Define:

$$\widehat{\mathcal{S}} = \left\{ v_b \in \mathbb{R}^{l_{n_x}} : B^m_n \mathcal{L}(v_b) \text{ is positive in } \mathcal{Z} \right\} \quad (25)$$

and note that $v_b = 0$ is in $\widehat{\mathcal{S}}$. We are interested in checking if $\widehat{\mathcal{S}}$ contains a coefficient that gives rise to a non-constant v.

Theorem 3.3 (Linear feasibility program for positivity): Suppose that there exists a positive integer $M \geq m$ such that the infimal value of the optimization problem:

$$\inf_{\epsilon, v_b, u, p_b} \ \epsilon$$
$$\text{subject to } \epsilon > 0, v_b \in \mathbb{R}^{l_{n_x}}, u \in \mathcal{N}_h, p_b \in \mathbb{R}^{m^+_n}, p_b > 0$$
$$\text{and } \ u + \mathcal{T}^{nM}_{h \leftarrow b} A^M_m \begin{bmatrix} \mathcal{L}(v_b) - E^+_m p_b \\ 0 \end{bmatrix} > \begin{bmatrix} \epsilon \\ 0 \end{bmatrix} \quad (26)$$

is zero. For each $\epsilon > 0$, let V^ϵ_b be the set of all v_b such that (ϵ, v_b, u, p_b) is feasible for some $u \in \mathcal{N}_h$ and $p_b > 0$. Let

$$v^0_b \in \bigcap_{\epsilon > 0} V^\epsilon_b$$

Then, $v^0_b \in \widehat{\mathcal{S}}$ and $v = B^l_{n_x} v^0_b$ solves the outer approximation problem.

The proof is omitted due to space limitations.

IV. Computational procedure for approximate verification

The satisfaction of specifications is checked as follows:

1) Fix positive integer l.
2) Fix positive integer M and solve the linear minimization problem in Theorem 3.3.
3) If the infimal value is not zero, then increase M and repeat the previous step.
4) If the infimal value is zero, then form v as in Theorem 3.3 to obtain an outer approximation $\mathcal{R}(v)$ for the reach set.
5) For each specification k, check if the intersection of the complement of specification satisfaction set Q_k and $\mathcal{R}(v)$ is empty.
6) If so, specifications are satisfied. Otherwise, refine outer approximation by increasing l and repeating the steps above.

Note that in Step 5, we use:

$$\mathcal{R}(v) \subset Q_k \Leftrightarrow (\mathcal{Z} \setminus Q_k) \bigcap \mathcal{R}(v) = \emptyset$$

Since $\mathcal{Z}$ and Q_k are both polytopes, $\mathcal{Z} \setminus Q_k$ is can be written as the union of a finite number of polytopes. So, the underlying computational problem in Step 5 is that of checking if the intersection of a polytope and a set of the form $\mathcal{R}(v)$ is empty. To formulate this problem, let $\mathcal{X}$ be a polytope $\mathcal{X} = \{x : a_i' x \leq b_i, i = 1, \cdots, L\}$ where a_i and b_i are given. Then, $\mathcal{X} \bigcap \mathcal{R}(v)$ is not empty if and only if the infimal value of

$$\inf_x a_i' x$$
$$\text{subject to } x \in \mathcal{R}(v)$$

is less than or equal to b_i for all i.

V. Extension to adaptive system with state-dependent switching

We describe how to apply the computational procedure presented in the previous sections to switching systems. Let $\{1, 2, \cdots, N\}$ be the set of discrete states. In the ith discrete state, the continuum part of the system evolves according to:

$$\dot{x} = f_i(x, w) \quad (27)$$

provided that the state $x(t)$ is not in any of the *autonomous jump sets* $\{\mathcal{A}_{i \to j}\}_{j=1}^N$. Here, f_i is a polynomial in (x, w) and $\mathcal{A}_{i \to j} \subset \mathbb{R}^{n_x}$. When $x(t)$ enters $\mathcal{A}_{i \to j}$ at some time t, then the system jumps instantaneously from the discrete state i to the discrete state j. In the process, the continuum state $x(t)$ is reset to a point in a *jump destination set*

$\mathcal{D}_{j \leftarrow i} \subset \mathbb{R}^{n_{x_j}}$ according to an *autonomous jump transition map* $\mathcal{J}_{j \leftarrow i} : \mathcal{A}_{i \rightarrow j} \rightarrow \mathcal{D}_{j \leftarrow i}$, that is,

$$x(t) = \mathcal{J}_{j \leftarrow i}\left(x(t)\right) \tag{28}$$

(without loss of generality, we take the destination set to be the image of the jump set under the jump transition map). See figure 2 and [20] for a complete description of these hybrid systems. A pair $(i, x(t))$, where i is the discrete state and $x(t)$ is the continuum state, will be referred to as the *hybrid state* of the system at time t. As a matter of convention, at a jump time t, the continuum part of the hybrid state at time t is taken to be the continuum state after jump transition. We assume that the jump and destination sets are convex sets, jump sets are pairwise disjoint, and the intersection between a jump set and a destination set is empty. These assumptions are needed for computational efficiency and to guarantee that any two consecutive jumps times are separated by a strictly positive time interval.

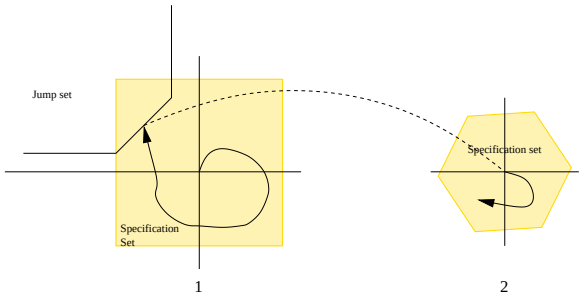

Fig. 2. A run of an example hybrid system with two discrete states and reset map equal to 0

In [5], a procedure to approximate a hybrid system with a finite state automaton is given. The approximation has the property that if the automaton satisfies the specifications, then so does the hybrid system. The approximate verification problem becomes a language-emptiness problem and can be solved using depth-first search [21], [22]. This procedure will never produce an "error-free" certificate when the actual system contains errors which is an important requirement in V&V of safety critical systems. The continuum dynamics considered in [5] is LTI, but the construction of automaton can be modified with the reach set computation procedures given in this paper. Essentially, the automaton states are initial condition set, specification sets, and partitions of jump destination sets, and the transition function is defined in terms of outer approximations of reach sets originating from the automaton states subject to exogenous inputs.

VI. Conclusions

This paper presented a computational procedure for the approximate verification of a class of adaptive control systems. The continuum dynamics of the adaptive systems considered are polynomial in both states and exogenous inputs. The transient specifications are given in terms of safe sets. Typical transient specifications that can be expressed in this manner are bounds on tracking error and parameter error, and control saturation limits. The computational procedure consists of outer approximating reach sets and checking behavioral inclusion. An extension to switching adaptive systems is also discussed.

There are two main computational difficulties with the outer approximation. The first difficulty stems from the use of Handelman functions to represent positive polynomials in a polytope. While we are able to give a necessary and sufficient condition for strict positivity in a polytope as a linear program, the number of terms required to represent a positive polynomial using Handelman functions can be very large. This means that very large linear programs will need to be solved reliably in practical verification problems. We have not been successful in finding an (useful) upper bound on the number of terms required. The second difficulty is in checking if the outer approximation $\mathcal{R}(v)$ is contained in the specification sets. It is due to the non-convexity of $\mathcal{R}(v)$. This difficulty can be alleviated by requiring that v be strictly positive (and, hence, a Lyapunov function).

References

[1] J. Buffington, et al, "Validation and verification of intelligent and adaptive control systems", *Proc. AIAA Unmanned Unlimited Conf.*, Sept., 2003.

[2] V. Blondel and J. Tsitsiklis, "A survey of computational complexity results in systems and control", *Automatica*, Vol. 36, No. 9, 2000.

[3] T. Henzinger, P. Kopke, A. Puri and P. Varaiya, "What's decidable about hybrid automata", *Proc. Ann. Sym. Theory Computing*, 1995.

[4] Special issue on hybrid systems, *Proc. of IEEE*, July, 2000.

[5] R. Prasanth, S. Bergstrom, J. Bošković and R. Mehra, "Verification of a class of hybrid systems using mathematical programming", *Proc. American Control Conf.*, 2003.

[6] D. Handelman "Representing polynomials by positive linear functions on compact convex polyhedra", *Pacific J. Math.*, 1988.

[7] J. Lasserre, http://www.laas.fr/ lasserre/

[8] P. Parillo and B. Sturmfels, "Minimizing polynomial functions", *DIMACS Series in Discrete Math. and Theo. Computer Sci.*, 2001

[9] P. Parillo, *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*, Ph.D Thesis, California Institute of Technology, 2000.

[10] V. Powers and B. Reznick, "Polynomials that are positive on an interval", preprint, 2000.

[11] K. Schmüdgen, "The k-moment problem for compact semi-algebraic sets", *Annals of Math.*, Vol. 289, pp.203-206, 1991.

[12] N. Shor, *Nondifferentiable optimization and polynomial problems*, Kluwer, 1998.

[13] B. Sturmfels, *Solving systems of polynomial equations*, AMS Lecture Notes, 2000.

[14] E. Clarke, O. Grumberg and D. Peled, *Model checking*, MIT Press, 2001.

[15] G. Holzmann, "The model checker SPIN", *IEEE Trans. on Softw. Eng.*, May 1997 (see also http://cm.bell-labs.com/cm/cs/who/gerard/)

[16] J. Bochnak, M. Coste and M. Roy, *Real algebraic geometry*, Springer, 1998.

[17] D. Cox, J. Little and D. O'Shea, *Using algebraic geometry*, Grad. Texts in Math., Vol. 185, Springer, New York, 1998.

[18] K. McMillan, *Symbolic model checking*, Kluwer, 1993.

[19] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1988.

[20] M. Branicky, V. Borkar and S. Mitter, "A unified framework for hybrid control: model and optimal control theory", *IEEE Trans. Auto. Con.*, January, 1998.

[21] J. Hopcroft and J. Ullman, *Introduction to automata theory, languages and computation*, Addison-Wesley, 1979.

[22] C. Papadimitriou and K. Steiglitz, *Combinatorial optimization*, Dover, 1998.