Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

Reachability Analysis of Hybrid Control Systems Using Reduced-Order Models

Zhi Han and Bruce Krogh

***Abstract*— Complexity of reachability computations for continuous and hybrid dynamic systems typically grows exponentially with respect to the dimension of the continuous state space. Consequently, reduced-order models usually need to be constructed to make reachability analysis tractable. Use of reduced-order models makes reachability-based verification unsound, however. This paper presents a method for incorporating bounds on errors due to model reduction into reachability analysis for a class of hybrid control systems so that the computed sets are guaranteed to be conservative (i.e., over-) approximations of the reachable sets for the original system. We also present an efficient method for computing error bounds due to model reduction for finite-time horizons that are less conservative than error bounds from the model-reduction literature. The effectiveness of the approach is illustrated with an example.**

I. INTRODUCTION

Reachability analysis is a major approach used in verification, controller synthesis and analysis of hybrid dynamical systems [17]. The main difficulty in reachability computation is the order of the system, i.e., the number of continuous state variables. As the order increases, the complexity of computation grows exponentially. Consequently, current verification tools based on reachability analysis are limited to systems with less than six to eight state variables.

To use existing verification tools for hybrid systems, one usually has to construct a reduced-order model of the continuous dynamics with which reachability analysis is performed. There are two problems with this approach. First, the reduced-order model is usually an approximation, so its trajectories do not match the trajectories of the original model exactly. This deviation, which is called the error of model reduction, is usually not accounted for in the verification procedure. Second, the reduced-order model is often tested using simulation to generate a finite set of trajectories. There is no guarantee that other trajectories are close approximations of the associated trajectories of the original system. These two problems make the verification result unsound because reachable sets computed using reduced-order models are not guaranteed to be conservative approximations of the reachable sets for the original system. Properties verified for the reduced-order model might be violated by the original model.

To address the above problems, we develop a reachability approximation procedure in this paper that accounts for the error introduced by model reduction. The key point of the procedure is the estimation of the upper bound of the error for all trajectories, which guarantees the soundness of verification results based reduced-order models.

Pappas and Tanner proposed the concept of *continuous abstractions* for LTI systems to construct reduced-order models that preserve reachability properties [16], [13]. Their approach differs from the model reduction method proposed here in two ways. First, the output trajectory of a continuous abstraction matches that of the full-order model exactly; a reduced-order model from standard model reduction algorithms usually introduces output error. Second, a continuous abstraction does not require the same input signal as the full-order model; the input to a standard reduced-order model is identical to the input for the original system. At this point continuous abstractions are more suitable for hierarchical modelling of linear control systems than for reachability analysis and verification [14].

The paper is organized as follows. Section II introduces the class of hybrid control systems considered in this paper, defines the reachability problem, and presents some results from the literature on projection-based model reduction algorithms. Section III presents our approach to reachability computations that account for the error introduced by model reduction. Details of the procedures for continuous-time and discrete-time hybrid control systems are given in section IV. In section V, we illustrate and evaluate the reachability procedure for a model of an electrical throttle control (ETC) system, a hybrid control system with seven continuous state variables. The concluding section summarizes the contributions of this paper.

II. BACKGROUND

A. Hybrid Control Systems and The Reach Set

We consider the class of hybrid control systems composed of a plant and a hybrid controller as shown in Fig. 1. The plant is a continuous-time linear time-invariant (LTI) system

$$\dot{x}(t) = Ax(t) + Bu(t),\ y = Cx(t)$$

or a discrete-time LTI system

$$x(t+1) = Ax(t) + Bu(t),\ y = Cx(t)$$

Following [19], we use the notation $S = \left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right]$ as a shorthand notation for an LTI system (the context will indicate whether it is a continuous- or discrete-time system).

A hybrid controller $C = (Q, q_0, T, f, G, I)$ consists of the following components:

Z. Han and B. Krogh are with Department of Electrical and Computer Engineering, Carnegie Mellon University, PA 15213 zhih—krogh@ece.cmu.edu

0-7803-8335-4/04/$17.00 ©2004 AACC

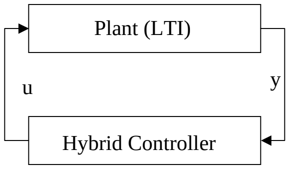


Fig. 1. A hybrid control system

- Q a finite set of control modes;
- q_0 an initial control mode;
- $G \in 2^Y$ a set of guard conditions over the plant output space $Y \subseteq R^m$;
- $I : q \rightarrow 2^Y$ a plant output invariant set associated with each control mode;
- $T : G \rightarrow (Q \rightarrow Q)$ a set of transitions associated with guard the conditions;
- $f : (Q \times Y) \rightarrow U$ a set of control laws $u = f(q, y)$.

We note that the controller in the above model implements an output feedback law and has only discrete-state dynamics. Controllers with internal LTI continuous dynamics can be converted into the above form if the continuous dynamics are the same in all control modes (e.g., if there is a state observer in the controller). In this case, the continuous dynamics of the controller can be incorporated into the plant model.

We assume the control state makes a transition any time a guard is enabled (i.e., we assume *urgent* semantics for the discrete state transitions in the controller). The output invariant set, $I(q)$, indicates *a priori* bounds on the value of the output signal when the controller is in mode q. We assume that for the reachable output values, if none of the guards are satisfied by the output $y(t)$ when the controller is in mode q, then $y \in I(q)$. (This condition is easily satisfied by making $I(q)$ the complement of the guards in mode q, but further knowledge of the reachable output values for the system may make it possible to make $I(q)$ smaller.) We use the invariants to construct the set of possible control values for each control mode. For each mode q, this set is given as

$$U(q) = \{u | u = f(q, y), y \in I_q\}.$$

These bounds on the plant control input will be used to estimate bound of the error in the computed reachable outputs for the closed-loop system. We use $\phi_u(\cdot)$ to denote a closed-loop trajectory of the control signal. Suppose at time t the controller is in mode q, then it is true that $\phi_u(t) \in U(q)$. We use $\mathcal{U}$ to denote the set of trajectories of closed-loop control signal, $\mathcal{U}$ is a subset of the admissible control set $\mathcal{U} \subseteq \{u(\cdot) | u(t) \in U(q)\ ,\ t \geq 0\}$.

For a hybrid control system, the reach set is typically defined in terms of the state variables of the plant. For the continuous-state reachability computations, we focus on the reachable set while the controller is in a particular mode. When the control mode switches, the reachability computations are resumed using the control input set defined for the new mode. In the following we focus on reachability computation for a given mode. Let X_0 denote the set of initial states of the state variables of plant when the controller is in mode q and let $\mathcal{U}$ denote the set of closed-loop control signals. In the procedure we proposed, we use existing tools to compute closed-loop reach set and use the bound of admissible control signal to estimate the error introduced by model reduction.

We define the reach set of the closed-loop control signals as follows.

Definition 1 (Reach set at a time instant): Given a continuous time LTI control system $S = \left[\begin{array}{c|c} \mathrm{A} & \mathrm{B} \\ \hline \mathrm{C} & 0 \end{array}\right]$, the reach set for a given set of controls $\mathcal{U}$ and initial set $X_0 \subset R^n$ is the set

$$\begin{aligned} Reach(S, \mathcal{U}, X_0, t) = \{x(t) | x(t) = \phi_x(t, x_0) \\ \equiv e^{At} x_0 + \int_0^t e^{A(t-\tau)} B \phi_u(\tau) d\tau, x_0 \in X_0, \phi_u(\cdot) \in \mathcal{U}\} \end{aligned}$$

. The output reach set is the set

$$\begin{aligned} Reach^o(S, \mathcal{U}, X_0, t) = \{y(t, x_0) | y(t, u, x_0) = Cx, \text{ where} \\ x \in Reach(S, \mathcal{U}, X_0, t\} \end{aligned}$$

· Reach set for discrete-time systems are defined in the similar way. In the rest of this paper, the *reachability problem* will refer to the output reachability problem. We define the reach set for a time interval as follows.

Definition 2 (Reach set for time interval): The reach set and output reach set for a time interval $[t_0, t_f]$ are defined as

$$Reach^{(o)}(S, \mathcal{U}, X_0, [t_0, t_f]) = \bigcup_{t \in \left[t_0, t_f\right]} Reach^{(o)}(S, \mathcal{U}, X_0, t).$$

This definition of the reach set for a time interval corresponds to the sets computed by reachability-based verification algorithms [2], [15], [3].

B. Model Reduction of LTI by Projection

Model reduction for linear time-invariant (LTI) systems has been studied since the 1970's [1], [11], [19], [12], [9]. Model reduction methods apply only to stable systems. For unstable systems, the system is decomposed into the stable and unstable part $S = S_{stable} + S_{unstable}$ and only the stable part is reduced. See [19], [1], [11] for further discussion on model reduction techniques. We introduce the formulation of projection and model reduction [1], [18]. It was pointed out that most of the widely-used model reduction methods can be formulated as projections [1].

A matrix $\pi \in R^{n \times n}$ is called a projection if $\pi = \pi^2$. If π is a projection, it can be written as $\pi = \pi_L \pi_R$, where $\pi_L \in R^{n \times m}$, $\pi_R \in R^{m \times n}$ and $\pi_R \pi_L = I_m$, where $m = rank(\pi) \leq n$. Projection-based model reduction methods construct the matrices for the reduced-order model as follows [1], [18]:

$$A_r = \pi_R A \pi_L,\ B_r = \pi_R B,\ C_r = C \pi_L$$

. For a state x in the original state space, the corresponding state in the projected state space is $x_r = \pi_R x$

In order to guarantee conservativeness, we need to determine a bound on the additive error between the original model and the reduced model, given by

$$e_r(t,u,x_0) = y(t,u,x_0) - y_r(t,u,\pi_R x_0).$$

The error trajectory is equivalent to the output trajectory of the following augmented system $S_{aug} = \left[\begin{array}{cc|c} A & & \mathrm{B} \\ & A_r & B_r \\ \hline C & -C_r & 0 \end{array}\right]$ for initial condition $\begin{bmatrix} x_0 \\ \pi_R x_0 \end{bmatrix}$ and input $u(\cdot)$.

The error bound computed from the augmented system is called *a posteriori* since it is computed after model reduction [8]. The *a priori* error bounds, which can be computed before creating the reduced-order model, have been derived for several model-reduction methods [5]. *a priori* error bounds can be computed from the solution of Lyapunov equations, which can be easily solved using existing efficient numerical packages. They are usually more conservative than *a posteriori* error bounds, however.

In this paper we apply one of the most commonly used projection-based model reduction methods, the balanced truncation [10], for which theoretical error bounds can be obtained [5].

*Theorem 1 (*A priori *error bound [11]):* Assume the original system is of order n and the reduced-order system of order k is obtained using balanced truncation. Let σ_i denote the i-th Hankel singular value of original system S. Then

$$\epsilon_r^{x_0=0}(\infty) \le \|u(\cdot)\|_\infty^{[0,\infty)}(4\sum_{i=r+1}^{n}(2i-1)\sigma_i) \qquad (1)$$

where i is the largest integer such that $\sigma_{i+1} \ge 2\sqrt{2}M_k$ and $M_k \equiv \sum_{i\ge 1}\sigma_{k+i}$

III. Reachability Approximation and Error Analysis

In this section we present a procedure to use reduced-order models to compute conservative approximation of the reach set for a hybrid control system. An outline of the procedure is shown in Fig. 2. Given an LTI plant and a hybrid controller, first a reduced-order model for the plant is created. Then it is composed with the controller to form a hybrid automaton [6], which is analyzed using a slight modification of the procedures in the verification tool CheckMate [2]. A bound on the error introduced by model reduction is computed, which is used by the reachability analysis routine to provide a conservative result. A sketch of the reachability procedure using the error bounds is shown in Fig. 3. We use S and S_r to denote the original system and the reduced-order system, respectively. First we compute the reach set of the reduced-order model using approximation algorithm reach_approx. Then the error of reduction is computed. The final step is to bloat the reach set of the reduced-order model to compensate for the error introduced by model reduction. Details of the procedures are given in the next section.

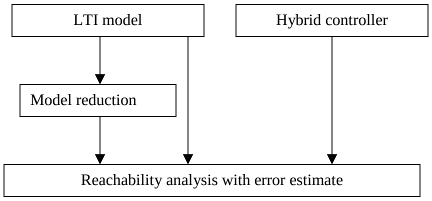


Fig. 2. Reachability analysis using reduced-order models.

Algorithms for reachability analysis compute reach sets in terms of state variables [2], [15], [7], [3]. In this paper, we consider the reach sets in terms of the output variables. They can be easily computed using the linear transformation matrix $C : R^n \rightarrow R^m$. The conservativeness of the output reach set is guaranteed by underlying algorithms, that is, $Reach^o(S,U,X_0) \subseteq Reach^o_{comp}(S,U,X_0)$ where $Reach^o_{comp}(S,U,X_0)$ denotes the computed result. Our objective is to use reduced-order models to compute these approximations. In order to be conservative, the reach sets are bloated to compensate for the error introduced by model reduction.

The result of bloating is a conservative approximation of the reach set of original system S. In the remainder of this section we focus on the subroutine error_estimate and discuss the other routines in the next section.

```
Reach($S$,$S_r$,U,$X_0$,$X_{r0}$)
        Reach = reach_approx($S_r$,U,$X_{r0}$);
        epsilon = error_estimate($S$,$S_r$,U, $X_0$, $X_{r0}$);
        R = bloat(Reach, epsilon);
        return(R)
```

Fig. 3. A reachability analysis procedure using reduced-order models

In the procedure in Fig. 3 we only compute the reach set for the reduced-order model, thereby avoiding the more expensive reachability algorithms for the full-order model. However, the routine $error\_estimate$ uses both the full-order and the reduced-order model. We show in the following that an error bound can be computed from simulations of the augmented LTI system $S_{aug} = \left[\begin{array}{cc|c} A & & \mathrm{B} \\ & A_r & B_r \\ \hline C & -C_r & 0 \end{array}\right]$. The computation time of simulations is negligible compared with the reachability computation.

The routine from CheckMate computes the reach set segments for each discrete state. For a specific state q, we assume the controller output $u(t) \in U(q) = \{u|u = f(q,y), y \in I_q\}$ is bounded. It is sufficient to have a method to compute an error bound for an arbitrary ℓ_∞ bounded control input $\mathcal{U}$. An upper bound of the Euclidean norm of error over all possible initial conditions and input signals can be formalized as the solution of the following optimal control problem:

$$\epsilon_r(t) = \max_{u(\cdot)\in\mathcal{U}, x_0\in X_0} \|e(x_0,u,t)\|$$

$$= \max_{u(\cdot)\in\mathcal{U}, x_0\in X_0} \|y(x_0,u,t) - y_r(\pi_R x_0, u, t)\| \quad (2)$$

The error bound is a function of time. We compute a more conservative error bound as the sum of two error bounds: $\epsilon_r^{x_0=0}$ is the error of the zero-state response, given by

$$\begin{aligned}\epsilon_r^{x_0=0}(t) &= \max_{u(\cdot)\in\mathcal{U}} \|e(x_0=0,u,t)\| \\ &= \max_{u(\cdot)\in\mathcal{U}} \|y(x_0=0,u,t) - y_r(x_{r0}=0,u,t)\|\end{aligned}$$

and $\epsilon_r^{u=0}$ is the error of the zero-input response, given by

$$\begin{aligned}\epsilon_r^{u=0}(t) &= \max_{x_0\in X_0,\ x_{r0}\in X_{r0}} \|e(x_0,u=0,t)\| \\ &= \max_{x_0\in X_0} \|y(x_0,u=0,t) - y_r(\pi_R x_0, u=0,t)\|\end{aligned}$$

The error bound for the system can be estimated as $\epsilon_r(t) \le \epsilon_r^{x_0=0}(t) + \epsilon_r^{u=0}(t)$ since the two systems are linear.

We now consider methods to estimate bound of output signal of a given LTI system at time t for the input set $\mathcal{U}$ and initial states X_0. Given the input signal in $\ell_\infty[0,t]$, the bound of the output signal at time t can be estimated using the following theorem.

Theorem 2 ([19]): For LTI system $S = \left[\begin{array}{c|c} \mathrm{A} & \mathrm{B} \\ \hline \mathrm{C} & 0 \end{array}\right]$, let y and u denote the output and input signals, respectively. Suppose the input is in $\ell_\infty[0,t]$ space, Then the norm of output of the system at time t is bounded by

$$\|y(t)\| \le \|y(0)\| + \|u(\cdot)\|_\infty^{[0,t]} \int_0^t \|Ce^{At}B\|dt \quad (3)$$

■

Considering the norm of a zero-input response, the upper bound can be computed directly as

$$\sup_{x(0)\in X_0} \|y(t)\| = \sup_{x_0\in X_0} \|Ce^{At}x(0)\| \quad (4)$$

Using (3) and (4) for the augmented system, the error of zero-state response is given by

$$\epsilon_r^{x_0=0}(t) \le \|u(\cdot)\|_\infty^{[0,t]} \int_0^t \|[\,C \quad -C_r\,]\begin{bmatrix} e^{tA} & \\ & e^{tA_r}\end{bmatrix}\begin{bmatrix} B \\ B_r\end{bmatrix}\|dt \quad (5)$$

The error of zero-input response is given by

$$\epsilon_r^{u=0}(t) \le \sup_{x_0\in X_0} \|[\,C \quad -C_r\,]\begin{bmatrix} e^{tA} & \\ & e^{tA_r}\end{bmatrix}\begin{bmatrix} x_0 \\ \pi_R x_0\end{bmatrix}\| \quad (6)$$

For discrete-time systems, the two error bounds can be computed as

$$\epsilon_r^{x_0=0}(t) \le \|u(\cdot)\|_\infty^{k} \sum_{i=0}^{t-1} \|[\,C \quad -C_r\,]\begin{bmatrix} A^i & \\ & A_r^i\end{bmatrix}\begin{bmatrix} B \\ B_r\end{bmatrix}\| \quad (7)$$

$$\epsilon_r^{u=0}(t) \le \sup_{x_0\in X_0} \|[\,C \quad -C_r\,]\begin{bmatrix} A^i & \\ & A_r^i\end{bmatrix}\begin{bmatrix} x_0 \\ \pi_R x_0\end{bmatrix}\| \quad (8)$$

Notice that the estimated error bounds vary with time t. The first error bound clearly increases with time, whereas the second bound could either increase or decrease as time increases.

In summary, the computation of e_r involves computing error bounds for zero-state response and zero-input response. The first computation involves integration of the norm of the impulse response of the augmented system, which can be obtained from a simulation. The second computation involves simulating all the vertices of the initial set X_0 when X_0 is polyhedral.

IV. Reachability Algorithm - Implementations

There are various tools developed to compute reach set in state space [2], [3], [7], [15]. Our descriptions are implementations of the algorithm in Fig. 3 for continuous-time and discrete-time LTIs. With the specific implementation, we investigate the underlying numeric computations and demonstrate the reduction of computation time by using reduced-order models. It has been shown in [15] that oriented rectangular hull (ORH) representation can reduce the computation time compared with the convex hull routines, especially for higher-order models. In this paper we show that with the ORH representation, reducing the model by several variables will further shorten the computation time significantly. This observation makes it attractive to use reduced-order models in reachability analysis.

A. Algorithm for Discrete-Time Systems

The computation of reach sets for a discrete-time control system can be performed using linear mapping and Minkowski sum operations [7]. Following [15], we implement the algorithm in Fig. 4 to compute reach set at time t_f. The first step is to compute the reachable states using the reduced-order model. This is done by successively computing the next step reachable set. The ORH is successively computed to avoid the rapid growth of faces. The second step is to estimate the error bound using (7) and (8). The third step is to compute the conservative reach set in output space. First, we transform each vertex to the output subspace. Then we compute an ORH. Since V is unitary, the $bloat\_r$ routine bloats the ORH by pushing each face using the error bound $\epsilon_r(k)$, where

$$bloat\_r\{x|\underline{d} \le V^T x \le \bar{d}\} = \{x|\underline{d} - V^T\epsilon \le V^T x \le \bar{d} + V^T\epsilon\}$$

The algorithm computes the reach set for a time instant t_f, it can be easily extended to compute the reach set for a time interval.

We analyze the computation time for algorithm in Fig. 4 by counting the number of flops (floating point operations). Each iteration consists of four steps: linear transform, axis determination and computing ORH. The required flops of all the operations are [4]

$$O(2^{2n_r}n_r^2) + O(n_r^2 2^{n_r}) + O(n_r^3) \approx O(n_r^2 2^{2n_r}) \quad (9)$$

The operations outside the for-loop consist of a linear transform, an error estimation and one ORH. Suppose the dimension of the output is k. The linear transform requires $O(n_r k 2^{n_r})$ operations. The error estimation consists of two simulations, which require $O(i_f n_r^2)$ flops. The computation

```
given: Original system as $[A, B, C]$, Reduced system
      $[A_r, B_r, C_r]$, $X_0$, $X_{r0}$, admissible control set $U$
      and a hybrid Controller $HC$
output: Conservative reach set $Y$ of the original
       system.
Algorithm:
$i$=0;
$X = X_{r0}$;
WHILE( i<t_f);
        $Vtx = vertices(X)$;
        $Vtx = A_r * V + B * control\_set(HC, X)$;
        $V_x = determine\_axis$(Vtx);
        $X = ORH_{V_x}(Vtx)$;
END
$\epsilon = estimate\_error(A, B, C, A_r, B_r, C_r, X_0, X_{r0}, U)$
$Vtx = vertices(X)$
$Vtx = C_r * Vtx$;
$V_y = determine\_axis(Vtx)$;
$Y = ORH_{V_y}(Vtx)$;
$Y = bloat\_r(Y, \epsilon)$
END
```

Fig. 4. Conservative Reachability Algorithm for Discrete-Time Systems

outside the for-loop can be estimated as

$$\begin{aligned} O(n_r k 2^{n_r}) + 2 \times O(i_f(n+n_r)^2) + O(2^{n_r}k^2) \\ \approx O(kn_r 2^{n_r} + i_f(n_r+n)^2) \quad (10) \end{aligned}$$

From (9) and (10), the total flops can be estimated as

$$i_f O(n_r^2 2^{2n_r}) + O(kn_r 2^{n_r} + i_f(n_r+n)^2)$$

The number of flops is super exponential in the dimension n_r. Therefore, using reduced-order models can decrease the computation effort exponentially. A comparison of computation times for the models of different orders for the example in section V is shown in Fig. 5.

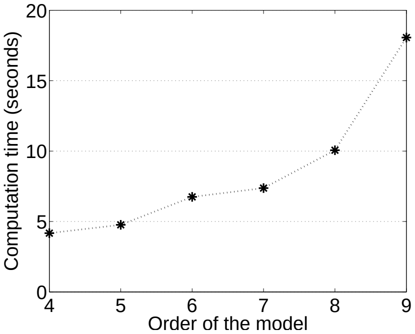


Fig. 5. Computation times for the discrete-time reachability procedure.

B. Algorithm for Continuous-Time Systems

To compute the reach set for a time interval $[0, t_f]$, existing tools like CheckMate [2] and *d/dt* [3] partition time into small intervals and compute polyhedral over-approximations of reach sets for each interval. Following the discussion in [2], [15], we implement the reachability algorithm for continuous-time systems shown in Fig. 6. We give the algorithm for a flow-pipe segment $[t, t+\delta]$. The algorithms to compute reach set for $[0, tf]$ can be easily constructed by successively computing each segment over the range.

```
Given: Original system as $[A, B, C]$, reduced system
        as $[A_r, B_r, C_r]$, $X_0$, $X_{r0}$, $t$, $\delta$, admissible control
      $U$ and a hybrid controller $HC$
Output: Reachable set Y for $[t, t+\delta]$
Algorithm:
//Step 1. Compute the reduced flow-pipe segment.
$V = vertices(X_{r0})$
$Vtx_t = evolve\_vertices(A_r, B_r, t, X_{r0}, HC)$
$Vtx_{t+\delta} = evolve\_vertices(A_r, B_r, t+\delta, X_{r0}, HC)$
$U = determine\_axis(Vtx_t \bigcup Vtx_{t+\delta})$
$X = ORH_U(V_t \bigcup V_{t+\delta})$
$X = bloat\_hull(X, A_r, B_r, X_{r0}, t, \delta, HC)$
//Step 2. Compute output reach set.
$\epsilon = estimate\_error(A, B, C, A_r, B_r, C_r, X_0, X_{r0}, U)$
$Vtx = vertices(X)$
$Vtx_y = C_r * Vtx$
$V_y = determin\_axis(Vtx_y)$
$Y = ORH_{V_y}(Vtx_y)$;
$Y = bloat\_r(Y, \epsilon)$
END
```

Fig. 6. Reachability Algorithm for Continuous-Time Systems

The algorithm in Fig. 6 has two parts. The first part is to compute the reach set of $[t, t+\delta]$. The algorithm was first proposed in [2] and later adapted in [15] to use ORH representation. The second part of the algorithm is the error estimation and the transformation of the reach set in state space to the output space.

There are two types of computations in the procedure: the computation of polytopes and the bloating of polytopes. From the discussion of the discrete-time algorithm, the time for polytope computation grows exponentially with the order. The bloating procedure involves solving the optimization problem [15]

$$\begin{aligned} max|min_{x_{r0} \in X_{r0}, \tau \in [t, t+\delta], u(\cdot) \in \mathcal{U}} U_i^T x_r(\tau) i = 1, \cdots n \\ s.t.\ x_r(t) = x_{r0} + \int_0^\tau Ax(s) + Bu(s) ds \end{aligned}$$

in order to guarantee the conservativeness of the computed ORH set. Experiments show that the time of solving the optimization problem does not vary much with the order. However the number of optimization problems increases linearly as the order grows. A comparison of the computation time is shown in Fig. 7 for the example in section V, where t_1 denotes the time of polytope computation and t_2 denotes the time of bloating routines.

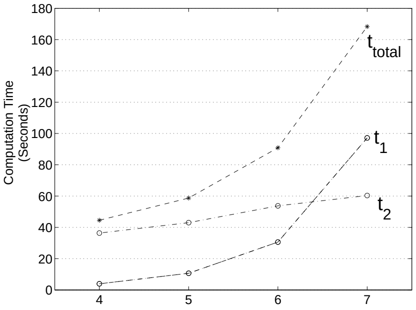
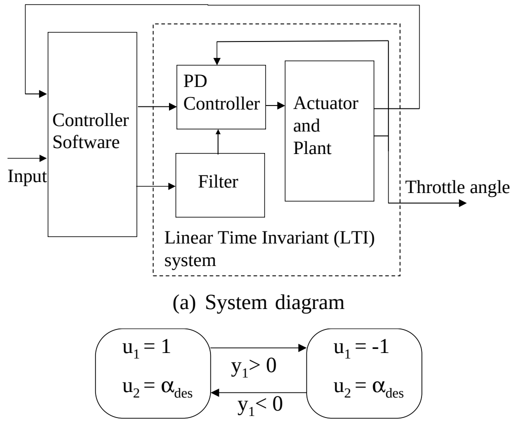


Fig. 7. Computation time for continuous-time reachbility.

V. Case Study: Electrical Throttle Control System

In this section we demonstrate the reachability procedure for a model of the electrical throttle control (ETC) system from the DARPA MoBIES Open Experimental Platform. [1] The plant is a continuous time LTI system with seven state variables.

The block diagram of the model is shown in Fig. 8(a). The subsystem inside dashed box of Fig. 8(a) is modelled as the plant. The output of the plant are the throttle angle α and angular velocity ω. The control inputs are $u_1(y) = \mathsf{sgn}(y_2)$ and angle command α_{des}. The sliding mode controller is a discrete-time controller with sampling period of 20ms.



(a) System diagram

(b) Switch Controller

Fig. 8. Hybrid model of the sliding mode controller for the ETC system.

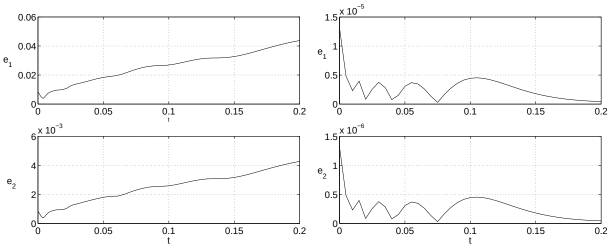
A series of reduced-order models were created using balanced truncation [10]. We performed reachability analysis to verify the rise-time requirement. The initial set is chosen to be $\{x | -0.005 \leq x_i \leq 0.005, i = 1 \cdots 7\} \cap \{x | -0.001 \leq y_i \leq 0.001, i = 1, 2\}$. The combined error $\epsilon(t)$ bound will be used for reachability analysis. Since the absolute value of y_1 and y_2 differ in magnitude, we estimate the reduction error for them separately to get more reasonable error bounds. As the order of the model decreases, the error bounds of the zero-state response become dominant in the total error. The error bounds of model reduction increase in magnitude as the order decreases. The *a priori* error bounds of balanced truncation for the model are shown in table I. For the model and the time horizon the *a posteriori* error bounds are much smaller than the *a priori* error bounds.

Order of the model	6	5	4
a priori error bound	0.0069	0.1116	0.7478
a posteriori error bounds for y_1, y_2	4.339e-07, 4.49e-07	0.00318, 0.000302	0.0478, 0.00534

TABLE I

ERROR BOUNDS OF THE REDUCED-ORDER MODELS

To use a reduced-order hybrid model to perform reachability analysis, we need to first verify that the reduced-order model is a good approximation. For this model, our criteria for a good approximation is that reduced order model have the same discrete transition sequence as the the original system for the given initial set. This property is verified using discrete-time reachability analysis: for the given initial set, the conservative result of the reduced-order model does not have more discrete transitions than the full-order model. This is verified by computing the reach set for both models at the sampling time.



(a) Error bounds for the fourth-order reduced model.

(b) Error bounds for the sixth-order reduced model.

Fig. 9. Error bounds for different reduced-order models.

Part of the reach set for the full-order model and the fifth-order reduced model are shown in Fig. 10. The sliding mode controller satisfies $u_1(i+1) = sgn(y_1(i))$. It can be verified that the sign of y_1 of the reach set computed from the reduced order model is consistent with the full-order model. Thus, the fifth-order model is a reasonable approximation of the original one.

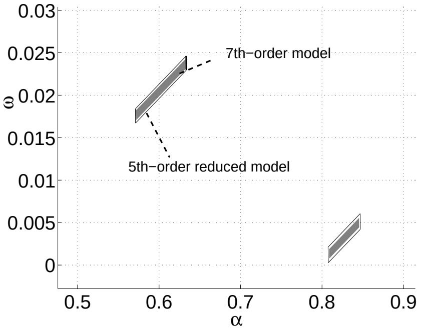


Fig. 10. Reach sets using the reduced-order models.

The reach set of the 4th-order model is shown in Fig. 11. The value of y_2 in set $Y(4)$ has both positive and negative values. Thus in the next step, the next location of the system should be positive or negative depending on the value of $y_2(4)$. The reduced-order model could make a transition, while the original model does not. We say that this reduced order model is not a good approximation of the original one.

The purpose of reachability analysis for the continuous time ETC system is to verify the rise-time and overshoot requirements. The rise-time requirement says all the trajectories enter the region $0.95 \leq y_2 \leq 1.05$ within 0.2 seconds. The overshoot requirement says that no trajectories enter the region $y_2 \geq 1.05$ during the time period. To verify this requirment, the reach set over $[0, 0.2]$ is computed using the algorithm in Fig. 6 with time step $\delta = 0.004s$. The reach set computed using the full-order model and fifth-order reduced model are shown in Fig. V. The reachability analysis verifies the rise-time and overshoot requirements.

[1] A description of the DARPA MoBIES Automotive Open Experimental Platforms can be found at http://vehicle.me.berkeley.edu/mobies/

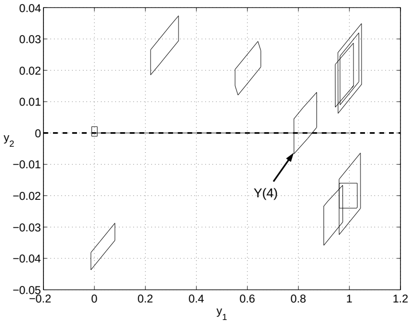
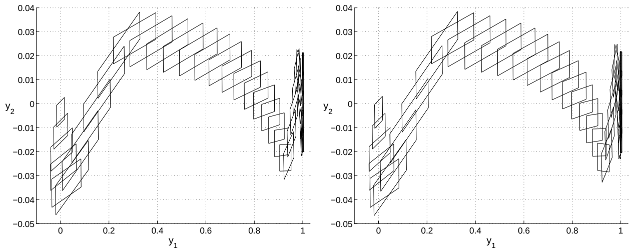


Fig. 11. Reach set of the 4th-order reduced model



(a) Reach set for the 7th-order full model.

(b) Reach set for the 5th-order reduced model.

Fig. 12. Reach sets computed by the continuous-time algorithm.

VI. Discussion

This paper considers the use of reduced-order models in reachability analysis for a class of hybrid control systems. We present algorithms to estimate the error bounds of model reduction. Combining these bounds with reachability algorithms, we demonstrate that the algorithm using reduced-order models is efficient in time compared with the algorithms using full-order models. To use reduced-order models in verification, the appropriate reduced-order should be chosen to achieve a trade-off between the approximation error and dimension of the model. This problem is demonstrated with the ETC example. The experiment data show that the less the order of the model, the less the computation time. However, as the dimension of the model decreases, the additive error introduced by model reduction is larger. Future work will focus on methods for verifying properties of more general classes of hybrid systems using reduced-order models.

Acknowledgements

This research was supported in part by Ford, the US Defense Advance Projects Research Agency (DARPA) contract nos. F33615-00-C-1701 and F33615-02-C-4029, US Army Research Office (ARO) contract no. DAAD19-01-1-0485, and the US National Science Foundation (NSF) contract no. CCR-0121547.

References

[1] A.C. Antoulas, D.C. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. *Contemporary Mathematics*, 280:193–219, 2000.

[2] A. Chutinan and B. H. Krogh. Compuational techniques for hybrid system verification. *IEEE Transaction on Automatic Control*, 48(1):64–75, Jan 2003.

[3] T. Dang. *Verification and Synthesis of Hybrid Systems*. PhD thesis, Verimag, Institut National Polytechnique de Grenoble, 2000.

[4] J. W. Demmel. *Applied numerical linear algebra*. Society for Industrial and Applied Mathematics (SIAM), Sep 1997.

[5] K. Glover and J. R. Partington. Bounds on the achievableaccuracy in model reduction. In *Modelling, Robustness and Sensitivity Reduction in Control Systems*, pages 95–199. Springer-Verlag, 1987.

[6] T. A. Henzinger. The theory of hybrid automata. In *Logic in Computer Science, 1996. LICS '96. Proceedings., Eleventh Annual IEEE Symposium on*, pages 278–292, Jul 1996.

[7] A. B. Kurzhanski and P. Varaiya. Reachability analysis for uncertain systems - the ellipsoidal technique. *Dynamics of Continuous, Discrete and Impulsive Systems Series B: Applications and Algorithms*, 9:347–367, Jan 2002.

[8] P. Mavrikis. *Nonlinear Model Reduction in Time Domain*. PhD thesis, Imperical College of Science Technology and Medicine, 1997.

[9] P. Mavrikis and R. B. Vinter. Trajectory-specific model reduction. In *Proc of 36th Conference on Decision and Control*, volume 4, pages 3323 –3328, Dec 1997.

[10] B. C. Moore. Principal component analysis in linear systems: Controllability, observability and model reduction. *IEEE Transaction on Automatic Control*, 26(1), Feb 1981.

[11] G. Obinata and B. D. O. Anderson. *Model Reduction for Control System Design*. Springer-Verlag, Oct 2001.

[12] A. Odabasioglu, M. Celik, and L. Pileggi. Prima: passive reduced-order interconnect macromodeling algorithm. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(8):645–654, Aug 1998.

[13] G. Pappas. Bisimilar linear systems. *Automatica*, Dec 2003. In press.

[14] G. Pappas, G. Lafferriere, and S. Sastry. Hierarchically consistent control systems. *IEEE Transactions on Automatic Control*, 45(6):1144–1160, Jun 2000.

[15] O. Stursberg and B. H. Krogh. On efficient representation and computation of reachable sets for hybrid systems. In *Hybrid Systems: Computation and Control HSCC'03*, Springer-Series: LNCS 2623, pages 482–497, 2003.

[16] H. G. Tanner and G. Pappas. Abstraction of constrained linear systems. In *Proceedings of the 2003 American Control Conference*, pages 3381–3386, Jun 2003.

[17] C. J. Tomlin, I. Mitchell, A. M. ayen, and M. Oishi. Computational techniques for the verification of hybrid systems. *Proceedings of the IEEE*, 91(7):986– 1001, Jul 2003.

[18] A. Varga. Model reduction routines for slicot. Technical report, NICONET, Jun 1999.

[19] K. Zhou, J. Doyle, and K. Glover. *Robust and optimal control*. Prentice Hall, Aug 1995.