43rd IEEE Conference on Decision and Control
December 14-17, 2004
Atlantis, Paradise Island, Bahamas



# Finite Automata Approximations with Error Bounds for Systems with Quantized Actuation and Measurement: A Case Study

Danielle C. Tarraf[1] Alexandre Megretski[2] Munther A. Dahleh[3]

*Abstract*— **We consider stable, discrete time, first order LTI systems with finite input alphabets and quantized outputs. We propose an algorithm for generating deterministic finite state machine approximations of these systems with computable bounds on approximation error, and we describe the conditions under which the bounds are valid.**

## I. INTRODUCTION

The results presented in this paper represent our first step towards investigating the use of deterministic finite state machines (FSMs) to approximate analog-state systems with quantized actuation and measurement in a manner that allows us to quantify the approximation error. In particular, we consider a simple case study where the plant we attempt to approximate consists of a discrete time, internally stable, first order LTI system with finite input alphabet and quantized output.

Systems with quantized measurement and actuation are practically important: The ubiquitous presence of digital computers in modern control systems has imposed on us the study of systems involving interacting continuous and discrete dynamics. Thus the field of hybrid systems has emerged as an active field of research, and the need for a systematic method to deal with these systems has been recognized and pursued.

In [1], Elia and Mitter consider the problem of quadratic stabilization of a single input single output LTI system with discrete actuation and measurement. They characterize the coarsest logarithmic state quantizer that allows stabilization and design a corresponding output feedback controller. In [2], Lunze et.al. consider autonomous systems with quantized outputs. They derive sufficient conditions for an exact representation as deterministic automata to exist, and describe how to partition the state space for that purpose. In [3], Lafferriere et.al. introduce a class of hybrid systems, referred to as O-minimal systems, for which successive refinement of an initial partition on the state space provably terminates and results in a finite bisimulation. In [4], Raisch and O'Young consider systems with discrete actuation and measurement and construct finite nondeterministic automata approximations by mapping strings of measurement and control symbols to automata states. They show that the behavior set of the approximating automata contains that of the original system and qualitatively prove that as the length of the strings increases, the approximation behavior becomes 'closer' to the original behavior. No quantitative measure of approximation accuracy or convergence results are given.

Our objective and approach are different in that we consider the system components, such as the quantizer, to be fixed and we do not insist on finding exact or almost exact discrete approximations. The results by Megretski in [5] show that the existing robust control framework can be extended to the discrete setup. Hence, we consider any approximating model with error bounds that are *computable* and *sufficiently small* for the task at hand (analysis or synthesis) to be good enough.

## II. PROBLEM STATEMENT

Consider a discrete-time, analog-state system $P$ with finite input and output alphabets, having the structure shown in Fig. 1. $H$ is a stable first order system ($|a| < 1$):

$$H : \begin{cases} x(t+1) = ax(t) + u(t) \\ v(t) = x(t) \end{cases} \tag{1}$$

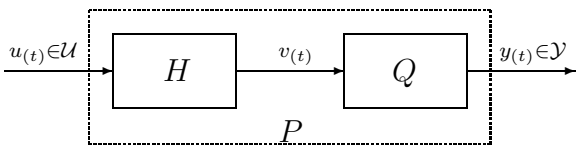


Fig. 1. Analog-State System with Finite Inputs and Quantized Outputs

The input of the system is restricted to take on integer values, $u(t) \in \mathcal{U} = \{-k, \ldots, -1, 0, 1, \ldots, k\}$ for some finite $k$. $Q$ is a given uniform quantizer with quantization step $l$:

$$Q : y_{(t)} = \begin{cases} \vdots & \\ -l & \text{if } \frac{-3l}{2} < v_{(t)} \leq \frac{-l}{2} \\ 0 & \text{if } -\frac{l}{2} < v_{(t)} < \frac{l}{2} \\ l & \text{if } \frac{l}{2} \leq v_{(t)} < \frac{3l}{2} \\ \vdots & \end{cases} \tag{2}$$

The initial state of $H$ is assumed to lie in compact set $I = [\frac{-k}{1-|a|}, \frac{k}{1-|a|}]$, which is an invariant subset of the state

[1] Department of Mechanical Engineering, Massachusetts Institute of Technology, *dtarraf@mit.edu*
[2] Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, *ameg@mit.edu*
[3] Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, *dahleh@mit.edu*

0-7803-8682-5/04/$20.00 ©2004 IEEE

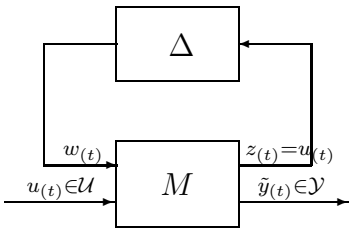


Fig. 2. Uncertain Finite State Machine Representation of $P$

space. The output $y_{(t)}$ takes on a finite number of successive values in $\mathcal{Y} \subset \{\ldots, -l, 0, l, \ldots\}$.

Under these assumptions, we seek to answer the following question: Is it possible to construct an 'uncertain deterministic finite state machine' $M$ and a corresponding uncertainty block $\Delta$ having 'small gain', such that the system consisting of the feedback interconnection of $M$ and $\Delta$ (Fig. 2) asymptotically predicts the output of plant $P$? The terms in quotations are precisely defined in Section III. In this setup, the input $z(t)$ to $\Delta$ is assumed to be $z(t) = u(t)$, and the output of $\Delta$ is restricted to a binary alphabet set $w(t) \in \{0, 1\}$.

The results presented in this paper are an algorithm to generate a machine $M$ and to verify whether the resulting $\Delta$ satisfies a given gain bound when both $M$ and the original plant $P$ start out with zero initial conditions. It is also shown that when $\Delta$ satisfies a given gain bound for zero initial conditions, it does so for arbitrary initial conditions provided the plant and the machine are 'externally stable', a notion that is defined in Section III. On the negative side (but not surprisingly), it is shown that when the original plant is not externally stable, it is not possible to generate machines for which the corresponding uncertainty block $\Delta$ has arbitrarily small gain in this setup.

## III. PRELIMINARY DEFINITIONS

### A. Uncertain Deterministic Finite State Machines

We begin by precisely defining what we mean by an "uncertain deterministic finite state machine".

*Definition 1:* An *uncertain deterministic finite state machine* $M$ is a set of finite alphabets $\{\mathcal{U}, \mathcal{W}, \mathcal{Y}, \mathcal{Z}, \mathcal{S}\}$ and a set of maps $\{f : \mathcal{S} \times \mathcal{U} \times \mathcal{W} \to \mathcal{S}, h : \mathcal{S} \to \mathcal{Y}, g : \mathcal{S} \times \mathcal{U} \to \mathcal{Z}\}$.

$\mathcal{U}$ and $\mathcal{W}$ are finite alphabet sets of possible instantaneous values of the control and disturbance inputs, respectively. Thus input signals $u$ and $w$ are understood to be strings over alphabets $\mathcal{U}$ and $\mathcal{W}$. Similarly, $\mathcal{Y}$ and $\mathcal{Z}$ are finite alphabet sets of possible instantaneous values of the measurement and cost outputs, respectively. $\mathcal{S}$ is the (finite) set of states of the finite state machine. $f$ is the state transition function. $h$ and $g$ are the output functions. $h$ is understood to define a partition on $\mathcal{S}$, while $g$ is an arbitrary memoryless function of the state and control inputs.

Even though the state transitions of $M$ corresponding to a particular input $(u_{(t)}, w_{(t)}) \in \mathcal{U} \times \mathcal{W}$ are deterministic, $M$ is 'uncertain' because it is driven by a disturbance input $w$ that we do not measure, predict or control.

### B. System Gain

In this section, a discrete-time 'system' is understood to mean a dynamical system that is amenable to a state-space representation of the form:

$$S : \begin{cases} x(t+1) = f(x(t), u(t)) \\ y(t) = h(x(t)) \end{cases} \tag{3}$$

Consider a discrete-time system $S$ whose input $u$ and output $y$ are sequences over finite input and output alphabets $\mathcal{U}$ and $\mathcal{Y}$, respectively. $\mathcal{U}$ and $\mathcal{Y}$ are assumed to be subsets of the set of integers containing 0.

*Definition 2:* $S$ is said to be *gain-bounded* if there exists finite non-negative constants $C$ and $\gamma$ such that

$$\sum_{t=0}^{T} |y(t)| \leq C + \gamma \sum_{t=0}^{T} |u(t)| \tag{4}$$

holds for all time $T \geq 0$, for all initial conditions of the system and for all admissible input sequences.

For a gain-bounded system, the greatest lower bound of $\gamma$ is called the system gain and denoted by $\gamma_o$.

### C. External Stability

Consider again a discrete-time system $S$ with finite input and output alphabets as described above.

*Definition 3:* $S$ is said to be *externally stable* if the following two conditions are satisfied:

(a) $S$ is gain bounded

(b) There exists a finite time $T > 0$ such that for any input $\{u(t)\}_{t=0}^{\infty}$ and for any two initial conditions $x(0) = x_1^o$ and $x(0) = x_2^o$, the corresponding outputs $\{y_1(t)\}_{t=0}^{\infty}$ and $\{y_2(t)\}_{t=0}^{\infty}$ satisfy $y_1(t) = y_2(t)$ for all $t \geq T$.

Thus, we are calling a system externally stable if it eventually forgets its past without blowing up in the process.

*Remark*: It is easy to construct a plant where $H$ is a nicely stable system but plant $P$ is not externally stable. For instance, the case where $a = 0.5$, $l = 4$ and $k = 5$ is one such system.

## IV. THE APPROXIMATION ALGORITHM

The first algorithm presented in this paper, which is described in this Section, takes a plant $P$ (Fig.1) and an integer parameter $m$ (to be described shortly) as its inputs and generates an uncertain deterministic finite state machine $M$ (Definition 1). The second algorithm presented in this paper, which is described in Section V, verifies whether the uncertainty block $\Delta$ associated with machine $M$ and plant $P$ satisfies a given gain bound when both $M$ and $P$ start out with zero initial conditions.

### A. Overview of the Algorithm

Define the state of the machine at time $t$ to be

$$\tilde{x}_{(t)} = \begin{pmatrix} u(t-1) \\ \vdots \\ u(t-m) \\ y(t) \\ y(t-1) \\ \vdots \\ y(t-m) \end{pmatrix} \tag{5}$$

for all feasible pairs of input/output sequences of the plant $P$, for some positive integer $m$. Since the input and output alphabets are finite by assumption, the resulting state set $\mathcal{S}_f$, $\mathcal{S}_f \subset \mathcal{U}^m \times \mathcal{Y}^{m+1}$, is also finite. The state transitions corresponding to a given value of $u_{(t)}$ are not expected to be unique in general, since we are losing information due to quantization and due to restriction of the memory of the machine to a finite length snapshot of the past. Hence, the machine $M_{ND}$ described so far has non-deterministic state transitions that can be described by a state transition function $f_{ND} : \mathcal{S}_f \times \mathcal{U} \rightarrow \mathcal{P}(\mathcal{S}_f) \setminus \emptyset$, where $\mathcal{P}(\mathcal{S}_f)$ is the power set of $\mathcal{S}_f$. This machine can be converted to a deterministic machine by introducing an additional input, disturbance input $w$, that is assumed to drive the non-unique transitions. If for every pair $(s,u) \in \mathcal{S} \times \mathcal{U}$, $f_{ND}(s,u)$ is a set of cardinality two at most, a binary alphabet $\mathcal{W} = \{0,1\}$ is sufficient for this purpose. The resulting machine is an uncertain deterministic finite state machine as in Definition 1.

### B. Details of the Algorithm

The algorithm for generating an approximating FSM model $M$ consists of the following steps:

1) *Selection of $m$*
   Larger values of $m$ should allow us to construct approximations of $P$ where the machine $M$ has more states and the uncertainty block $\Delta$ is smaller. However, there are practical limitations to how large $m$ can be made before the computations involved become infeasible.
2) *Generation of the set of feasible states*
   A state $s_i = (u_{1,i}, \ldots, u_{m,i}, y_{0,i}, y_{1,i}, \ldots, y_{m,i})'$ is feasible if there exists at least one state $x_o \in I$ such that $Q(x_o) = y_{m,i}$, $Q(ax_o + u_{m,i}) = y_{m-1,i}$, $\ldots$,$Q(a^m x_o + a^{m-1}u_{m,i} + \ldots + u_{1,i}) = y_{0,i}$. The set of feasible states is denoted by $\mathcal{S}_f$.
3) *Generation of the 1-step state transitions*
   For every possible input value $u_j \in \mathcal{U}$ and for every feasible machine state $s_i$, any feasible state of the form $s_k = (u_j, u_{1,i}, \ldots, u_{m-1,i}, y_k, y_{0,i}, \ldots, y_{m-1,i})'$ for some $y_k \in \mathcal{Y}$ is a potential end state for the transition. Hence in general, the 1-step transitions corresponding to a given state and input are not unique. Function $f_{ND}$ describes these non-deterministic transitions.
4) *Conversion to a deterministic machine*
   To deal with the ambiguous transitions, we introduce an additional input, $w \in \mathcal{W} = \{0,1\}$ thus converting the machine to one with deterministic state transitions, described by function $f$ defined as follows:

$$f \doteq \begin{cases} f(s,u,0) = s_k \\ f(s,u,1) = s_k \quad \text{if } f_{ND}(s,u) = \{s_k\} \\ \\ f(s,u,0) = s_k \\ f(s,u,1) = s_l \quad \text{if } f_{ND}(s,u) = \{s_k, s_l\} \end{cases} \tag{6}$$

5) *State aggregation*
   The last step is to aggregate indistinguishable states. Two machine states are said to be indistinguishable if their corresponding outputs and their one-step transitions are identical. Formally, $s_i$ and $s_j$ are indistinguishable if (i) $h(s_i) = y_{0,i}$ and $h(s_j) = y_{0,j}$ are equal and (ii) $f(s_i,u,w) = f(s_j,u,w)$ for all $(u,w) \in \mathcal{U} \times \mathcal{W}$. In this case, the two states can be collapsed into one and the input/output behavior of the resulting machine is identical to the original one.

The uncertain deterministic finite state machine generated by this algorithm is an approximate model $M$ of plant $P$. Note that beyond step (2), the computations involve only the discrete-state system, whether for determining the 1-step state transitions or for aggregating the states. This could have positive consequences beyond this simple case study, as we are not having to compute the images of entire subsets of the continuous state space. Also, note that the elements of $\mathcal{W} = \{0,1\}$ and $f_{ND}(s,u) = \{s_k, s_l\}$ are arbitrarily matched in step (4) of this algorithm. This could be potentially improved in later work. Finally, the sparsity of the (transition) matrices involved could be potentially exploited in step (5) when dealing with large machines.

### C. Properties of the Algorithm

In order for this approach to be acceptable, it should allow us to construct machines that meet the minimum requirements that $1 \leq card(f_{ND}(s,u)) \leq 2$. This is shown to be the case in this section.

*Proposition 1:* For any input $u_j \in \mathcal{U}$ and for any feasible state $s_i$, there exists at least one value $y_k \in \mathcal{Y}$ such that state $s_k = (u_j, u_{1,i}, \ldots, u_{m-1,i}, y_k, y_{0,i}, \ldots, y_{m-1,i})'$ is also feasible.

*Proof:* Since $s_i$ is a feasible state, there exists some $x_i \in I$ such that $Q(x_i) = y_{m,i}$, $\ldots$,$Q(a^m x_i + a^{m-1}u_{m,i} + \ldots + u_{1,i}) = y_{0,i}$. Let $x_k = ax_i + u_{m,i}$ and let $y_k = Q(a^m x_k + a^{m-1}u_{m-1,i} + \ldots + u_j)$. $x_k \in I$ and $a^m x_k + a^{m-1}u_{m-1,i} + \ldots + u_j \in I$ since $I$ is invariant. Moreover, $x_k$ satisfies $Q(x_k) = y_{m-1,i}$, $\ldots$, $Q(a^m x_k + a^{m-1}u_{m-1,i} + \ldots + u_j) = y_k$ by construction.

Given a plant $P$, let $\mathcal{S}_p$ be the set of all potential states of the machine for a particular choice of $m$, $\mathcal{S}_p = \mathcal{U}^m \times \mathcal{Y}^{m+1}$. Define the following relation on $\mathcal{S}_p$: $s_i \in \mathcal{S}_p$ and $s_j \in \mathcal{S}_p$ are related, $s_i \circ s_j$, if $u_{1,i} = u_{1,j}$, $\ldots$, $u_{m,i} = u_{m,j}$, $y_{1,i} =$

$y_{1,j}, \ldots, y_{m,i} = y_{m,j}$. That is, $s_i$ and $s_j$ are related if they "remember" the same pair of input/output strings of length $m$. This relation is clearly symmetric, reflexive and transitive, and hence defines a partition on $\mathcal{S}_p$ consisting of $n_e = card(\mathcal{U})^m card(\mathcal{Y})^m$ equivalence classes, $\mathcal{S}_p = \cup_{i=1}^{n_e} E_i$.

*Proposition 2:* At most two elements in each equivalence class $E_i$ are feasible states of the machine.

*Proof:* Suppose that three elements in equivalence class $E_i$, say $s_1$, $s_2$, and $s_3$, are feasible states. Then there exists at least three states of the original system, say $x_1$, $x_2$ and $x_3$, all in $I$, such that:

$$Q(a^m x_1 + a^{m-1} u_{m,1} + \ldots + u_{1,1}) = y_{0,1} \quad (7)$$

$$Q(a^m x_2 + a^{m-1} u_{m,2} + \ldots + u_{1,2}) = y_{0,2} \quad (8)$$

$$Q(a^m x_3 + a^{m-1} u_{m,3} + \ldots + u_{1,3}) = y_{0,3} \quad (9)$$

where $y_{0,1}$, $y_{0,2}$ and $y_{0,3}$ are distinct. Two of the arguments of $Q$ must fall at a distance of more than one quantizer interval apart, say those corresponding to $x_1$ and $x_3$:

$$|(a^m x_1 + \ldots + u_{1,1}) - (a^m x_3 + \ldots + u_{1,3})| > l \quad (10)$$

We also have that:

$$|(a^{m-1} x_1 + \ldots + u_{2,1}) - (a^{m-1} x_3 + \ldots + u_{2,3})| < l \quad (11)$$

since $y_{1,1} = y_{1,3}$ by assumption. Thus we have that $|a^{m-1}(x_1 - x_3)| < l$ and $|a^m(x_1 - x_3)| > l$, which is impossible when $|a| < 1$.

*Corollary 1:* For any input $u(t) \in \mathcal{U}$, the machine, starting from any state $\tilde{x}(t) \in \mathcal{S}_f$, can transition to at most two states.

## V. DESCRIPTION OF UNCERTAINTY

In fact, what is driving the non-unique transitions of the finite state machine $M_{ND}$ described in the previous section are the unmodeled dynamics. The unmodeled dynamics are lumped into an uncertainty block $\Delta$ whose binary output is $w$, the disturbance input to $M$, and whose structure is given in Fig. 3. The boolean comparator is a memoryless system that outputs 0 when its inputs are equal and outputs 1 otherwise. It is clear that $\Delta$ is not a finite state machine in general, but is rather a system with complicated discrete and continuous dynamics that we do not want to explicitly model. Uncertainty set $\mathcal{D}$ is a set of systems mapping $u \rightarrow w$ to which uncertainty block $\Delta$ belongs. What we seek is to characterize $\mathcal{D}$ by finding some gain constraints that are satisfied by the input/output behavior of all its elements. A procedure for computing conservative constraints that hold for all inputs and for zero initial conditions is presented in this section, as well as conditions under which these constraints hold for all initial conditions.

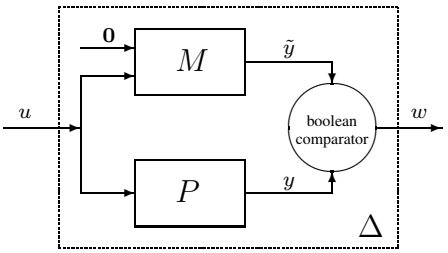


Fig. 3. The structure of $\Delta$

### A. The Intuition Behind this Description

The structure and connectivity of the states of machine $M_{ND}$ provide some information about the constraints that govern the input/output behavior of $\Delta$. For a given feasible state $s_i$ of $M_{ND}$ and a given input value $u_j \in \mathcal{U}$, if the one-step state transition $f_{ND}(s_i, u_j)$ is a singleton, then we know for sure that the output of the machine $M$ generated by the approximation algorithm and the output of the original system $P$ have to match. Hence, the corresponding output of $\Delta$ is a 0. On the other hand, if $f_{ND}(s_i, u_j)$ is a pair of states, then two possibilities exist: either the outputs of $M$ and $P$ match (output of $\Delta$ is 0), or they do not match (output of $\Delta$ is 1). Moreover, it is *not* the case that transition to one of the end states always corresponds to a mismatch and transition to the other end state always corresponds to a match. Thus, in the absence of other information, we assume that in the worst case scenario, transition to either end state always results in a mismatch. We can then formulate conservative constraints on the input/output behavior of $\Delta$ by searching through the simple cycles[1] of $M_{ND}$ to find the cycle corresponding to the worst possible (i.e. highest) ratio of cumulative error because of mismatched outputs to cumulative input required to drive the machine through the cycle. Let the ratio corresponding to the worst cycle be $\gamma$ and let $N$ be the least common multiple of the lengths of all simple cycles with at least one possible mismatch. The following inequality holds for any non-negative integer $\tau$:

$$\sum_{t=0}^{N\tau} |w_{(t)}| \leq \gamma \sum_{t=0}^{N\tau} |u_{(t)}| \quad (12)$$

(12) implies a constraint of the form (4), since we have:

$$\sum_{t=0}^{T} |w_{(t)}| \leq \sum_{t=0}^{N\tau_0} |u_{(t)}| + \sum_{N\tau_0+1}^{T} |w_{(t)}| \quad (13)$$

for some non-negative integer $\tau_0$ ($\tau_0 > \frac{T}{N} - 1$). Hence, (12) implies (14):

$$\sum_{t=0}^{T} |w_{(t)}| \leq C + \gamma \sum_{t=0}^{T} |u_{(t)}| \quad (14)$$

[1]A simple cycle is one that doesn't repeat any (state,input) pair except for the first and last

for all $T \geq 0$ and for some positive constant $C$, $C < N$.

### B. Computing a Gain Bound for Zero Initial Conditions

A brute force search through all simple cycles of $M_{ND}$ in order to compute an upper bound for the gain of $\Delta$ is practically infeasible for all but the simplest machines. To get around this issue, we propose the following algorithm based on the analysis theorem presented in [5]. This algorithm does not explicitly compute a gain bound for $\Delta$ as described in the previous section. Rather, it allows us to verify whether a given value of $\gamma$ is a valid gain bound by checking feasibility of a corresponding linear program.

Consider the subset $\mathcal{S}_{LP}$ of $\mathcal{S}_f^2$ defined as follows:

$$\mathcal{S}_{LP} \doteq \{(s, s^+) \in \mathcal{S}_f^2 | \exists u \in \mathcal{U} \text{ s.t. } s^+ \in f_{ND}(s, u)\} \quad (15)$$

Define two integer-valued functions on $\mathcal{S}_{LP}$; the *control cost*, $\tilde{u}$, defined by the rule:

$$\tilde{u}(s, s^+) \doteq \{u | s^+ \in f_{ND}(s, u)\} \quad (16)$$

and the *approximation penalty*, $\tilde{w}$, defined by the rule:

$$\tilde{w}(s, s^+) \doteq \begin{cases} 0 & \text{if } f_{ND}(s, \tilde{u}) = \{s^+\} \\ 1 & \text{otherwise} \end{cases} \quad (17)$$

Given a value of $\gamma$ whose validity we wish to verify, we search for a real-valued storage function $V$ on $\mathcal{S}_f$, positive and bounded, such that inequality (18) holds for every pair of states $(s, s^+) \in \mathcal{S}_{LP}$:

$$\gamma |\tilde{u}(s, s^+)| - \tilde{w}(s, s^+) \geq V(s^+) - V(s) \quad (18)$$

This involves solving a linear program in $V$. If a solution exists, we would have succeeded in proving that when the plant $P$ and the approximating machine $M$ both start out with zero initial conditions, all the corresponding pairs of input/output signals of $\Delta$ satisfy the following inequality:

$$\sum_{t=0}^{T} |w_{(t)}| \leq V(\tilde{x}_{(0)}) + \gamma \sum_{t=0}^{T} |u_{(t)}| \quad (19)$$

On the other hand, if the linear program turns out to be infeasible, that does not necessarily indicate that $\gamma$ is not a valid gain bound for $\Delta$; we can only conclude that we are unable to prove that $\gamma$ is a valid gain bound.

### C. Validity of the Gain Bound for Arbitrary Initial Conditions

Consider the setup in Fig. 4 where $S_1$ is understood to be an arbitrary discrete-time system as defined in Section III and $S_2$ is understood to be a deterministic finite state machine with state set $\mathcal{S}$.

*Theorem 1:* Suppose there exists non-negative constants $C$ and $\gamma$ such that for every input string $\{u(t)\}_{t=0}^{\infty}$, the output of $S_1$ corresponding to initial condition $x(0) = x^o$ and the output of $S_2$ corresponding to initial condition $\tilde{x}(0) = s^o$ are such that the following inequality is satisfied for all $T \geq 0$:

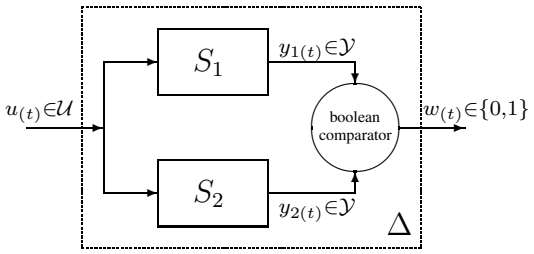


Fig. 4. The error system of $S_1$ and $S_2$

$$\sum_{t=0}^{T} w(t) \leq C + \gamma \sum_{t=0}^{T} |u(t)| \quad (20)$$

A sufficient condition for $\gamma$ to be a valid gain bound for $\Delta$ is that $S_1$ and $S_2$ are externally stable.

*Proof:* Let $\{w^o(t)\}_{t=0}^{\infty}$ be the output of $\Delta$ corresponding to input $\{u(t)\}_{t=0}^{\infty}$ and initial conditions $(x^o, s^o)$, and let $\{w^*(t)\}_{t=0}^{\infty}$ be the output of $\Delta$ corresponding to the same input and an arbitrary pair of initial conditions, say $(x^*, s^*)$. $S_1$ and $S_2$ are externally stable, hence there exists finite times $T_1$ and $T_2$ after which the initial state is forgotten, in the sense described in Definition 3. Thus, $w^o(t)$ and $w^*(t)$ are identical for all times $t \geq \tau$, where $\tau = max(T_1, T_2)$. Let $C' = C + \tau$. All valid input/output signals of $\Delta$ thus satisfy (20) when constant $C$ is replaced by $C'$.

*Theorem 2:* If $S_1$ is gain-bounded but not externally stable, it is not possible to find a machine $S_2$ such that the gain of system $\Delta$ is arbitrarily small.

*Proof:* Suppose this is not true: then for every $\gamma_\epsilon$, there exists a machine $S_{2,\epsilon}$ such that $\gamma_\epsilon$ is a valid gain bound for the resulting system $\Delta$. Since $S_1$ is externally unstable but gain-bounded, there exists a pair of values of the initial state, say $x^o$ and $x^*$, an input sequence $\{u^{o*}(t)\}_{t=0}^{\infty}$ and a finite positive constant $\tau$ such that for any $t > 0$, there exists at least one $t_o \in [t, t+\tau]$ at which the corresponding outputs, $y^o(t_o)$ and $y^*(t_o)$ are not equal. Let $\tau_o$ be the smallest such value of $\tau$. Let $\{w^o(t)\}_{t=0}^{\infty}$ and $\{w^*(t)\}_{t=0}^{\infty}$ be the outputs of $\Delta$ corresponding to input $\{u^{o*}(t)\}_{t=0}^{\infty}$, to some initial state (say $s_o \in \mathcal{S}$) of $S_2$ and to initial states $x^o$ and $x^*$ of $S_1$, respectively. By assumption, both pairs of input/output signals $(\{u^{o*}(t)\}_{t=0}^{\infty}, \{w^o(t)\}_{t=0}^{\infty})$ and $(\{u^{o*}(t)\}_{t=0}^{\infty}, \{w^*(t)\}_{t=0}^{\infty})$ satisfy inequality (20). Hence we have:

$$\sum_{t=0}^{\infty} (|w^o(t)| + |w^*(t)| - 2\gamma_\epsilon |u^{o*}(t)|) \leq 2C \quad (21)$$

Since $w^o(t)$ and $w^*(t)$ have to differ for at least one $t_o \in [t, t+\tau_o]$, for any $t \geq 0$, we also have:

$$\sum_{t}^{t+\tau_o} (|w^o(t)| + |w^*(t)|) \geq 1 \quad (22)$$

TABLE I

SIMULATION RESULTS: L=2, K=1

| | $a = 0.551$ | $a = 0.612$ | $a = 0.671$ |
|---|---|---|---|
| m | 2 | 3 | 5 |
| $n_{feas}$ | 47 | 155 | 2723 |
| $n_{agg}$ | 13 | 19 | 115 |
| $\gamma$ | 1 | 1 | 1 |

which leads to a contradiction when $\gamma_\epsilon < \frac{1}{2k(\tau_o+1)}$.

## VI. SIMULATION RESULTS

We implemented the algorithms and ran simulations in order to check whether we could construct, for a given plant $P$, an approximate FSM model $M$ such that the gain of the corresponding $\Delta$ is bounded by 1. The results of these simulations for the case where the quantization interval is $l = 2$ and the input alphabet is $\mathcal{U} = \{-1, 0, 1\}$ are presented in Table I. $n_{feas}$ is the number of feasible states of the machine and $n_{agg}$ is the number of reduced states after aggregation. The machines reported correspond to the smallest value of $m$ for which a gain bound of $\gamma = 1$ could be verified. The simulations show that larger values of $m$ are needed for plants with larger values of $|a|$ in order to guarantee similar gain bounds for $\Delta$. This is to be expected since systems with larger values of $|a|$ are less stable (internally) and hence remember their pasts longer. Moreover, the simulations indicate that a large computational effort is typically needed for state aggregation, highlighting the importance of computationally efficient aggregation algorithms and/or suggesting that alternative approaches where we build up some of the states of the machine as needed might be more practical. We also ran simulations in which we applied randomly generated input sequences of length $10,000$ to the original system, starting from a zero initial condition, and to the machine generated by this algorithm again starting from a zero initial state and with disturbance input $w$ identically 0. We directly compared the outputs of the two systems to get a feel for how well the machines are approximating the original systems 'on average'. The approximate models seemed to perform better than expected (with the values of the computed 'average gains', computed as the ratio of cumulative error to cumulative input, being only about 10 to 20% of the verified gain bounds), hence confirming the belief that the gain bounds we are computing are conservative.

## VII. CONCLUSIONS AND FUTURE WORK

We considered a simple case study of a stable first order plant with quantized input and output. We presented two algorithms, one for generating a deterministic finite state machine $M$ that approximates the original plant, and another that verifies whether the resulting uncertainty block $\Delta$ satisfies a given gain bound. The system consisting of the feedback interconnection of $M$ and $\Delta$ asymptotically predicts the output of the original plant if the plant and the machine both start out with zero initial states. We identified a property, external stability, that guarantees that the machine $M$ remains an equally good approximation of the original plant when the plant and the machine are arbitrarily initialized. We also showed that when the plant is not externally stable, it is not possible to find a pair $(M, \Delta)$ where $\Delta$ has arbitrarily small gain and the feedback interconnection of $M$ and $\Delta$ asymptotically predicts the output of the plant regardless of initial conditions.

In this paper, we did not address the issue of how to formally verify external stability of a given system, or the lack thereof. This question will be addressed in future work. Alternative approaches that would allow us to find less conservative gain bounds will also be explored. Another research direction is in investigating alternative frameworks that would enable us to approximate systems that are gain bounded but not externally stable to an arbitrary degree of accuracy. Future work will also focus on moving beyond this simple case study to systems where this approach could be of more practical value (for instance, when the internal dynamics are nonlinear or hybrid). We hope to be able to identify interesting classes of systems for which this approach is feasible. Finally, another research effort will be in the direction of finding more efficient computational algorithms for state aggregation and system analysis, since the systems we expect to be dealing with will be large, as the simulations for this simple case study indicate.

## REFERENCES


[1] N. Elia and S.K. Mitter, "Stabilization of Linear Systems with Limited Information", *IEEE Transactions on Automatic Control*, vol. 46, no. 9, September 2001, pp. 1384-1400.

[2] J. Lunze, B. Nixdorf and J. Schröder, "Deterministic Discrete-Event Representation of Continuous-variable Systems, *Automatica*, vol. 35, March 1999, pp. 395-406.

[3] G. Lafferriere, G. Pappas and S. Sastry, "O-Minimal Hybrid Systems", *Mathematics of Control, Signals and Systems*, vol. 13, no. 1, March 2000, pp. 1-21.

[4] J. Raisch and S.D. O'Young, "Discrete Approximation and Supervisory Control of Continuous Systems", *IEEE Transactions on Automatic Control*, vol. 43, no. 4, April 1998, pp. 569-573.

[5] A. Megretski, "Robustness of Finite State Automata", *in Proceedings of the Mohammed Dahleh Memorial Symposium*, Santa Barbara, CA, February 2002.