ABSTRACTION DES TRAJECTOIRES D'UN SYSTEME CONTINU EN AUTOMATES TEMPORISES

ABSTRACTION OF CONTINUOUS SYSTEM TRAJECTORIES INTO TIMED AUTOMATA

Arnaud HÉLIAS\textsuperscript{1,2,*}, François GUERRIN\textsuperscript{1} and Jean-Philippe STEYER\textsuperscript{2}

\textsuperscript{1}CIRAD, Gdor team, Station de la Bretagne, BP 20, 97408 Saint-Denis, Reunion Island, France, guerrin@cirad.fr
\textsuperscript{2}INRA, Laboratory of environmental biotechnology, av. des étangs, 11100 Narbonne, France, steyer@ensam.inra.fr

Résumé: Cet article traite de la représentation de la dynamique d’un système continu par un formalisme à événements discrets et temps continu, à des fins d’analyse de système. Le système continu est d’abord approximé à l’aide d’intervalles numériques, puis traduit dans le formalisme des automates temporisés grâce à l’introduction de seuils partitionnant les domaines de valeurs des variables d’état. La détection des franchissements de seuil est caractérisée par deux instants correspondant à la date au plus tôt et à la date au plus tard de franchissement. Cette approche est brièvement illustrée sur un procédé pilote de traitement d’eaux usées de 1 m\textsuperscript{3}.

Mots clés: Approximation de modèle, Système discret, Système continu, Intervalles, Seuils, Automates, Traitement des déchets.

Abstract: This paper deals with the representation of continuous system dynamics into a continuous-time discrete-event formalism to the end of system analysis. The continuous system is first approximated by using numerical intervals and then translated into the timed automata formalism thanks to with the introduction of thresholds partitioning the state variables’ domains. The detection of thresholds crossing is characterised by two time instants corresponding respectively to the earliest and latest crossing dates. This approach is briefly illustrated on a 1 m\textsuperscript{3} wastewater treatment pilot plant.

Keywords: Model approximation, Discrete system, Continuous system, Intervals, Thresholds, Automata, Waste treatment.

1. INTRODUCTION

A classical approach to analyse the interaction between continuous and discrete elements within a system is to represent it with a unique formalism. In the Qualitative reasoning and Hybrid dynamical system communities, this problem was often addressed. However, Struss (2002) underlined three main difficulties: (i) explosion in the number of discrete states, (ii) rounding errors of intermediate variables and (iii) thresholds detection. Moreover, time is not often clearly represented. This study is based on a timed discrete-event representation similar to those developed by Henzinger et al. (1998), Kowalewski et al. (1999), Lunze (1999), Supavatanakul et al. (2003); however it differs from several points:

− as the continuous system dynamics is assumed to be partially known, the resulting imprecision is represented by numerical intervals bounding both initial states and input variables values;
− thresholds, partitioning the state-variable domains according to expert knowledge are used as a basis of the state space discretisation;

\textsuperscript{*}Current author’s address: INRA, Food Process Engineering and Microbiology Laboratory, BP 01, 78850 Thiverval-Grignon, France, arnaud.helias@grignon.inra.fr.
to avoid combinatorial explosion, timed automata are purposely built to answer specific questions, according to some initial state and the properties to be checked.

The paper is organised as follows. After a brief recall on timed automata, one of the most widespread timed discrete formalism, the continuous system under study is described in Section 2. The discretisation procedure is then detailed in Section 3 and illustrated by an application example in Section 4. Finally, this approach is briefly discussed in Section 5 and conclusions are drawn in Section 6.

2. SYSTEM FORMAL DESCRIPTION

Timed automata

Introduced by Alur and Dill (1994), a timed automaton is composed by a finite state machine and the expression of continuous time. This formalism allows timed constraints to be introduced using variables named clocks.

Clock. Let \( X \), with \( X \) a finite set, is a clock whose value grows linearly with time (i.e., \( x(t + \varepsilon) = x(t) + \varepsilon \) with \( \varepsilon \) a positive time lapse and \( x \)'s derivative always equal to 1).

Clock constraint. A clock constraint is an atomic constraint conjunction. Taking \( x \) and \( y \) two clocks, \( c \in \mathbb{Q} \) a constant and \( \# \) a symbol relation from the set \( \{<, \leq, =, \geq, >\} \), the set \( \Phi(X) \) of the atomic constraints \( \varphi \) is defined by the grammar \( \varphi := x \# c | x < y \# c | \varphi_1 \wedge \varphi_2 \). These constraints are possibly associated to the locations or edges of the automaton.

Timed automaton. A timed automaton \( G \) is defined by a 6-tuple \((S, P, X, E, A, \text{Inv})\) with:

- \( S \) a finite set of locations with \( s_0 \in S \) the initial location;
- \( P \) a mapping which associates to each \( s \in S \), a set of atomic propositions valid at this location;
- \( X \) a finite set of clocks;
- \( E \) a finite set of labels;
- \( A \subseteq S \times E \times \Phi(X) \times 2^X \times S \) a finite set of edges; each edge \( a \) is a 5-tuple \((s, e, \varphi, \delta, s')\) with \( s \in S \) the start location and \( s' \in S \) the destination location, \( \varphi \in \Phi(X) \) a clock constraint named guard that must be satisfied to trigger the discrete transition and \( \delta \subseteq X \) the clock subset to be reinitialised during the transition;
- \( \text{Inv}(s) : S \rightarrow \Phi(X) \) a mapping that associates to each location a timed constraint \( \phi' \), named invariant. The system can remain in the same location as long as the invariant is true.

Continuous system

We consider a non-linear piecewise continuous system described by an ordinary differential equation (ODE) system such as:

\[
\begin{align*}
\dot{\xi} &= f(\xi, \zeta) \\
\xi(t_0) &= \xi_0
\end{align*}
\]

where \( \xi \in \mathbb{R}^n \), with \( \Xi \) a \( n \)-dimensional state space, \( \zeta = g(p, t) \), \( \zeta \in Z \subseteq \mathbb{R}^m \), with \( Z \) a \( m \)-dimensional input space, and \( p \in P \) a parameter. It is assumed that we have only a partial knowledge of both the initial state \( \xi(t_0) \) and the input variables values \( \zeta \) that can be estimated by:

\[
\begin{align*}
\forall i, \xi_i(t_0) \leq \xi_i(t_0) & \leq \xi_i(t_0) \\
\forall j, \forall t, \xi_j(t) & \leq \xi_j(t) & \xi_j(t) \leq \xi_j(t)
\end{align*}
\]

with \( i \in [1, ..., n] \), \( j \in [1, ..., m] \), \( (\zeta_i^-, \zeta_i^+) \in Z_i^2 \) and \( (\zeta_j^-, \zeta_j^+) \in Z_j^2 \). (\( \Xi \subset \mathbb{R}^n \) and \( Z_j \subset Z_j \)).

Uncertainty representation. We define \( \xi^- \) and \( \xi^+ \) according to the influence of the interval bounds on the derivative’s values (taking \( k = [1, ..., m] \setminus \{j\} \)):

\[
\begin{align*}
f(\xi^-, \zeta) & \leq f(\xi, \zeta) \leq f(\xi, \zeta) & \Rightarrow \xi^- = \zeta, \xi^+ = \zeta \\
f(\xi^-, \zeta) & \geq f(\xi, \zeta) \geq f(\xi, \zeta) & \Rightarrow \xi^- = \zeta, \xi^+ = \zeta
\end{align*}
\]

From (4) we can rewrite (1) as the double ODE system:

\[
\begin{align*}
\dot{\xi}^- &= f(\xi^-, \zeta^-) \\
\xi^-(t_0) &= \xi_0 \\
\dot{\xi}^+ &= f(\xi^+, \zeta^+) \\
\xi^+(t_0) &= \xi_0
\end{align*}
\]

With the property \( \forall t, \xi^-(t) \leq \xi(t) \leq \xi^+(t) \), Eq. (5) can approximate the system (1) by introducing uncertainty on the initial state (Eq. (2)) and the system inputs (Eq. (3)) if the cooperativity property is verified on the system’s dynamics (Smith 1995). Cooperativity simply states that the off-diagonal elements of the Jacobian matrix of a dynamic system are positive or equal to zero (additional details can be found in Gouzé et al., (2000) or Moor and Raisch (2002)).

Threshold

To represent the continuous system dynamics in a discrete formalism, the first step is to discretise the state space. For this, each \( \xi_i \)'s domain is partitioned according to thresholds into a finite number of intervals that can be considered as qualitative states. Thresholds are defined from expert knowledge; e.g., a stock level can be qualified as “insufficient”,...
“low”, “medium”, “high” and “critical”. \( L_i \) denotes the ordered set of thresholds in the \( i \)th dimension, that is, with \( i \in \{ 1, \ldots, n \} \) and \( \Omega \in \mathbb{N} : \\
\( (L_i, <) = \{ l_{i,0}, \ldots, l_{i,\Omega} \} \) 
(7)

The continuous state space \( \Xi \) is divided according to the \( \prod_i \Omega \) thresholds by the mapping:
\n\[
D_\xi : \Xi \rightarrow \{ 1, \ldots, \Omega \} \times \ldots \times \{ 1, \ldots, \Omega \}
\]

with the assumption that \( \xi_i \in [l_{i,0}, l_{i,\Omega}] \), \( \forall i \). A rectangular partition is thus obtained.

3. DISCRETE APPROXIMATION

The idea is (i) not to consider as discrete states the cells partitioning the state space themselves but, rather, the cells’ faces and (ii) determine the state transitions from the simulation output of the continuous system within each cell as was done by Kowalewski et al., 1999. These authors defined a grid on each cell’s face and simulated all the trajectories from or bounded to each grid-point. In comparison, our approach depends upon each considered initial state and focuses on handling uncertainty by propagating the interval \( \left[ \xi_i^-, \xi_i^+ \right] \) within the cells by simulating Eq. (5).

Trajectories within a cell

Thresholds crossing. Let \( \tau \) be the time-point when the system (1) reaches \( v \), a threshold value defined in the \( i \)th dimension, i.e., \( \xi_i(\tau) = v, v \in \mathbb{R} \). Because \( \xi_i \) is estimated by the interval \( \left[ \xi_i^-, \xi_i^+ \right] \), the time window within which crossing the \( v \) threshold may occur can be described by \( \tau^- \) and \( \tau^+ \) such that \( \xi_i^- (\tau^-) = \xi_i^+ (\tau^+) = v \). From Eq. (5), the following implications can then be set:
\n\[
\left( \xi_i^- (\tau^-) > 0 \right) \land \left( \xi_i^+ (\tau^+) > 0 \right) \Rightarrow \tau^+ < \tau^-
\]
\n(9)
\[
\left( \xi_i^- (\tau^-) < 0 \right) \land \left( \xi_i^+ (\tau^+) < 0 \right) \Rightarrow \tau^+ > \tau^-
\]

(10)

Timed automaton representation. Let \( \tau_{\min} = \min(\tau^-, \tau^+) \), \( \tau_{\max} = \max(\tau^-, \tau^+) \) and \( F \) the mapping that associates a label to each threshold crossing:
\n\[
F(v) \rightarrow \begin{cases} 
\{ \psi^+ \} & \text{if } \tau_{\min} = \tau^+ \\
\{ \psi^- \} & \text{if } \tau_{\min} = \tau^-
\end{cases}
\]

(12)

with the symbols \( \Delta \) and \( \nabla \) denoting threshold crossing with increasing and decreasing trends respectively.

\( \tau_{\min} \) and \( \tau_{\max} \) being approximated on \( \mathbb{Q} \) (cf. the clock constraint definition), the system trajectory between two thresholds \( w \) and \( v \) is represented within the time window \( [\tau_{\max}, \tau_{\min}] \). This window can be modelled in the timed automata formalism as follows (cf. Fig. 1):

- \( s_1 \) and \( s_2 \) are two locations with associated labels "\( \psi^+ \)" and "\( \psi^- \)" respectively,
- \( w \) is a clock,
- \( x \leq \tau_{\max} \) is the invariant of \( s_1 \),
- \( (s_1, \emptyset, x \geq \tau_{\max}) \) is the edge.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{Fig1.png}
\caption{Timed automaton for describing a system’s trajectory from thresholds \( w \) to \( v \) with an increasing trend.}
\end{figure}

Generalisation

For sake of clarity, the index \( i \) is omitted in notations for all the variables in the following sections. Applying the above described approach to the lower bound of Eq. (5) in the \( i \)th dimension allows one to obtain \( \Sigma^- \), the ordered set of time instants when \( \xi^- \) crosses the thresholds and \( \Sigma^- \), the ordered set of threshold crossings:
\n\[
\left( \Sigma^-, < \right) = \{ \tau_j^- : \xi^-(\tau_j^-) = l_{\omega} \}
\]
\n(13)
\[
\left( \Sigma^-, < \right) = \{ \tau_j^- : \xi^-(\tau_j^-) = l_{\omega} \}
\]

(14)

with \( j = 1, \ldots, J \), \( J = \text{card} (\Sigma^-) = \text{card} (\Sigma^+) \) and \( \omega \in \{ 1, \ldots, \Omega \} \). In the following, the mapping that associates \( \tau_{\min} \) or \( \tau_{\max} \) to each \( \tau_j^- \) according to the sign of the derivative is denoted \( A^- \) :
\n\[
A^- : \tau_j^- \rightarrow \begin{cases} 
\tau_{\min} & \text{if } \xi^-(\tau_j^-) < 0 \\
\tau_{\max} & \text{if } \xi^-(\tau_j^-) > 0
\end{cases}
\]

(15)

A similar approach can be used on the upper bound to obtain \( \Sigma^+ \), \( \Sigma^+ \) and associated labels by the mapping \( A^+ \) :

Building the timed automaton associated to the system’s trajectory in the \( i \)th dimension

From Eqs. (13-15), the following sets can be defined:
\( B, \) the ordered set of time instants when occur threshold crossings by \( \xi^- \) and \( \xi^+ \) with label \( \tau_{\text{min}} \) \((k_B = 1, \ldots, K_B, \ K_B = \text{card}(B) = \text{card}(D)):\)

\[
(B, \leq) = \left\{ b_{k_B} \in T^{-} \mid \xi^- (b_{k_B}) = \tau_{\text{min}} \right\} \\
\bigcup \left\{ b_{k_B} \in T^{+} \mid \xi^+ (b_{k_B}) = \tau_{\text{min}} \right\}
\]

(16)

\(- C, \) the ordered set of time instants when occur threshold crossings by \( \xi^- \) and \( \xi^+ \) with label \( \tau_{\text{max}} \) \((k_C = 1, \ldots, K_C, \ K_C = \text{card}(C)):\)

\[
(C, \leq) = \left\{ c_{k_C} \in T^{-} \mid \xi^- (c_{k_C}) = \tau_{\text{max}} \right\} \\
\bigcup \left\{ c_{k_C} \in T^{+} \mid \xi^+ (c_{k_C}) = \tau_{\text{max}} \right\}
\]

(17)

\(- D, \) the ordered set of threshold crossings associated to the elements of \( B:\)

\[
(D, \leq) = \left\{ d_{k_B} \in \Sigma^{-} \mid \xi^- (d_{k_B}) = d_{k_B} \right\} \\
\bigcup \left\{ d_{k_B} \in \Sigma^{+} \mid \xi^+ (d_{k_B}) = d_{k_B} \right\}
\]

(18)

\(- \Sigma_0^V, \) the set of threshold crossings with decreasing trends at the initial time \((k_{x^v} = 1, \ldots, K_{x^v}, \ K_{x^v} = \text{card}(\Sigma_0^V)):\)

\[
\Sigma_0^V = \left\{ \sigma_{0,k_{x^v}}^V \mid \sigma_{0,k_{x^v}}^V \in L \wedge (\xi^- < \sigma_{0,k_{x^v}}^V < \xi^+) \right\}
\]

\[
\bigcup \left\{ \sigma_{0,k_{x^v}}^V \mid \sigma_{0,k_{x^v}}^V \in L \wedge (\xi^- = \sigma_{0,k_{x^v}}^V \wedge (\xi^- \leq 0)) \right\}
\]

(19)

\[- \Sigma_0^A, \) the set of threshold crossings with an increasing trend at the initial time \((k_{x^a} = 1, \ldots, K_{x^a}, \ K_{x^a} = \text{card}(\Sigma_0^A)):\)

\[
\Sigma_0^A = \left\{ \sigma_{0,k_{x^a}}^A \mid \sigma_{0,k_{x^a}}^A \in L \wedge (\xi^- < \sigma_{0,k_{x^a}}^A \wedge (\xi^- \geq 0)) \right\}
\]

\[
\bigcup \left\{ \sigma_{0,k_{x^a}}^A \mid \sigma_{0,k_{x^a}}^A \in L \wedge (\xi^- = \sigma_{0,k_{x^a}}^A) \wedge (\xi^- \geq 0) \right\}
\]

(20)

Clock set. The clock set is singleton since the automaton is obtained from only one simulation of the continuous model.

**Location set.** Let us denote:

\(- \), a mapping that associates a location to each element of \( D, \Sigma_0^V \) and \( \Sigma_0^A \). From \( \), one gets \( S = \{ s_k \}_{k=1}^{K_B} \cup S_0 \) with \( K_B + K_{x^v} + K_{x^a} \) and \( S_0 \), the location corresponding to the initial state of the continuous system.

\(- P, \) a mapping that associates a proposition to each element of \( S \) denoting which threshold is crossed and the trend (with \( \omega \in \Omega \) and with \( P(s_n) \rightarrow \xi^0 \)):

\[
\begin{align*}
\{( \xi^- (d_{k_B}) \rightarrow s_k) & \wedge (d_{k_B} = \omega) \wedge (d_{k_B} = \Sigma^+ ) \Rightarrow \xi^0 \} \\
\{( \xi^- (d_{k_B}) \rightarrow s_k) & \wedge (d_{k_B} = \omega) \wedge (d_{k_B} = \Sigma^- ) \Rightarrow \xi^0 \}
\end{align*}
\]

(21)

**Forward edges.** Edges that can reach a new location at the earliest time point \( \tau_{\text{min}} \) are named Forward edges. Let \( s_n \in K_B \), a location corresponding to a threshold crossing with increasing trend, and its associated proposition \( \xi^0 \). The preceding location can reflect the following cases (cf. Fig. 2):

a) crossing a lower threshold with increasing trend,
b) crossing the same threshold with decreasing trend,
c) the initial state whenever no threshold has already been crossed and (i) the threshold is within the initial state interval, or (ii) the initial state interval lies below the threshold.

The same reasoning can be made the other way around about threshold crossing with a decreasing trend (i.e., about a location with a proposition \( \xi^0 \)).

![Fig. 2. Different cases of threshold crossing with increasing trend.](image)

Based on this, a set of locations can be obtained which satisfy one of these conditions. The preceding location is the element to which is associated the nearest \( \tau_{\text{min}} \) of the \( \tau_{\text{min}} \) of \( s_n \). Formally, \( a_i \), the forward edge of \( s_n \) is defined by the 5-tuple
\( (s_n, \mathcal{D}, \varphi, \mathcal{D}, s_s) \) with \( \varphi := x \geq h_n^{-1} \). Let the ordered set \( S' \subseteq S \), the union of the different cases described by the Fig. 2:

\[
(S'_i, \leq) = \begin{cases} 
(P(s_i) = t_i^+ \Rightarrow s_i \subseteq P(s_i) = t_i^-) \\
(P(s_i) = t_i^- \Rightarrow s_i \subseteq P(s_i) = t_i^+) 
\end{cases} 
\]

\[
(S'_i, \leq) = \begin{cases} 
(P(s_i) = t_i^+ \Rightarrow s_i \subseteq P(s_i) = t_i^-) \\
(P(s_i) = t_i^- \Rightarrow s_i \subseteq P(s_i) = t_i^+) 
\end{cases} 
\]

(23) (24)

where \( k = 1, \ldots, n - 1 \) (i.e., only reachable locations before \( s_n \) are taken). The preceding location \( s_m \subseteq S' \) is the latest that could be reached. The set of locations of \( S \) forward edges is named \( A^1 \).

A similar approach can be used about the latest time points (i.e., threshold crossing at \( t_{\text{max}} \)) to determine the location’s invariants. Moreover, when a location has no next location, a Backward edge is defined. Figure 3 shows this case: the system crosses a threshold successively twice, and the edge with guard \( x = t_{\xi i}^- \) is a backward edge. Doing this, the Inv mapping, that associates the invariants to the locations, and \( A^1 \), the set of backward edges, are defined. A complete description of this approach can be found in Hélias, (2003).

From (16-27), the continuous system dynamics in the \( j^{th} \) dimension is approximated by the timed automaton \( G = (S, \mathcal{P}, X, \mathcal{E}, A, \text{Inv}) \) where:

- \( S = \{ s_k \}_{k=1}^{n} \tag{19} \cup S_{\hat{i}} \) is a set of locations,
- \( \mathcal{P}_j \) is a mapping that determines, for each location, the threshold crossing and its trend,
- \( X_i \) is a singleton composed of the clock \( x_i \),
- \( \mathcal{E} \), the set of labels, is an empty set,
- \( A_j = A_{ij} \cup A_{2j} \) is the set of edges,
- \( \text{Inv} \) is a mapping that associates the invariants to the locations.

4. EXAMPLE: APPROXIMATION OF A WASTEWATER ANAEROBIC DIGESTION MODEL

To briefly illustrate this approach, the procedure previously described is used on an anaerobic digestion process. A more complete description of this example can be found in Hélias et al., (2004). The aim is to predict possible dysfunctions of a wastewater treatment plant by a discrete approximation of a biological reaction model in front of expert partial knowledge of the system’s dynamics. Anaerobic digestion is a biological process used for carbon removal from wastewater. The principle is to transform organic matter into biogas (i.e., that is a methane (CH\(_4\)) and carbon dioxide (CO\(_2\)) mixture) in absence of oxygen. This complex process can be modelled in two steps: the acidogenic and methanogenic phases. A simplified version of the mass balance analytical model of our process (Bernard et al., 2001) is used:

\[
\begin{aligned}
\dot{S}_1 &= D(S_{in} - S_1) - k_1 \mu_1 X_1 \\
\dot{S}_2 &= D(S_{in} - S_2) + k_2 \mu_1 X_1 - k_3 \mu_2 X_2 \\
\dot{Z} &= D(Z_{in} - Z) \\
\dot{C} &= D(C_{in} - C) + k_3 \mu_1 X_1 + k_2 \mu_2 X_2
\end{aligned}
\]

(28)

with \( k_p = k_3 - k_6 \frac{P_{\text{CO}_2}}{P_{\text{CH}_4} - P_{\text{CO}_2}} \), \( S_1, S_2, Z, \) and \( C \) are the organic matter concentration, the volatile fatty acids concentration, the total alkalinity and the total inorganic carbon, respectively, \( k_1 \) to \( k_6 \) are the yield coefficients for the reactions and \( \mu_1 \) and \( \mu_2 \) are the growth rates of the bacterial biomass \( X_i \) according to \( S_i \) (in the acidogenic phase) and bacterial biomass \( X_2 \) according to \( S_2 \) (in the methanogenic phase) respectively. These growth rates are assumed to follow a Monod-type kinetics, i.e. \( \mu = \mu_{max} \frac{S}{S + K_s} \) with \( \mu_{max} \) the maximum bacterial growth rate and \( K_s \) the half-saturation constant. Finally, \( P_{\text{CH}_4} \) is the total pressure in the reactor, \( P_{\text{CO}_2} \) is the partial pressure of CO\(_2\), \( D \) is the dilution rate according to input flow and \( S_{in} \), \( S_{in} \), \( Z_{in} \), and \( C_{in} \) the different influent concentrations. Because of a relation holding between \( P_{\text{CH}_4} \) and \( P_{\text{CO}_2} \), the cooperativity property holds in this model. Thus, intervals bounding influent and biomass concentrations can be used to handle imprecision, following the approach described in Section 2. This model is run with parameters measured on an anaerobic digestion pilot plant

\(1\) If the location is an initial state threshold crossing, then the guard is of the form \( x = t_{\xi i}^- \).
The inputs of the process (i.e., influent concentrations) and the initial state are thus represented by intervals. Qualitative states resulting from expert knowledge (e.g., $S_2$ can be assigned labels “normal”, “high” or “critical”) also available are used in this approach. The plant operation is apprehended on a short time period (a few days) by combining information on different variables. For example, an important organic overload (i.e. dysfunction due to an inhibition of the reaction by $S_3$) can be associated to a “low” $Z$ value and a “high” $S_2$ value according to thresholds specifically set in these variables’ domains. To check the possible occurrence of this situation, we use the model-checking tool on the obtained timed automaton. This approach has been used to check about fifteen operating states during a four days period (Hélias et al., 2004). The complete abstraction procedure (system simulation, threshold crossing detection, timed automata creation), with 11 thresholds defined on 4 variable domains, takes less than ten seconds on a personal computer with a CPU running at 1.7 GHz. This procedure has been implemented by combining the Matlab® environment and the Kronos model-checker (Yovine, 1997). The resulting automaton is composed of 24 locations and 66 edges.

5. DISCUSSION
The final size of the automaton is set according to the system’s dimension, the number of thresholds and the crossing number. Consequently, this approach is inappropriate when:
- the system oscillates around a threshold value,
- a large number of thresholds are defined.

The idea is not to create an automaton to check all the possible properties of the continuous system but, rather, to create the automaton specifically suited to check a given property according to purposely defined thresholds.

Whereas in the classical definition of timed automata the clock constraints compare the clock values using rational numbers, integers must be used in the Kronos model-checker. Consequently, in our approach, the threshold crossing time values must be rounded. Thus, if the time unit used in the model is not carefully chosen according to the system dynamics, some additional imprecision may be introduced.

6. CONCLUSION
This paper has described a procedure aimed at representing a continuous system’s dynamics as a timed automaton, where:
- some elements (inputs, initial state) are estimated by intervals, which comes down to reformulate the system as extremal ODE systems (upper and lower bounds $\xi^{-}$ and $\xi^{+}$ approximating the real dynamics),
- dynamical envelopes stemming from the simulation of this extremal ODE systems are translated into the timed automata formalism (discrete model with continuous time) according to expert defined thresholds on the state variables’ domain.

The principal characteristics of this approach are:
- taking threshold crossings themselves as qualitative discrete states (disregarding the interval between two consecutive thresholds);
- defining the discrete state transitions from the piecewise simulation of the extremal ODE systems within consecutive thresholds;
- starting the procedure from (i) an initial state, (ii) imprecise values of inputs and (iii) threshold definitions, all defined in accordance to the property to be checked in order to reduce the state explosion.

This approach was briefly illustrated on an anaerobic digestion model. The timed discrete-event representation makes model-checking possible to check properties of the continuous system’s dynamics. An interesting extension would be to couple this representation to the discrete part of a wastewater treatment process in order to check properties on mixed continuous and discrete elements.

ACKNOWLEDGEMENTS
This work was partially supported by the Région Réunion and the European Social Fund. A first version of this paper was published at IFAC Workshop on Discrete Event Systems, WODES’04, 22-24 September 2004.

REFERENCES
Diagnosis. Carcassonne (France).


