RISK ASSESSMENT FOR SAFE RESTART OF ANAEROBIC DIGESTION PROCESSES

Belynda BRAHIMI[1], Hamid DEMMOU[1], Arnaud HÉLIAS[2], Jean Philippe STEYER[2]

[1]*LAAS/CNRS, 7 Avenue du colonel Roche, 31077 Toulouse cedex 4, France*
[2]*LBE/INRA, Avenue des Etangs, 11100 Narbonne, France*

Abstract: For several years, a 1 m^3 fixed bed anaerobic digestion process has been operated for the treatment of wine distillery wastewater. This reactor has been fully instrumented with the following variables available on-line: pH, temperature, liquid and gas flow rates, gas composition (*i.e.,* CH_4, CO_2 and H_2), concentration of bicarbonate, chemical oxygen demand, total organic carbon, volatile fatty acids and partial and total alkalinity. This paper deals with the problem of restarting the reactor after a period of time with missing information from the sensors. An approach has been developed to determine – while managing inherent uncertainty – the evolution of the inputs between the date of information loss and that of information recovered. It is then used together with a timed automata formalism and model checking capabilities to assess and analyse the induced risk in order to restart the process in safe conditions. *Copyright © 2005 IFAC*

Keywords: Anaerobic digestion, wastewater, bioprocess control, timed automata, diagnosis, model checker, risk assessment.

1. INTRODUCTION

The anaerobic digestion (AD) process is based on a complex ecosystem of anaerobic bacterial species that degrade the organic matter. It presents very interesting advantages compared to the traditional aerobic treatment: high capacity to degrade difficult substrates at high concentrations, very low sludge production, low energy requirements, and a possibility for energy recovery through methane combustion.

At the industrial scale, AD processes have been widely used since the late seventies and today, more than 1,400 digesters are referenced through the world (Totzke, 1999). Many of these processes are used for the treatment of residues from agro-industrial (*i.e.,* sugar, corn processing…) and beverage industries (*i.e.,* beer, wine, canning, distilleries…) although many other types of wastes from other origins are considered (*e.g.,* petrochemical, wood processing …).

However, many industrials are still reluctant to use anaerobic treatment plants in spite of their efficiency because they can become unstable under some circumstances. Disturbances like variations of the process operating conditions can lead to a destabilization of the process. This is due to the accumulation of intermediate toxic compounds resulting in biomass elimination. The reaactor takes then several weeks to several months to recover (see for example (Lardon *et al.,* 2004)). During this period, the unit cannot be used anymore and the wastewater is rejected into the natural environment (*e.g.,* rivers) without any treatment. It is therefore a great challenge for instrumentation and control sciences to make this process more reliable and usable at industrial scale. The first step – and maybe the most important one – is to follow dynamically (*i.e.,* using on-line sensors) the key process variables. The second step is to rebuild the trajectory of the system in case of information loss (sensor failure for example). The goal is clearly to assist the operators and the engineers in the task of diagnosis before restarting the reactor, because their decision depends on the history of the system.

2. PROCESS DESCRIPTION

The schematic layout of the up-flow anaerobic fixed bed reactor used in the present study is shown in Figure 1. It is a real pilot-scale process located in the LBE laboratory. The working volume of the reactor is 0.948 m^3. The dilution of the influent is performed by adding water to raw industrial vinasses (*i.e.,* wine distillery wastewater) in a 200 L buffer tank connected to the input line of the reactor. For a detailed description of the process and its instrumentation, see (Steyer *et al.,* 2002).

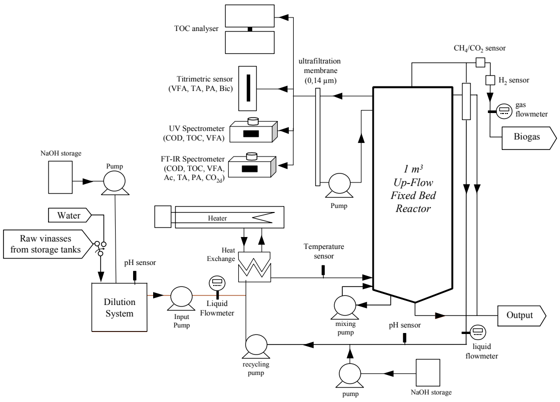


Figure 1. Schematic layout of the process

3. MODEL DESCRIPTION

Since the late seventies, various models have been proposed in the literature for AD processes. The first model included a single bacterial population (Graef and Andrews, 1974). The representation of the process was improved by considering three stages: the solubilization of organic compounds, acidogenesis and methanogenesis (Hill and Bart, 1977), or even a four-population model with two acidogenesis reactions and two-methanization reactions (Mosey, 1980). Thereafter, these models have been improved in order to get closer to the complexity of the process. The resulting models include several bacterial populations and various substrates, so the number of parameters may become very large (*i.e.,* up to 26 dynamic state concentration variables and 19 biochemical kinetic processes as in Batstone *et al.,* 2002). Because of the difficulty to precisely calibrate and validate these models, their direct use for monitoring and control purposes becomes difficult.

On the contrary, a simple model was chosen in the present study assuming that two main bacterial populations are present (Bernard *et al.,* 2001). From these considerations, a mass-balance-based model consisting of six ordinary differential equations was derived. The state variables of this model are linked to on-line measurements of the gas flow rates. By locating the biological variability in dedicated terms, namely the kinetic reaction rates, the model circumvents the lack of reliability of the bacterial growth modelling. The use of such models for monitoring and control design has been proven to be reliable over the years because it minimizes the number of assumptions during model construction (Steyer and Bernard, 2003). This mass-balance model forms also the basis for a software sensor that uses the available on line measurements of gases flow rates.

A drastic simplification of the digester ecosystem (*Cf.* Figure 2) was thus assumed during the model development. In fact, the corresponding reaction scheme was considered as a summary of the main mass transfer throughout the digester, *i.e.,* the two steps included in the model were:

- *Acidogenesis:* the population of acidogenic bacteria (X_1) consumes the organic substrate (S_1) and produces CO_2 and volatile fatty acids (S_2) through an acidogenesis step:

$$S_1 \rightarrow X_1 + S_2 + CO_2$$

- *Methanogenesis:* the second population (X_2) uses the volatile fatty acids in a methanogenesis step as substrate for growth and produces CO_2 and methane.

$$S_2 \rightarrow X_2 + CH_4 + CO_2$$

The total inorganic carbon is stored in the medium as bicarbonate and dissolved CO_2 form. A variable Z represents the total alkalinity within the digester.

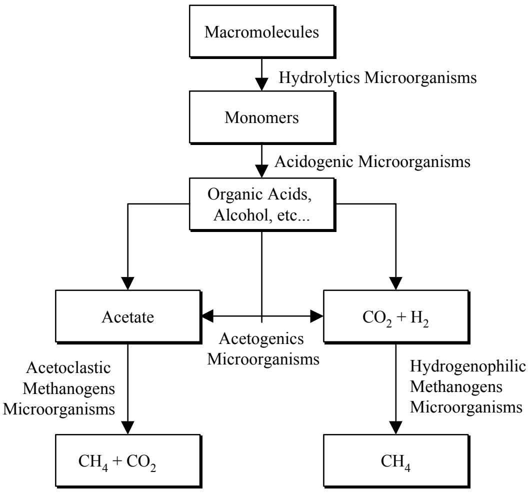

Macromolecules
Hydrolytics Microorganisms
Monomers
Acidogenic Microorganisms
Organic Acids, Alcohol, etc...
Acetate
CO$_2$ + H$_2$
Acetogenics Microorganisms
Acetoclastic Methanogens Microorganisms
Hydrogenophilic Methanogens Microorganisms
CH$_4$ + CO$_2$
CH$_4$

Figure 2. Main biological pathways in the anaerobic digestion reaction scheme

The model equations are then the following:

$$\begin{cases} \dot{X}_1 = \left(\mu_1 - \alpha D\right) X_1 \\ \dot{X}_2 = \left(\mu_2 - \alpha D\right) X_2 \\ \dot{S}_1 = D\left(S_1^{in} - S_1\right) - k_1 \mu_1 X_1 \\ \dot{S}_2 = D\left(S_2^{in} - S_2\right) - k_2 \mu_1 X_1 - k_3 \mu_2 X_2 \\ \dot{Z} = D\left(Z^{in} - Z\right) \\ \dot{C} = D\left(C^{in} - C\right) - q_{CO_2} + k_4 \mu_1 X_1 + k_5 \mu_2 X_2 \end{cases} \qquad (1)$$

The inputs are the influent concentration of organic substrate, volatile fatty acids (VFA), alkalinity and inorganic carbon, which are denoted S_1^{in}, S_2^{in}, Z^{in} and C^{in} respectively. Bacterial growth rates were chosen as a Monod model for X_1 and a Haldane model for X_2:

$$\begin{cases} \mu_1(S_1) = \mu_{1\max}\dfrac{S_1}{K_{S1}+S_1} \\ \mu_2(S2) = \mu_{2\max}\dfrac{S_2}{K_{S2}+S_2+\left(S_2/K_{I2}\right)^2} \end{cases} \quad (2)$$

3. HYBRID MODELLING FOR DIAGNOSIS

As said previously, the main drawback of the anaerobic treatment plants is related to their instability and the difficulty to recover from an unexpected disturbance. This paper focuses on the problem of recovering the system in case of sensors failure. More precisely, two questions have to be answered in order to assess the risk associated to the restart of the process: (i) how long the microbial populations spent in specific states (sometimes critical ones)? and (ii) what is the trajectory followed from a starting point corresponding to the date of information loss (sensor failure occurrence) to the arriving point where the data is totally recovered?

In other words, the type, the amplitude and the duration of process inputs disturbances the system had to face during this interval of information loss have to be determined and analysed. The approach we have developed combines continuous simulations with uncertainty management of the previously presented model and discrete analysis with a model checker of timed automata of the system. The continuous simulation aims to verify the validity of the inputs while the discrete analysis aims to check the reachability of some states.

3.1 Temporal abstraction for hybrid modelling

In the following, we consider a generalisation of model (1) that is a non-linear piecewise continuous system described by an ordinary differential equation (ODE) system such as:

$$\begin{cases} \dot{\xi} = f(\xi,\zeta) \\ \xi(t_0) = \xi_0 \end{cases} \quad (3)$$

where $\xi \in \Xi \subseteq \Re^n$, with Ξ a n-dimensional state space, $\zeta = g(p,t)$, $\zeta \in Z \subseteq \Re^m$, 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 ζ that can be estimated by:

$$\forall i, \xi_i^-(t_0) \le \xi_i(t_0) \le \xi_i^+(t_0) \quad (4)$$

$$\forall j, \forall t, \zeta_j^-(t) \le \zeta_j(t) \le \zeta_j^+(t) \quad (5)$$

with $i \in \{1,...,n\}$, $j \in \{1,...,m\}$, $(\xi_i^-,\xi_i^+) \in \Xi_i^2$ and $(\zeta_j^-,\zeta_j^+) \in Z_j^2$ ($\Xi_i \subset \Xi$ and $Z_j \subset Z$).

3.1.1 Uncertainty representation.

To handle the uncertainty due to the lack of information on the initial state and input variables, we define $\zeta^<$ and $\zeta^>$ according to the influence of the interval bounds on the derivative's values (taking $k = \{1,...,m\} \setminus \{j\}$):

$$\begin{matrix} f_i(\xi,\zeta_k,\zeta_j^-) \le f_i(\xi,\zeta) \le f_i(\xi,\zeta_k,\zeta_j^+) \Rightarrow \zeta_j^< = \zeta_j^-, \zeta_j^> = \zeta_j^+ \\ f_i(\xi,\zeta_k,\zeta_j^-) \ge f_i(\xi,\zeta) \ge f_i(\xi,\zeta_k,\zeta_j^+) \Rightarrow \zeta_j^< = \zeta_j^+, \zeta_j^> = \zeta_j^- \end{matrix} \quad (6)$$

From (6) we can rewrite (3) as the double ODE system:

$$\begin{cases} \dot{\xi}^- = f(\xi^-,\zeta^<) \\ \xi^-(t_0) = \xi_0^- \\ \\ \dot{\xi}^+ = f(\xi^+,\zeta^>) \\ \xi^+(t_0) = \xi_0^+ \end{cases} \quad (7)$$

With the property $\forall t, \forall i, \xi_i^-(t) \le \xi_i(t) \le \xi_i^+(t)$, Eq. (7) can approximate the system (3) by introducing uncertainty on the initial state (Eq. (4)) and the system inputs (Eq. (5)) 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)).

3.1.2 *Thresholds*

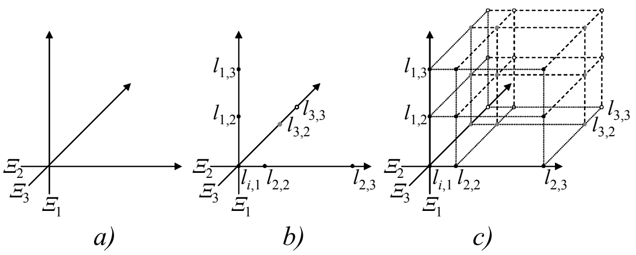
To represent the continuous system dynamics in a discrete formalism, the first step is to divide the state space. For this, each ξ_i's domain is partitioned according to thresholds into a finite number of intervals that can be considered as qualitative states. Thresholds are defined from expert knowledge; *e.g.*, a liquid flow can be qualified as "low", "medium", "high" and "critical". L_i denotes the ordered set of thresholds in the i^{th} dimension, that is, with $i \in \{1,...,n\}$ and $\Omega_i \in$ □:

$$(L_i,<) = \{\ l_{i,0},...,l_{i,\Omega_i}\ \} \quad (8)$$

The continuous state space Ξ is divided according to the $\prod_i \Omega_i$ thresholds by the mapping:

$$D_X : \Xi \to \{1,...,\Omega_1\} \times ... \times \{1,...,\Omega_i\} \quad (9)$$

with the assumption that $\xi_i \in [l_{i,0}, l_{i,\Omega_i}], \forall i$. A rectangular partition is thus obtained (*Cf.* Figure 3).



a) *b)* *c)*

Figure 3. Translation of a continuous state space (a) by thresholds (b) into a discrete space state (c).

3.1.3 *Discret approximation*

The idea is then (i) to consider as discrete states the faces of the cell 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 face of the cell and simulated all the trajectories from each grid-point or bounded to it. In comparison, our approach depends upon each considered initial state and focuses on handling uncertainty by propagating the interval $\left[\xi^-, \xi^+\right]$ within the cells by simulating Eq. (7).

3.1.4 *Timed automaton representation*

Introduced by Alur and Dill (1994), a timed automaton is composed of a finite state machine and the expression of continuous time. This formalism allows timed constraints to be introduced using variables named clocks. In the present study, the system trajectory between two thresholds w and v is represented by the time window $[\tau_{min}, \tau_{max}]$ which can be modelled in the timed automata formalism (*Cf.* figure 4 where $w < v$) by:

- s_1 and s_2, two locations with associated propositions "w^Δ" and "v^Δ" respectively,
- x, a clock,
- $x \le \tau_{max}$, the invariant of s_1,
- $(s_1, \varnothing, x \ge \tau_{min}, \varnothing, s_2)$, the edge.

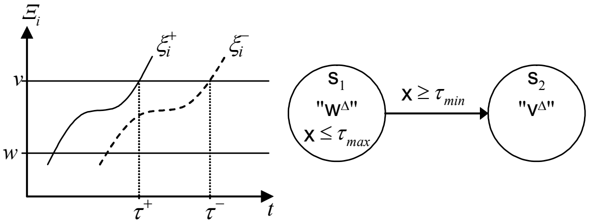


Figure 4. Timed automaton representing a trajectory from threshold w to v.

Using this approach and performing the synchronised product with Kronos (more details about Kronos can be found in Yovine 1997) of all the timed automata generated for each dimension of the system, yields a single timed automaton. This automaton has one clock without reset and it globally represents the system dynamics for the considered time period. More complete description of the methodological aspects can be found in (Helias *et al.,* 2004)

4. METHOD FOR DETERMINING THE INPUTS

4.1 *Type of inputs of the reactor*

Expert knowledge on the process makes it possible to identify some relevant types of inputs. Typical inputs are made of irregular successive steps of different amplitude. Therfore only three types of inputs have been considered in case of sensor information loss: a constant, positive or negative step.

4.2 *Introducing thresholds*

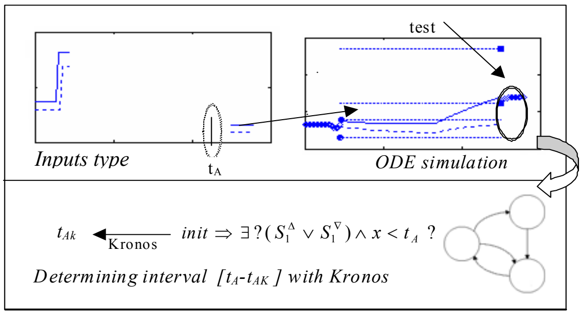
The thresholds are defined so that the starting point (*i.e.,* sensors information loss) and the arrival point (*i.e.,* sensors information recovery) are delimited by a given interval. This is done after the temporal abstraction procedure previously described has been applied and the automata for each dimension (S_1, S_2, C, Z, DCO, q_{gas}) have been obtained. These thresholds will help to determine the duration of the starting and arriving states using the obtained automata and the Kronos model checker. In particular, the existence of an input evolution from the starting point to a given arrival process state will be checked. If we consider that the predicate of the starting state is *init* and S_1 a threshold delimiting the values at the starting point, the question submited to the Kronos model checker will be: Does a path exist so that the threshold is overstepped after t_D time units?, which is expressed by the following formula:

$$init \Rightarrow \exists \lozenge \,(S_1^\Delta \vee S_1^\nabla) \wedge x > t_D$$

4.3 *First step of the method*

- *First phase*

The values of the unknown process inputs at the starting date t_D are expressed as intervals describing the imprecision $In_{Di}=[In^-_{Di}\ In^+_{Di}]$. The thresholds fix intervals delimiting the output values to be considered: two for the starting state and two for the arrival state. The system is then simulated (from date t_D to date t_A) with a constant value of inputs equal to the limits of the input intervals. The simulated outputs of the system at date t_A are then compared to the intervals associated to the real values obtained after information recovery. If this verification is positive and if the inputs at date t_D are equal to inputs at date t_A then we conclude that the system has been submitted to constant inputs during the time (t_A-t_D) when the sensors information was lost. On the opposite, we use the timed automata of each dimension and, using Kronos, check the time the system holds in the starting state with constant inputs. (*Cf.* Figure 5). We obtain a duration d_i for each input that expresses the time the input i has been constant. A new start date $t_{Dk} = min\ (t_D + d_i)$ is defined.



test
Inputs type
t_A
ODE simulation
t_{Ak}
Kronos
$init \Rightarrow \exists ?(S_1^\Delta \vee S_1^\nabla) \wedge x < t_A$?
Determining interval $[t_A\text{-}t_{AK}]$ with Kronos

Figure 5. Schematic representation of the approach

- *Second phase*

This phase aims to determine the time the system has spent in the arrival state with constant inputs. The ODE system is simulated with constant inputs value equal to the limits of the input intervals $In_{Ai}=[In^-_{Ai}\ In^+_{Ai}]$ until the arrival date t_A. The outputs of the system at date t_A are then compared to the intervals associated to the real values. If the simulated outputs are outside the intervals, a new simulation is processed with new initial values. As for the first phase, timed automata are used to represent the process behaviour in a discrete space. We obtain a duration a_i for each input that expresses the time the input i has been constant. A new arrival date is defined as $t_{Ak} = max\ (t_A - a_i)$.

- *Third phase*

It consists on the use of the dates $t_{Dk=}\tau_D$ and $t_{Ak}=\tau_A$ in order to build the inputs of the process. These dates are considered as discontinuities corresponding to a positive or a negative step. For example, if $In_{Di} < In_{Ai}$, then we deduce that a positive step of the inputs is present between t_D and t_A. As for the other phases, the system is simulated and verification with Kronos and test are processed on the output values ($Out_{Ai} \in\ [Out^-_i(t_A)\ Out^+_i(t_A)]$). In this phase, the objective is to refine the intervals containing the discontinuity. This process is thus repeated until the test on the outputs is negative.

4.4 *Second step of the method*

The objective is to refine the interval of the input. It consists on dividing the interval $]\tau_D\ \tau_A[$ found in the third phase of the precedent step. We obtain two intervals: $]\ \tau_D\ \frac{\tau_A - \tau_D}{2}\ [$, $[\frac{\tau_A - \tau_D}{2}\ \ \tau_A[$. For each of these intervals, a simulation of the ODE model and the test on the outputs are processed. This process is continued until a satisfying interval is found.

5. APPLICATION TO THE AD PROCESS

The data we used to illustrate our method are real measurements recorded on the Anaerobic Digestion (AD) process described in section 2. We focus on the interval from t_D = 49h (that we consider as the start date when the information is lost) and t_A=90h (arrival date) (Table 1). The initial states are as follows: $S_1(t_0) \in [3\ \ 4]$, $S_2(t_0) \in [60\ \ 90]$, $Z(t_0) \in [100\ \ 140]$, $C(t_0) \in [60\ \ 75]$. The thresholds are chosen so as to avoid any intersection between those delimiting the starting state and those delimiting the arrival state. They are chosen from 10% to 50% of the output-measured values.

5.1 *First step*

- First phase

The system is simulated with constant inputs corresponding to the limit of the intervals $Cste_{Di} = \left[In^-_{Di}\ In^+_{Di}\right]$ (See figure 6). Simulation results are presented in figure 7 where one can see the data available before information loss (*i.e.,* before 49h) and after information recovery (*i.e.,* after 90 h). Thin lines corresponds to the model simulation with the indication of the thresholds crossings (the thresholds are the constant values represented by dotted lines in Figure 7).

We can see in figure 7 that the outputs C and Z leave very fast their initial discrete states and the result of the verification with Kronos gives the interval [49 50[.

	Input values	Input values
	at t_D= 49	at t_A=90
	$(In_{Di}=[\ In^-_{Di}\ In^+_{Di}])$	$(In_{Ai}=[\ In^-_{Ai}\ In^+_{Ai}])$
S_{1in}	[17.4 21.7]	[11.31 13.81]
S_{2in}	[118.75 131.25]	[59.21 65.48]
Z_{in}	[198 253]	[44 88]
C_{in}	[34 51]	[17 25.54]

	Output Values	Output Values
	at t_D = 49	at t_A=90
	(Out_{Di})	(Out_{Ai})
S_1	3.42	2.73
S_2	73.42	33.32
Z	126.25	159.41
C	66	132.72
DCO	8.11	4.86
Q_{gas}	412.68	208

Table 1: Measured input and output values

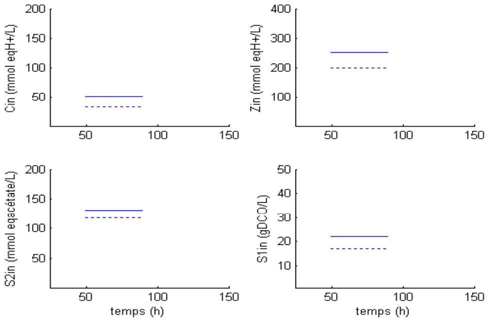


Figure 6: Interval of the inputs

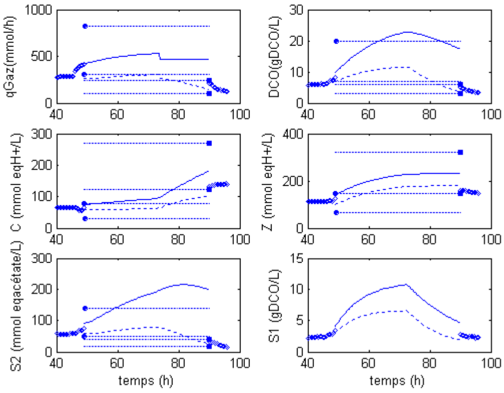


Figure 7: Simulated outputs

- Second phase

Similarly, when checking the arrival states during the second phase, we found the interval]86 90].

- Third phase

In the precedent phases we found t_{D0}=50 and t_{A0}=86. It should be noted that In_{Di}>In_{Ai} which means that there exists a negative step in the input. This is illustrated in figure 8 where are presented the new inputs envelopes.

This process is iterated four times and finally the interval [64 78] is obtained. This result implies that the system stayed for 15 hours in the starting state and for 12 hours in the arrival state.

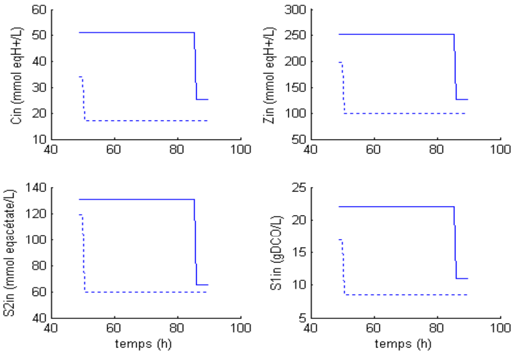


Figure 8: Form of the deduced inputs

5.2 *Second step*

Now that an interval has been found [64 78], we can proceed by dividing it until we obtain satisfying inputs (See Figure 9). We can then deduce that the discontinuity on the process inputs occurred between 69 and 73h and that the system stayed for 20 hours in the start state and for 17 hours in the arrival state. This is in fact exactly what happened in practice (Process inputs were indeed recorded but they were considered as unknown in the present application).

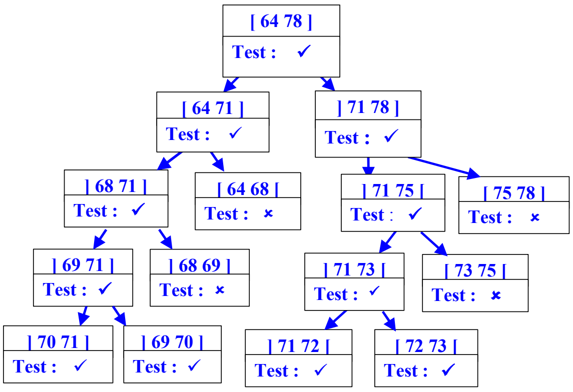


Figure 9: intervals of the second step

These inputs are thus used for final simulation of the model and the simulated model outputs are discretised and the encountered discrete states of the process are analysed using Kronos (see Hélias *et al.*, 2004 for an example of such an analysis). The end user is then advised about the possible discrete process states – and their duration – encountered during the information loss. This knowledge is very important to decide upon the best control strategy to apply after information recovery.

6. CONCLUSION

In this paper we described a method that helps for safe restart of an anaerobic digester, after a period of time where some information (inputs and sate variables) are unknown. In order to operate a safe restart, it is important to have some information related to the behaviour of the system during this period of time of information loss. After the presentation of the physical plant and its modelling, we introduced a method based on both the continuous and the discrete model (automata obtained by temporal abstraction) that aims to explain the behaviour of the system. The inputs of the systems have been limited to three types given by the experts. The first step of the method permits to identify the inputs variation during the information loss. A second step refines the interval of time where a change of the inputs occurred.

REFERENCES

Batstone D.J., Keller J., Angelidaki I., Kalyuzhnui S., Pavlostathis S.G., Rozzi A., Sanders W., Siegrist H., Vavilin V. (2002) "*Anaerobic Digestion Model No.1 (ADM1)",* IWA publishing, London, UK.

Bernard O., Hadj-Sadok Z., Dochain D., Genovesi A., Steyer J.P. (2001) "Dynamical model development and parameter identification for anaerobic wastewater treatment process", *Biotech. & Bioeng.*, **75**(4), 424-439.

Gouzé J.L., Rapaport A., Hadj-Sadok M.Z (2000). "Interval observers for uncertain biological systems", *Ecological Modelling*. **133**(1-2): 45-56.

Graef S., Andrews J. (1974) "Mathematical modeling and control of anaerobic digestion", *Wat.Res.*, **8**, 262-289.

Hélias A., Guerrin F., Steyer J.P. (2004) "Abstraction of continuous system trajectories into timed automata", *7th IFAC Workshop on Discrete Event Systems, WODES'04,* Reims, France, 22-24 Sept. 2004.

Hill D., Bart C. (1977) "A dynamic model for simulation of animal waste digestion", *J. Water Pollut. Control Assoc.*, **10**, 2129-2143.

Kowalewski S., Engell S., Prußig J., Stursberg O. (1999) "Verification of logic controller for continuous plants using timed condition/event-system models", *Automatica*. **35**(3): pp. 505-518.

Lardon L., Punal A., Steyer J.P. (2004) "On-line diagnosis and uncertainty management in biological wastewater treatment processes", *Journal of Process Control*, **14**, 747-763.

Moor T., Raisch J (2002). "Abstraction based controller synthesis for high order monotone continuous systems", *LNCIS* **279,** 247-265.

Mosey F. (1980) "Mathematical modelling of the anaerobic digestion process: Regulatory mechanisms for the formation of short-chain volatile fatty acids from glucose", *Wat. Sci. Technol.*, **15**, 209-232.

Smith H.L. (1995) *Monotone dynamical systems: An introduction to the theory of competitive and cooperative systems*. Mathematical Surveys and Monographs. **41**. 174 p.

Steyer J.P., Bouvier J.C., Conte T., Gras P., Sousbie P. (2002) "Evaluation of a four year experience with a fully instrumented anaerobic digestion process", *Wat. Sci. Technol.*, **45**(4-5), 495-502.

Steyer J.P., Bernard O. (2003) "An example of the benefits obtained from the long term use of mathematical models in biological systems", *4th MATHMOD Int. Symposium on Mathematical Modelling,* Vienna, Austria, February 5-7, 2003, 6 pages on CDROM.

Yovine S. (1997) "Kronos: A verification tool for Real-Time Systems", *Journal of Software Tools for Technology Transfer*, **1**, 123-133.