A DECLARATIVE FRAMEWORK FOR THE CHARACTERISATION OF CYCLIC BEHAVIOUR FOR A CLASS OF HYBRID CONTROL SYSTEMS

Janan Zaytoon, Véronique Carré-Ménétrier, Omar Nour El Alaoui, François Gellot

Laboratoire d'Automatique et de Micro-électronique, Faculté des Sciences, Moulin de la Housse B.P. 1039, 51687 REIMS Cedex 2, FRANCE.
Tel: +33.3.26.91.32.26, Fax: +33.3.26.91.31.06, E-mail: janan.zaytoon@univ-reims.fr

Abstract: This article proposes a hybrid framework based on the use of Constraint Logic Paradigm for the analysis and synthesis of cyclic behaviour for a class of hybrid systems that combines discrete configurations and differential equations. For a given periodical control signal, the analysis problem consists in determining whether the controlled-system behaviour is cyclic, and to characterise the parameters of the working cycle if it exists. The synthesis approach aims at determining the parameters of the periodical controller so as to reinforce a specified cyclic behaviour of the closed-loop system. An example related to a batch system is used for illustration. *Copyright ©2000 IFAC*

Keywords: Control synthesis, Hybrid Dynamic Systems, Cyclic behaviour, Constraint Logic Paradigm (CLP), Petri nets (PN).

1. BACKGROUND

Hybrid systems are dynamical systems, where the behaviour of interest is determined by interacting continuous and discrete dynamics (Antsaklis, et al., 1998). In this paper, we focus on a class of cyclic-behaviour hybrid systems that combines discrete configurations and differential equations, and is subject to periodical discrete-event control signals, i.e. a controller that generates a sequence of periodical events. This class of hybrid systems arises in varied contexts such as batch processes, electromechanical and industrial electronics, traffic control, industrial process control, agro-alimentary industry, etc.

The goal for controlling this class of hybrid systems is to find some (periodic) control policy that reinforces cyclic-behaviour specifications. The characterisation of cyclic behaviour focuses on two complementary approaches referred to as analysis and control synthesis. For a given model of the plant, a given periodical control signal and a given initial values for the continuous variables, the analysis problem consists in determining whether the modelled-system behaviour is cyclic (or not), and to characterise the parameters of the working cycle if it exists. The synthesis approach aims at determining the parameters of the periodical controller so as to reinforce a specified cyclic behaviour of the closed-loop system.

Hybrid extensions of Petri nets (PN) offer some remarkable facilities for the modelling of this class of hybrid systems. These facilities include the graphical representation and the ability to capture concurrency, synchronisation and conflicts, allowing to model systems with continuous flows in an intuitive way. Preliminary works of the authors (Robert et al. 1998; Nourelalaoui et al. 1999) have therefore investigated the use of Hybrid PN (Le Bail et al., 1991) and Mixed PN (Valentin, 1998; Champagnat et al., 1997) for the modelling and analysis of electromechanical and batch systems belonging to the mentioned class of hybrid systems. The formal analysis tools provided by Hybrid PN were shown to be suitable to identify the existence of functioning cycles, and to determine their period. However, the use of Hybrid PN is limited to systems with linear evolutions (i.e., dx/dt is constant). Mixed PN were therefore used to deal with systems having complex continuous dynamics. In these nets, non-linear differential equations can be associated with the discrete states of the PN. However, the cyclic behaviour of the resulting model can only be analysed by simulation, and the interpretation of simulation results is usually very complex, even in the case of simple systems that are subject to simple periodic control signals.

The work presented in this paper aims therefore at proposing a computer aided framework that benefits from the modelling facilities of Mixed PN while providing analysis and synthesis backbone for this model. The different elements of this framework, which is based on the use of Constraint Logic Paradigm (CLP) (Benhamou et al., 1993; Codonet, 1995), are detailed in section 2. An illustrative example of a batch system is presented in section 3.

2. HYBRID FRAMEWORK BASED ON CLP

A declarative framework based on Constraints Logic Programming (CLP) is proposed here to provide a formal support for the analysis and the synthesis of cyclic behaviour for systems modelled using mixed PN. In the first place, mixed PN are translated into CLP clauses. The reversibility related to the declarative aspect of CLP, and the underlying mechanisms used by its inference motor (matching, unification, instantiation, backtracking, ...) allows the study of the cyclic behaviour of the resulting CLP model both from the analysis and the synthesis points of view. The resolution is computed as an answer to a Prolog4 program which incorporates the dynamics of the problem as well as the logical constraints and goals. The resolver of Prolog4 is used to solve algebraic/differential equations and takes into account constraints expressed in terms of real and Boolean variables, as well as other complex data structures. This resolver determines the search space required for the resolution (bindings) of the free (non-bounded) variables of the system, which correspond to the required analysis results or to the parameters to be synthesised.

2.1. CLP Structure

The first step consists of rewriting the mixed PN and the sequence of periodic control events (within a control period) into CLP clauses.

Each place of the net together with the set of associated differential equations will be represented by a clause that represents the system configuration corresponding to the place. This clause is : *Place*(K, X_K, T_i, T_o, F_K), where K is the number of the place, X_K is the set of associated continuous variables, T_i is the set of input transitions of K, T_o is the set of output transitions, and F_K is the set of algebraic and differential equations associated to K. These equations are considered as constraints on the evolutions of the continuous variables associated to the place.

A clause is also associated to each of the transitions of the PN to capture its input and output places, its firing condition (guard), the triggering control event and its jump function (which initialises the values of the continuous variables when the transition is fired). In the proposed framework, only PN of the state-graph type are used; that is, each transition has only one input and one output place. A transition is rewritten to the clause: *Transition*(T, P_i, P_o, $Guard_T$, $Jump_T$), where T is the number of the transition, P_i is the input place and P_o is the output place of T. $Guard_T$ represents the enabling condition of the transition, given in terms of internal and/or control events, whereas $Jump_T$ is the set of functions that initialises the values of continuous variables when the transition occurs.

The control scheduler comprises the control events associated with their occurrence instants within the control period. It is rewritten into a CLP clause of the form : *Scheduler*([e_1,θ_1], [e_2,θ_2], ..., [e_n,θ_n]), where e_i is the event occurring at date θ_i.

The variables involved in the CLP clauses representing the places, the transitions and the scheduler can be instantiated or unbounded according to whether the model will be used for analysis or synthesis.

2.2. Analysis

Given a fully instantiated CLP structure of the modelled system and a periodical control signal, the analysis approach is used to certify whether the controlled system exhibits cyclic behaviour or not, and to determine the parameters of the cycle if it exists. Analysis (Fig. 1) proceeds by executing the CLP model subject to the extracted periodic control events, and verifying whether there exists a configuration that is activated twice with the same initial values of continuous variables. The matching/unification mechanism of CLP allows an efficient search for equivalent configurations inserted in the knowledge base through a previous iteration of the analysis procedure. When such an equivalent configuration exists, the analysis procedure terminates and the parameters of the identified cycle are calculated. The number of iterations of the analysis procedure is minimal since the resolver is only required to calculate the initial and the final values of the continuous variables for each of the consecutively activated configurations of the system.

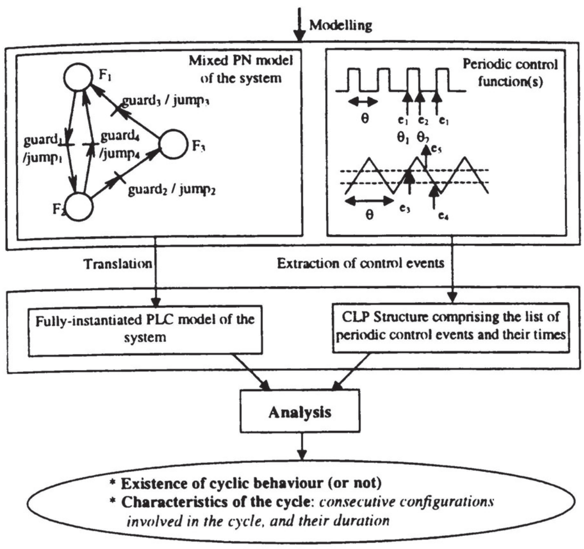

Fig 1. Analysis of cyclic behaviour

A simplified CLP-like formulation of the analysis procedure is given by the following clause:

Analysis (C, θ, Xi_C) *: -*
C_Evolution$(Xi_C, Xf_C, F_C, T, Next)$,
Insert$(state(C, \theta_C, Xi_C))$,
Search$(state(C, \theta_C, Xi_C))$,
D_Evolution$(T, Jump_T, Xf_C, Xi_{Next}, \theta_C)$,
Analysis$(Next, \theta + \theta_C, Xi_{Next})$

C (respectively, *Next*) is a reference to the current (next) configuration, θ is a global variable representing the time. Xi_C (Xi_{Next}) is a set giving the values of the variables of C (*Next*) when the configuration is activated, and Xf_C is the set of values of the variables of C when the configuration is to be deactivated. The analysis procedure goes through the base of facts, which comprises the completely instantiated clauses: *Place, Transition* and *Scheduler.*

Starting from the clauses *Place, Transition, Scheduler* corresponding to the initial configuration C, the clause *C_Evolution* detects the first occurrence of a control or an internal event, and the corresponding transition T (leaving C). The date of occurrence of the first event corresponds to the sojourn time for the current configuration, θ_c. This date is also used to determine the set of values of the variables when the configuration C is deactivated, Xf_C, by applying the equation related to the current configuration, F_C. The output transition T is also used to identify the following configuration, *Next.*

The configuration C and its associated parameters are inserted in the knowledge base using the following structure: $state(C, \theta_c, Xi_C)$. The clause, *Search*, goes through the knowledge base to determine whether the currently inserted *state* in the knowledge base has identical values of C, Xi_C and θ_c with respect to a previously inserted *state*. When a match is found, the set of configurations inserted between these two *states* represent the identified functioning cycle of the controlled converter. In this case, the cycle time is given by the sum of the sojourn times in these configurations. To compensate for eventual numerical imprecision of the mathematical resolver, a precision factor is taken into account when comparing the values of Xi_C and θ_c for two configurations instead of strict equivalence.

If a match is found, the analysis terminates, by virtue of another (terminating) *Analysis* clause, not represented here. Otherwise, the clause *D_Evolution* applies the jump function of T, $Jump_T$, to determine the values of continuous variables when the next configuration is initialised Xi_{Next}, on the basis of the values of variables (Xf_C) when the previous configuration is deactivated. Finally, the analysis is iterated again with a current configuration equal to the next configuration of the previous iteration, and a time $\theta = \theta + \theta_c$.

To avoid infinite iterations in the case of non-cyclic systems, our current research work aims at establishing heuristics to determine a limit on the number of program iterations. This limit depends on the complexity of the model, the precision factor introduced to compensate for calculation rounding errors, and on the existence of certain relationships between the values of continuous variables when a discrete configuration is reactivated.

2.3. Synthesis

The synthesis problem aims at determining a number of control and/or model parameters that reinforce a (completely or partially) specified cyclic behaviour of the system, given in terms of an ordered sequence of configurations (Fig. 2). The parameters related to these configurations (activation interval, values of variables at the instants when the configuration is activated and/or when it is deactivated) may either be fully or partially specified. Many parameters can be synthesised in the proposed approach. In the PN model, some of the unbounded variables used in the transition guards or in the jump functions may be calculated by the synthesis procedure. This allows, for example, to determine the required current value (and hence the calibrating of a current sensor) to trigger a control action in an electronic converter. Control parameters to be synthesised are related to the determination of the periodical control events and their times. These events are initially given either as an unordered list of possible events, or as a (totally or partially) ordered sequence of events comprising the control period. The occurrence time of some of these events may also be specified in advance.

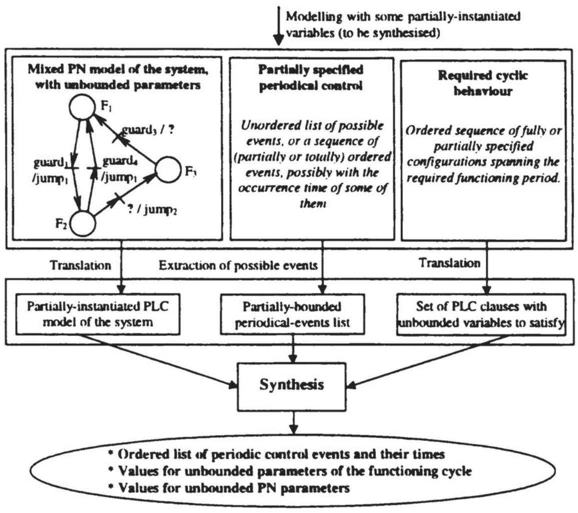

Fig 2. Synthesis

The synthesis approach consists of finding the most general instantiation of unbounded variables to execute the specified sequence of configurations through the transition structure of the CLP model of the mixed PN. The general form of the synthesis clause is the following:

Synthesize ($[C_1, C_2, \ldots\ldots, C_n]$, $[P_1, P_2, \ldots\ldots, P_p]$) :-
Discrete_transition($T_C, C_1, C_2, GUARD_{TC}, JUMP_{TC}$),
C_Evaluate($Xi_{C1}, Xf_{C1}, T, \theta_{C1}, F_{C1}$),
D_Evaluate($Jump_T, Xf_{C1}, Xi_{C2}, \theta_{C1}$),
Discrete_transition($T_C, C_2, C_3, GUARD_{TC}, JUMP_{TC}$),
....
....
Discrete_transition($T_C, C_n, C_1, GUARD_{TC}, JUMP_{TC}$),
C_Evaluate($Xi_{Cn}, Xf_{Cn}, T, \theta_{Cn}, F_{Cn}$),
D_Evaluate($Jump_T, Xf_{Cn}, Xi_{C1}, \theta_{Cn}$),

The list $[C_1 \ldots C_n]$ correspond to the consecutive configurations comprising the required cycle. Each parameter in the list $[P_1 \ldots P_p]$ refers to a variable (used in the sub-clauses) whose value is known in advance or is to be determined by the synthesis procedure. The *Discrete_Transition* clause is used to identify the set of possible transitions, T_C, between the current and the next configuration, and to capture the guard and the jump functions of these transitions. The clause *C_Evaluate* is then used to determine the transition $T \in T_C$ that will be taken to bring the system to the next configuration, as well as the sojourn time, θ_{Ck}, in the current configuration. This clause proceeds by applying the equation related to the current configuration, F_{Ck}, and detecting the first time instant at which a guard of an output transition becomes *true*. This instant allows to determine the sojourn time and to identify the output transition, T, to be taken. The jump function of this transition, $Jump_T$, is applied by the clause *D_Evaluate* to determine the values of continuous variables when the next configuration is initialised, on the basis of the values of variables when the previous configuration is left. Thus, the clause *C_Evaluate* corresponds to the invocation of the resolver of CLP to determine the most general instantiation of the variables Xf_{Ck}, and the clause *D_Evaluate* determines the most general instantiation for the variables $Xi_{C(k+1)}$. The automatic backtracking of CLP is exploited to search for the alternative possible transition sequences that satisfies the synthesis problem. Optimisation criteria can also be used to choose the best possible solution.

3. EXAMPLE

The two-tank process presented in Fig. 3 is used for illustration. The control objective is to maintain the level in tank 2, x_2, within the range $x_{2refmin} \le x_2 \le x_{2refmax}$, and to satisfy the following constraints: $x_{1min} \le x_1 \le x_{1max}$ and $x_{2min} \le x_2 \le x_{2max}$. The flow through the valve is given by $Q_1 = q_1.u$, where u is a Boolean control variable: the valve is controlled to be closed (opened) if u=0 (u=1). The levels in tank 1 and in tank 2 are given by x_1 and x_2, respectively, whereas the parameters R_1, S_1, R_2 and S_2 represent the internal characteristics of the corresponding tanks.

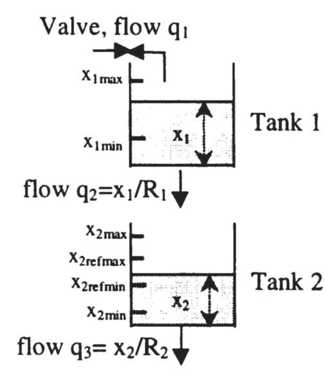

Fig. 3. A two-tank process

The system has two distinct configurations represented by places P_1 (closing the valve) and P_2 (opening the valve) of the mixed PN depicted in Fig. 4. The intermediate place P_3 represents the states where $x_2 \ge x_{2refmax}$ and $x_1 \le x_{1min}$ (see the guard associated to the input transition of P_3) which should entail both the closing of V_1 (because $x_2 \ge x_{2refmax}$) and the opening of this valve (since $x_1 \le x_{1min}$). To alleviate this ambiguity, the valve is ordered to open in P_3, since $x_1 \le x_{1min}$ corresponds to a violation of a safety constraint. The output transition of P_3 is fired when x_1 attains a certain level, x_{1m}, that will be used below as a parameter to analyse. The place P_4 has a similar role to that of P_3, but it is related to closing the valve when $x_1 \ge x_{1max}$ and $x_2 \le x_{2refmin}$.

The dynamic behaviour of the system is non-linear and is described by algebraic-differential equations associated to the places of the PN. The parameter δ used by the guards of transitions between P_1 and P_3 is a tolerance level that is introduced to compensate for the delay between an order (closing or opening the valve) and its consequential effect on the level x_2. The coupled variables x_1 and x_2 have no discontinuities, so an identity jump function (not represented in Fig. 4) is associated to all the transitions.

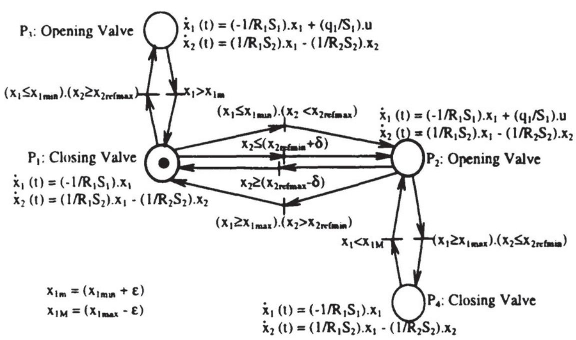

Fig. 4. Mixed PN model

Figure 5 presents some analysis results for different initial values of x_1 and x_2, and for different values of x_{1m} and x_{1M}, as indicated in the figure. Figures 5a and 5b correspond to initial values within the range ($x_{1min} \leq x_1 \leq x_{1max}$ and $x_{2refmin} \leq x_2$), whereas this range is ($x_1 \leq x_{1max}$ and $x_2 \leq x_{2refmin}$) in Fig. 5c and 5d. The values of system parameters are the following: R_1=20 s/m^2, S_1=1 m^2, R_2=18 s/m^2, S_2=2,5 m^2, q_1=0.03 m^3/s, x_{1min}=0.1 m, x_{1max}=0.4 m, x_{2max}=0.4 m, x_{2min}=0.1 m, $x_{2refmin}$=0.23 m, $x_{2refmax}$=0.26 m, δ=0.01 m.

In Fig. 5a and 5b, we note that a periodic cycle is established after a transition phase whose duration depends on the value of x_{1m}, whereas Fig. 5c and 5d show the influence of the choice of x_{1M} on the establishment of cyclic behaviour. For example, the list of the consecutive states inserted by the *Analysis* clause is given in Fig. 5a. This list highlights a functioning cycle because states 16 and 14 represent the same configuration with identical values for the sojourn time and the continuous variables at initialisation.

Starting from a given configuration, different parameters can be synthesised to reinforce a desired cyclic behaviour. In Fig. 6, the desired functioning cycle is given as an alternation between the two configurations corresponding to the opening and the closing of the valve.

- In Fig 6a, knowing that the level in tank 1 is situated somewhere between x_{1max}=0.4 et x_{1min}=0.15, the synthesis procedure is used to determine the initial value of x_2, the value of this variable at the commutation instant, and the sojourn time in each configuration.
- In Fig 6b, the limit values are given (x_{1max}, x_{1min}, $x_{2refmax}$ and $x_{2refmin}$ have the same values as for the previous case) and the required functioning cycle is characterised by an equivalent sojourn time in each of the two configurations. The synthesis procedure is used here to determine the sojourn time, the flow q_1, the initial values for the levels in the two tanks and their values at the commutation instant.

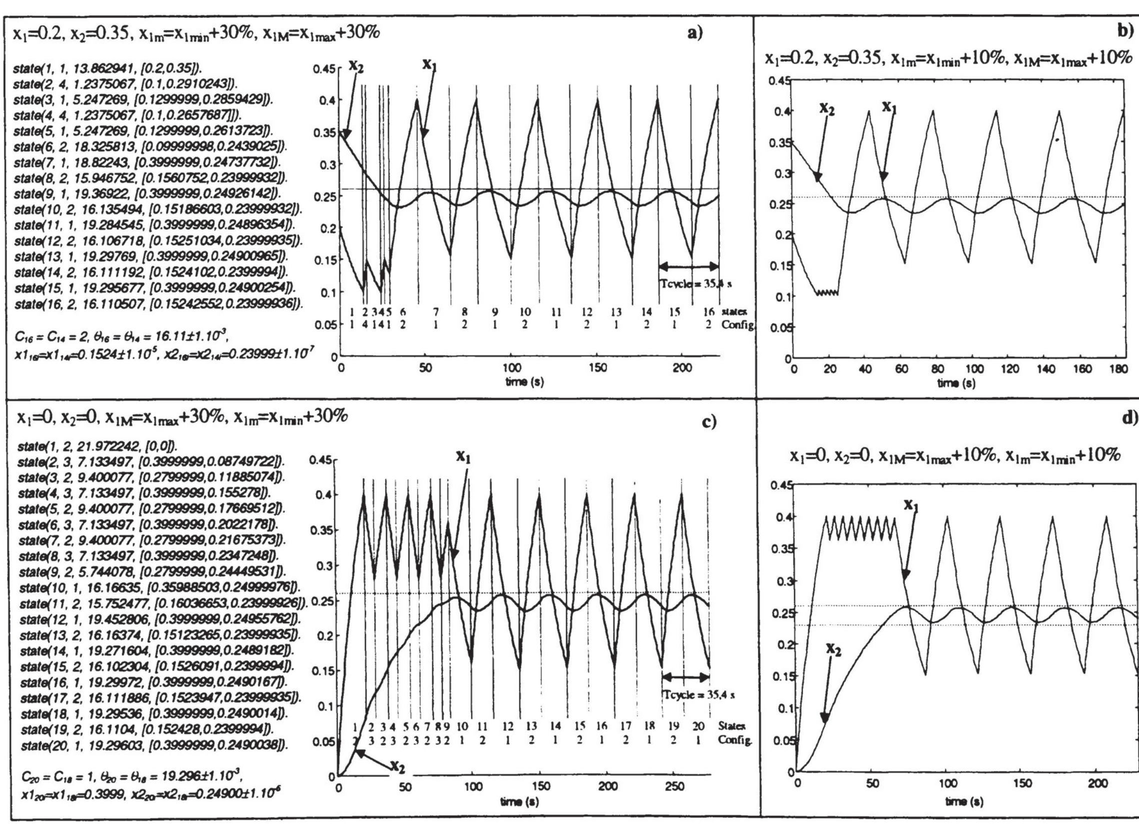

Fig. 5. Analysis results

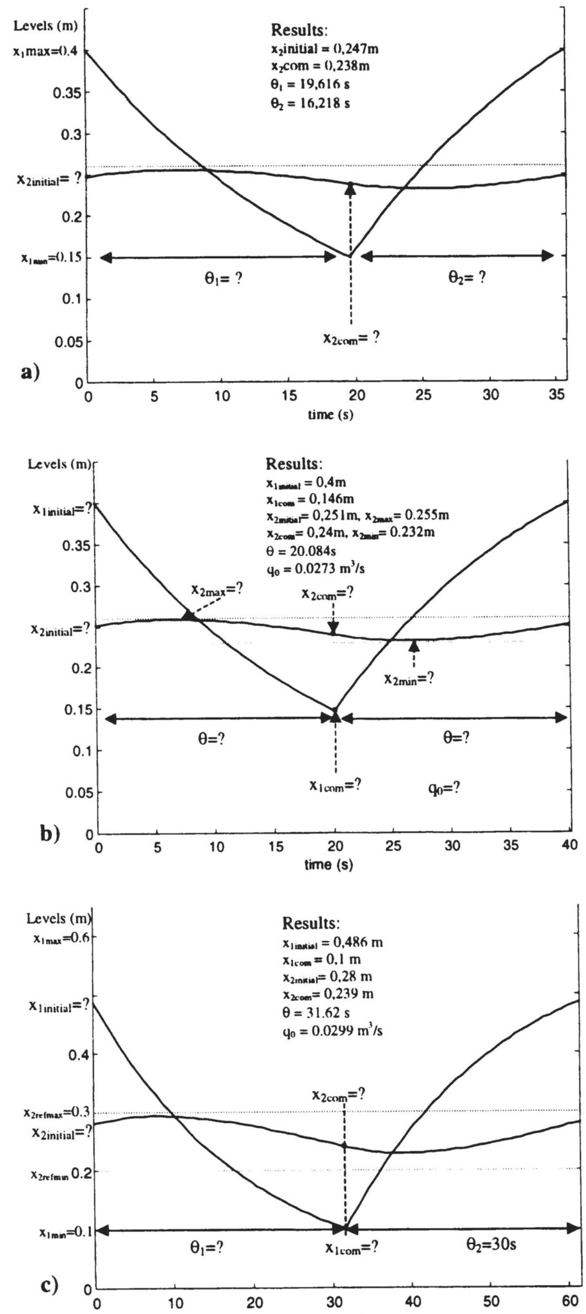

Fig. 6. Synthesis results

- In Fig 6c, the sojourn time in the opening configuration is given, and the parameters to synthesise are the sojourn time in the closing configuration as well as the different levels for the tanks.

Internal tank parameters, such as the diameter, can also be synthesised, but this requires a linearisation of the exponential terms in the equations.

4. CONCLUSION

This paper has presented a declarative framework, based on the used of Constraint Logic Paradigm, for the characterisation of cyclic-behaviour hybrid systems. The reversibility related to the declarative aspect of CLP, and the underlying mechanisms used by its inference motor allows the study of the cyclic behaviour both from the analysis and the synthesis points of view.

Current research work is undertaken to establish heuristics allowing to determine the number of iterations required for analysis, and to formalise the research of optimal solutions for the synthesis framework. Symbolic integration techniques are also investigated so as to solve non-linear differential equations within the proposed framework.

REFERENCES

Antsaklis, P., X. Koutsoukos and J. Zaytoon (1998), On hybrid control of complex systems: a survey. *European Journal of Automation APII-JESA*, **32** (9-10), 1023-1045

Benhamou, F. and A. Colmerauer (1993), In: *Constraint Logic Programming*, Selected Research, MIT Press, Cambridge-USA.

Champagnat, R., P. Esteban, H. Pingaud and R. Valette (1997), Modelling hybrid systems by means of high-level Petri nets: Benefits and limitations. *IFAC CIS'97 Conference*, vol 1, Belfort-France, 469-474.

Codonet, P (1995). Programmation logique avec contraintes : une introduction, *APII*, **14** (6), 665-692.

LeBail, J., H. Alla and R. David (1991). Hybrid Petri Nets, *European Control Conference*, Grenoble-France, 1472-1477.

Nourelalaoui, O., F. Gellot, V. Carré-Ménétrier and G. Villermain-Lecolier (1999). Analyse de systèmes électrotechniques à l'aide des réseaux de Petri en vue de leur commande. *3rd Int. Conference on Industrial Automation*, Montréal-Canada, 18.1-18.4.

Robert, B., O. Nourelalaoui, F. Gellot and V. Carré-Ménétrier (1998). Le modèle réseaux de Petri en électrotechnique. *3rd Int. Conference on Automation of Mixed Processes ADMP'98*, Reims-France, 383-389

Valentin-Roubinet, C. (1998). Modelling of hybrid systems: DAE supervised by Petri nets the example of a gas storage, *3rd Int. Conference on Automation of Mixed Processes ADMP'98*, Reims-France, 142-149.