Preprints of the 2nd IFAC Conf. on Analysis and Design of Hybrid Systems (Alghero, Italy), 7-9 June 2006

TIMED DISCRETE EVENT CONTROL OF A PARALLEL PRODUCTION LINE WITH CONTINUOUS OUTPUT [1]

Dmitry Gromov * **Stephanie Geist** * **Jörg Raisch** *, **

* *Fachgebiet Regelungssysteme*
Technische Universität Berlin, Germany
Email: *gromov@control.tu-berlin.de*
** *Systems and Control Theory Group*
Max-Planck-Institut für Dynamik komplexer technischer Systeme, Magdeburg, Germany

Abstract: In this paper we present an approach to formulate and solve certain scheduling tasks using timed discrete event control methods. To demonstrate our approach, we consider a special class of systems: a cyclically operated chemical plant with parallel reactors using common resources and a continuous output. This problem was motivated by a benchmark proposed within the EU Network of Excellence HYCON. For this class of systems, we show how to pose the control problem within a discrete event framework by modelling system components as multirate timed automata. Safety and nonblocking are investigated. These properties have to be achieved in the presence of a class of bounded errors/disturbances.

Keywords: Parallel Production Line, Scheduling, Timed Automata

1. INTRODUCTION

In this contribution, we investigate a *"parallelised" production line with resource constraints and continuous output*. Such a plant has been proposed as a case study by the Université Catholique de Louvain for the EU Network of Excellence HYCON (Simeonova *et al.*, 2005). In this example two parallel reactors sharing resources as, e.g., reactants, hot steam, cool water, are considered. The reactors are discharged into a shared storage tank that has a continuous outflow. The plant is cyclically operated. For this hybrid system, the goal is to assure non-conflicting work of these reactors and to prevent over- and underfilling of the tank.

We present an approach to the scheduling problem guaranteeing non-blocking and safety despite disturbances, using a timed automata formulation of the problem. Motivated by the HYCON case study we consider a generalised problem consisting of an arbitrary number of parallel reactors, an arbitrary number of shared resources and one storage tank.

Describing scheduling problems in a discrete event framework allows a very intuitive way of problem formulation. All system components including the resources can be considered as subsystems which can be easily described by timed automata and subsequently composed to form the overall problem.

[1] Work partially done in the framework of the HYCON Network of Excellence, contract number FP6-IST-511368

The advantage of using a formal approach, as opposed to heuristic strategies, is that desired properties can be guaranteed.

It is well known that only certain classes of timed automata systems are computationally tractable. For these classes, however, there are various numerical and symbolical methods for analysis and computation. Numerical methods are described in (Pettersson, 1999; Bengtsson and Yi, 2004; Bozga *et al.*, 1998) and symbolical methods are presented, for instance, in (Asarin *et al.*, 1995).

This paper is arranged as follows: in Section 2, we give a formal description of the overall problem. A short introduction to multirate timed automata is given in Section 3. In Section 4, the modelling of the system components is described. In Section 5, we address control issues and give some remarks on implementation. Finally, in Section 6, we apply our approach to the HYCON benchmark example described in (Simeonova *et al.*, 2005).

2. PROBLEM STATEMENT

Figure 1 presents a schematic view of the chemical plant considered in the sequel. The system consists of n parallel reactors with k common resources, e.g., reactants, cold/hot water supplies and pumps. The reactors are discharged into one tank that acts as an output buffer and has the continuous output flow $F_{out,t}$. The volumetric flow $F_{out,r}$ during the discharging of a reactor is fixed, the output flow of the tank $F_{out,t}$ can be adjusted within a given range. In each reactor the same process is performed. The goal is to assure non-conflicting use of resources and to keep the level of the tank volume between given values v_{max} and v_{min}. Furthermore, due to safety reasons, the tank outflow may not be interrupted.

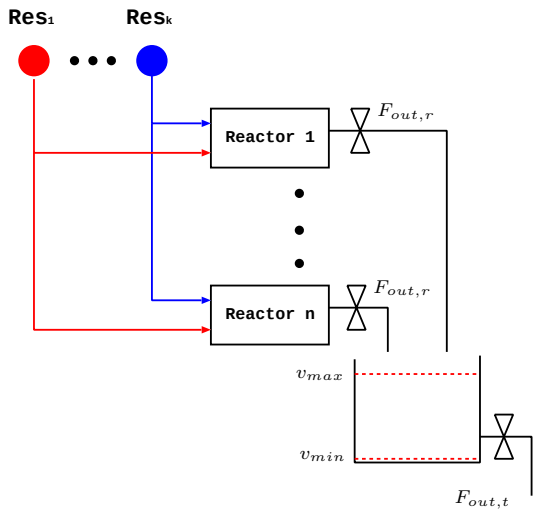

Fig. 1. A parallelised production line with resource constraints

A production cycle in the j-th reactor consists of a set of operations: $O_j = \{o_{ij}\}$, $i = 1 \dots m$, e.g., heating, cooling, reaction, discharging and so one, which, in turn, are characterised by their processing times d_{ij}. The temporal ordering of these operations is given. We will also consider a case when the processing times of operations are not fixed, but an upper and lower bound is known: $d_{ij} \in [d_i^* - \underline{d}_i; d_i^* + \overline{d}_i]$. These deviations in processing times may be caused by disturbances.

There are a set of resources R and sets of "resource-sensitive" operations $O'_j \subset O_j$, $j = 1, \dots, n$. A map $r_j : O'_j \to R$ associates a resource to each operation $o_{ij} \in O'_j$. Here we assume that these maps are injective, i.e. resources are used only once within the reaction cycle of a reactor but multiple use by different reactors is allowed.

Due to technological or safety requirements some operations must be processed without delay. These operations are grouped into tasks $K_j^l = \{o_{ij}^l\}, K_j^{l_1} \cap K_j^{l_2} = \emptyset$. Each task must contain at least one resource-sensitive operation, i.e. $K_j^l \cap O'_j \neq \emptyset$ where, within a task, delays are not permitted. We assume that an isolated operation also forms a task if it is resource-sensitive. Otherwise, it can be joined with the neighbour task. Hence, each operation belongs to some task, $\bigcup\limits_{j,l} K_j^l = O = \bigcup\limits_{j} O_j$. We denote the set of all tasks by $K = \{K_j^l\}, j = 1, \dots, n, l = 1, \dots, r$.

3. MULTIRATE TIMED AUTOMATA

Timed automata (TA) (Alur and Dill, 1994) are finite automata augmented with continuous clocks whose values grow uniformly at each discrete state. The set of clock variables is denoted by X. A clock valuation ν for the set X assigns a real value to each clock. Clocks can be reset to zero at certain transitions. There are also *clock constraints* $\Phi(X)$ defined over X in the following way:

$$\phi := x \leq c \mid x < c \mid x \geq c \mid x > c \mid \phi_1 \wedge \phi_2.$$

That means that each clock constraint can be represented as an union of inequalities. A transition may be equipped with a clock constraint which is interpreted as an enabling or guard condition. Clock constraints attached to locations can be interpreted as invariants. In this paper, a transition condition with upper time limit ∞ occurs only for the transitions whose switching can be controlled. Hence, we can assume that these transitions switch at the first possible time instant.

We consider an extended class of timed automata, namely *multirate timed automata* (Alur

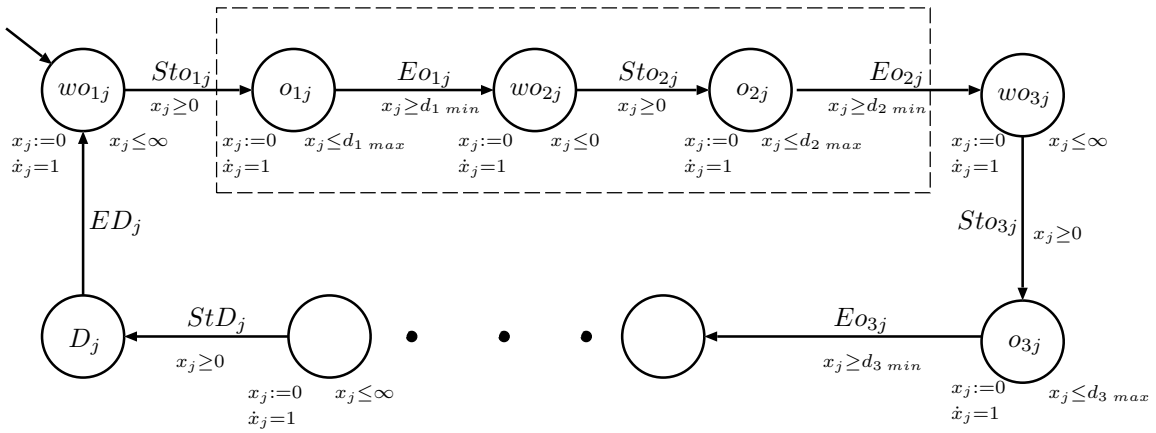

Fig. 2. A timed automaton model of reactor j

et al., 2000). A multirate TA is a tuple $\mathcal{T} = (L, L_0, \Sigma, X, I, E, c, \lambda)$, where

- L is a finite set of locations, and $L_0 \subseteq L$ is a set of initial locations,
- Σ is a finite set of labels (events),
- X is a finite set of clock variables,
- I is a map that associates some clock constraints in $\Phi(X)$ to each location, i.e. $I : L \rightarrow \Phi(X)$,
- $E \subseteq L \times \Sigma \times \Phi(X) \times L$ is a set of transitions,
- $c : L \times X \rightarrow \mathbb{R}$ is a function that defines the rate of change of each clock in a certain location. Thus, the clock dynamics in location l can be described by a simple differential equation $\dot{x}_i = c_{l,i} = const$. If $c_{l,i}$ is equal to 1 for all indices l, i, we deal with the standard timed automaton.
- $\lambda : E \rightarrow 2^X$ associates to each transition a set of clocks to be reset.

Furthermore, we will use the standard definition of the product for timed automata (Alur and Dill, 1994). Let $A_1 = (L_1, L_{01}, \Sigma_1, X_1, I_1, E_1, f_1, \lambda_1)$ and $A_2 = (L_2, L_{02}, \Sigma_2, X_2, I_2, E_2, f_2, \lambda_2)$ be two timed automata. Assume that the clock sets X_1 and X_2 are disjoint. Then, the product, denoted $A_1||A_2$, is the timed automaton $(L, L_0, \Sigma, X, I, E, f, \lambda)$, where $L = L_1 \times L_2$, $L_0 = L_{01} \times L_{02}$, $\Sigma = \Sigma_1 \cup \Sigma_2$, $I = I_1 \cap I_2$ and $X = X_1 \cup X_2$. The transition structure E and the reset function λ are defined by the following rules:

(1) $\sigma \in \Sigma_1 \cap \Sigma_2$: for every $e_1 = (l_1, \sigma, \phi_1, l_1') \in E_1$ and $e_2 = (l_2, \sigma, \phi_2, l_2') \in E_2$, $e = ((l_1, l_2), \sigma, \phi_1 \wedge \phi_2, (l_1', l_2')) \in E$ and $\lambda(e) = \lambda_1(e_1) \cup \lambda_2(e_2)$.

(2) $\sigma \in \Sigma_1 \setminus \Sigma_2$: for every $e_1 = (l_1, \sigma, \phi_1, l_1') \in E_1$ and $l_2 \in L_2$, $e = ((l_1, l_2), \sigma, \phi_1, (l_1', l_2)) \in E$ and $\lambda(e) = \lambda_1(e_1)$.

(3) $\sigma \in \Sigma_2 \setminus \Sigma_1$: for every $e_2 = (l_2, \sigma, \phi_2, l_2') \in E_2$ and $l_1 \in L_1$, $e = ((l_1, l_2), \sigma, \phi_2, (l_1, l_2')) \in E$ and $\lambda(e) = \lambda_2(e_2)$.

4. TIMED AUTOMATON MODEL OF THE PLANT

4.1 Reactors

The first step is the modelling of the reactors using timed automata. Since the operation sequence is identical in each reactor, they can be described in a uniform way, as shown in Fig. 2.

There are two types of locations: wo_{ij} and o_{ij} which mean "wait before i-th operation starts in reactor j" and "i-th operation is active in reactor j". The events Sto_{ij} and Eo_{ij} denote start and end of the i-th operation in the j-th reactor, respectively. Fig.2 is to be interpreted as follows: in location o_{ij}, the progress of time is measured by a clock modelled by $\dot{x}_j = 1$, and the clock is reset to zero when the location is entered, i.e. when event Sto_{ij} occurs. We are only allowed to stay in the location if $x_j \leq d_{i\ max}$ holds (invariant). The event Eo_{ij} may only occur if $x_j \geq d_{i\ min}$ holds (guard). Hence, the transition between location o_{ij} to location $wo_{(i+1)j}$ has to happen when $d_{i\ min} \leq x_j \leq d_{i\ max}$. In location wo_{ij}, there are two possibilities: either invariant and guard of the outgoing transition enforce an immediate switch to the next location (an example for this case is location wo_{2j} in Fig.2), or invariant and guard allow an arbitrary stay within the location (an example for this case is location wo_{3j} in Fig.2). If a wo_{ij}-location is of the former type, $o_{(i-1)j}$ and o_{ij} belong to the same task (see the dashed box in Fig.2).

The last operation, denoted by D_j, is the discharging of reactor j. Note that the operation "discharging" always represents a task since the output tank can be interpreted as an external resource.

4.2 Resources

The next step is to model the restrictions on resource availability. The simplest way is to build a finite automaton for each resource R_i and the corresponding operations $o_{ij} = r_j^{-1}(R_i) \in O'_j,\ j = 1, \ldots, n$ as shown in Fig.3. The depicted timed automaton represents a simple rule: a resource sensitive operation can be simultaneously carried out in one reactor only.

To enforce uniqueness of the solution, a sequence of reactors is predetermined. As all reactors are equal, this does not restrict generality.

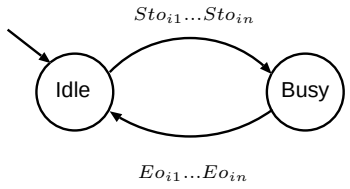

Fig. 3. A finite automaton modelling the availability of resource i

4.3 Output tank

The last element of the plant is the output tank. Its timed automaton model model is presented in Fig. 4. The transitions StD and ED denote "start discharging" and "discharging is finished". The clock variable v models the amount of liquid in the tank. Here, $a = F_{out,r} - F_{out,t}$ and $b = -F_{out,t}$, where $F_{out,r}$ is the volumetric rate of the flow from any reactor j to the tank, whereas $F_{out,t}$ is the volumetric rate of the output flow from the tank. If the value of v becomes too small or too big, the automaton goes to one of the locations modelling a forbidden situation.

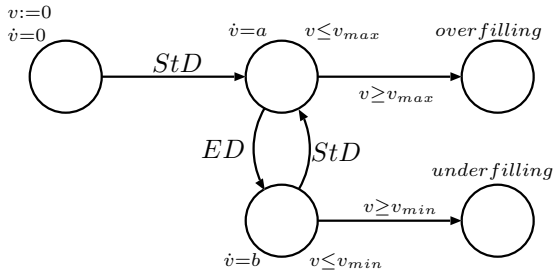

Fig. 4. A timed automaton modelling the output tank

5. CONTROL ISSUES

Two specifications have to be enforced: 1. the closed-loop system must be non-blocking; 2. the closed-loop system must be safe, i.e. the locations *underfilling* and *overfilling* must be rendered unreachable..

In the following we present several possible control strategies to enforce the specifications. Note that we can provide a formal guarantee for the specifications to hold, even if the actual design process contains some heuristics.

5.1 Non-blocking

It is obvious that the synchronous product of the n reactor models (Fig.2) and the k resource availability models (Fig.3) may give rise to blocking. This may happen if a resource, say R_i, is being allocated by an operation o_{ij} in reactor j, an operation $o_{(i-1)k}$ is finished in reactor k and operation o_{ik} belonging to the same task as $o_{i-1,k}$ attempts to allocate R_i. In this situation, o_{ik} must start at the same time as $o_{(i-1)k}$ finishes, which is clearly impossible as the corresponding resource is being used by another reactor. To avoid this situation, the start of tasks has to be delayed appropriately. This is being done by assigning one timed automaton to each task $K_i^l = \{o_{\mu j}^l\},\ \mu = 1, 2, \ldots$. We assume that the index μ describes the temporal ordering of operations within the task (Fig.5). These automata are similar to the resource models, but with additionally introduced time constraints. The first location is added because in the beginning of the process the operation can start immediately.

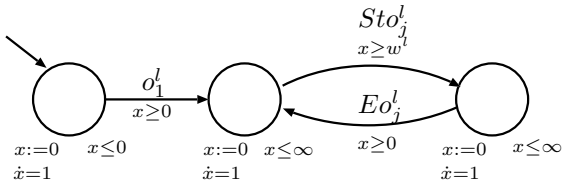

Fig. 5. A timed automaton modelling resource availability

The parameter w^l can be found as a solution of a simple scheduling problem. There is a fixed relation between the temporal position of an operation within the task and within the overall sequence of operations in the reactor. The latter is denoted by i and given by $i = \sum_{q=1}^{l-1} |K_j^q| + \mu$. The start and end times of $o_{\mu j}^l$ are $s_{\mu j}^l$ and $f_{\mu j}^l$ and its durations is $d_{\mu j}^l$. The goal is to ensure that

$$\min_{\mu: o_{\mu j}^l \in O'_j} (s_{\mu,j+1}^l - f_{\mu j}^l) = 0, \qquad (1)$$

where $s_{\mu 1}^l = s_{11}^l + \sum_{k=1}^{\mu-1} d_{k1}^l,\ f_{\mu 1}^l = s_{\mu 1}^l + d_{\mu j}^l,$

$s_{\mu,j}^l = f_{1(j-1)}^l + w^l + \sum_{k=2}^{\mu-1} d_{kj}^l,\ d_{\mu j}^l = d_\mu^l,\ j > 1.$

This can be done easily. The situation becomes more complicated if we suppose that the processing durations are only known unprecisely, i.e.,

$d^l_{\mu j} \in [d^*_i - \underline{d}_i; d^*_i + \overline{d}_i]$. Condition (1), then takes the form

$$\min_{\mu: o^l_{\mu j} \in O'_j} \min_{d^l_{\mu j}, d^l_{\mu(j+1)} \in [d^*_i - \underline{d}_i; d^*_i + \overline{d}_i]} (s^l_{\mu(j+1)} - f^l_{\mu j}) = 0. \quad (2)$$

The main difference, compared to the previous case is that the processing time of the first operation is a priori unknown. Thus, w^l is calculated in a worst-case fashion and is therefore, conservative.

5.2 Safety

In a first step, we form the synchronous product of the reactor models (Fig.2), the extended resource availability models (Fig.5) and the tank model (Fig.4). As we have previously determined suitable waiting times w^l, in this step only nonblocking schemes are considered. If all processing times are known, using standard verification procedures, we can check whether the locations "Overfilling" and "Underfilling" can be reached for a given fixed outflow rate. If yes, it is an easy exercise to suitably adjust the outflow rate.

The analysis shows that for fixed and physically reasonable processing times a fixed outflow can be determined such that safety is guaranteed.

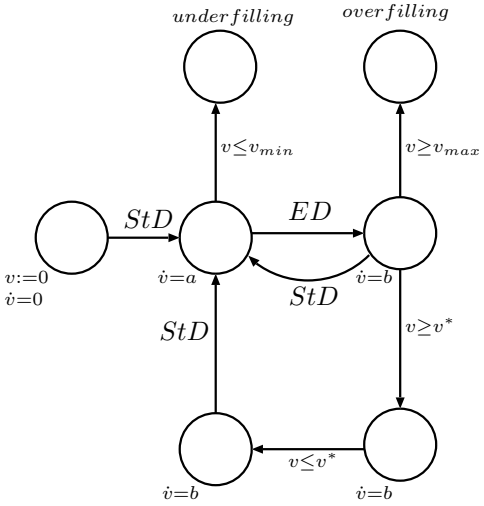

Fig. 6. A modified timed automaton of the output tank

If processing times are known imprecisely, one needs an additional degree of freedom. One possibility to achieve an additional degree of freedom is to introduce additional waiting times for the discharging of the reactors. This is illustrated in Fig.6. The figure shows that the start of a discharging process is only allowed if the liquid volume is below a certain threshold. In this way, overfilling can be avoided. To avoid underfilling, we need the maximal cycle duration $t_{c,max}$, i.e. the maximal time between two discharging operations in the same reactor provided the processing times of all operations take maximal values. $t_{c,max}$ can be obtained by applying verification procedures to the product of reactor models and resource availability models for uncertain processing times. Then, the outflow of the output tank can be set to

$$F_{out,t} = \frac{nF_{out,r}d_d}{t_{c,max}},$$

where d_d is the duration of the discharging operation.

An alternative is to switch the output rate online between several values $F_{out,ti}, i = 1, q$. In this case, the tank model has to be modified according to Fig.7.

To implement the necessary verification procedures we can use slightly modified versions of standard algorithms (see, e.g. (Pettersson, 1999; Bengtsson and Yi, 2004; Bozga *et al.*, 1998) and references therein). This procedure can be considered as the computation of the set of all reachable states of the timed automaton under consideration. Obviously, this set will consist of a set of locations and sets of clock valuations associated with these locations. Fortunately, these sets of clock valuations can be represented through the union of parallelograms on the space X, which makes the procedure computationally tractable.

6. EXAMPLE

We now consider the specific example described in detail in (Simeonova *et al.*, 2005). The plant consists of two reactors. In each reactor the following sequence of operations is performed: filling ($d^*_1 = 0.17$h), heating ($d^*_2 = 0.45$h), temperature regulation ($d^*_3 = 3.44$h), cooling ($d^*_4 = 0.92$h) and discharging ($d^*_5 = 0.17$h). The operations filling, heating and cooling are resource-sensitive. The set of operations has been partitioned into three tasks: K^1_j={filling}, K^2_j={heating, temperature regulation, cooling} and K^3_j={discharging}, $j = 1, 2$. The time for heating is only known imprecisely: $d_2 \in [d^*_2 - \underline{d}_2, d^*_2 + \overline{d}_2]$ where $\underline{d}_2 = \overline{d}_2 = 0.13$h. The minimal and maximal volume of liquid in the tank is $V_{min} = 0$ and $V_{max} = 50\text{m}^3$.

We applied the method presented in the previous section to obtain a solution which guarantees safety and nonblocking for all possible variations of parameters. For example, the particular schedule for the worst case $d_2 = d^*_2 + \overline{d}_2$ is shown in Fig.8.

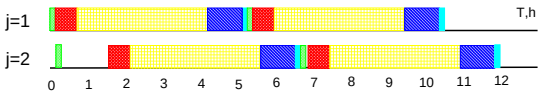

Fig. 8. Resulting schedule for $d_{2j} = d^*_2 + \overline{d}_2$, $j = 1, 2$

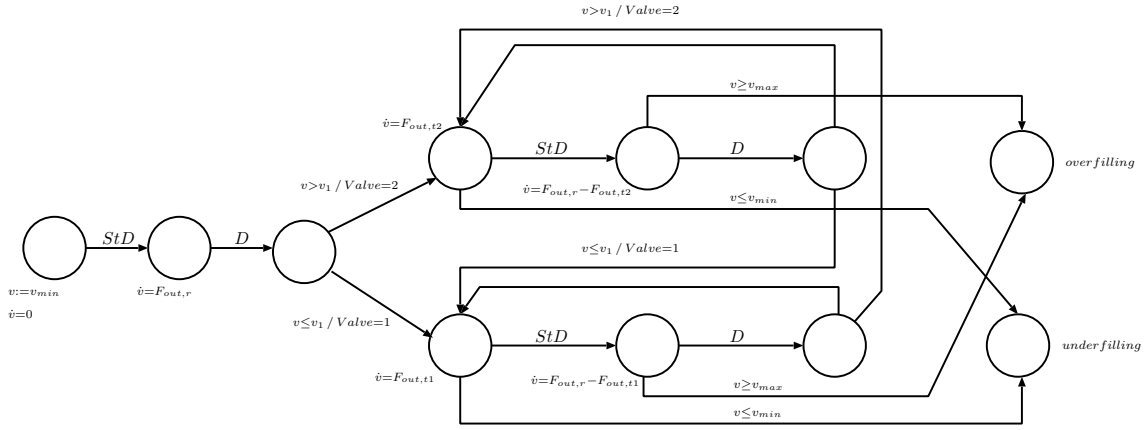

Fig. 7. A timed automaton modelling the output tank

The maximal admissible constant tank outflow is $F_{out,t} = 10.23\text{m}^3/\text{h}$. This results in the change of the liquid volume in the tank as shown in Fig.9.

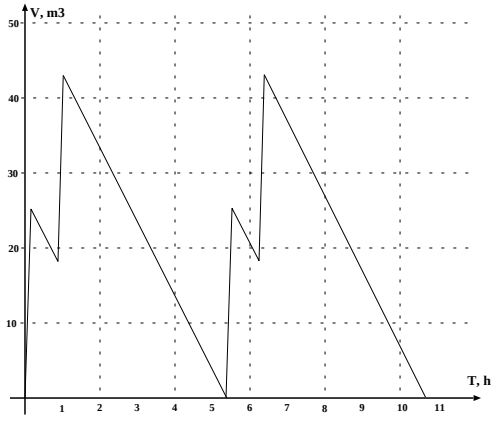

Fig. 9. Liquid volume in the tank

7. CONCLUSIONS AND FUTURE WORK

In this contribution, we investigated the use of timed automata for the scheduling of a class of parallel production lines. We have addressed the case when uncertainties regarding certain operating times are present. Although the prescribed approach contains heuristic elements in the design procedure, we can guarantee that non-blocking and safety are obtained. We have applied this procedure to a specific process which has been suggested as a benchmark problem within the EU Network of excellence HYCON.

REFERENCES

Alur, R. and D.L. Dill (1994). A theory of timed automata. *Theoretical Computer Science* **126**, 183–235.

Alur, R., T.A. Henzinger, G. Lafferriere and G.J. Pappas (2000). Discrete abstractions of hybrid systems. *Proceedings of the IEEE* **88**(7), 971–984.

Asarin, E., O. Maler and A. Pnueli (1995). Symbolic controller synthesis for discrete and timed systems. In: *Hybrid Systems II*. pp. 1–20. LNCS 999. Springer.

Bengtsson, J. and W. Yi (2004). Timed automata: Semantics, algorithms and tools. In: *Lecture Notes on Concurrency and Petri Nets* (W. Reisig and G. Rozenberg, Eds.). LNCS 3098. Springer–Verlag.

Bozga, M., C. Daws, O. Maler, A. Olivero, S. Tripakis and S. Yovine (1998). Kronos: A model-checking tool for real-time systems.. In: *Computer Aided Verification, CAV '98, Vancouver, Canada* (A.J. Hu and M.Y. Vardi, Eds.). LNCS 1427. Springer. pp. 546–550.

Pettersson, P. (1999). Modelling and Verification of Real-Time Systems Using Timed Automata: Theory and Practice. PhD thesis. Uppsala University.

Simeonova, I., F. Warichet, G. Bastin, D. Dochain and Y. Pochet (2005). On line scheduling of chemical plants with parallel production lines and shared resources : a feedback implementation. In: *Proceedings IMACS World Congress, Paris*.