Copyright © 2002 IFAC
15th Triennial World Congress, Barcelona, Spain

SCHEDULING OF HYBRID SYSTEMS: MULTI PRODUCT BATCH PLANT

Boštjan Potočnik [*], **Alberto Bemporad** [***,**], **Fabio Danilo Torrisi** [**], **Gašper Mušič** [*,1], **Borut Zupančič** [*]

[*] *Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, SI-1000 Ljubljana, Slovenia*
[**] *Automatic Control Laboratory, ETH - Swiss Federal Institute of Technology, CH-8092 Zürich, Switzerland*
[***] *Dip. Ingegneria dell'Informazione, Università di Siena, Via Roma 56, I-53100 Siena, Italy*

Abstract: The paper proposes a solution to a class of scheduling problems where the goal is to minimize the schedule (production) time. The algorithm, which takes into account a model of a hybrid system described as MLD (*mixed logical dynamical*) system, is based on performance driven reachability analysis. The algorithm abstracts the behavior of the hybrid system by building a tree of evolution. Nodes of the tree represent reachable states of a process, and the branches connect two nodes if a transition exists between the corresponding states. To each node a cost function value is associated and based on this value, the tree exploration is driven. As soon as the tree is explored, the global solution to the scheduling problem is obtained. *Copyright ©2002 IFAC*

Keywords: Hybrid systems, Scheduling, Optimal control, Reachability analysis, Branch-and-bound methods

1. INTRODUCTION

New methods and advanced technology enable the automation of industrial processes to outgrow basic low-level control functions. Higher levels usually include discrete event dynamics. The traditional control approaches, however, are mostly dealing with continuous dynamics. Hybrid methods deal with interactions between continuous and discrete event dynamics.

In this paper we apply the *mixed logical dynamical* (MLD) modeling framework to a class of scheduling problems. Several control approaches, based on MLD descriptions of a process were proposed in the literature. A model predictive control technique is presented in (Bemporad and Morari, 1999*a*), which is able to stabilize MLD system on desired reference trajectories and where online optimization procedures are solved through *mixed integer quadratic programming* (MIQP). Verification of hybrid systems is presented in (Bemporad and Morari, 1999*b*) and (Bemporad *et al.*, 2000*a*). Optimal control (scheduling) based on reachability analysis is addressed in (Bemporad *et al.*, 2000*b*).

In this paper we address a class of scheduling problems for plants modeled as MLD systems, where the goal is to minimize the schedule time. A similar problem is addressed in (Blömer and

[1] Partially supported by the Ministry of Education, Science and Sport of the Republic of Slovenia, grant J2-2417-1538-01

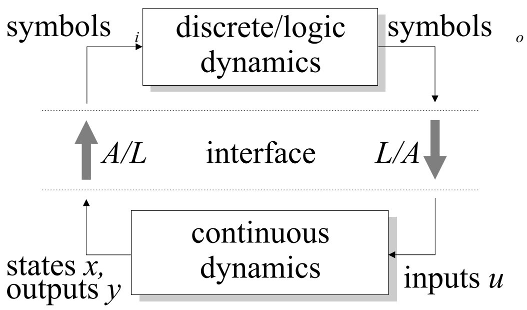

Fig. 1. Hybrid control system. Discrete and continuous dynamic interact through interfaces.

Günther, 1998), where the mixed integer linear programming (MILP) is used, while scheduling approaches for the batch process are addressed in (Reklaitis, 1995).

2. MLD SYSTEMS

Hybrid systems are the combination of logic, finite state machines, linear discrete-time dynamic systems and constraints. The interaction between continuous and discrete/logic dynamic is shown in Fig. 1, where both parts are connected through interfaces. The MLD modeling framework is based on an idea of translating logic relations, discrete/logic dynamics, A/L (analog to logic), L/A conversion and logic constraints into mixed integer linear inequalities. The inequalities are combined with the continuous dynamical part, described by linear difference equations in the following relations (Bemporad and Morari, 1999*a*)

$$\mathbf{x}(k+1)=\mathbf{A}\mathbf{x}(k)+\mathbf{B}_1\mathbf{u}(k)+\mathbf{B}_2\boldsymbol{\delta}(k)+\mathbf{B}_3\mathbf{z}(k) \quad (1a)$$
$$\mathbf{y}(k)=\mathbf{C}\mathbf{x}(k)+\mathbf{D}_1\mathbf{u}(k)+\mathbf{D}_2\boldsymbol{\delta}(k)+\mathbf{D}_3\mathbf{z}(k) \quad (1b)$$
$$\mathbf{E}_2\boldsymbol{\delta}(k)+\mathbf{E}_3\mathbf{z}(k)\leq \mathbf{E}_1\mathbf{u}(k)+\mathbf{E}_4\mathbf{x}(k)+\mathbf{E}_5 \quad (1c)$$

where $\mathbf{x} = [\mathbf{x}_c, \mathbf{x}_l]' \in \mathbb{R}^{n_c} \times \{0,1\}^{n_l}$ is a vector of continuous and logic states, $\mathbf{u} = [\mathbf{u}_c, \mathbf{u}_l]' \in \mathbb{R}^{m_c} \times \{0,1\}^{m_l}$ are the inputs, $\mathbf{y} = [\mathbf{y}_c, \mathbf{y}_l]' \in \mathbb{R}^{p_c} \times \{0,1\}^{p_l}$ are the outputs and $\boldsymbol{\delta} \in \{0,1\}^{r_l}$, $\mathbf{z} \in \mathbb{R}^{r_c}$ are logic and continuous auxiliary variables respectively. Inequalities (Eq. 1c) can include also physical constraints over continuous variables (states and inputs). Given the current state $\mathbf{x}(k)$ and input $\mathbf{u}(k)$, the time evolution of (1) is determined by solving $\boldsymbol{\delta}(k)$ and $\mathbf{z}(k)$ from (1c), and then updating $\mathbf{x}(k+1)$ and $\mathbf{y}(k)$ from (1a, 1b). For a more detailed description, the reader is referred to (Bemporad and Morari, 1999*a*) and (Torrisi *et al.*, 2000).

3. SCHEDULING OF HYBRID SYSTEMS

In (Bemporad *et al.*, 2000*b*) and (Bemporad and Morari, 1999*a*) the authors present procedures for optimal control of hybrid processes described in the MLD form. Optimal control amounts to finding the control sequence $\mathbf{u}_0^{k_{fin}-1} = \{\mathbf{u}(0), \ldots, \mathbf{u}(k_{fin}-1)\}$, which transfers the initial state x_0 to the final state x_f in a finite time $T = k_{fin} \cdot T_s$ (T_s is the sampling time) minimizing a performance index. We will refer to "scheduling" rather than "optimal control" to emphasize the fact that we have to determine the timing of proper production tasks, where the switching between different tasks is modeled by discrete decision variables. The stress will be given on the scheduling of hybrid systems described in MLD form, where the goal is to minimize the production time. The problem will be tackled by extending tools from reachability analysis. The whole idea is based on the selection of a proper cost function as will be explained in Section 3.5.

3.1 Complexity of the scheduling problem

The solution to a scheduling problem is the final time $T = k_{fin} \cdot T_s$ and the optimal control sequence $\mathbf{u}_0^{k_{fin}-1} = \{\mathbf{u}(0), \ldots, \mathbf{u}(k), \ldots, \mathbf{u}(k_{fin}-1)\}$ where $\mathbf{u}(k)$ represent the input to the system at step k. If the system has m_l discrete inputs and no continuous inputs, that means $\mathbf{u}(k) \in \{0,1\}^{m_l}$ and $\mathbf{u}_0^{k_{fin}-1} \in \{0,1\}^{m_l \cdot k_{fin}}$. Because all the inputs are discrete, there are $2^{m_l \cdot k_{fin}}$ possible combinations for $\mathbf{u}_0^{k_{fin}-1}$. Hence the scheduling problem is NP-hard and the computational time required to solve the problem grows exponentially with the problem size.

3.2 Scheduling based on reachability analysis

In general, all the combinations of inputs are not feasible, because of the constraints (1c). One way to rule out infeasible inputs is to use reachability analysis. The idea for hybrid systems with continuous inputs presented in (Bemporad *et al.*, 2000*b*) and (Bemporad *et al.*, 2000*a*) is here extended to hybrid systems with discrete inputs.

Through reachability analysis it is possible to extract the reachable states of the system, although enumerating all for them would not be effective, as many of them will be far away from the optimal trajectory. Therefore it is reasonable to combine reachability analysis with procedures which can detect reachable states not leading to the optimal solution and remove them from the exploration procedure. The whole procedure is a kind of *branch and bound* strategy. The procedure involves the generation of a tree of evolution, as described below.

3.3 Reachability analysis

Let $x_i(k)$ be the state at step k. Reachability analysis computes all the possible states $x_{r_j}(k+1)$ which are reachable at the next time step. If the system has m_l discrete inputs, then 2^{m_l} possible next states may exist. However, because of the constraints (1c), only a smaller number of states can actually be reached. The reachable states can be computed using the algorithm for reach-set computation described in (Bemporad *et al.*, 2000*a*).

3.4 Tree of evolution

A "tree of evolution" (see Fig. 2) abstracts the possible evolution of the system over a horizon of k_{fin} steps. The nodes of the tree represent reachable states and branches connect two nodes if a transitions exists between corresponding states. For a given root node $\mathcal{V}_0$, representing the initial state x_0, the reachable states are computed and inserted into the tree as nodes $\mathcal{V}_i$. A cost value J_i is associated to each new node. A new node is selected based on the associated cost value J_i and new reachable states are computed (Bemporad *et al.*, 2000*b*). More about the cost function and node selection criteria will be presented in the following section. The construction of the tree of evolution continues in depth first until one of the following conditions occurs:

- The step horizon limit k_{max} has been reached.
- The value of the cost function at the current node is greater than the current optimal one ($J_i \geq J_{opt}$, where initially $J_{opt} = \infty$).
- A feasible solution has been found ($x_k = x_f$).

A node which satisfies one of the above conditions is labelled as explored. If a node satisfies the first or the third condition, the associated value of the cost function J_i becomes the current optimal one ($J_{opt} = J_i$), the step instance k becomes the current optimal one ($k_{opt} = k$) and the control sequence $\mathbf{u}_0^{k_{fin}-1}$ which leads from the initial node $\mathcal{V}_0$ to the current node $\mathcal{V}_i$ becomes the current optimizer. The exploration continues until there are no more unexplored nodes in the tree and the temporary control sequence $\mathbf{u}_0^{k_{fin}-1}$ becomes the optimal one.

3.5 Cost function and node selection criterion

The selection of the cost function and the node selection criterion have a great influence on the size of the tree of evolution and, indirectly, on the time efficiency of the scheduling algorithm. The best node selection criterion is to propagate the tree of evolution in a direction that minimizes

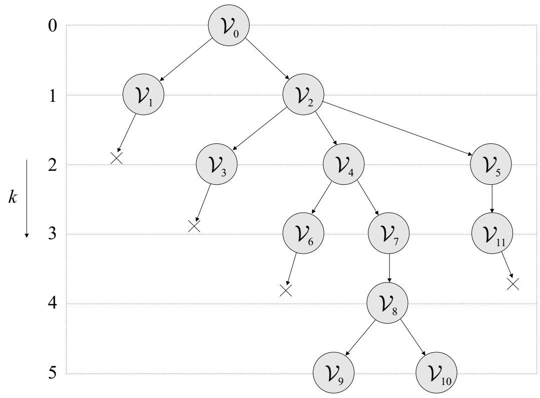

Fig. 2. Tree of evolution

the value of the cost function. At the same time the cost value J_i associated with a node is used to detect nodes which are not going to lead to the optimal solution. To achieve that, the cost function must have certain properties that we describe below.

As the goal is to minimize the total production time we choose the following cost function:

$$J_i = h(\mathbf{x}) + g(k) \tag{2}$$

where $h(\mathbf{x})$ is a rewarding function (the "quality" of the product) and $g(k)$ is a penalty function (the influence of the production time). Detecting nodes not going to lead to the optimal solution before the step instance reaches the current optimal one ($k < k_{opt}$) can be achieved by increasing the cost value J_i with function $g(k)$ (penalizing) and decreasing with function $h(\mathbf{x})$ (rewarding). When the cost value becomes greater then the current optimal one ($J_i \geq J_{opt}$), we want to ensure that by continuing the exploration no better solution than the current one can be found. To achieve that, the cost function (2) has to be monotonically increasing, i.e. in the next steps the cost value can only increase. To this end, we impose

$$\Delta J_i = \frac{\partial J_i}{\partial \mathbf{x}} \Delta \mathbf{x} + \frac{\partial J_i}{\partial k} \Delta k \geq 0 \tag{3}$$

4. A CASE STUDY: MULTI PRODUCT BATCH PLANT

We applied the scheduling algorithm to a multi product batch plant, designed and built at the Process Control Laboratory of the University of Dortmund (Bauer, 2000; Bauer *et al.*, 2000).

4.1 Description of the plant

In the batch process two liquid products, one of blue colour, one green, are produced from three liquid substances (yellow, red, white). The chemical reaction behind the change of colours is

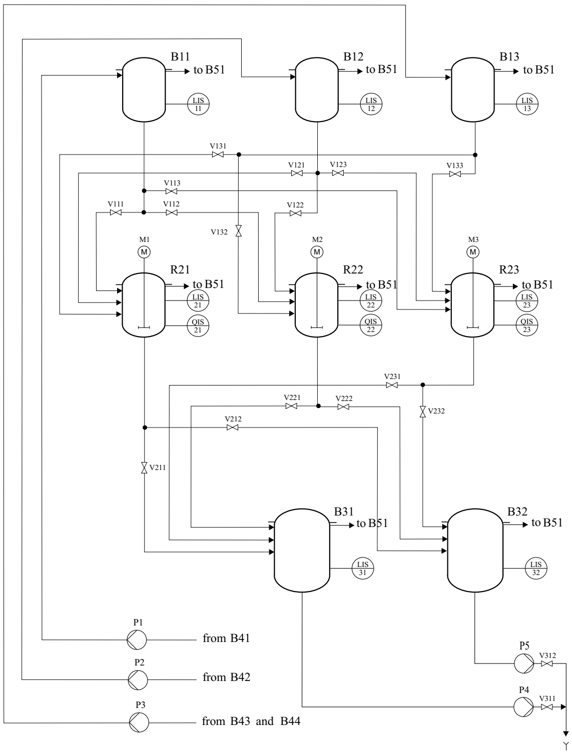

Fig. 3. Multibatch process

the neutralization of the diluted hydrochloric acid (HCl) with the diluted sodium hydroxide (NaOH). The diluted hydrochloric acid is mixed with two different pH indicators to make the acid look yellow or red. During the neutralization reaction the pH indicators change colour when the pH reaches approximately pH 7. The first indicator changes from yellow to blue, while the second from red to green.

The plant consists of three different layers, as shown in Fig. 3. The upper layer consists of the buffering tanks B11, B12 and B13 which are used for holding the raw materials, “Yellow”, “Red” and “White” respectively. Each of the buffer tanks is used exclusively for one raw material and can hold two batches of liquid. The middle layer consists of three reactors R21, R22 and R23. Each reactor can be filled from any raw material buffer tank. This means that each reactor can produce either ”Blue” or “Green”. The production is done by first filling the reactor with one batch of “Yellow” or “Red” and then neutralizing it with one batch of “White”. The lower layer consists of two buffer tanks B31 and B32 in which the products are collected from the middle layer. Each of them is used exclusively for “Blue” or “Green” and can contain three batches of product.

The processing times of the plant (data of the scheduling problem), presented in Table 1, are based on following assumptions: the sampling time is $T_s = 1$ s and one batch of raw material is 0.85 l (one batch of product is then 1.7 l). For more detailed description the reader is referred to (Bauer, 2000) and (Bauer *et al.*, 2000).

Table 1. Processing times

Processes	**Times**
Pumping 1 batch “Yellow” into B11	12 sec.
Pumping 1 batch “Red” into B12	12 sec.
Pumping 1 batch “White” into B13	12 sec.
Draining 1 batch “Yellow” into R21	15 sec.
Draining 1 batch “Red” into R21	11 sec.
Draining 1 batch “White” into R21	10 sec.
Draining 1 batch “Yellow” into R22	12 sec.
Draining 1 batch “Red” into R22	13 sec.
Draining 1 batch “White” into R22	9 sec.
Draining 1 batch “Yellow” into R23	12 sec.
Draining 1 batch “Red” into R23	14 sec.
Draining 1 batch “White” into R23	13 sec.
Draining 1 batch “Blue” from R21 into B31	12 sec.
Draining 1 batch “Green” from R21 into B32	13 sec.
Draining 1 batch “Blue” from R22 into B31	12 sec.
Draining 1 batch “Green” from R22 into B32	12 sec.
Draining 1 batch “Blue” from R23 into B31	12 sec.
Draining 1 batch “Green” from R23 into B32	12 sec.
Pumping 3 batches “Red” out of B31	30 sec.
Pumping 3 batches “Green” out of B32	30 sec.

4.2 MLD Model

The system was described in HYSDEL and transformed into MLD form. HYSDEL is a language and a tool (Torrisi *et al.*, 2000) for generating hybrid models in MLD form. The tool returns a MATLAB script defining the matrices of the corresponding MLD form. The code for the multi product batch plant can be found in (Potočnik, 2001).

The HYSDEL tool generates a MLD model of the form (1), where $\mathbf{x}(k) \in \mathbb{R}^{28} \times \{0,1\}^6$, $\mathbf{u}(k) \in \{0,1\}^{12}$, $\boldsymbol{\delta}(k) \in \{0,1\}^{85}$, and $\mathbf{z}(k) \in \mathbb{R}^{40}$. Matrices $\mathbf{A}$, $\mathbf{B_1}$, ..., $\mathbf{C}$, ..., $\mathbf{D_3}$ have suitable dimensions. Matrices $\mathbf{E_1}$ to $\mathbf{E_5}$ define 511 inequalities. As can be seen the matrices are quite large and building them without an automated tool like HYSDEL would be extremely difficult and tedious.

4.3 Scheduling of a multi product batch plant

Problem formulation:

For a given initial condition schedule the production of “Blue” and “Green” to minimize the production time.

We assume that the raw material batches are delivered at fixed, given times, which are known in advance (see Table 2). The degrees of freedom for the scheduling are the times at which one batch of “Yellow”, “Red” or “White” is emptied into a reactor and the selection of a reactor in which the raw material will be emptied.

Table 2. Delivery times in [min:sec] for raw material

Yellow	**Red**	**White**	
0:00	0:10	0:00	4:20
0:30	0:40	0:30	5:30
1:10	1:50	2:00	6:20
4:20	5:30	2:20	6:55
5:40	6:50	3:00	7:10
6:45	7:50	3:35	8:00

The goal is to produce 6 batches of "Blue" and "Green" constrained by the deliver times of batches "Yellow", "Red" and "White" given in Table 2.

4.4 Complexity of the problem.

The solution of a scheduling problem is a control sequence $\mathbf{u}_0^{T-1}$. At each time 9 inputs can influence the system $\mathbf{u}(k) \in \{0,1\}^9$(valves V111-V133, see Fig. 3). The first three inputs are predefined by the deliver times. The minimal scheduling time can be estimated from Table 1 and 2, although such a time may not be feasible. The last batch of white is delivered after 480 seconds. Additionally $12 + 9 + 12 = 33$ seconds must be added to finish the production, so the minimal time cannot be smaller than $T_{min} = 513s$ and $\mathbf{u}_0^{k_{fin}-1} \in \{0,1\}^{9\cdot 513=4617}$. Because all the inputs are logical, 2^{4617} possible combinations of the solution vector $\mathbf{u}_0^{k_{fin}-1}$ exist and searching the solution through all the combinations is practically impossible.

4.5 Scheduling of the multi product batch plant.

The goal is to minimize the production time of "Blue" and "Green" for given initial conditions. According to the cost function and node selection criterion introduced earlier, we use the following cost function

$$J_i = (V_{max} - v)F + k, \tag{4}$$

where V_{max} is constant defined as $V_{max} = 3 \times$ *volume of all products*, v represents the current value of material pumped/emptied into the first, second and third layer (see Fig. 3), k is the current step and F is a factor, whose properties will be explained later. The goal (final state $\mathbf{x}_f$) is reached when $v = V_{max}$ i.e. the predefined number (volume) of batches is produced. The cost function value at the feasible solution is

$$J_i = k. \tag{5}$$

According to (3), the cost function (4) has to be monotonically increasing

Table 3. Tree of the evolution

Interval	Param. F	Tree size [nodes]	Time effi. [h:min:sec]
$k \in [0, 40]$	1.90	32205	0:22:26
$k \in [0, 40]$	1.80	36852	0:25:48
$k \in [0, 50]$	1.90	497799	5:48:10
$k \in [0, 50]$	1.80	608826	7:11:23

$$\Delta J_i = \frac{\partial J_i}{\partial v}\Delta v + \frac{\partial J_i}{\partial k}\Delta k = -F\Delta v + \Delta k \geq 0 \tag{6}$$

and hence the parameter F must satisfy the condition

$$F \leq \frac{\Delta k}{\Delta v}, \tag{7}$$

where $\Delta k = T_s = 1$. The value for Δv is defined through the estimation of maximum change of volume in the system not taking into account delivery times. If all three raw materials are delivered, two reactors are filled and the third is emptied into a buffer tank at the same time, i.e. $3 \cdot 0.85/12$ (see Table 1) for filling all the three tanks in upper layer, $0.85/9$ and $0.85/11$ represent filling two reactors using maximum flow (shortest deliver time) and $1.7/12$ emptying reactor into buffer tank, then the following estimation for Δv is obtained

$$\Delta v = 3\frac{0.85}{12} + \frac{0.85}{9} + \frac{0.85}{11} + \frac{1.7}{12} = 0.526. \tag{8}$$

Hence it follows that

$$F \leq \frac{\Delta k}{\Delta v} = \frac{1}{0.526} = 1.90. \tag{9}$$

Regarding the node selection criterion, it is reasonable to choose the node which leads to the best (current) optimal solution, i.e., the node with the smallest associated cost function value J_i at step k (the influence of step instance k is the same, but the influence of v is greater).

5. RESULTS

5.1 Scheduling on a bounded interval.

The scheduling algorithm is run for the time interval $k \in [0, 50]$ and $k \in [0, 40]$ and for two values of the parameter F ($F = 1.80$ and $F = 1.90$). The corresponding trees of evolution are presented in Table 3.

The obtained results illustrate that parameter F has a great influence on the tree size and indirectly on the time needed to solve the scheduling problem.

5.2 Complete production scheduling.

The complexity of the scheduling problem grows exponentially with the scheduling horizon. Due to

Table 4. Production time [min:sec] and corresponding reactor)

Blue	Reactor	Green	Reactor
0:24	R22	0:42	R21
2:29	R23	2:12	R22
3:50	R21	3:12	R22
4:44	R22	5:42	R21
6:32	R22	7:22	R21
7:09	R22	8:13	R21

these limitations suboptimal algorithms, based on additional knowledge on the process, are preferred and in most cases give satisfactory results in acceptable time.

Here the same algorithm will be applied with a constraint over the tree size of 100000 nodes (≈ 1 hour) and parameter F set to 1.90. The solution, production starting times and corresponding reactors, is presented in Table 4 and in Fig. 4. The first and in this case the final suboptimal solution was obtained in just 27 seconds. The production of 6 batches of "Blue" and "Green", can be achieved in 515 seconds. The lower bound to the global optimal solution (513 seconds) is missed only by 2 seconds.

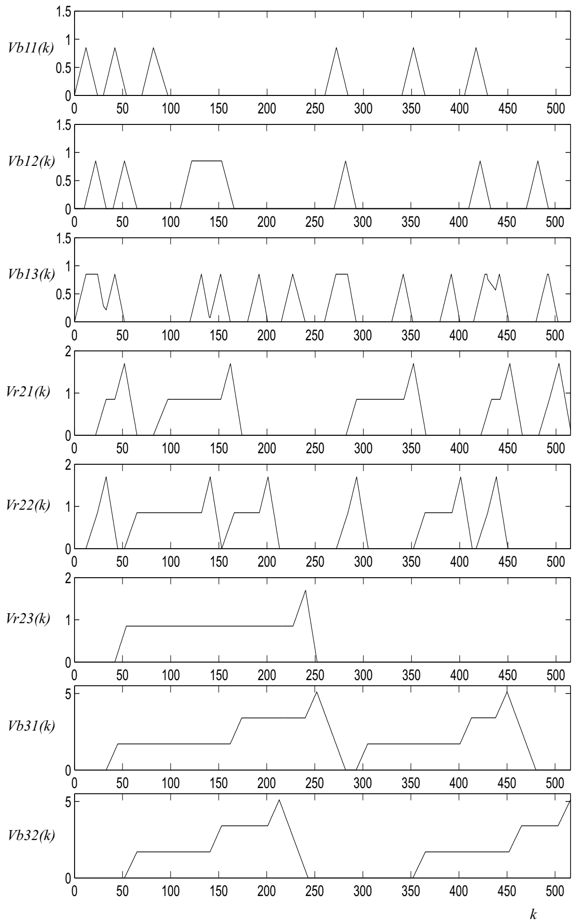

Fig. 4. Volumes in the tanks

6. CONCLUSIONS

In this paper we addressed a class of scheduling problems where the goal is to minimize the total production time. The problem was solved by combining reachability analysis and a branch and bound technique.

REFERENCES

Bauer, N. (2000). A demonstration plant for the control and scheduling of multi-product batch operations. Tech. report - VHS case study 7. University of Dortmund. Dortmund.

Bauer, N., S. Kowalewski, G. Sand and T. Löhl (2000). A case study: Multi product batch plant for the demonstration of control and scheduling problems. In: *ADPM2000 Conference Proceedings*. pp. 383–388.

Bemporad, A. and M. Morari (1999*a*). Control of systems integrating logic, dynamic, and constraints. *Automatica* **35**(3), 407–427.

Bemporad, A. and M. Morari (1999*b*). *Hybrid Systems: Computation and Control. F.W. Vaandrager and J.H. van Schuppen Eds.*. Chap. Verification of Hybrid Systems via Mathematical Programming, pp. 31–45. Vol. Lecture Notes in Computer Science Vol. 1569. Springer Verlag.

Bemporad, A., F.D. Torrisi and M. Morari (2000*a*). *Hybrid Systems: Computation and Control, B.H. Krogh, N. Lynch Eds.*. Chap. Optimization Based Verification and Stability Characterization of Piecewise Affine and Hybrid Systems, pp. 45–58. Vol. Lecture Notes in Computer Science Vol. 1790. Springer Verlag.

Bemporad, A., L. Giovanardi and F.D. Torrisi (2000*b*). Performance driven reachability analysis for optimal scheduling and control of hybrid systems. In: *Proc. 39th IEEE Conference on Decision and Control*. Sydney, Australia. pp. 969–974.

Blömer, F. and H.O. Günther (1998). Scheduling of a multi-product batch process in the chemical industry. *Computers in industry* **36**(3), 245–259.

Potočnik, B. (2001). Hysdel code of a multi product batch plant. `http://msc.fe.uni-lj.si/potocnik`.

Reklaitis, G.V. (1995). Scheduling approaches for the batch plant industries. *ISA Transactions* **34**(4), 349–358.

Torrisi, F.D., A. Bemporad and D. Mignone (2000). Hysdel - a tool for generating hybrid models. Tech. Report AUT00-03. Automatic Control Lab. ETH Zürich. Switzerland. `http://control.ethz.ch/~hybrid/hysdel`.