

## HIERARCHICAL HYBRID CONTROL OF A MULTIPRODUCT BATCH PLANT

Thomas Moor\* Jörg Raisch†

\* *Lehrstuhl für Regelungstechnik*  
*Universität Erlangen-Nürnberg, Germany*  
*thomas.moor@rt.eei.uni-erlangen.de*

† *Lehrstuhl für Systemtheorie technischer Prozesse, Otto-von-Guericke Universität,*  
*and Max-Planck-Institut für Dynamik komplexer technischer Systeme,*  
*Magdeburg, Germany, raisch@mpi-magdeburg.mpg.de*

**Abstract:** This contribution demonstrates the potential of a hierarchical hybrid control framework developed in a previous paper (Moor *et al.*, 2003) by applying it to a specific process control example. The example is the control of a multiproduct batch plant with both continuous and discrete inputs. The control problem is to produce a desired amount of each product with minimal operating costs while observing safety and quality requirements.  
*Copyright © 2005 IFAC*

**Keywords:** hierarchical control, hybrid systems, supervisory control synthesis, multiproduct batch process.

### 1. INTRODUCTION

It is common engineering knowledge that suitable decomposition techniques form a necessary ingredient for any systematic treatment of complex, or large scale, control problems. Hierarchical approaches, where several control layers interact, are an attractive way of problem decomposition as they provide an extremely intuitive control architecture. Of course, one needs to guarantee that the different control levels interact “properly” and do indeed enforce the overall specifications for the considered plant model.

In (Moor *et al.*, 2003), the authors have extended their abstraction-based approach to hybrid control systems design (Moor and Raisch, 1999; Moor *et al.*, 2002) to include multilevel control structures. For the simplest case, a two-level controller, they have described a bottom-up synthesis procedure which meets the above requirement. Its elegance stems from the fact that the specifications for the lower control level can be considered a suitable abstraction, which may be used as a basis for the synthesis of the high-level controller. Formulating specifications for the lower control level may rely on engineering intuition. In fact, the approach

allows one to encapsulate engineering intuition within a formal framework hence exploiting positive aspects of intuition while preventing misguided aspects from causing havoc within the synthesis step.

In the context of discrete event systems (DES) and hybrid systems, where the “curse of dimensionality” seems to be particularly prohibitive, a number of hierarchical concepts have been discussed in the literature. Our approach has been inspired by the hierarchical DES theory developed in (Wong and Wonham, 1996), but is technically quite different as it is based on an input/output structure to adequately represent both time and event-driven dynamics for hybrid systems. There is also a strong conceptual link to (Leduc *et al.*, 2001), where, as in (Pappas *et al.*, 2000; Caines and Wei, 1998), the preservation of fundamental properties across levels of abstraction is a prime concern.

It is the purpose of this paper to demonstrate the potential of the hierarchical synthesis framework (Moor *et al.*, 2003) by applying it to a specific hybrid control problem, which is simple enough to serve as an illustration for the main ideas, but of enough complexity to make it hard to handle for unstructured synthesis methods.

This contribution is organised as follows: in Section 2, a discontinuously operated multi-product batch plant is introduced. Section 3 provides earlier results on hierarchical control synthesis for hybrid systems and shows how an additional optimal performance objective can be addressed. Finally, in Section 4, the resulting procedure is applied to the multiproduct batch control problem, consisting of “hard” quality and safety constraints and a low operating cost requirement.

## 2. A DISCONTINUOUSLY OPERATED MULTIPRODUCT PLANT

In the chemical industries, discontinuously operated multiproduct plants are widely used for the production of fine, or speciality, chemicals. A typical task is to maximise profit while guaranteeing safety and certain lower bounds for quality. In the sequel, a specific example for a multiproduct batch control problem is described.

The plant is used to produce three kinds of colour pigments, using similar production methods (Fig. 1): from one of the storage tanks B1, B2, or B3, solvent is pumped into either a large reactor R1 or a small reactor R2. Reactant  $A_i$ ,  $i = 1, 2, 3$ , is added to start reaction  $i$  delivering the desired product:  $A_i \xrightarrow{k_{P_i}} P_i$ . It is accompanied by a parallel reaction  $A_i \xrightarrow{k_{W_i}} W_i$  resulting in the waste product  $W_i$ . If, at the end of the reaction step, the concentration of  $W_i$  is above a given threshold  $W_{i,max}$ , product quality is unacceptable and the batch is spoilt. For the duration of the reaction, there are two control inputs: the feed rate of the reactant and the heating/cooling rate for the reactor.

After the reaction is finished, the contents of the reactors is filtered through either F1, F2, or F3, and the solvent is collected in the corresponding tank B1, B2, or B3. The solvent can subsequently be fed back into either of the two reactors. If, in any of the filters, darker colours are filtered before lighter ones (say P3 before P1 or P2 and P2 before P1), an additional clean-

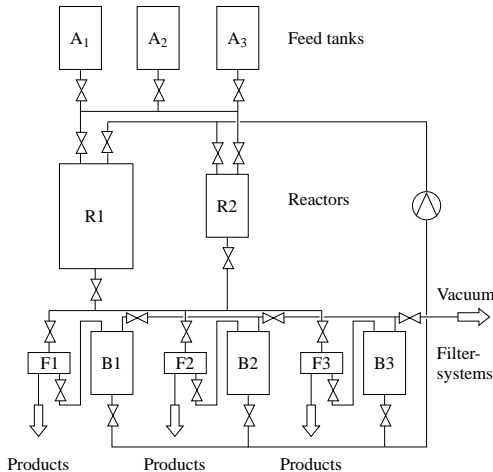


Fig. 1. Example plant.

ing process between the two filtration tasks is needed, taking time  $t_c$ . The feed rates into the reactors are discrete-valued control inputs as are the decision variables (realized by discrete valve positions) that determine whether a particular reactor is emptied through a particular filter system. Heating/cooling rates for R1 and R2 are continuous-valued control inputs. The overall aim is to produce the demanded product volumes with minimal operating costs, while satisfying quality constraints (upper bounds for the concentration of waste products) and safety constraints (upper bounds for reactor temperatures).

### Model of chemical reaction

For simplicity, the following assumptions are made.

- All reactions are first order.
- The volume of reactant  $A_i$ , product  $P_i$  and waste product  $W_i$ ,  $i = 1, 2, 3$ , is negligible compared to overall reactor volume. The latter can therefore be considered constant during dosing and reaction.
- The time constants for heating/cooling of the reactors are small compared to the reaction time constants. The reactor temperatures can therefore be considered to be the manipulated variables.

The model equations can be then be derived from component balances (Bequette, 1998):

$$\frac{d}{dt}c_{A_i}(t) = \frac{q(t)}{V} - (k_{P_i}(t) + k_{W_i}(t))c_{A_i}(t), \quad (1)$$

$$\frac{d}{dt}c_{P_i}(t) = k_{P_i}(t)c_{A_i}(t), \quad (2)$$

$$\frac{d}{dt}c_{W_i}(t) = k_{W_i}(t)c_{A_i}(t), \quad (3)$$

where  $V$  is the volume of the considered reactor,  $q$  is the dosing rate (in kmol/h), and  $c_{A_i}$ ,  $c_{P_i}$ ,  $c_{W_i}$  are reactant, product and waste concentration in the  $i$ th production process (in kmol/m<sup>3</sup>),  $i = 1, 2, 3$ . The reaction rates  $k_{P_i}(t)$  and  $k_{W_i}(t)$  are determined by the temperature  $T$ , where one refers to the Arrhenius law

$$k_{P_i}(t) = k_{P_{i0}}e^{-\frac{E_{P_i}}{RT(t)}}, \quad k_{W_i}(t) = k_{W_{i0}}e^{-\frac{E_{W_i}}{RT(t)}},$$

with the activation energy  $E_-$  and the ideal gas constant  $R$ . Defining  $u(t) := k_{W_1}(t)$ ,  $\beta_i := E_{P_i}/E_{W_1}$ ,  $\delta_i := E_{W_i}/E_{W_1}$ ,  $\alpha_i := k_{P_{i0}}/k_{W_{i0}}^{\beta_i}$ , and  $\gamma_i := k_{W_{i0}}/k_{W_{i0}}^{\delta_i}$ , Eqs. (1) – (3) can be rewritten as

$$\frac{d}{dt}c_{A_i}(t) = \frac{q(t)}{V} - (\alpha_i u(t)^{\beta_i} + \gamma_i u(t)^{\delta_i})c_{A_i}(t), \quad (4)$$

$$\frac{d}{dt}c_{P_i}(t) = \alpha_i u(t)^{\beta_i}c_{A_i}(t), \quad (5)$$

$$\frac{d}{dt}c_{W_i}(t) = \gamma_i u(t)^{\delta_i}c_{A_i}(t). \quad (6)$$

Note that, by definition,  $\delta_1 = \gamma_1 = 1$ , and that  $u$  is a strictly monotonically increasing function in  $T$  and can therefore be considered as scaled temperature with unit [1/h].

### 3. HIERARCHICAL CONTROL IN A BEHAVIOURAL FRAMEWORK

The purpose of this section is to outline earlier work (Moor *et al.*, 2003) on hierarchical supervisory control for hybrid systems. The control architecture consists of a low-level plant, an intermediate layer and a high-level supervisor; see Fig. 2. All major entities are represented by behaviours, i.e., sets of signals on which the respective system can evolve.

*Definition 1.* (Willems, 1991) A behaviour  $\mathfrak{B}$  over a signal space  $W$  is a set of maps  $w: \mathbb{N}_0 \rightarrow W$ ; i.e.,  $\mathfrak{B} \subseteq W^{\mathbb{N}_0}$ .

The low-level plant behaviour  $\mathfrak{B}_p^L$  over  $W_L = U_L \times Y_L$  represents a detailed plant model that may involve both continuous and discrete variables, i.e., real vector valued variables and variables that take their values from a finite set, respectively. The control objective is represented by a specification behaviour  $\mathfrak{B}_{\text{spec}}^L$  over  $W_L$  that consists of all low-level signals that are acceptable for the overall closed-loop configuration.

The intermediate layer,  $\mathfrak{B}_{\text{im}} \subseteq (W_H \times W_L)^{\mathbb{N}_0}$ , represents the relation between low-level signals over  $W_L$  and high-level signals over  $W_H = U_H \times Y_H$ . This behavioural relation allows for a variety of interesting cases:  $\mathfrak{B}_{\text{im}}$  may implement low-level controllers to switch between, it may also implement measurement aggregation and/or a transformation from *clock time* (driven by a fixed sampling period) to *logic time* (driven by events).

The high-level supervisor  $\mathfrak{B}_{\text{sup}}^H \subseteq W_H^{\mathbb{N}_0}$  is meant to enforce the low-level specification via the intermediate layer. Formally, the overall configuration satisfies the specification  $\mathfrak{B}_{\text{spec}}^L$  if

$$w^L \in \mathfrak{B}_p^L, w^H \in \mathfrak{B}_{\text{sup}}^H, (w^H, w^L) \in \mathfrak{B}_{\text{im}} \implies w^L \in \mathfrak{B}_{\text{spec}}^L. \quad (7)$$

There are two general questions regarding the hierarchical control architecture under consideration that have been discussed in (Moor *et al.*, 2003): (3.1) what are appropriate conditions for the proposed system interconnection and how can they be guaranteed to hold? (3.2) given  $\mathfrak{B}_p^L$  and  $\mathfrak{B}_{\text{spec}}^L$ , how can one actually design  $\mathfrak{B}_{\text{im}}$  and  $\mathfrak{B}_{\text{sup}}^H$ ?

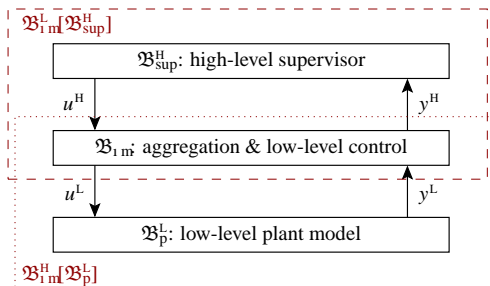


Fig. 2. Plant (supervisor) perspective, dashed (dotted)

#### 3.1 Admissibility criteria

Conditions for hierarchical system interconnection are derived from the simple interconnection of plant and supervisor, where the plant is assumed to conform to a weakened version of Willems' *I/O behaviours*.

A supervisor  $\mathfrak{B}_{\text{sup}}$  is connected to a plant  $\mathfrak{B}_p$  in order to *enforce* a specification  $\mathfrak{B}_{\text{spec}}$ , i.e., one aims for  $\mathfrak{B}_p \cap \mathfrak{B}_{\text{sup}} \subseteq \mathfrak{B}_{\text{spec}}$ . However, it needs to be assured that: (a) any restrictions on the plant output shall only be imposed indirectly by restricting the plant input; and (b) at any time there must be possible future evolution, so the closed-loop must not “get stuck”. If both conditions are fulfilled, one says that  $\mathfrak{B}_{\text{sup}}$  is *admissible to*  $\mathfrak{B}_p$ ; see (Moor *et al.*, 2003) for a technical definition. Furthermore,  $\mathfrak{B}_{\text{sup}}$  *solves* the control problem  $(\mathfrak{B}_p, \mathfrak{B}_{\text{spec}})_{\text{cp}}$  if  $\mathfrak{B}_{\text{sup}}$  is admissible and enforces the specification  $\mathfrak{B}_{\text{spec}}$ .

In the proposed hierarchical architecture, there are two system interconnections. From the perspective of the low-level plant, the intermediate layer together with the high-level supervisor plays the role of a compound supervisor  $\mathfrak{B}_{\text{im}}^L[\mathfrak{B}_{\text{sup}}^H]$  over  $W_L$  (dashed box in Fig. 2):

$$\mathfrak{B}_{\text{im}}^L[\mathfrak{B}_{\text{sup}}^H] := \{w^L \in W_L^{\mathbb{N}_0} \mid (\exists w^H \in W_H^{\mathbb{N}_0}) [(w^H, w^L) \in \mathfrak{B}_{\text{im}} \wedge w^H \in \mathfrak{B}_{\text{sup}}^H]\}. \quad (8)$$

From the perspective of the high-level supervisor, the compound plant  $\mathfrak{B}_{\text{im}}^H[\mathfrak{B}_p^L]$  over  $W_H$  (dotted box in Fig. 2) is given by

$$\mathfrak{B}_{\text{im}}^H[\mathfrak{B}_p^L] := \{w^H \in W_H^{\mathbb{N}_0} \mid (\exists w^L \in W_L^{\mathbb{N}_0}) [(w^H, w^L) \in \mathfrak{B}_{\text{im}} \wedge w^L \in \mathfrak{B}_p^L]\}. \quad (9)$$

For both system interconnections the admissibility conditions need to be satisfied. In (Moor *et al.*, 2003), it is shown under moderate technical assumptions that generic implementability and input/output characteristics propagate through the layers of the hierarchy. Results include that the admissibility conditions are satisfied for intermediate layers that implement low-level control, measurement aggregation and/or causal transformations of the time axis.

#### 3.2 A bottom-up design

A bottom-up-design of  $\mathfrak{B}_{\text{im}}$  and  $\mathfrak{B}_{\text{sup}}^H$  can be done as follows. In the first step, the *intended* relationship between high-level signals and low-level signals is represented by the specification  $\mathfrak{B}_{\text{spec}}^{\text{HL}} \subseteq (W_H \times W_L)^{\mathbb{N}_0}$ ; e.g.,  $\mathfrak{B}_{\text{spec}}^{\text{HL}}$  denotes the set of all signal pairs  $(w^H, w^L)$  that confers the desired effect of high-level control actions on the low-level plant  $\mathfrak{B}_p^L$  and the desired scheme of measurement aggregation.

To ensure that  $w^H$  and  $w^L$  are in fact related in the intended way, one requires the intermediate layer  $\mathfrak{B}_{\text{im}}$  to enforce the specification  $\mathfrak{B}_{\text{spec}}^{\text{HL}}$  when interconnected with the low-level plant  $\mathfrak{B}_p^L$ :

$$\{(w^H, w^L) \in \mathfrak{B}_{\text{im}} \mid w^L \in \mathfrak{B}_p^L\} \subseteq \mathfrak{B}_{\text{spec}}^{\text{HL}}. \quad (10)$$

Suppose  $\mathfrak{B}_{\text{im}}$  has been designed according to Eq. (10). In principle, one could then base the design of  $\mathfrak{B}_{\text{sup}}^{\text{H}}$  on the compound high-level plant  $\mathfrak{B}_{\text{im}}^{\text{H}}[\mathfrak{B}_{\text{p}}^{\text{L}}]$ . However, from a computational point of view — particularly for hybrid systems — it is preferable to use an abstraction  $\tilde{\mathfrak{B}}_{\text{p}}^{\text{H}}$  of  $\mathfrak{B}_{\text{im}}^{\text{H}}[\mathfrak{B}_{\text{p}}^{\text{L}}]$  that does not explicitly depend on the low-level dynamics or the implemented low-level controllers. Such an abstraction together with a suitable high-level specification  $\tilde{\mathfrak{B}}_{\text{spec}}^{\text{H}}$  can be obtained from Eq. (10):

$$\tilde{\mathfrak{B}}_{\text{p}}^{\text{H}} := \{w^{\text{H}} \mid (\exists w^{\text{L}}) [(w^{\text{H}}, w^{\text{L}}) \in \mathfrak{B}_{\text{spec}}^{\text{HL}}]\}; \quad (11)$$

$$\tilde{\mathfrak{B}}_{\text{spec}}^{\text{H}} := \{w^{\text{H}} \mid (\forall w^{\text{L}}) [(w^{\text{H}}, w^{\text{L}}) \in \mathfrak{B}_{\text{spec}}^{\text{HL}} \Rightarrow w^{\text{L}} \in \mathfrak{B}_{\text{spec}}^{\text{L}}]\}. \quad (12)$$

It follows immediately that any high-level supervisor that enforces  $\tilde{\mathfrak{B}}_{\text{spec}}^{\text{H}}$  when connected to  $\tilde{\mathfrak{B}}_{\text{p}}^{\text{H}}$  will also enforce the low-level specification via  $\mathfrak{B}_{\text{im}}$ :

$$\begin{aligned} \tilde{\mathfrak{B}}_{\text{p}}^{\text{H}} \cap \mathfrak{B}_{\text{sup}}^{\text{H}} &\subseteq \tilde{\mathfrak{B}}_{\text{spec}}^{\text{H}} \\ \Rightarrow \mathfrak{B}_{\text{p}}^{\text{L}} \cap \mathfrak{B}_{\text{im}}^{\text{L}}[\mathfrak{B}_{\text{sup}}^{\text{H}}] &\subseteq \mathfrak{B}_{\text{spec}}^{\text{L}}. \end{aligned} \quad (13)$$

### 3.3 Minimising the cost of closed-loop operation

In addition to a specification that *must* hold, many applications come with the control objective of minimising a certain cost function. This issue can be addressed by a straightforward extension to the hierarchical framework.

Consider a cost function  $\gamma^{\text{L}}$  — typically additive over time and positive — to assign the cost  $\gamma^{\text{L}}(w^{\text{L}})$  to the trajectory  $w^{\text{L}}$ :

$$\gamma^{\text{L}}: \mathfrak{B}_{\text{p}}^{\text{L}} \cap \mathfrak{B}_{\text{spec}}^{\text{L}} \rightarrow \mathbb{R} \quad (14)$$

The maximum cost

$$\max_{w^{\text{L}}} \gamma^{\text{L}}(w^{\text{L}}) \quad (15)$$

over all low-level closed-loop trajectories  $w^{\text{L}} \in \mathfrak{B}_{\text{p}}^{\text{L}} \cap \mathfrak{B}_{\text{sup}}^{\text{L}}$  is interpreted as the performance index of a supervisor  $\mathfrak{B}_{\text{sup}}^{\text{L}}$ . Note that when the initial state of the plant is given and the supervisor is sufficiently restrictive, the closed-loop trajectory is unique and the maximum in Eq (15) becomes obsolete. This is the case for the multiproduct batch plant; see also Section 4. In general, an optimal supervisor  $\mathfrak{B}_{\text{sup}}^{\text{L}}$  is characterised by

$$\min_{\mathfrak{B}_{\text{sup}}^{\text{L}}} \max_{w^{\text{L}}} \gamma^{\text{L}}(w^{\text{L}}), \quad (16)$$

$$\text{s.t. } \mathfrak{B}_{\text{sup}}^{\text{L}} \text{ solves } (\mathfrak{B}_{\text{p}}^{\text{L}}, \mathfrak{B}_{\text{spec}}^{\text{L}})_{\text{cp}}, \quad w^{\text{L}} \in \mathfrak{B}_{\text{p}}^{\text{L}} \cap \mathfrak{B}_{\text{sup}}^{\text{L}}.$$

Suppose that the intermediate layer  $\mathfrak{B}_{\text{im}}$  has been designed as outlined in Section 3.2. A pessimistic high-level cost function can then be defined as

$$\begin{aligned} \gamma^{\text{H}}(w^{\text{H}}) &:= \\ \max_{w^{\text{L}}} \{ \gamma^{\text{L}}(w^{\text{L}}) \mid (w^{\text{H}}, w^{\text{L}}) &\in \mathfrak{B}_{\text{im}}, \quad w^{\text{L}} \in \mathfrak{B}_{\text{p}}^{\text{L}} \}, \end{aligned} \quad (17)$$

leading to an optimal high-level controller  $\mathfrak{B}_{\text{sup}}^{\text{H}}$  with

$$\min_{\mathfrak{B}_{\text{sup}}^{\text{H}}} \max_{w^{\text{H}}} \gamma^{\text{H}}(w^{\text{H}}), \quad (18)$$

s.t.  $\mathfrak{B}_{\text{sup}}^{\text{H}}$  solves  $(\tilde{\mathfrak{B}}_{\text{p}}^{\text{H}}, \tilde{\mathfrak{B}}_{\text{spec}}^{\text{H}})_{\text{cp}}$  and  $w^{\text{H}} \in \tilde{\mathfrak{B}}_{\text{p}}^{\text{H}} \cap \mathfrak{B}_{\text{sup}}^{\text{H}}$ .

Note that the overall controller, i.e., the interconnection  $\mathfrak{B}_{\text{im}}^{\text{L}}[\mathfrak{B}_{\text{sup}}^{\text{H}}]$  of  $\mathfrak{B}_{\text{im}}$  and the optimal  $\mathfrak{B}_{\text{sup}}^{\text{H}}$ , does not necessarily form an optimal solution to the original problem (16). This is for two reasons: (i) the introduction of  $\mathfrak{B}_{\text{im}}$  reduces the available degrees of freedom; (ii) in (18), the behaviour  $\mathfrak{B}_{\text{im}}^{\text{L}}[\mathfrak{B}_{\text{p}}^{\text{L}}]$  has been replaced by its abstraction  $\tilde{\mathfrak{B}}_{\text{p}}^{\text{H}}$ , resulting in over-approximation of actual costs. On the positive side, the problem (18) may turn out computationally tractable in situations where (16) is not. Note that, despite the tradeoff between computational effort and closed-loop performance, the bottom-up design method guarantees that the “hard” specification  $\mathfrak{B}_{\text{spec}}^{\text{L}}$  holds.

## 4. APPLICATION TO THE MULTIPRODUCT BATCH PLANT

This section reports results from a case study in which a hierarchical control scheme has been designed for the multiproduct batch plant in Section 2. The overall control architecture is illustrated by Fig. 3, to be discussed in more detail. Some statistics are given to demonstrate the computational efficiency of the proposed approach.

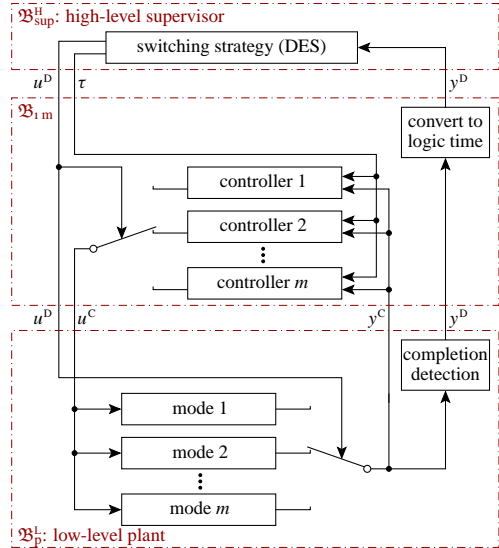


Fig. 3. Control architecture (subsystems merged)

### 4.1 Low-level plant model

The low-level plant model represents the continuous dynamics of filter- and reaction-processes depending on the mode of operation. Low-level signals are considered to evolve w.r.t. clock time, generated from a suitably fast sampling period  $\Delta > 0$  for quasi-continuous operation.

Observe from Fig. 1 that after a reaction is finished, the respective reactor has to be emptied, i.e., its contents has to be filtered before the reactor can be reused in another production step. Neglecting the time required to fill a reactor, there are at most two concurrent operations performed by the example plant. Thus, the low-level plant can be modelled by two subsystems, each of which is being used for one out of three chemical reactions schemes or a subsequent filtering process. As a low-level signal space we choose

$$W_L = W_{L1} \times W_{L2},$$

where each component corresponds to one subsystem.

The possible modes of operation for the subsystem  $j \in \{1, 2\}$  consist of the three chemical reactions and the filtering processes. The latter can use any nontrivial combination of the three filters. Including a filter cleaning mode and an “idle” mode, this gives a total of  $3 + 2 + 7 = 12$  possible modes for each subsystem; they can be conveniently encoded as a discrete event input with range

$$U_{Dj} = \{P1, P2, P3, \text{Clean}, \text{Idle}, \\ F001, F010, F011, \dots, F111\}. \quad (19)$$

While in one of the chemical reaction modes  $P_i$ , low-level dynamics are modelled by a sampled version of the ODEs (4), (5), (6). The parameters are as follows:  $\beta_1 = 0.5$ ,  $\alpha_1 = 2.0\text{h}^{-0.5}$ ,  $\beta_2 = 0.4$ ,  $\alpha_2 = 2.0\text{h}^{-0.6}$ ,  $\beta_3 = 0.5$ ,  $\alpha_3 = 3.0\text{h}^{-0.5}$ ,  $\delta_i = \gamma_i = 1$ ,  $i = 1, 2, 3$ ; the initial concentrations at the beginning of each reaction are all zero:  $c_{A_i0} = c_{P_i0} = c_{W_i0} = 0$ ,  $i = 1, 2, 3$ . The product concentrations required at the end of each reaction are  $c_{P_1e} = 10\text{kmol/m}^3$ ,  $c_{P_2e} = 8\text{kmol/m}^3$ ,  $c_{P_3e} = 12\text{kmol/m}^3$ , and the bounds for the waste concentrations  $c_{W_1}$ ,  $c_{W_2}$ ,  $c_{W_3}$  are  $2\text{kmol/m}^3$ ,  $1.5\text{kmol/m}^3$ , and  $3\text{kmol/m}^3$ , respectively. The volumes of reactor R1 and R2 are  $5\text{m}^3$  and  $2.5\text{m}^3$ , respectively. The (on/off) dosing signal  $q_j$  can take values in the set  $\{0, 12\text{kmol/h}\}$ , and the control signal  $u_j$  is required to “live” within the interval  $[0.01\text{h}^{-1}, 3.0\text{h}^{-1}]$ , where the upper bound results from safety requirements. The signal  $(q_j, u_j)$  is seen as an additional low-level input with range  $U_{Cj} \subseteq \mathbb{R}^2$ . The continuous state is assumed to be measured as a plant output with range  $Y_{Cj} \subseteq \mathbb{R}^3$ .

For filtering, an integrator models the progress of time, where the integration constant depends on the number of filters used and the volume of the respective reactor. The time to empty the smaller of the two reactors through one filter is  $c_{\text{ff}} = 6\text{h}$ . If two or three filters are being used simultaneously, this reduces to 3h and 2h. For the larger reactor, filtering takes twice as long. The continuous input  $U_{Cj}$  is ignored in filtering mode.

The completion of either operation corresponds to reaching a target region within the continuous state space. This is indicated by a discrete low-level output  $y^{Dj}$  which can take values in  $\{\text{Busy}, \text{Done}\}$ .

Finally, the signal space of subsystem  $j \in \{1, 2\}$  is composed as the product

$$W_{Lj} = U_{Lj} \times Y_{Lj}, \quad U_{Lj} = U_{Dj} \times U_{Cj}, \quad Y_{Lj} = Y_{Dj} \times Y_{Cj}.$$

With the above parameters, the typical time to finish a reaction step is between 5h and 10h, with filtering taking at least another two hours.

#### 4.2 Low-level specification and cost function

It is required that the mode of operation changes only immediately after completion of the previous operation; chemical reactions and filtering alternate per subsystem; each filter can only be used by one subsystem at a time; the filters are cleaned appropriately; the demanded products are in fact produced. Note that the low-level specification refers to the discrete event components only and, thus, is a typical discrete event specification. Formally, one has  $\mathfrak{B}_{\text{spec}}^L = \mathfrak{B}_{\text{spec}}^D \times (U_C \times Y_C)^{\mathbb{N}_0}$  for some behaviour  $\mathfrak{B}_{\text{spec}}^D$  over  $U_D \times Y_D$ . A finite automaton realisation of  $\mathfrak{B}_{\text{spec}}^D$  turns out with an overall number of  $1.16 \times 10^6$  states.

The cost function refers to the  $U_{Cj}$  components of the low-level signal only. It is defined as an integral cost on the chemical processes that takes into account the energy for heating, the feed rate and the progress of time. For a low-level trajectory  $w^L$  let

$$\gamma^L(w^L) := \int_0^{T_f} u_1(t) + u_2(t) \\ + 0.05q_1(t) + 0.05q_2(t) + 0.15 dt, \quad (20)$$

where  $T_f$  denotes the time at which all demanded products have been delivered.

Given the low-level plant model, the specification and the cost function, the optimisation problem (16) amounts to a nonlinear mixed discrete-continuous dynamic program with 2 continuous inputs; a lumped discrete input that can take  $(12 \times 2)^2 = 576$  values; 6 continuous state variables; the discrete state set with  $1.16 \times 10^6$  states; a time horizon of about 50h. We did not find this to be computationally tractable for off-the-shelf optimisation software.

#### 4.3 Intermediate layer

The design of  $\mathfrak{B}_{\text{im}}$  is based on a set of low level controllers to run the chemical reactions. Each controller runs a particular process within a prescribed time  $\tau_j \in T$  and does so at minimal cost. Thus, low-level controller design is local in the sense that it only refers to one individual process and one individual subsystem at the time. The corresponding dynamic program has 1 binary input, 1 continuous input and 3 continuous state variables. For the example, the software package GAMS takes about 45 seconds on a desktop computer to perform the optimisation for a fixed  $\tau_j$ .

By discretisation of  $T$  with  $T = \{1\text{h}, 2\text{h}, \dots, 10\text{h}\}$ , one ends up with a finite number  $(2 \times 3 \times 10 = 60)$

of low-level controllers. The optimal cost  $\gamma_{1,i}(\tau_1)$  for reactor R1 to produce one batch of the product  $P_i$  is given for illustration:

$\tau_1$	< 5h	5h	6h	7h	8h	9h	10h
$\gamma_{1,1}$	$\infty$	$\infty$	3.70	3.42	3.28	3.21	3.17
$\gamma_{1,2}$	$\infty$	2.81	2.58	2.46	2.40	2.36	2.32
$\gamma_{1,3}$	$\infty$	$\infty$	4.16	3.71	3.58	3.51	3.49

Obviously, the low-level optimisation does not depend on the demanded overall amount of products. Consequently, this design step only needs to be performed once over the life-cycle of the plant.

High-level control actions consist of modes from  $U_{D_j}$  and timing parameters from  $T$ . Because the timing for the filter process and the idle operation are determined by the mode, there are  $3 \times 10 + 9 = 39$  relevant high-level control actions per subsystem to be encoded in  $U_H$ . The completion component from the low-level subsystems serves as high-level measurement; i.e.,  $Y_H = \{\text{Busy}, \text{Done1}, \text{Done2}\}$ . While the low-level signals refer to clock time, the high-level signal refers to logic time, where events are triggered by changes in the  $Y_{D_j}$ -components.

#### 4.4 High-level supervisor

The interconnection  $\mathfrak{B}_{\text{im}}^H[\mathfrak{B}_p^L]$  of the low-level plant with the intermediate layer can be modelled as a hybrid system. Rather than computing an abstraction, the proposed method from Section 3 is applied to derive an abstraction directly from  $\mathfrak{B}_{\text{spec}}^H$ . The external behaviour w.r.t. the discrete variables  $u^{D_j}$  and  $y^{D_j}$  of each subsystem together with the respective low-level controllers can be modelled as a timed DES (TDES); see, e.g., (Brandin and Wonham, 1994). To obtain a DES realisation of an abstraction  $\mathfrak{B}_p^H$  of  $\mathfrak{B}_{\text{im}}^H[\mathfrak{B}_p^L]$ , the individual TDESs are composed by a synchronous product. Subsequently, tick events can be removed by language projection. Note that the first instance where a composition of subsystems needs to be computed occurs after the individual subsystems have undergone considerable simplification.

The high-level-specification  $\tilde{\mathfrak{B}}_{\text{spec}}^H$  can be directly obtained from  $\mathfrak{B}_{\text{spec}}^D$  by Eq. (12). Together with the high-level abstraction  $\tilde{\mathfrak{B}}_p^H$ , one obtains a transition system with  $17 \times 10^6$  states and an average of 13.1 relevant input events per state. Since every high-level input event corresponds to a low-level mode that will be completed at a known cost, the high-level cost function  $\gamma^H$  is additive over high-level logic time. Thus, the high-level optimisation problem (18) can be solved by standard methods from dynamic programming.

On a desktop computer the synthesis of the high-level supervisor takes 61 minutes, and, hence, can be integrated in an automated production environment. For illustration, Fig. 4 shows the obtained closed-loop

operation to produce  $12.5m^3$ ,  $12.5m^3$  and  $7.5m^3$  of the respective product. The overall cost amounts to 27.5.

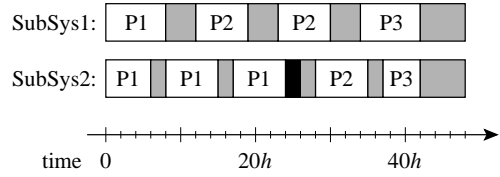


Fig. 4. Optimal schedule (filter processes grey, cleaning black)

## 5. CONCLUSION

This contribution extends the hierarchical design approach described in (Moor *et al.*, 2003) to take into account optimal performance objectives. The potential of the proposed method is demonstrated by successful applying it to a demanding process control problem, which we did not find tractable by using off-the-shelf optimisation software.

**Acknowledgements:** We gratefully acknowledge discussions with D. Gromov, B.V. Mishra and A. Kienle.

## 6. REFERENCES

- Bequette, B.W. (1998). *Process Dynamics: Modeling, Analysis and Simulation*. Prentice Hall.
- Brandin, B.A. and W.M. Wonham (1994). Supervisory control of timed discrete-event systems. *IEEE TAC* **39**, 329–342.
- Caines, P.E. and Y.J. Wei (1998). Hierarchical hybrid control systems: a lattice theoretic formulation. *IEEE TAC* **43:4**, 501–508.
- Leduc, R.J., B.A. Brandin, W.M. Wonham and M. Lawford (2001). Hierarchical interface-based supervisory control: Serial case. In: *Proc. of the 40th IEEE CDC*. pp. 4116–4121.
- Moor, T. and J. Raisch (1999). Supervisory control of hybrid systems within a behavioural framework. *Systems and Control Letters* **38**, 157–166.
- Moor, T., J. Raisch and J.M. Davoren (2003). Admissibility criteria for a hierarchical design of hybrid control systems. In: *Proc. IFAC Analysis and Design of Hybrid Systems (ADHS)*. pp. 389–394.
- Moor, T., J. Raisch and S.D. O’Young (2002). Discrete supervisory control of hybrid systems based on  $l$ -complete approximations. *Journal of Discrete Event Dynamic Systems* **12**, 83–107.
- Pappas, G.J., G. Lafferriere and S. Sastry (2000). Hierarchically consistent control systems. *IEEE TAC* **45:6**, 1144–1160.
- Willems, J.C. (1991). Paradigms and puzzles in the theory of dynamic systems. *IEEE TAC* **36**, 258–294.
- Wong, K.C. and W.M. Wonham (1996). Hierarchical control of discrete-event systems. *Discrete Event Dynamic Systems* **6**, 241–306.