Hierarchical Hybrid Control Synthesis and its Application to a Multiproduct Batch Plant

Jörg Raisch[1][2] and Thomas Moor[3]

[1] Lehrstuhl für Systemtheorie technischer Prozesse, Otto-von-Guericke Universität, Postfach 4120, D-39016 Magdeburg, Germany, raisch@mpi-magdeburg.mpg.de

[2] Systems and Control Theory Group, Max Planck Institute for Dynamics of Complex Technical Systems

[3] Lehrstuhl für Regelungstechnik, Friedrich-Alexander Universität, D-91058 Erlangen, Germany thomas.moor@rt.ee.uni-erlangen.de

Summary. This paper presents a hierarchical approach to hybrid control systems synthesis. It is set in a general behavioural framework and therefore allows a combination of continuous and discrete layers in the resulting overall controller. Compared to unstructured approaches, it drastically reduces computational complexity and hence enlarges the class of continuous-discrete control problems that can be addressed using methods from hybrid systems theory. The potential of our approach is demonstrated by applying it to a multiproduct batch control problem.

1 Introduction

In hybrid dynamical systems, discrete-event components (realised, e.g., by finite automata) and continuous components (realised, e.g., by ordinary differential equations) interact in a nontrivial way. The fundamental problems in analysing and synthesing such systems stem from the nature of their state sets. While the state space of a continuous system usually exhibits vector space structure, and the state set of a discrete event system (DES) is often finite, the state set of the overall system inherits none of theses amenities: as a product of the constituting state sets, it is neither finite nor does it exhibit vector space structure.[4] Hence, neither methods from discrete event systems theory, which rely on exploiting finiteness, nor concepts from continuous control theory carry over readily to hybrid problems. Nevertheless, as inherently hybrid application problems are very common, hybrid control systems have become an increasingly popular subject during the last decade

[4] An obvious consequence from this structural fact is – and this links our contribution to the theme of M. Zeitz's birthday symposium – that hybrid systems are nonlinear by nature.

(e.g. [2, 1, 6]). A considerable part of hybrid systems research has gone into investigating approximation-based approaches (e.g. [7, 5, 16, 9]). There, the core idea is to approximate continuous dynamics by discrete event systems, and hence to transform the hybrid control problem into a purely discrete one. Of course, care has to be taken to guarantee that the resulting (discrete event) control system enforces the specifications not only for the discrete approximation but also for the underlying hybrid system. In [10, 14], the authors of this contribution developed an approximation-based synthesis approach which is based on the notion of l-complete abstractions. Like other approximation-based methods, our approach suffers from the "curse of complexity": state sets of approximating DESs may become very large, and, as the subsequent control synthesis step involves forming the product of approximating DESs and a realisation of the specifications, computational effort can become excessive even for seemingly "small" applications. Obviously, complexity also represents a major problem in other control contexts, and it is common engineering knowledge that suitable decomposition techniques form a necessary ingredient for any systematic treatment of complex control problems. Hierarchical approaches are a particularly attractive way of problem decomposition as they provide an extremely intuitive control architecture.

This contribution presents a hierarchical synthesis framework which is general enough to encompass both continuous and discrete levels and is therefore especially suited for hybrid control problems. It is based on two previous (rather technical) conference papers [13, 12]. To keep exposition reasonably straightforward, we focus on the case of two control layers. Unlike heuristic approaches, our synthesis framework guarantees that the control layers interact "properly" and do indeed enforce the overall specifications for the considered plant model. 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, our approach allows 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 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 in [19], but is technically quite different because we employ an input/output structure to adequately represent both time and event driven dynamics for hybrid systems. There is also a strong conceptual link to [8], where, as in [15, 4] and in our work, the preservation of fundamental properties across levels of abstraction is of prime concern.

We demonstrate the potential of our hierarchical synthesis framework by applying it to a multiproduct batch control problem which is simple enough to serve as illustration for our 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, we briefly summarise our abstraction-based approach to hybrid control systems synthesis. In Section 3, it is shown how this approach can be extended to a two-level control structure. Section 4 describes the multiproduct batch control problem. Finally, in Section 5, we apply our results to this test case.

2 Abstraction Based Supervisory Control

The purpose of this section is to briefly summarise key results from our earlier work [10, 14]. They apply to the scenario depicted in Fig. 1. There, the plant

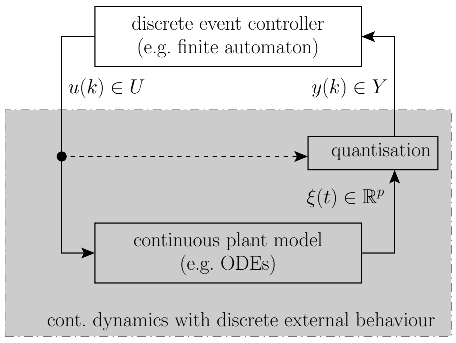

discrete event controller
(e.g. finite automaton)
$u(k) \in U$
$y(k) \in Y$
quantisation
$\xi(t) \in \mathbb{R}^p$
continuous plant model
(e.g. ODEs)
cont. dynamics with discrete external behaviour

Fig. 1. Continuous plant under discrete control.

model is continuous, realised, e.g., by a set of ODEs, but communicates with its environment exclusively via discrete events. Input events from the set U may switch the continuous dynamics, and output events from a set Y are typically generated by some sort of quantisation mechanism. Hence both the input and the output signal are sequences of discrete events, denoted by u and y, respectively. Note that we do *not* need to specify at this point whether events occur at equidistant instants of time ("time-driven sampling", "clock time") or at instants of time that are defined by the plant dynamics, e.g., by continuous signals crossing certain thresholds ("event-driven sampling", "logic time"). In J.C. Willems' terminology (e.g. [18]), the (external) behaviour of a dynamic system is the set of external signals that the system can evolve on. Hence, with $w := (u, y)$ and $W := U \times Y$, the external plant behaviour $\mathfrak{B}_\mathrm{p}$ is a set of maps $w\colon \mathbb{N}_0 \to W$; i.e. $\mathfrak{B}_\mathrm{p} \subseteq W^{\mathbb{N}_0}$, where $\mathbb{N}_0$ is the set of nonnegative integers and $W^{\mathbb{N}_0} := \{w\colon \mathbb{N}_0 \to W\}$ represents the set of all sequences in W.

To clarify the input/output structure, we use a slightly weakened version of Willems' I/O-behaviours:

Definition 1. $\mathfrak{B} \subseteq W^{\mathbb{N}_0}$ *is a* (strict) I/- behaviour with respect to (U, Y)*, if*

(i) the input is free*, i.e.* $\mathcal{P}_\mathrm{U}\mathfrak{B} = U^{\mathbb{N}_0}$ *and*

(ii) the output does (strictly) not anticipate the input[5], *i.e.*

$$\mathcal{P}_{\mathrm{U}}\tilde{w}|_{[0,k]} = \mathcal{P}_{\mathrm{U}}\hat{w}|_{[0,k]} \\ \Rightarrow (\exists\, w \in \mathfrak{B})\mathcal{P}_{\mathrm{Y}}w|_{[0,k]} = \mathcal{P}_{\mathrm{Y}}\tilde{w}|_{[0,k]} \text{ and } \mathcal{P}_{\mathrm{U}}w = \mathcal{P}_{\mathrm{U}}\hat{w}$$

for all $k \in \mathbb{N}_0$, $\tilde{w}$, $\hat{w} \in \mathfrak{B}$; *for the* strict *case the premiss on the l.h.s. is weakened to* $\mathcal{P}_{\mathrm{U}}\tilde{w}|_{[0,k)} = \mathcal{P}_{\mathrm{U}}\hat{w}|_{[0,k)}$.

Item (ii) in Def. 1 says that we can change the future (and, in the strict case, the present) of the input without affecting present and past of the output.

We now focus on the role of a controller, or supervisor, evolving on the same signal space as the plant model. Adopting the concepts of supervisory control theory for DESs [17] to the behavioural framework, the task of a supervisor $\mathfrak{B}_{\mathrm{sup}} \subseteq W^{\mathbb{N}_0}$ is to restrict the plant behaviour $\mathfrak{B}_{\mathrm{p}} \subseteq W^{\mathbb{N}_0}$ such that the closed loop behaviour contains only acceptable signals. The closed loop behaviour $\mathfrak{B}_{\mathrm{cl}}$ is $\mathfrak{B}_{\mathrm{p}} \cap \mathfrak{B}_{\mathrm{sup}}$, because a signal $w \in \mathfrak{B}_{\mathrm{p}}$ "survives" closing the loop if and only if it is also in $\mathfrak{B}_{\mathrm{sup}}$. We collect all acceptable signals in the specification behaviour $\mathfrak{B}_{\mathrm{spec}}$ and say that the supervisor $\mathfrak{B}_{\mathrm{sup}}$ *enforces the specification* if $\mathfrak{B}_{\mathrm{cl}} \subseteq \mathfrak{B}_{\mathrm{spec}}$. It is immediately clear that any supervisor must exhibit two additional properties: (i) it must respect the I/O structure of the plant, i.e., it may restrict the plant input but then has to accept whatever output event the plant generates; (ii) it must ensure that, at any instant of time, there is a possible future evolution for the closed loop. This is formalised by the following definition:

Definition 2. *A supervisor* $\mathfrak{B}_{\mathrm{sup}} \subseteq W^{\mathbb{N}_0}$ *is* admissible *to* $\mathfrak{B}_{\mathrm{p}} \subseteq W^{\mathbb{N}_0}$ *if*

(i) $\mathfrak{B}_{\mathrm{sup}}$ *is* generically implementable, *i.e.* $k \in \mathbb{N}_0$, $w|_{[0,k]} \in \mathfrak{B}_{\mathrm{sup}}|_{[0,k]}$, $\tilde{w}|_{[0,k]} \in W^{k+1}$, $\tilde{w}|_{[0,k]} \approx_{\mathrm{y}} w|_{[0,k]}$ *implies* $\tilde{w}|_{[0,k]} \in \mathfrak{B}_{\mathrm{sup}}|_{[0,k]}$; *and*
(ii) $\mathfrak{B}_{\mathrm{p}}$ *and* $\mathfrak{B}_{\mathrm{sup}}$ *are* non-conflicting, *i.e.* $\mathfrak{B}_{\mathrm{p}}|_{[0,k]} \cap \mathfrak{B}_{\mathrm{sup}}|_{[0,k]} = (\mathfrak{B}_{\mathrm{p}} \cap \mathfrak{B}_{\mathrm{sup}})|_{[0,k]}$ *for all* $k \in \mathbb{N}_0$.

This leads to the following formulation of supervisory control problems.

Definition 3. *Given a plant* $\mathfrak{B}_{\mathrm{p}} \subseteq W^{\mathbb{N}_0}$, $W = U \times Y$, *and a specification* $\mathfrak{B}_{\mathrm{spec}} \subseteq W^{\mathbb{N}_0}$, *the pair* $(\mathfrak{B}_{\mathrm{p}}, \mathfrak{B}_{\mathrm{spec}})_{\mathrm{cp}}$ *is a* supervisory control problem. *A supervisor* $\mathfrak{B}_{\mathrm{sup}} \subseteq W^{\mathbb{N}_0}$ *that is admissible to* $\mathfrak{B}_{\mathrm{p}}$ *and that enforces* $\mathfrak{B}_{\mathrm{spec}}$ *is said to be a* solution *of* $(\mathfrak{B}_{\mathrm{p}}, \mathfrak{B}_{\mathrm{spec}})_{\mathrm{cp}}$.

In [10, 11], we adapt the set-theoretic argument of [17] to show the unique existence of the *least restrictive solution* for our class of supervisory control

[5] The *restriction* operator $(\,\cdot\,)|_{[k_1,k_2)}$ maps sequences $w \in W^{\mathbb{N}_0}$ to finite strings $w|_{[k_1,k_2)} := w(k_1)w(k_1+1)\cdots w(k_2-1) \in W^{k_2-k_1}$, where $W^0 := \{\epsilon\}$ and ϵ denotes the *empty string.* For closed intervals, $(\,\cdot\,)|_{[k_1,k_2]}$ is defined accordingly.

For $W = U \times Y$, the symbols $\mathcal{P}_{\mathrm{U}}$ and $\mathcal{P}_{\mathrm{Y}}$ denote the *natural projection* to the resp. component, i.e. $\mathcal{P}_{\mathrm{U}}w = u$ and $\mathcal{P}_{\mathrm{Y}}w = y$ for $w = (u, y)$, $u \in U^{\mathbb{N}_0}$, $y \in Y^{\mathbb{N}_0}$. We use $\tilde{w}|_{[0,k]} \approx_{\mathrm{y}} w|_{[0,k]}$ as an abbreviation for the two strings to be identical up to the last output event, i.e. $\mathcal{P}_{\mathrm{U}}\tilde{w}|_{[0,k]} = \mathcal{P}_{\mathrm{U}}w|_{[0,k]}$ and $\mathcal{P}_{\mathrm{Y}}\tilde{w}|_{[0,k)} = \mathcal{P}_{\mathrm{Y}}w|_{[0,k)}$.

problems. Note that the least restrictive solution may be trivial, i.e. $\mathfrak{B}_{\mathrm{sup}} = \emptyset$, as this is admissible to the plant and, because of $\mathfrak{B}_{\mathrm{p}} \cap \emptyset \subseteq \mathfrak{B}_{\mathrm{spec}}$, enforces the specifications. Obviously, only nontrivial solutions are of interest. Therefore, if $\emptyset$ turns out to be the least restrictive solution of $(\mathfrak{B}_{\mathrm{p}}, \mathfrak{B}_{\mathrm{spec}})_{\mathrm{cp}}$, one would conclude that the specifications are "too strict" for the given plant model.

If both $\mathfrak{B}_{\mathrm{p}}$ and $\mathfrak{B}_{\mathrm{spec}}$ could be realised by finite automata, we could easily compute (a realisation of) the least restrictive solution of $(\mathfrak{B}_{\mathrm{p}}, \mathfrak{B}_{\mathrm{spec}})_{\mathrm{cp}}$ by appropriately modifying standard DES tools. While a finite automaton realisation of $\mathfrak{B}_{\mathrm{spec}} \in W^{\mathbb{N}_0}$ is quite common for finite W, the hybrid plant is in general *not* realisable on a finite state space. In [16, 10], we suggest to approach this problem by replacing $\mathfrak{B}_{\mathrm{p}}$ with an *abstraction*[6] $\mathfrak{B}_{\mathrm{ca}}$ that is realised by a finite automaton. We can then readily establish a solution $\mathfrak{B}_{\mathrm{sup}}$ of the (purely discrete) control problem $(\mathfrak{B}_{\mathrm{ca}}, \mathfrak{B}_{\mathrm{spec}})_{\mathrm{cp}}$. Clearly, because of $\mathfrak{B}_{\mathrm{p}} \cap \mathfrak{B}_{\mathrm{sup}} \subseteq \mathfrak{B}_{\mathrm{ca}} \cap \mathfrak{B}_{\mathrm{sup}} \subseteq \mathfrak{B}_{\mathrm{spec}}$, the resulting supervisor also enforces the specifications for the original plant $\mathfrak{B}_{\mathrm{p}}$. To show that $\mathfrak{B}_{\mathrm{sup}}$ is also admissible to $\mathfrak{B}_{\mathrm{p}}$ and hence solves the original (hybrid) control problem $(\mathfrak{B}_{\mathrm{p}}, \mathfrak{B}_{\mathrm{spec}})_{\mathrm{cp}}$, we employ the notion of *completeness*:

Definition 4. *[18] A behaviour $\mathfrak{B} \subseteq W^{\mathbb{N}_0}$ is* complete *if*

$$w \in \mathfrak{B} \quad \Leftrightarrow \quad \forall\, k \in \mathbb{N}_0 : \; w|_{[0,k]} \in \mathfrak{B}|_{[0,k]} \,.$$

Hence, to decide whether a signal w belongs to a complete behaviour, it is sufficient to look at "finite length portions" of w. The external behaviour induced by a finite state machine is an example for completeness. $\mathfrak{B} = \{w \in \mathbb{R}^{\mathbb{N}_0} |\ \lim_{k\to\infty} w(k) = 0\}$, on the other hand, is *not* complete. As a consequence of the following proposition, admissibility of a supervisor is independent of the particular plant dynamics provided that all involved behaviours are complete:

Proposition 1. *[11] Let $\mathfrak{B}_{\mathrm{p}} \subseteq W^{\mathbb{N}_0}$ be a complete I/- behaviour and $\mathfrak{B}_{\mathrm{sup}} \subseteq W^{\mathbb{N}_0}$ be complete and generically implementable. Then $\mathfrak{B}_{\mathrm{p}}$ and $\mathfrak{B}_{\mathrm{sup}}$ are nonconflicting.*

For the remainder of this paper, we restrict consideration to complete behaviours. Theorem 1 then follows immediately from Proposition 1.

Theorem 1. *Let $\mathfrak{B}_{\mathrm{ca}} \subseteq W^{\mathbb{N}_0}$, $W = U \times Y$, be an abstraction of an I/- behaviour $\mathfrak{B}_{\mathrm{p}} \subseteq W^{\mathbb{N}_0}$, let $\mathfrak{B}_{\mathrm{spec}} \subseteq W^{\mathbb{N}_0}$, and let $\mathfrak{B}_{\mathrm{sup}} \subseteq W^{\mathbb{N}_0}$ be a nontrivial solution of the supervisory control problem $(\mathfrak{B}_{\mathrm{ca}}, \mathfrak{B}_{\mathrm{spec}})_{\mathrm{cp}}$. If $\mathfrak{B}_{\mathrm{p}}$ and $\mathfrak{B}_{\mathrm{sup}}$ are complete then $\mathfrak{B}_{\mathrm{sup}}$ is a nontrivial solution of $(\mathfrak{B}_{\mathrm{p}}, \mathfrak{B}_{\mathrm{spec}})_{\mathrm{cp}}$.*

In practice, to make our approach work, a sequence of increasingly refined abstractions $\mathfrak{B}_l$, $l = 1, 2, \ldots$, of $\mathfrak{B}_{\mathrm{p}}$, i.e. $\mathfrak{B}_{\mathrm{p}} \subseteq \ldots \mathfrak{B}_{l+1} \subseteq \mathfrak{B}_l \ldots \subseteq \mathfrak{B}_1$, is employed. In [16, 10], we suggest l-complete approximations as candidate abstractions $\mathfrak{B}_l$. One then begins with the "least accurate" abstraction $\mathfrak{B}_1$,

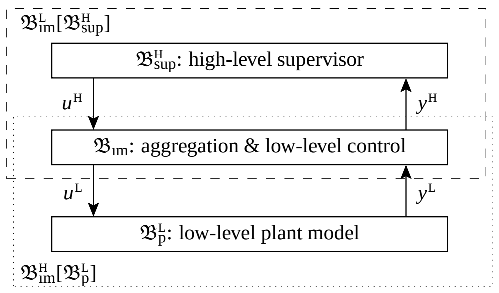
[6] $\mathfrak{B}_{\mathrm{ca}}$ is said to be an abstraction of $\mathfrak{B}_{\mathrm{p}}$, if $\mathfrak{B}_{\mathrm{p}} \subseteq \mathfrak{B}_{\mathrm{ca}}$.

checks whether a non-trivial solution of $(\mathfrak{B}_1, \mathfrak{B}_{\rm spec})_{\rm cp}$ exists and, if this is not the case, turns to the refined abstraction $\mathfrak{B}_2$. In this way, refinement and discrete control synthesis steps alternate until either a nontrivial solution to $(\mathfrak{B}_l, \mathfrak{B}_{\rm spec})_{\rm cp}$ and hence to $(\mathfrak{B}_{\rm p}, \mathfrak{B}_{\rm spec})_{\rm cp}$ is found or computational resources are exhausted. Unfortunately, the latter case often turns out be true. This is the motivation for a hierarchical extension of our approach.

3 Hierarchical Control

3.1 Control architecture

To simplify exposition, we concentrate on the two-level control architecture shown in Fig. 2. Low-level control is implemented by an intermediate layer



$\mathfrak{B}^{\rm L}_{\rm im}[\mathfrak{B}^{\rm H}_{\rm sup}]$
$\mathfrak{B}^{\rm H}_{\rm sup}$: high-level supervisor
$u^{\rm H}$
$y^{\rm H}$
$\mathfrak{B}_{\rm im}$: aggregation & low-level control
$u^{\rm L}$
$y^{\rm L}$
$\mathfrak{B}^{\rm L}_{\rm p}$: low-level plant model
$\mathfrak{B}^{\rm H}_{\rm im}[\mathfrak{B}^{\rm L}_{\rm p}]$

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

$\mathfrak{B}_{\rm im}$ communicating with the plant[7] $\mathfrak{B}^{\rm L}_{\rm p}$ via low-level signals $u^{\rm L}$ and $y^{\rm L}$ and with the high-level supervisor $\mathfrak{B}^{\rm H}_{\rm sup}$ via high-level signals $u^{\rm H}$ and $y^{\rm H}$. Apart from implementing low-level control mechanisms corresponding to high-level commands $u^{\rm H}$, the intermediate layer $\mathfrak{B}_{\rm im}$ aggregates low-level measurement information $y^{\rm L}$ to provide high-level information $y^{\rm H}$ to $\mathfrak{B}^{\rm H}_{\rm sup}$. Aggregation may be both in signal space and in time, i.e. the time axis for high-level signals may be "coarser" than for low-level signals. Note that in this scenario $\mathfrak{B}_{\rm im}$ is a behaviour on $W_{\rm H} \times W_{\rm L}$, where $W_{\rm H} := U_{\rm H} \times Y_{\rm H}$ and $W_{\rm L} := U_{\rm L} \times Y_{\rm L}$ represent the high and low-level signal sets.

From the perspective of the (low-level) plant $\mathfrak{B}^{\rm L}_{\rm p}$, interconnecting $\mathfrak{B}_{\rm im}$ and $\mathfrak{B}^{\rm H}_{\rm sup}$ provides the overall controller. Its external behaviour is denoted by $\mathfrak{B}^{\rm L}_{\rm im}[\mathfrak{B}^{\rm H}_{\rm sup}]$ and, as indicated by the dashed box in Figure 2, evolves on the low-level signal space $W_{\rm L}$. The behaviour $\mathfrak{B}^{\rm L}_{\rm im}[\mathfrak{B}^{\rm H}_{\rm sup}]$ is given by the projection of $\mathfrak{B}_{\rm im}$ into $W_{\rm L}^{\mathbb{N}_0}$ with the internal high-level signal restricted to $\mathfrak{B}^{\rm H}_{\rm sup}$:

[7] To make notation easier to read, all high-level signals, signal sets and behaviours will be indicated by a sub- or superscript "H", while low-level entities will be characterised by a sub- or superscript "L". As the plant evolves on a physical, i.e. low-level signal set, its behaviour will be denoted by $\mathfrak{B}^{\rm L}_{\rm p}$ from now on.

$$\mathfrak{B}^{\text{L}}_{\text{IM}}[\mathfrak{B}^{\text{H}}_{\text{sup}}] := \{w^{\text{L}} |\, (\exists\, w^{\text{H}} \in \mathfrak{B}^{\text{H}}_{\text{sup}})[\,(w^{\text{H}},\, w^{\text{L}}) \in \mathfrak{B}_{\text{IM}}\,]\}. \tag{1}$$

Clearly, we require the overall controller $\mathfrak{B}^{\text{L}}_{\text{IM}}[\mathfrak{B}^{\text{H}}_{\text{sup}}]$ to be a (nontrivial) solution of the original control problem $(\mathfrak{B}^{\text{L}}_{\text{p}},\, \mathfrak{B}^{\text{L}}_{\text{spec}})_{\text{cp}}$. In particular, the overall controller is required to enforce the specification, i.e. we need

$$\mathfrak{B}^{\text{L}}_{\text{p}} \cap \mathfrak{B}^{\text{L}}_{\text{IM}}[\mathfrak{B}^{\text{H}}_{\text{sup}}] \subseteq \mathfrak{B}^{\text{L}}_{\text{spec}}\,. \tag{2}$$

We now re-examine Fig. 2: from the perspective of the high-level supervisor $\mathfrak{B}^{\text{H}}_{\text{sup}}$, interconnecting the intermediate layer $\mathfrak{B}_{\text{IM}}$ with the (low-level) plant model $\mathfrak{B}^{\text{L}}_{\text{p}}$ provides a compound high-level plant model. Its external behaviour is denoted by $\mathfrak{B}^{\text{H}}_{\text{IM}}[\mathfrak{B}^{\text{L}}_{\text{p}}]$ and, as indicated by the dotted box in Fig. 2, evolves on the high-level signal set W_{H}. The behaviour $\mathfrak{B}^{\text{H}}_{\text{IM}}[\mathfrak{B}^{\text{L}}_{\text{p}}]$ is the projection of $\mathfrak{B}_{\text{IM}}$ into $W_{\text{H}}^{\mathbb{N}_0}$ with the internal low-level signal restricted to $\mathfrak{B}^{\text{L}}_{\text{p}}$:

$$\mathfrak{B}^{\text{H}}_{\text{IM}}[\mathfrak{B}^{\text{L}}_{\text{p}}] := \{w^{\text{H}} |\ (\exists\, w^{\text{L}} \in \mathfrak{B}^{\text{L}}_{\text{p}})[\,(w^{\text{H}},\, w^{\text{L}}) \in \mathfrak{B}_{\text{IM}}\,]\}\,. \tag{3}$$

By the same argument as before, the high-level supervisor $\mathfrak{B}^{\text{H}}_{\text{sup}}$ is required to be admissible to the compound high-level plant model $\mathfrak{B}^{\text{H}}_{\text{IM}}[\mathfrak{B}^{\text{L}}_{\text{p}}]$, i.e. $\mathfrak{B}^{\text{H}}_{\text{sup}}$ must be generically implementable, and $\mathfrak{B}^{\text{H}}_{\text{IM}}[\mathfrak{B}^{\text{L}}_{\text{p}}]$ and $\mathfrak{B}^{\text{H}}_{\text{sup}}$ must be non-conflicting.

We summarise our discussion of Figure 2 in the following definition.

Definition 5. *The pair* $(\mathfrak{B}_{IM},\, \mathfrak{B}^{\text{H}}_{\text{sup}})_{\text{tl}}$ *is a* two-level hierarchical solution *of the supervisory control problem* $(\mathfrak{B}^{\text{L}}_{\text{p}},\, \mathfrak{B}^{\text{L}}_{\text{spec}})_{\text{cp}}$ *if*

(i) $\mathfrak{B}^{\text{L}}_{\text{p}} \cap \mathfrak{B}^{\text{L}}_{IM}[\mathfrak{B}^{\text{H}}_{\text{sup}}] \subseteq \mathfrak{B}^{\text{L}}_{\text{spec}}$*, and*

(iia) $\mathfrak{B}^{\text{L}}_{IM}[\mathfrak{B}^{\text{H}}_{\text{sup}}]$ *is admissible to* $\mathfrak{B}^{\text{L}}_{\text{p}}$*, and*

(iib) $\mathfrak{B}^{\text{H}}_{\text{sup}}$ *is admissible to* $\mathfrak{B}^{\text{H}}_{IM}[\mathfrak{B}^{\text{L}}_{\text{p}}]$*.*

We will now investigate how to make sure that the admissibility conditions (iia) and (iib) in Def. 5 hold. More precisely, we will discuss which property of $\mathfrak{B}_{\text{IM}}$ will help to enforce these conditions. To structure the discussion, we first address the case of uniform time scales on both signal levels. A layer suitably mediating between different time scales will be investigated subsequently.

Uniform time scales – type I intermediate layer

From Fig. 2 it is obvious that u^{H} and y^{L} can be interpreted as inputs of the intermediate layer $\mathfrak{B}_{\text{IM}}$, while u^{L} and y^{H} are outputs. It is therefore natural to require that $\mathfrak{B}_{\text{IM}}$ is a (strict) I/- behaviour w.r.t. $(U_{\text{H}} \times Y_{\text{L}},\, Y_{\text{H}} \times U_{\text{L}})$. If it is also complete, I/- and completeness properties are passed from $\mathfrak{B}^{\text{L}}_{\text{p}}$ to to $\mathfrak{B}^{\text{H}}_{\text{IM}}[\mathfrak{B}^{\text{L}}_{\text{p}}]$. Formally, this can be stated as

Lemma 1. *If* $\mathfrak{B}_{IM}$ *is a complete strict I/- behaviour w.r.t.* $(U_{\text{H}} \times Y_{\text{L}},\, Y_{\text{H}} \times U_{\text{L}})$*, and if* $\mathfrak{B}^{\text{L}}_{\text{p}}$ *is a complete I/- behaviour w.r.t.* $(U_{\text{L}},\, Y_{\text{L}})$*, then* $\mathfrak{B}^{\text{H}}_{IM}[\mathfrak{B}^{\text{L}}_{\text{p}}]$ *is a complete I/- behaviour w.r.t.* $(U_{\text{H}},\, Y_{\text{H}})$*.*

The same property of $\mathfrak{B}_{\text{IM}}$ ensures that completeness and generic implementability carry over from $\mathfrak{B}^{\text{H}}_{\text{sup}}$ to $\mathfrak{B}^{\text{L}}_{\text{IM}}[\mathfrak{B}^{\text{H}}_{\text{sup}}]$. Formally:

Lemma 2. *If $\mathfrak{B}_{\imath m}$ is a complete strict I/- behaviour w.r.t. $(U_{\mathrm{H}} \times Y_{\mathrm{L}},\ Y_{\mathrm{H}} \times U_{\mathrm{L}})$, and if $\mathfrak{B}^{\mathrm{H}}_{\mathrm{sup}}$ is complete and generically implementable, then $\mathfrak{B}^{\mathrm{L}}_{\imath m}[\mathfrak{B}^{\mathrm{H}}_{\mathrm{sup}}]$ is complete and generically implementable.*

From Lemma 1 and 2, we can immediately deduce the following important statement: if the plant model $\mathfrak{B}^{\mathrm{L}}_{\mathrm{p}}$ is a complete I/- behaviour, if the intermediate layer $\mathfrak{B}_{\mathrm{im}}$ is a complete strict I/-behaviour, and if the high-level supervisor is both complete and generically implementable, then the admissibility Conditions (iia) and (iib) are satisfied.

Multiple time scales – type II intermediate layer

High-level and low-level signals are often defined on different time scales. Typically, in technical realisations, low-level signals "live" on a discrete time axis that is obtained by (fast) equidistant sampling. High-level signals mostly live on a time axis that is generated by low-level signals. A common scenario is that y^{L} produces events, e.g. when certain thresholds are crossed. y^{H} is then a sequence of events, and its time axis is constituted by event times. We assume that high-level commands are immediately issued after the occurrence of a high-level measurement event, hence u^{H} lives on the same time axis as y^{H}. This scenario is illustrated in Fig. 3. We call the resulting high-level time a *dynamic time scale* and formally define this notion as follows:

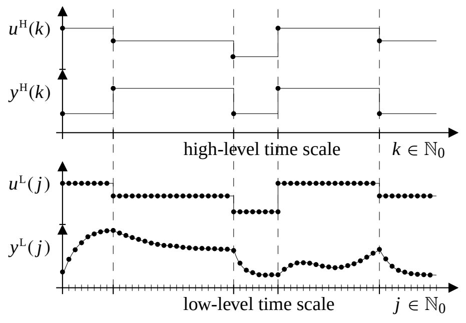


Fig. 3. Two time scales.

Definition 6. *Let $T\colon Y_{\mathrm{L}}^{\mathbb{N}_0} \to \mathbb{N}_0{}^{\mathbb{N}_0}$. The operator T is said to be a* dynamic time scale *if T is strictly causal*[8] *and if the* time transformation $T(y^{\mathrm{L}})\colon \mathbb{N}_0 \to \mathbb{N}_0$ *is surjective and monotonically increasing for all $y^{\mathrm{L}} \in Y_{\mathrm{L}}^{\mathbb{N}_0}$.*

For a fixed low-level signal y^{L}, the time transformation $T(y^{\mathrm{L}})$ maps low-level time $j \in \mathbb{N}_0$ to high-level time $k \in \mathbb{N}_0$. By requiring that T itself is a strictly

[8] Recall that an operator $H\colon U^{\mathbb{N}_0} \to Y^{\mathbb{N}_0}$, i.e. an operator mapping signals u to signals y, is called *strictly causal* if $\tilde{u}|_{[0,k)} = \hat{u}|_{[0,k)} \quad \Rightarrow \quad H(\tilde{u})|_{[0,k]} = H(\hat{u})|_{[0,k]}$ for all $k \in \mathbb{N}_0$, $\tilde{u}$, $\hat{u} \in U^{\mathbb{N}_0}$.

causal operator, we ensure that at any instant of time the transformation $T(y^{\mathrm{L}})$ only depends on the strict past of y^{L}.

We focus on measurement aggregation operators that are causal with respect to a dynamic time scale:

Definition 7. *The operator $F\colon Y_{\mathrm{L}}^{\mathbb{N}_0} \to Y_{\mathrm{H}}^{\mathbb{N}_0}$ is said to be* causal w.r.t. T *if T is a dynamic time scale and if*

$$\tilde{y}^{\mathrm{L}}|_{[0,j]} = \hat{y}^{\mathrm{L}}|_{[0,j]} \quad \Rightarrow \quad F(\tilde{y}^{\mathrm{L}})|_{[0,k]} = F(\hat{y}^{\mathrm{L}})|_{[0,k]} \tag{4}$$

for $k = T(\tilde{y}^{\mathrm{L}})(j)$ and all $j \in \mathbb{N}_0$, $\tilde{y}^{\mathrm{L}}$, $\hat{y}^{\mathrm{L}} \in Y_{\mathrm{L}}^{\mathbb{N}_0}$.

We still have to link high-level control signals u^{H} to low-level control signals u^{L}. This is done via a sample-and-hold device that is triggered by the time transformation $T(y^{\mathrm{L}})$, i.e. successive high-level control actions are passed on to the lower level whenever a high-level measurement is generated. Formally, this is expressed by $u^{\mathrm{L}} = u^{\mathrm{H}} \circ T(y^{\mathrm{L}})$.

In summary, an intermediate layer $\mathfrak{B}_{\mathrm{im}}$ mediating between low-level and high-level time is completely defined by a dynamic time scale T and a measurement aggregation operator F that is causal w.r.t. T:

$$\mathfrak{B}_{\mathrm{im}} := \{(u^{\mathrm{H}}, y^{\mathrm{H}}, u^{\mathrm{L}}, y^{\mathrm{L}})|\ y^{\mathrm{H}} = F(y^{\mathrm{L}}) \ \text{ and } \ u^{\mathrm{L}} = u^{\mathrm{H}} \circ T(y^{\mathrm{L}})\}\,. \tag{5}$$

It can be shown that (5) represents a complete behaviour and, like the intermediate layer discussed in the previous section, preserves the input/output structure of the plant and generic implementability of the supervisor.

Lemma 3. *For $\mathfrak{B}_{im}$ given by* (5) *and $\mathfrak{B}_{\mathrm{p}}^{\mathrm{L}}$ a complete I/- behaviour w.r.t. $(U_{\mathrm{L}},\, Y_{\mathrm{L}})$, it follows that $\mathfrak{B}_{im}^{\mathrm{H}}[\mathfrak{B}_{\mathrm{p}}^{\mathrm{L}}]$ is a complete I/- behaviour w.r.t. $(U_{\mathrm{H}},\, Y_{\mathrm{H}})$.*

Lemma 4. *If $\mathfrak{B}_{im}$ is given by* (5)*, and if $\mathfrak{B}_{\mathrm{sup}}^{\mathrm{H}}$ is complete and generically implementable, it follows that $\mathfrak{B}_{im}^{\mathrm{L}}[\mathfrak{B}_{\mathrm{sup}}^{\mathrm{H}}]$ is complete and generically implementable.*

In most practical situations, we will have to combine the two types of intermediate layers discussed on the previous pages. It is intuitively clear that combinations of type I and type II layers will also preserve the input/output structure of $\mathfrak{B}_{\mathrm{p}}^{\mathrm{L}}$ and generic implementability of $\mathfrak{B}_{\mathrm{sup}}^{\mathrm{H}}$ across the resulting intermediate layer. We will therefore omit a formal treatment (for this, the interested reader is referred to [13]) and conclude the discussion on structural properties of $\mathfrak{B}_{\mathrm{im}}$ by collecting the relevant facts in the following proposition.

Proposition 2. *If the plant model $\mathfrak{B}_{\mathrm{p}}^{\mathrm{L}}$ is a complete I/- behaviour, if the intermediate layer $\mathfrak{B}_{im}$ is an arbitrary combination of type I layers (i.e. complete strict I/-behaviours) and of type II layers (i.e. given by* (5)*), and if the high-level supervisor is both complete and generically implementable, then the admissibility Conditions (iia) and (iib) are satisfied.*

We are now in a position to discuss how to design $\mathfrak{B}_{\mathrm{im}}$ and $\mathfrak{B}_{\mathrm{sup}}^{\mathrm{H}}$ (subject to the above constraints) such that $\mathfrak{B}_{\mathrm{p}}^{\mathrm{L}} \cap \mathfrak{B}_{\mathrm{im}}^{\mathrm{L}}[\mathfrak{B}_{\mathrm{sup}}^{\mathrm{H}}] \subseteq \mathfrak{B}_{\mathrm{spec}}^{\mathrm{L}}$, the last remaining condition from Definition 5, is also satisfied and $(\mathfrak{B}_{\mathrm{im}},\, \mathfrak{B}_{\mathrm{sup}}^{\mathrm{H}})_{\mathrm{tl}}$ therefore forms a two-level hierarchical solution of the control problem $(\mathfrak{B}_{\mathrm{p}}^{\mathrm{L}},\, \mathfrak{B}_{\mathrm{spec}}^{\mathrm{L}})_{\mathrm{cp}}$.

3.2 A bottom-up design procedure

We suggest an intuitive bottom-up procedure where we first design appropriate low-level control $\mathfrak{B}_{\text{IM}}$ and then find a suitable high-level supervisor $\mathfrak{B}^{\text{H}}_{\text{sup}}$.

In a first step, we formalise the *intended* relation between high-level and low-level signals by the specification $\mathfrak{B}^{\text{HL}}_{\text{spec}} \subseteq (W_{\text{H}} \times W_{\text{L}})^{\mathbb{N}_0}$. Hence $\mathfrak{B}^{\text{HL}}_{\text{spec}}$ denotes the set of all signal pairs $(w^{\text{H}},\, w^{\text{L}})$ that represent the desired effect of high-level control on the low-level plant $\mathfrak{B}^{\text{L}}_{\text{p}}$ and, by implication, on the high-level measurement. To ensure that w^{H} and w^{L} are related in the intended way, we require the intermediate layer $\mathfrak{B}_{\text{IM}}$ to enforce the specification $\mathfrak{B}^{\text{HL}}_{\text{spec}}$ when connected to the low-level plant $\mathfrak{B}^{\text{L}}_{\text{p}}$. This condition is expressed by:

$$\{(w^{\text{H}},\, w^{\text{L}}) \in \mathfrak{B}_{\text{IM}} |\; w^{\text{L}} \in \mathfrak{B}^{\text{L}}_{\text{p}}\} \subseteq \mathfrak{B}^{\text{HL}}_{\text{spec}}\,. \tag{6}$$

Suppose we have designed $\mathfrak{B}_{\text{IM}}$ to enforce (6), perhaps using classical continuous control methods. In principle, we could then base the design of $\mathfrak{B}^{\text{H}}_{\text{sup}}$ on the compound high-level plant $\mathfrak{B}^{\text{H}}_{\text{IM}}[\mathfrak{B}^{\text{L}}_{\text{p}}]$. However, from a computational point of view — particularly for hybrid systems — it is preferable to use an abstraction $\tilde{\mathfrak{B}}^{\text{H}}_{\text{p}}$ of $\mathfrak{B}^{\text{H}}_{\text{IM}}[\mathfrak{B}^{\text{L}}_{\text{p}}]$ that does not explicitly depend on the low-level dynamics or the precise nature of the implemented low-level control scheme. A suitable abstraction $\tilde{\mathfrak{B}}^{\text{H}}_{\text{p}}$ and a high-level specification $\tilde{\mathfrak{B}}^{\text{H}}_{\text{spec}}$ expressing $\mathfrak{B}^{\text{L}}_{\text{spec}}$ in terms of high-level signals can be derived from (6):

$$\tilde{\mathfrak{B}}^{\text{H}}_{\text{p}} := \{w^{\text{H}} |\, (\exists\, w^{\text{L}})[\,(w^{\text{H}},\, w^{\text{L}}) \in \mathfrak{B}^{\text{HL}}_{\text{spec}}]\,\}\,; \tag{7}$$

$$\tilde{\mathfrak{B}}^{\text{H}}_{\text{spec}} := \{w^{\text{H}} |\; (\forall\; w^{\text{L}})\,[\,(w^{\text{H}},\, w^{\text{L}}) \in \mathfrak{B}^{\text{HL}}_{\text{spec}} \quad\Rightarrow\quad w^{\text{L}} \in \mathfrak{B}^{\text{L}}_{\text{spec}}\,]\,\}\,. \tag{8}$$

In other words, the abstraction $\tilde{\mathfrak{B}}^{\text{H}}_{\text{p}}$ of the compound high-level plant model is just the projection of the specification $\mathfrak{B}^{\text{HL}}_{\text{spec}}$ onto its high-level signal components. Then, as desired, the resulting high-level control problem $(\tilde{\mathfrak{B}}^{\text{H}}_{\text{p}},\, \tilde{\mathfrak{B}}^{\text{H}}_{\text{spec}})_{\text{cp}}$ does *not* depend on the actual low-level plant under low-level control, $\mathfrak{B}^{\text{H}}_{\text{IM}}[\mathfrak{B}^{\text{L}}_{\text{p}}]$, but only on the specification $\mathfrak{B}^{\text{HL}}_{\text{spec}}$ of the preceeding low-level design step.

It follows immediately that any (nontrivial) solution $\mathfrak{B}^{\text{H}}_{\text{sup}}$ of the high-level control problem $(\tilde{\mathfrak{B}}^{\text{H}}_{\text{p}},\, \tilde{\mathfrak{B}}^{\text{H}}_{\text{spec}})_{\text{cp}}$ will enforce the original low-level specification $\mathfrak{B}^{\text{L}}_{\text{spec}}$ when connected to $\mathfrak{B}^{\text{H}}_{\text{IM}}[\mathfrak{B}^{\text{L}}_{\text{p}}]$, the plant model under low-level control:

$$\tilde{\mathfrak{B}}^{\text{H}}_{\text{p}} \cap \mathfrak{B}^{\text{H}}_{\text{sup}} \subseteq \tilde{\mathfrak{B}}^{\text{H}}_{\text{spec}} \qquad \Longrightarrow \qquad \mathfrak{B}^{\text{L}}_{\text{p}} \cap \mathfrak{B}^{\text{L}}_{\text{IM}}[\mathfrak{B}^{\text{H}}_{\text{sup}}] \subseteq \mathfrak{B}^{\text{L}}_{\text{spec}}\,. \tag{9}$$

Hence Condition (i) from Def. 5 also holds, and the pair $(\mathfrak{B}_{\text{IM}},\, \mathfrak{B}^{\text{H}}_{\text{sup}})_{\text{tl}}$ is a *two-level hierarchical solution* of the overall control problem $(\mathfrak{B}^{\text{L}}_{\text{p}},\, \mathfrak{B}^{\text{L}}_{\text{spec}})_{\text{cp}}$.

If $\tilde{\mathfrak{B}}^{\text{H}}_{\text{p}}$ and $\tilde{\mathfrak{B}}^{\text{H}}_{\text{spec}}$ can be realised by finite automata, a slight modification of standard DES methods, e.g. [17], may be used to synthesise $\mathfrak{B}^{\text{H}}_{\text{sup}}$. Such a situation is to be expected when all continuous signals are "handled" by the lower-level control scheme within $\mathfrak{B}_{\text{IM}}$. Otherwise, another abstraction step is required; see e.g. [10].

The "degree of freedom" in the proposed bottom-up approach is the specification $\mathfrak{B}^{\text{HL}}_{\text{spec}}$. In general, its choice can be guided by the same engineering intuition that we would use in a hierarchical ad hoc design. However, unlike heuristic approaches, our method encapsulates intuition in a formal framework where we can prove that the composition of high-level controller and intermediate layer forms a valid solution of the original problem.

3.3 Minimising the cost of closed-loop operation

In addition to a “hard” specification (i.e., a specification that *must* hold), many applications come with the control objective of minimising a certain cost function. To address this issue, we describe a straightforward extension of the hierarchical framework outlined in Section 3.1 and 3.2.

From the low-level perspective, we want to solve the control problem:

$$\min_{\mathfrak{B}^{\mathrm{L}}_{\mathrm{sup}}} \max_{w^{\mathrm{L}}} \gamma^{\mathrm{L}}(w^{\mathrm{L}}) \,\text{s.t.}\, \mathfrak{B}^{\mathrm{L}}_{\mathrm{sup}} \text{ solves } (\mathfrak{B}^{\mathrm{L}}_{\mathrm{p}},\, \mathfrak{B}^{\mathrm{L}}_{\mathrm{spec}})_{\mathrm{cp}}\,,\; w^{\mathrm{L}} \in \mathfrak{B}^{\mathrm{L}}_{\mathrm{p}} \cap \mathfrak{B}^{\mathrm{L}}_{\mathrm{sup}}\,, \tag{10}$$

where $\gamma^{\mathrm{L}}\colon \mathfrak{B}^{\mathrm{L}}_{\mathrm{p}} \cap \mathfrak{B}^{\mathrm{L}}_{\mathrm{spec}} \to \mathbb{R}$ is a (typically additive over time and positive) function to associate the cost $\gamma^{\mathrm{L}}(w^{\mathrm{L}})$ with each low-level plant signal w^{L} that satisfies the “hard” specification $\mathfrak{B}^{\mathrm{L}}_{\mathrm{spec}}$. 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 (10) becomes obsolete. This will be the case in our multiproduct batch application; see also Sections 4 and 5.

Suppose that the intermediate layer $\mathfrak{B}_{\mathrm{im}}$ has been designed as outlined in Sections 3.1 and 3.2. We then define a pessimistic high-level cost function by

$$\gamma^{\mathrm{H}}(w^{\mathrm{H}}) := \max_{w^{\mathrm{L}}}\{\gamma^{\mathrm{L}}(w^{\mathrm{L}})|\; (w^{\mathrm{H}},\, w^{\mathrm{L}}) \in \mathfrak{B}_{\mathrm{im}},\; w^{\mathrm{L}} \in \mathfrak{B}^{\mathrm{L}}_{\mathrm{p}}\}\,, \tag{11}$$

and seek an optimal solution $\mathfrak{B}^{\mathrm{H}}_{\mathrm{sup}}$ for the high-level min-max problem

$$\min_{\mathfrak{B}^{\mathrm{H}}_{\mathrm{sup}}} \max_{w^{\mathrm{H}}} \gamma^{\mathrm{H}}(w^{\mathrm{H}}) \,\text{s.t.}\, \mathfrak{B}^{\mathrm{H}}_{\mathrm{sup}} \text{ solves } (\tilde{\mathfrak{B}}^{\mathrm{H}}_{\mathrm{p}},\, \tilde{\mathfrak{B}}^{\mathrm{H}}_{\mathrm{spec}})_{\mathrm{cp}} \text{ and } w^{\mathrm{H}} \in \tilde{\mathfrak{B}}^{\mathrm{H}}_{\mathrm{p}} \cap \mathfrak{B}^{\mathrm{H}}_{\mathrm{sup}}\,. \tag{12}$$

Note that the overall controller, i.e. the interconnection $\mathfrak{B}^{\mathrm{L}}_{\mathrm{im}}[\mathfrak{B}^{\mathrm{H}}_{\mathrm{sup}}]$ of $\mathfrak{B}_{\mathrm{im}}$ and the optimal $\mathfrak{B}^{\mathrm{H}}_{\mathrm{sup}}$ does not necessarily form an optimal solution to the original problem (10). This is for two reasons: (i) the introduction of $\mathfrak{B}_{\mathrm{im}}$ reduces the available degrees of freedom; (ii) in the high-level control problem (12), the behaviour $\mathfrak{B}^{\mathrm{H}}_{\mathrm{im}}[\mathfrak{B}^{\mathrm{L}}_{\mathrm{p}}]$ has been replaced by its abstraction $\tilde{\mathfrak{B}}^{\mathrm{H}}_{\mathrm{p}}$, resulting in over-approximation of actual costs. On the positive side, we expect the problem (12) to be computationally tractable in situations where (10) is not. We want to emphasise that, despite the tradeoff between computational effort and closed-loop performance, our bottom-up design method guarantees the “hard” specification $\mathfrak{B}^{\mathrm{L}}_{\mathrm{spec}}$ to hold.

4 Discontinuously Operated Multiproduct Batch Plant

In the chemical industries, discontinuously operated multiproduct plants are widely used for the production of fine, or specialty, chemicals. In the sequel, we describe a specific example for a multiproduct batch control problem. It is idealised to a certain extent but general enough to capture most of the problems that characterise multiproduct batch plants. The plant is used to produce three kinds of colour pigments (Fig. 4): 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: $A_i \xrightarrow{k_{Pi}} P_i$. It is accompanied by a parallel reaction $A_i \xrightarrow{k_{Wi}} W_i$ resulting in the waste product W_i. If concentration of W_i crosses a given threshold $W_{i,max}$, product quality becomes 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.

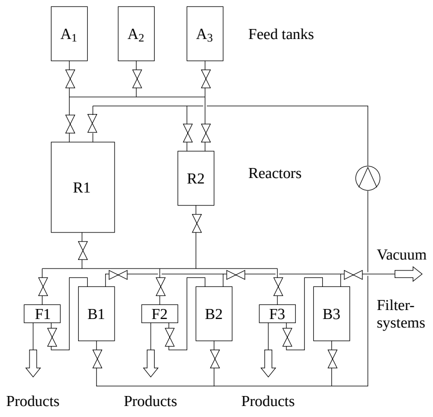


Fig. 4. Example plant.

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 the reactors. If, in any of the filters, darker colours are filtered before lighter ones (say P3 before P1 or P2, and P2 before P1), a cleaning 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 (realised by discrete valve positions) that determine whether a particular reactor is emptied through a particular filter. Heating/cooling rates 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 waste concentrations) and safety constraints (upper bounds for reactor temperatures). For simplicity, the following assumptions are made for the reactions:

1. all reactions are first order.
2. 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.

3. the time constants for heating/cooling of the reactors are small compared to the reaction time constants. The (scaled) reactor temperatures can therefore be considered to be the manipulated variables.

With these assumptions, the model equations can be easily derived from component balances:

$$\frac{d}{dt}c_{A_i}(t) = \frac{q(t)}{V} - (k_{Pi}(t) + k_{Wi}(t))\, c_{A_i}(t), \tag{13}$$
$$\frac{d}{dt}c_{P_i}(t) = k_{Pi}(t)c_{A_i}(t), \tag{14}$$
$$\frac{d}{dt}c_{W_i}(t) = k_{Wi}(t)c_{A_i}(t), \tag{15}$$

where V is the volume of the considered reactor, q the corresponding dosing rate (in kmol/h), and c_{A_i}, c_{P_i}, c_{W_i} are reactant, product and waste concentration in the ith reaction (in kmol/m^3), $i = 1, 2, 3$. The reaction kinetics

$$k_{Pi}(t) = k_{Pi_0} e^{-\frac{E_{Pi}}{R\theta(t)}}, \tag{16}$$
$$k_{Wi}(t) = k_{Wi_0} e^{-\frac{E_{Wi}}{R\theta(t)}} \tag{17}$$

are determined by temperature θ. Defining $u(t) := k_{W1}(t)$, $\beta_i := E_{Pi}/E_{W1}$, $\delta_i := E_{Wi}/E_{W1}$, $\alpha_i := k_{Pi_0}/k_{W1_0}^{\beta_i}$, $\gamma_i := k_{Wi_0}/k_{W1_0}^{\delta_i}$, we rewrite (13)–(15) as

$$\frac{d}{dt}c_{A_i}(t) = \frac{q(t)}{V} - \left(\alpha_i u(t)^{\beta_i} + \gamma_i u(t)^{\delta_i}\right) c_{A_i}(t), \tag{18}$$
$$\frac{d}{dt}c_{P_i}(t) = \alpha_i u(t)^{\beta_i} c_{A_i}(t), \tag{19}$$
$$\frac{d}{dt}c_{W_i}(t) = \gamma_i u(t)^{\delta_i} c_{A_i}(t). \tag{20}$$

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

5 Hierarchical Control of Multiproduct Batch Plant

5.1 Low-level plant model

The low-level plant model represents the continuous dynamics of filter and reaction processes in the various modes of operation. We consider low-level signals to evolve w.r.t. clock time, using a suitably small sampling period.

Note 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 plant. Thus, our low-level plant model consists of two subsystems, each of which is being used for one out of three chemical reaction schemes or a subsequent filtering process. As a low-level signal space we choose $W_{\mathrm{L}} = W_{\mathrm{L}1} \times W_{\mathrm{L}2}$, 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 cleaning and an "idle" mode, this gives a total of $3 + 2 + 7 = 12$ modes for each subsystem; they can be conveniently encoded as a discrete input $u^{\mathrm{D}j}$, $j = 1, 2$, with range

$$U_{\mathrm{D}j} = \{\mathrm{P1},\ \mathrm{P2},\ \mathrm{P3},\ \mathrm{Clean},\ \mathrm{Idle},\ \mathrm{F001},\ \mathrm{F010},\ \mathrm{F011},\ \ldots,\ \mathrm{F111}\}\,. \tag{21}$$

While in one of the reaction modes Pi, low-level dynamics is modelled by a sampled version of the ODEs (18), (19), (20). The parameters are as follows: $\beta_1 = 0.5$, $\alpha_1 = 2.0\mathrm{h}^{-0.5}$, $\beta_2 = 0.4$, $\alpha_2 = 2.0\mathrm{h}^{-0.6}$, $\beta_3 = 0.5$, $\alpha_3 = 3.0\mathrm{h}^{-0.5}$, $\delta_i = \gamma_i = 1$, $i = 1, 2, 3$; 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 $c_{P_1e} = 10\mathrm{kmol/m}^3$, $c_{P_2e} = 8\mathrm{kmol/m}^3$, $c_{P_3e} = 12\mathrm{kmol/m}^3$, and the bounds for the waste concentrations c_{W_1}, c_{W_2}, c_{W_3} are $2\mathrm{kmol/m}^3$, $1.5\mathrm{kmol/m}^3$, and $3\mathrm{kmol/m}^3$, respectively. The volumes of reactor R1 and R2 are $5\mathrm{m}^3$ and $2.5\mathrm{m}^3$, respectively. The (on/off) dosing signal q_j can take values in the set $\{0, 12\mathrm{kmol/h}\}$, and the control signal u_j is required to "live" within the interval $[0.01\mathrm{h}^{-1}, 3.0\mathrm{h}^{-1}]$, where the upper bound results from safety requirements. The signal $(q_j,\, u_j)$ is seen as an additional low-level input $u^{\mathrm{C}j}$ with range $U_{\mathrm{C}j} \subseteq \mathbb{R}^2$. We assume the continuous state is measured as a plant output $y^{\mathrm{C}j}$ with range $Y_{\mathrm{C}j} \subseteq \mathbb{R}^3$.

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 $c_{\mathrm{tf}} = 6\mathrm{h}$. 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 $u^{\mathrm{C}j}$ 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 $y^{\mathrm{D}j}$ which can take values in $\{\mathrm{Busy},\ \mathrm{Done}\}$. The signal space of subsystem $j \in \{1,\ 2\}$ is then given by the product

$$W_{\mathrm{L}j} = U_{\mathrm{L}j} \times Y_{\mathrm{L}j}\,, \quad U_{\mathrm{L}j} = U_{\mathrm{D}j} \times U_{\mathrm{C}j}\,, \quad Y_{\mathrm{L}j} = Y_{\mathrm{D}j} \times Y_{\mathrm{C}j}\,.$$

This is illustrated in Fig. 5, which summarises the overall control architecture. Note that in Fig. 5, the two subsystems are shown merged. 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.

5.2 Low-level specification and cost function

The low-level specification $\mathfrak{B}^{\mathrm{L}}_{\mathrm{spec}}$ for the multiproduct batch example includes the following requirements: (i) the mode of operation may only change imme-

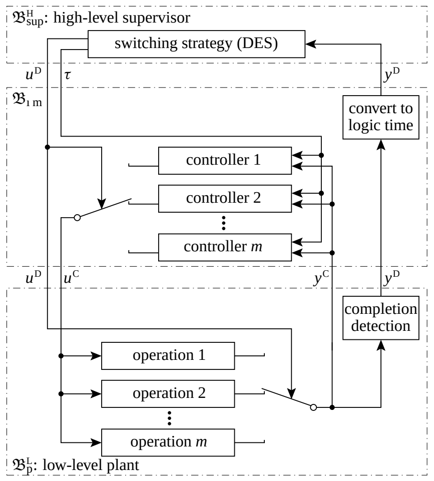


Fig. 5. Control architecture (subsystems merged).

diately after the previous operation has been completed; (ii) chemical reactions and filtering alternate in each subsystem; (iii) each filter can only be used by one subsystem at a time; (iv) the filters have to be cleaned when need arises (see Section 4); (v) the demanded product volumes are produced. Note that $\mathfrak{B}^{\mathrm{L}}_{\mathrm{spec}}$ contains only discrete requirements and therefore represents a typical discrete event specification, albeit in clock time. Formally, we therefore have $\mathfrak{B}^{\mathrm{L}}_{\mathrm{spec}} = \mathfrak{B}^{\mathrm{D}}_{\mathrm{spec}} \times (U_{\mathrm{C}} \times Y_{\mathrm{C}})^{\mathbb{N}_0}$ for some behaviour $\mathfrak{B}^{\mathrm{D}}_{\mathrm{spec}}$ over $U_{\mathrm{D}} \times Y_{\mathrm{D}}$, implying that the continuous low-level signals are not restricted[9] by $\mathfrak{B}^{\mathrm{L}}_{\mathrm{spec}}$.

In a finite realisation of $\mathfrak{B}^{\mathrm{D}}_{\mathrm{spec}}$, the state needs to track the following: current major mode of operation (5^2 possibilities: three reactions, filtering, or cleaning in both subsystems); current allocation of filters (2^3 possibilities: three filters which can be allocated to either subsystem); recent usage of filters (3^3 possibilities: three filters, each of which could have been used for either of three products); product volumes produced so far (6^3 possibilities: this number results from the additional assumption that only integer multiples of a minimum batch size are allowed and that the maximum demand for each of the three products is known. In our example, $2.5\mathrm{m}^3$ is the minimum batch size and $12.5\mathrm{m}^3$ the maximum demand.) This amounts to 1.16×10^6 states.

The integral cost function γ^{L} only refers to the $U_{\mathrm{C}j}$ components of the low-level signal. It includes energy cost (heating), material cost (feed rates) and an overhead cost depending on time spent. For a low-level signal w^{L}, let

$$\gamma^{\mathrm{L}}(w^{\mathrm{L}}) := \int_0^{T_{\mathrm{f}}} (u_1(t) + 0.05 q_1(t))\, dt + \int_0^{T_{\mathrm{f}}} (u_2(t) + 0.05 q_2(t))\, dt + \int_0^{T_{\mathrm{f}}} 0.15\, dt\,, \tag{22}$$

[9] Note that safety and quality restrictions are imposed indirectly – the former via the restricted range for $u^{\mathrm{C}j} = (q_j, u_j)$, the latter via the completion signals $y^{\mathrm{D}j}$, which are linked to the $y^{\mathrm{C}j}$ reaching their target regions.

where T_{f} denotes the time when all demanded products have been delivered.

Given low-level plant model, specification and cost function, one could try to solve the optimisation problem (10) as it stands. This amounts to a nonlinear mixed discrete-continuous dynamic program with 2 continuous input signals (u_1, u_2), discrete input signals ($u^{\mathrm{D}j}$, q_j, $j = 1, 2$) that altogether can take $(12 \times 2)^2 = 576$ values, 6 continuous state variables, and a discrete state set with 1.16×10^6 elements. Over an adequate time horizon (about 50h), we found this computationally intractable for off-the-shelf optimisation software. Instead we apply the hierarchical procedure outlined in Section 3.2.

5.3 Hierarchical design – low-level control

Recall that low-level control is based on the specification $\mathfrak{B}^{\mathrm{HL}}_{\mathrm{spec}}$ representing the intended relation between high-level and low-level signals. For example, we introduce high-level control symbols signifying the commands "run reaction i in subsystem j such that the batch is finished at minimal cost within time $\tau_j \in T = \{1h, 2h, \ldots 10h\}$". This is implemented by $3 \times 2 \times 10 = 60$ low-level controllers that can be selected by high-level control (see Fig. 5). Obviously, low-level controller design is local in the sense that it only refers to one reaction process and one subsystem at a time. The corresponding dynamic program has 1 binary input (q_j), 1 continuous input (u_j) and 3 continuous state variables (concentrations). It can be solved numerically by standard optimisation software. As an illustration, the minimal cost $\gamma_{1,i}(\tau_1)$ for reactor R1 to produce one batch of product Pi is given in Table 1. As low-level optimisation does not depend on the demanded overall amount of products, this design step only needs to be performed once over the life-cycle of the plant.

Table 1. Minimal cost for reactor R1 to produce one batch.

τ_1	< 5h	5h	6h	7h	8h	9h	10h
$\gamma_{1,1}$	∞	∞	3.70	3.42	3.28	3.21	3.17
$\gamma_{1,2}$	∞	2.81	2.58	2.46	2.40	2.36	2.32
$\gamma_{1,3}$	∞	∞	4.16	3.71	3.58	3.51	3.49

As indicated above, high-level control actions consist of modes from $U_{\mathrm{D}i}$ 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}. As high-level measurement, we choose the completion component from the low-level subsystems, i.e. $Y_{\mathrm{H}} = \{\text{Busy, Done1, Done2}\}$. While low-level signals "live" on clock time, high-level signals evolve on logic (event-driven) time, where events are triggered by changes in the $Y_{\mathrm{D}j}$-components.

5.4 Hierarchical design – high-level control

Connecting the intermediate layer (i.e., the low-level control and measurement aggregation mechanism described above) to the plant model, results in a hybrid system with external behaviour $\mathfrak{B}^{\mathrm{H}}_{\mathrm{im}}[\mathfrak{B}^{\mathrm{L}}_{\mathrm{p}}]$. To design a suitable high-level supervisor, we need a discrete abstraction of $\mathfrak{B}^{\mathrm{H}}_{\mathrm{im}}[\mathfrak{B}^{\mathrm{L}}_{\mathrm{p}}]$, a high-level "image" of the original specification $\mathfrak{B}^{\mathrm{L}}_{\mathrm{spec}}$, and a high-level cost function γ^{H}.

As pointed out in Section 3.2, we can derive a suitable abstraction $\tilde{\mathfrak{B}}^{\mathrm{H}}_{\mathrm{p}}$ directly from the intermediate layer specification $\mathfrak{B}^{\mathrm{HL}}_{\mathrm{spec}}$. The specified external behaviour of each subsystem under low-level control (w.r.t. the discrete variables $u^{\mathrm{D}j}$ and $y^{\mathrm{D}j}$) can be modelled as a timed discrete event system (TDES, see e.g. [3]), where time is still clock time. To obtain a DES realisation of an abstraction $\tilde{\mathfrak{B}}^{\mathrm{H}}_{\mathrm{p}}$ of $\mathfrak{B}^{\mathrm{H}}_{\mathrm{im}}[\mathfrak{B}^{\mathrm{L}}_{\mathrm{p}}]$, we form the synchronous product of the individual TDESs and remove tick events (which "count" the progress of clock time) by language projection. Note that in our design the first instance where a composition of subsystems needs to be computed occurs after the individual subsystems have undergone considerable simplification.

According to (8), the high-level-specification $\tilde{\mathfrak{B}}^{\mathrm{H}}_{\mathrm{spec}}$ can be directly obtained from $\mathfrak{B}^{\mathrm{D}}_{\mathrm{spec}}$ by transforming clock time to logic time. Together with the high-level abstraction $\tilde{\mathfrak{B}}^{\mathrm{H}}_{\mathrm{p}}$, we obtain a transition system with 17×10^6 states and an average of 13.1 relevant input events per state. Since every high-level input event corresponds to a low-level mode (either chemical reaction or filtering) that will be completed at a known cost, the high-level cost function γ^{H} is additive over high-level logic time (i.e. cost per transitions). Thus, the high-level optimisation problem (12) can be solved using standard methods from dynamic programming. On a decent desktop computer, synthesis of the high-level supervisor takes 61 minutes. Fig. 6 shows the obtained closed-loop operation to produce $12.5m^3$, $12.5m^3$ and $7.5m^3$ of the products P1, P2 and P3, respectively. The overall cost amounts to 27.5.

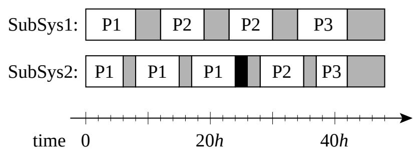


Fig. 6. Optimal schedule (filter processes grey, cleaning black).

6 Conclusions

We discussed a hierarchical extension of our behavioural approach to hybrid control synthesis. In particular, we provided conditions for intermediate layers ensuring that all control layers interact in the desired sense. These conditions imply that if each layer enforces its specification for an appropriate plant model or abstraction, the resulting overall controller is guaranteed to enforce the overall specification for the underlying plant model. We also discussed how to add optimal performance objectives to hard specifications. We demonstrated the potential of our approach through a multiproduct batch example, which we found intractable using off-the-shelf optimisation software.

Acknowledgement. We dedicate this paper to Michael Zeitz on the occasion of his 65th birthday. He has always been a role model for us in both research and teaching. We thank A. Kienle for helpful discussions on the multiproduct batch example, and B.V. Mishra and D. Gromov for their help in solving the low-level optimisation problems in Section 5.3. The first author gratefully acknowledges funding by Deutsche Forschungsgemeinschaft (DFG) in the context of Research Unit (Forschergruppe) 468. Part of this work has also been done in the framework of the HYCON Network of Excellence, contract number FP6-IST-511368.

References

1. P. Antsaklis, W. Kohn, M. Lemmon, A. Nerode, and S. Sastry, editors. *Hybrid Systems V*, LNCS 1567. Springer-Verlag, 1999.
2. P. J. Antsaklis and A. Nerode, editors. *IEEE Transactions on Automatic Control, Special Issue on Hybrid Systems*, volume 43. 1998.
3. B.A. Brandin and W.M. Wonham. Supervisory control of timed discrete-event systems. *IEEE Transactions on Automatic Control*, 39:329–342, 1994.
4. P.E. Caines and Y.J. Wei. Hierarchical hybrid control systems: a lattice theoretic formulation. *IEEE Transactions on Automatic Control*, 43:4:501–508, 1998.
5. J.E. Cury, B. Krogh, and T. Niinomi. Synthesis of supervisory controllers for hybrid systems based on approximating automata. *IEEE Trans. Autom. Contr.*, 43:564–568, 1998.
6. S. Engell, G. Frehse, and E. Schnieder, editors. *Modelling, Analysis, and Design of Hybrid Systems.* LNCIS 279. Springer-Verlag, 2002.
7. X. Koutsoukos, P.J. Antsaklis, J.A. Stiver, and M.D. Lemmon. Supervisory control of hybrid systems. *Proceedings of the IEEE*, 88:1026–1049, July 2000.
8. R.J. Leduc, B.A. Brandin, W.M. Wonham, and M. Lawford. Hierarchical interface-based supervisory control. In *Proc. 40th CDC*, pp. 4116–4121, 2001.
9. J. Lunze, B. Nixdorf, and H. Richter. Hybrid modelling of continuous-variable systems with application to supervisory control. In *Proc. ECC*, 1997.
10. T. Moor and J. Raisch. Supervisory control of hybrid systems within a behavioural framework. *Systems and Control Letters*, 38:157–166, 1999.
11. T. Moor and J. Raisch. Think continuous, act discrete: DES techniques for continuous systems. *Proc. 10th Mediterranean Conf. Contr. and Autom.*, 2002.
12. T. Moor and J. Raisch. Hierarchical hybrid control of a multiproduct batch plant. In *Proc. 16th IFAC World Congress*, Prague, Czech Republic, 2005.
13. T. Moor, J. Raisch, and J.M. Davoren. Admissibility criteria for a hierarchical design of hybrid control systems. In *Proc. IFAC Conference on the Analysis and Design of Hybrid Systems (ADHS'03)*, pages 389–394, 2003.
14. T. Moor, J. Raisch, and S.D. O'Young. Discrete supervisory control of hybrid systems based on l-complete approximations. *Discrete Event Dynamic Systems*, 12:83–107, 2002.
15. G.J. Pappas, G. Lafferriere, and S. Sastry. Hierarchically consistent control systems. *IEEE Transactions on Automatic Control*, 45:6:1144–1160, 2000.
16. J. Raisch and S.D. O'Young. Discrete approximation and supervisory control of continuous systems. *IEEE Trans. Autom. Contr.*, 43:569–573, 1998.
17. P.J. Ramadge and W.M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77:81–98, 1989.
18. J.C. Willems. Models for dynamics. *Dynamics Reported*, 2:172–269, 1989.
19. K.C. Wong and W.M. Wonham. Hierarchical control of discrete-event systems. *Discrete Event Dynamic Systems*, 6:241–306, 1996.

Index