A TWO-LEVEL HYBRID CONTROL STRATEGY FOR THE START-UP OF A COUPLED DISTILLATION PLANT

A. Itigin[1], J. Raisch[2,3], T. Moor[4], A. Kienle[3,5]

[1] Institut für Systemdynamik und Regelungstechnik, Universität Stuttgart,
Pfaffenwaldring 9, D-70550 Stuttgart, FR Germany, email: itigin@isr.uni-stuttgart.de
[2] Lehrstuhl für Systemtheorie technischer Prozesse, Otto-von-Guerecke Universität Magdeburg,
Postfach 4120, D-39016 Magdeburg, FR Germany, email: raisch@mpi-magdeburg.mpg.de
[3] Max-Planck-Institut für Dynamik komplexer technischer Systeme,
Sandtorstrasse 1, D-39106 Magdeburg, FR Germany
[4] Research School of Information Science and Engineering, Australian National University,
Canberra ACT 0200, Australia, email: thomas.moor@anu.edu.au
[5] Lehrstuhl für Automatisierungstechnik/Modellbildung, Otto-von-Guerecke Universität Magdeburg,
Postfach 4120, D-39016 Magdeburg, FR Germany, email: kienle@mpi-magdeburg.mpg.de

Keywords: Hierarchical control, hybrid systems, behavioural systems theory, nonlinear dynamical systems, distillation columns

Abstract

In this contribution, we suggest an efficient hierarchical hybrid control scheme for the start-up of a distillation plant consisting of two coupled distillation columns. The overall control task is decomposed into a set of low-level and high-level subtasks. Each of these subtasks can be solved with much less effort than the original specifications. Low-level tasks are solved by means of continuous feedback, whereas the high-level task is solved by a discrete supervisory controller operating on quantised measurement information and switching between continuous low-level controllers. The approach presented is set within a behavioural framework; it guarantees that the chosen interaction of low-level and high-level control does indeed enforce the specifications.

1 Introduction

The conceptual and computational effort for solving complex control problems can often be considerably reduced by imposing an appropriate structure. For example, hierarchical control is an attempt to handle complex problems by decomposing them into smaller subproblems and reassembling their solutions in a hierarchical structure. Not surprisingly, it has been a popular topic within both academia and industry. In practice, heuristic approaches have been preferred. While they usually succeed in "breaking" the control task into problems of feasible dimension, they cannot guarantee that the overall solution does indeed meet the specifications. Formal approaches, on the other hand, have mostly been restricted to a small class of problems; typical assumptions are linear time-invariant plant models and quadratic cost functions, see e.g. [9]. There are a number of exceptions though, most notably [11] (dealing with purely discrete systems) and [1, 4]. The latter study abstractions that are equivalent with respect to a given problem, i.e. the problem can be solved on the abstraction level if and only if the underlying problem is solvable.

In this contribution, we present an efficient hierarchical two-level hybrid control strategy for the start-up procedure of the methanol-ethanol-propanol separation process in a distillation plant consisting of two coupled columns. It is based on the conceptual framework in [5]. There, we addressed the start-up problem for the main (stand-alone) column of the distillation plant; it was solved using a purely discrete two-level controller. The latter was synthesised on the basis of two discrete abstractions of the continuous process model [6, 2], each with different granularity. A straightforward transfer of this approach to the control problem at hand is not possible, though, as the state dimension of the continuous plant model is considerably higher. Following [3], continuous low-level controllers are added within the controller hierarchy. They implement "elementary manoeuvres" in the continuous plant state space, which substantially facilitates synthesising an abstraction-based discrete controller (supervisor) at the top level. By switching between continuous low-level controllers at appropriate times, the discrete supervisor ensures that the overall specifications are met.

This contribution is organised as follows: Section 2 motivates the use of J.C. WILLEMS' behavioural systems theory, e.g. [10], as a framework for our approach. Section 3 addresses the problem of hierarchical control within this framework. In Section 4, which constitutes the main part of this contribution, our approach is applied to synthesise a two-level control scheme for the automatic start-up of a coupled distillation plant.

2 Behavioural systems theory

J.C. WILLEMS' behavioural systems theory [10], provides a highly intuitive framework for investigating hybrid phenomena. In particular, the interaction between dynamical systems, e.g. the interaction between controller and a plant model, can be formalised in a natural and familiar way in terms of set intersection.

In J.C. WILLEMS' behavioural framework, a dynamical system is understood as an object interacting with its environment via external signals. It is formally represented as a triple $(T, W, \mathcal{B})$, where T is the time "axis" and W denotes the external signal "space". Let $W^T := \{w | w : T \to W\}$ represent the set of all functions mapping T into W or, in other words, the set of all signals evolving on the time "axis" T within the set W. Then, the *behaviour* $\mathcal{B} \subseteq W^T$ is defined as the set of external signals which are compatible with the internal dynamics of the system.

Of course, when performing actual calculations, a finite-dimensional representation of $(T, W, \mathcal{B})$ is needed. Behaviours are, however, an extremely intuitive way of "thinking" about systems and their interaction. This is illustrated by a standard feedback configuration: consider a system ("plant model") with input $u(\tau) \in U$, (measurable) output $y(\tau) \in Y$, $\tau \in T$, and behaviour $\mathcal{B}_p \subseteq (U \times Y)^T$. It is to be controlled by feeding back y to u via a second system ("the controller") with behaviour $\mathcal{B}_c$. Then, the closed loop behaviour is given by $\mathcal{B}_{cl} = \mathcal{B}_p \cap \mathcal{B}_c$ — only signal pairs (u, y) that are compatible with the dynamics of both plant model and controller "survive" closing the loop. In the simplest case, closed loop specifications can be formulated as a "legal" set $\mathcal{B}_s \subset (U \times Y)^T$ of signal pairs. The control task is then to "enforce"

$$\emptyset \neq \mathcal{B}_{cl} \subseteq \mathcal{B}_s \tag{1}$$

by finding and realizing a suitable controller $C = (T, U \times Y, \mathcal{B}_c)$.

Now, suppose that controller synthesis for a system $P = (T, U \times Y, \mathcal{B}_p)$ is inconvenient (because, for example, realizations of P are tricky to handle). Hence, we want to perform the synthesis step on the basis of an approximation, or abstraction, $P_a = (T, U \times Y, \mathcal{B}_a)$. Clearly, we want

$$\emptyset \neq \mathcal{B}_a \supseteq \mathcal{B}_p. \tag{2}$$

to hold. If this condition were violated, P could respond to a given input signal with an unacceptable measurement signal which would not be predictable by the abstracted model P_a. Hence, this unacceptable phenomenon could not be suppressed by a control strategy based on P_a – the abstraction would be useless as far as control synthesis is concerned. The "abstraction condition" (2) implies $\mathcal{B}_a \cap \mathcal{B}_c \subseteq \mathcal{B}_s \Rightarrow \mathcal{B}_p \cap \mathcal{B}_c \subseteq \mathcal{B}_s$. One also needs to ensure that P and C are nonblocking. Assume this can be done (and in many scenarios this is straightforward or even trivial) – then, a controller which enforces the specification for the abstracted model P_a will also make the underlying model P obey the specification.

3 Synthesis of a hierarchical control scheme

The starting point of our investigations is a continuous plant model $P = (T, U \times Y, \mathcal{B}_p)$ and a specification $\mathcal{B}_s \subseteq (U \times Y)^T$. In our particular control problem, the start-up of a coupled distillation system, a first principles model is so complex that finding a suitably accurate discrete abstraction involves too much computational effort. Instead of a "monolithic" abstraction based controller, we suggest a two-level hybrid hierarchical strategy. On the lower level, we implement a number of suitable continuous feedback loops realising certain "elementary manoeuvres" in the continuous plant state space. If these manoeuvres are chosen appropriately, the approximation of the resulting closed loop configuration by a discrete model requires much less effort. Then, on the basis of this discrete model, a high-level discrete supervisory controller is synthesised. This discrete controller enforces the overall specification by switching between low-level controllers at appropriate points of time.

Suppose we want to implement m "elementary manoeuvres" with desired behaviours $\mathcal{B}_s^1, \ldots, \mathcal{B}_s^m \subseteq (U \times Y)^T$. These can be interpreted as low-level subtasks. They should meet two main requirements: (i) each of them should be achievable with much less effort than the original task, and (ii) certain suitable temporal arrangements of "elementary manoeuvres" should provide a solution for the overall control problem. In order to formalise the second requirement, we need some more notation. Let $w_1 := (u_1, y_1)$, $w_2 := (u_2, y_2)$, $w_1, w_2 \in (U \times Y)^T$, denote two external signals. Then $w_1 \underset{\chi}{\wedge} w_2$ denotes *concatenation* of w_1 and w_2 at time χ, [10]. It is defined as

$$(w_1 \underset{\chi}{\wedge} w_2)(\tau) := \begin{cases} w_1(\tau) & \text{for } \tau < \chi, \\ w_2(\tau) & \text{for } \tau \geq \chi. \end{cases} \tag{3}$$

By writing $\mathcal{B}_1 \underset{\chi}{\wedge} \mathcal{B}_2$, this can be extended to behaviours $\mathcal{B}_1, \mathcal{B}_2 \subseteq (U \times Y)^T$ in an obvious way. Then, we introduce *y-continuous concatenation* $\mathcal{B}_1 \underset{\chi}{\bar{\wedge}} \mathcal{B}_2 := \{(u, y) | (u, y) \in \mathcal{B}_1 \underset{\chi}{\wedge} \mathcal{B}_2, \lim_{\tau \to \chi - 0} y(\tau) = \lim_{\tau \to \chi + 0} y(\tau)\}$ consisting of all signal pairs $(u, y) \in \mathcal{B}_1 \underset{\chi}{\wedge} \mathcal{B}_2$ with y continuous at time χ. Requirement (ii) can now be formulated as follows: for any initial condition, there is at least one set of indices $i_1, i_2, \ldots \in K := \{1, \ldots, m\}$ and a corresponding set of switching times $\chi_1 < \chi_2 < \ldots$ such that

$$\mathcal{B}_s^{i_1} \underset{\chi_1}{\bar{\wedge}} \mathcal{B}_s^{i_2} \underset{\chi_2}{\bar{\wedge}} \ldots \subseteq \mathcal{B}_s. \tag{4}$$

Suppose the low-level subtasks can be solved by conventional continuous controllers $C_j = (T, U \times Y, \mathcal{B}_c^j)$, i.e.

$$\mathcal{B}_c^j \cap \mathcal{B}_p \subseteq \mathcal{B}_s^j \tag{5}$$

holds for $j = 1, \ldots, m$. Then, we can assemble the overall low-level controller $\tilde{C} = (T, U \times Y \times K, \tilde{\mathcal{B}}_C)$ from the controllers $C_j, j = 1, \ldots, m$. A particular low-level controller C_j is activated when the discrete signal $k(\tau) = j$ is applied to $\tilde{C}$. Hence,

$$\mathcal{B}_c^j = \{(u, y) | (u, y, k) \in \tilde{\mathcal{B}}_c, k(\tau) = j, \forall \tau \in T\}. \tag{6}$$

The overall low-level specification behaviour $\tilde{\mathcal{B}}_s \subseteq (U \times Y)^T$ is given by

$$\tilde{\mathcal{B}}_s = \bigcup_{\substack{\forall i_j \in K, \\ \forall \chi_j < \chi_{j+1}, \\ \forall j \in \mathbb{N}}} \mathcal{B}_s^{i_1} \underset{\chi_1}{\bar{\wedge}} \mathcal{B}_s^{i_2} \underset{\chi_2}{\bar{\wedge}} \ldots \tag{7}$$

It contains all signals which meet the requirements of different low-level subtasks in a piecewise manner.

Let us introduce a projection mapping $s : (U \times Y \times K)^T \to (U \times Y)^T$ and its inverse s^{-1}. They are defined as $s((u,y,k)) := (u,y)$ and $s^{-1}((u,y)) := \{(u,y,k)|s((u,y,k)) = (u,y)\}$. These mappings can be naturally extended to behaviours. Obviously, the overall low-level controller enforces the low-level specification $\tilde{\mathcal{B}}_s$ as (5) implies

$$s(\tilde{\mathcal{B}}_c) \cap \mathcal{B}_p \subseteq \tilde{\mathcal{B}}_s. \tag{8}$$

The resulting low-level closed loop configuration has input k and output y (see Fig. 1); its behaviour $\tilde{\mathcal{B}}_{cl}$ is given by

$$\tilde{\mathcal{B}}_{cl} := p(\tilde{\mathcal{B}}_c \cap s^{-1}(\mathcal{B}_p)), \tag{9}$$

where $p : (U \times Y \times K)^T \to (Y \times K)^T$ is another projection operator defined by $p((u,y,k)) := (y,k)$.

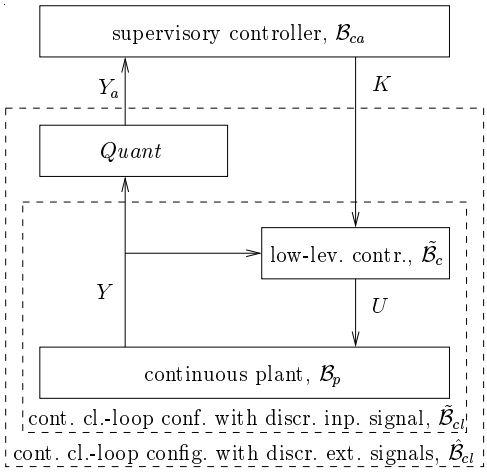

Figure 1: Two-level hierarchical control architecture.

The top level supervisor to be synthesised "sees" a quantised version y_a of the plant output y. The corresponding (finite) signal set is denoted Y_a. Moreover, the supervisor will only accept inputs at instants of time when the quantised measurement changes, and it is only allowed to change its output at these instants of time. Hence, its behaviour "lives" in $(Y_a \times K)^{\mathbb{N}_0}$. The behaviour of the low-level closed loop configuration on this domain is given by

$$\hat{\mathcal{B}}_{cl} := quant(\tilde{\mathcal{B}}_{cl}), \tag{10}$$

where the map $quant : (Y \times K)^T \to (Y_a \times K)^{\mathbb{N}_0}$ is defined by

$$quant((y,k)) = (y_a, k'), \tag{11}$$

where

$$y_a(i) = Quant(y(\tau_i)), \tag{12}$$
$$k'(i) = k(\tau_i), \tag{13}$$

and τ_i are the time instants, where the signal y has entered into another quantisation cell, i.e. $Quant(y(\tau_i)) \neq Quant(y(\tau_i - \varepsilon))$. Finally, the map $Quant : Y \to Y_a$ partitions Y into finitely many quantisation cell and may depend on the current value of k.

Although $\hat{\mathcal{B}}_{cl}$ evolves in discrete time on a discrete external signal set, its state space is not discrete – clearly, it involves the continuous plant state space and the state space of the low-level controllers. The last step is therefore to come up with a discrete abstraction of $\hat{\mathcal{B}}_{cl}$, denoted by $\mathcal{B}_{cla}$. It is realised by a finite state machine and has to satisfy $\mathcal{B}_{cla} \supseteq \hat{\mathcal{B}}_{cl}$.

The top level subtask $\mathcal{B}_{sa} \subseteq (Y_a \times K)^{\mathbb{N}_0}$ is concerned with finding a correct sequential arrangement of low-level subtasks. We require

$$s(p^{-1}(quant^{-1}(\mathcal{B}_{sa}))) \cap \tilde{\mathcal{B}}_s \subseteq \mathcal{B}_s \tag{14}$$

which amounts to "tightening" the original specification. In other words, the combination of high- and low-level subtasks is more restrictive than the original overall task. However, this is clearly a price we expect to pay for simplifying the original problem.

We now seek a discrete supervisory controller $C_a = (\mathbb{N}_0, Y_a \times K, \mathcal{B}_{ca})$ such that

$$\mathcal{B}_{ca} \cap \mathcal{B}_{cla} \subseteq \mathcal{B}_{sa}. \tag{15}$$

If such a controller exists, the behaviour of the resulting two-level hybrid controller is given by

$$\mathcal{B}_c = s(p^{-1}(quant^{-1}(\mathcal{B}_{ca}) \cap \tilde{\mathcal{B}}_c). \tag{16}$$

It enforces the overall specification for the plant model as

$$\begin{aligned}
\mathcal{B}_c \cap \mathcal{B}_p &= s(p^{-1}(quant^{-1}(\mathcal{B}_{ca})) \cap \tilde{\mathcal{B}}_c) \cap \mathcal{B}_p \\
&= s(p^{-1}(quant^{-1}(\mathcal{B}_{ca})) \cap \tilde{\mathcal{B}}_c \cap \tilde{\mathcal{B}}_c) \cap \mathcal{B}_p \\
&= s(p^{-1}(quant^{-1}(\mathcal{B}_{ca})) \cap \tilde{\mathcal{B}}_c \cap \tilde{\mathcal{B}}_c \cap s^{-1}(\mathcal{B}_p)) \cap \mathcal{B}_p \\
&\subseteq s(p^{-1}(quant^{-1}(\mathcal{B}_{ca})) \cap p^{-1}(\tilde{\mathcal{B}}_{cl})) \cap s(\tilde{\mathcal{B}}_c) \cap \mathcal{B}_p \\
&\subseteq s(p^{-1}(quant^{-1}(\mathcal{B}_{ca}) \cap \tilde{\mathcal{B}}_{cl})) \cap \tilde{\mathcal{B}}_s \\
&\subseteq s(p^{-1}(quant^{-1}(\mathcal{B}_{ca}) \cap quant^{-1}(\hat{\mathcal{B}}_{cl}))) \cap \tilde{\mathcal{B}}_s \\
&= s(p^{-1}(quant^{-1}(\mathcal{B}_{ca} \cap \hat{\mathcal{B}}_{cl}))) \cap \tilde{\mathcal{B}}_s \\
&\subseteq s(p^{-1}(quant^{-1}(\mathcal{B}_{ca} \cap \mathcal{B}_{cla}))) \cap \tilde{\mathcal{B}}_s \\
&\subseteq s(p^{-1}(quant^{-1}(\mathcal{B}_{sa}))) \cap \tilde{\mathcal{B}}_s \subseteq \mathcal{B}_s.
\end{aligned} \tag{17}$$

Hence, if blocking can be ruled out (i.e. if the plant model P, low-level controllers C_j, $j = 1, \ldots, m$, and top level controller C_a can always "agree" on at least one common signal pair (u,y,k)), the overall control problem is solved.

4 Start-Up of a Distillation Plant

In this section, we consider synthesising a control strategy for the automatic start-up of a distillation plant, see Fig. 2. The plant consists of two coupled distillation columns serving for the separation of methanol (MeOH), ethanol (EOH) and propanol (POH). The columns are operated at the University of Stuttgart in a pilot plant scale. The main column

is about 10m high and consists of 40 bubble cap trays (numbered by $z = 2, \ldots, 41$ from bottom to top), a reboiler ($z = 1$) and a condenser ($z = 42$). The side column is smaller, it has a height of about 2m and consists of 10 bubble cap trays ($z = 1, \ldots, 10$) and a condenser ($z = 11$). The side column is coupled to the main column by vapour side draw and therefore has no reboiler.

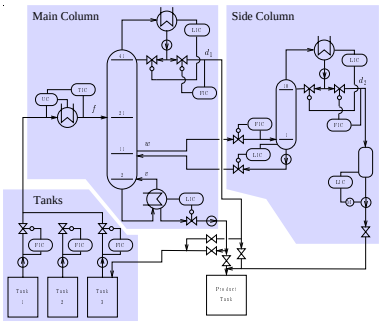

Figure 2: Distillation plant.

The following steps can be distinguished during conventional start-up of the distillation plant: in the first step, thermal and hydrodynamic start-up is carried out. During this step, the columns are operated at total reflux and reboil without feed. In the second step, the reboil, distillate and feed flow rates in both columns are adjusted to their steady state operating point values. At the end of this step, formation of characteristic concentration and temperature fronts in both columns is completed. Now, in the third step, the concentration fronts move very slowly towards their steady state positions. This is illustrated by the simulation results shown in Fig. 3. These results have been obtained with the distillation plant model in DAE form to be described subsequently. Start-up is considered to be finished once the plant state is close enough to the desired steady state operating point, where almost pure products can be withdrawn: methanol from the top of the main column, propanol from the bottom of the main column and ethanol from the top of the side column. While the first and the second step of start-up are usually quite fast, the third step is extremely slow, cf. Fig. 3. This is especially annoying in a research environment, where one usually wants to perform quite a few experiments within a limited time, and start-up has therefore to be repeated fairly frequently. Hence, we try to speed up the third step of the start-up procedure by applying a suitable control strategy.

We use the following constant molar overflow model based on material balances only:

main column, material balances:

$$n_z^{L,I}\dot{\xi}_{i,z}^I = f_{z+1}^{L,I}\xi_{i,z+1}^I - f_z^{L,I}\xi_{i,z}^I + f_{z-1}^{V,I}\eta_{i,z-1}^I - f_z^{V,I}\eta_{i,z}^I + \begin{cases} f\xi_{i,F} & \text{if } z = 21 \text{ (feed)}, \\ (w-d_2)\xi_{i,1}^{II} & \text{if } z = 11 \text{ (liquid from side column)}, \\ -w\eta_{i,11}^I & \text{if } z = 11 \text{ (vapour side draw)}, \\ 0 & \text{else}, \end{cases} \quad (18)$$

side column, material balances:

$$n_z^{L,II}\dot{\xi}_{i,z}^{II} = f_{z+1}^{L,II}\xi_{i,z+1}^{II} - f_z^{L,II}\xi_{i,z}^{II} + f_{z-1}^{V,II}\eta_{i,z-1}^{II} - f_z^{V,II}\eta_{i,z}^{II} + \begin{cases} w\eta_{i,11}^I & \text{if } z = 1 \text{ (vapour from side column)}, \\ 0 & \text{else}, \end{cases} \quad (19)$$

summation conditions:

$$\sum_{i=1}^{3}\eta_{i,z}^{I,II} = \sum_{i=1}^{3}\xi_{i,z}^{I,II} = 1, \quad (20)$$

vapour–liquid equilibrium:

$$\eta_{i,z}^{I,II} = \frac{p_i^V(t_z^{I,II})}{p}\xi_{i,z}^{I,II}, \quad (21)$$

Antoine-equation:

$$p_i^V = 10^{\left(A_i - \frac{B_i}{t_z^{I,II} - 273.15 + C_i}\right)}. \quad (22)$$

Upper indices I, II denote variables and parameters belonging to the main column (I) and side column (II). The index z denotes the tray number, $z = 1, \ldots, 42$, for the main column and $z = 1, \ldots, 11$, for the side column. Products (methanol, ethanol, propanol) are indexed by $i = 1, 2, 3$. The product mole fractions in the liquid and in the vapour are denoted by ξ and η. f^L denotes the liquid molar flow rate, f^V the vapour flow rate and n_L the molar liquid holdup. Values of f^L, f^V and n^L for each tray are given in Table 1. f denotes the feed, v the reboil, w the vapour side draw and d_1 the distillate flow rate in the main column, d_2 the distillate flow rate in the side column, cf. Fig. 2. The total pressure is denoted by p, the vapour pressure by p^V. The former is constant, $p = 760$ torr, and the latter is calculated by the *Antoine-equation* (22), [8], with parameters $A_{1\div3} = (8.08097\ 8.11220\ 7.74416)$, $B_{1\div3} = (1582.271\ 1592.864\ 1437.686)$, $C_{1\div3} = (239.726\ 226.184\ 198.463)$. Temperature is denoted by t. The temporal evolution of t on each tray can be calculated from the algebraic equation (20).

		z	f_{z+1}^L	f_z^L	f_{z-1}^V	f_z^V	n_z^L [mol]
main col.	condenser	42	0	$v-w$	$v-w$	0	4.17
	feed tray	22-41	$v-w-d_1$	$v-w-d_1$	$v-w$	$v-w$	1.0
		21	$v-w-d_1$	$v-w-d_1+f$	$v-w$	$v-w$	1.0
	side draw	12-20	$v-w-d_1+f$	$v-w-d_1+f$	$v-w$	$v-w$	1.0
		11	$v-w-d_1+f$	$v-d_1-d_2+f$	v	$v-w$	1.0
	reboiler	2-10	$v-d_1-d_2+f$	$v-d_1-d_2+f$	v	v	1.0
		1	$v-d_1-d_2+f$	$-d_1-d_2+f$	0	v	114.0
side col.	condenser	11	0	w	w	0	3.08
	feed tray	2-10	$w-d_2$	$w-d_2$	w	w	0.2
		1	$w-d_2$	$w-d_2$	0	w	0.2

Table 1: Flow rates and liquid holdups.

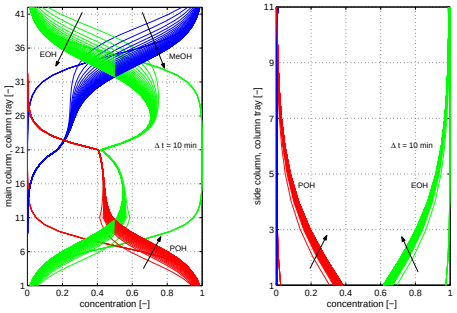

Figure 3: Third step of start-up, open loop.

Although this model is fairly complex from a control synthesis point of view, a chemical engineer might still consider it to be too simple to adequately describe a distillation column during start-up. However, because we will concentrate our efforts on the third step of start-up, where thermal start-up has already been finished and boiling point conditions have been

established at each tray of both columns, the assumptions stated above are justified.

All major external flows of the plant could be used as control inputs. For our application example, we use two of them: $u := (d_1, v)$, $d_1(\tau) \in D_1 := [0, 0.5\text{kmol/h}]$ and $v(\tau) \in V := [0, 1\text{kmol/h}]$. The measurement variables are given by two temperatures, one in the stripping section, on tray $z = 5$, and one in the rectifying section of the main column, on tray $z = 36$: $y := (t_5, t_{36})$. For the third step of start-up $t_5(\tau) \in T_5 := [t_{5s} - 5K, t_{5s} + 5K]$, $t_{36}(\tau) \in T_{36} := [t_{36s} - 5K, t_{36s} + 5K]$, where $t_{5s} := 366.9\text{K}$ and $t_{36s} = 340.0\text{K}$ denote the operating point steady state values of t_5 and t_{36}. Time is continuous: $\tau \in T := \mathbb{R}^+$. Then the "behavioural" formulation of the continuous low-level plant model is given by $P = (\mathbb{R}^+, (D_1, V) \times (T_5, T_{36}), \mathcal{B}_p)$.

Our experience with start-up of the distillation plant [5], shows that for the third step of start-up the measurements t_5 and t_{36} indicate reliably whether the plant is "close enough" to the desired operating point: this is achieved if (t_5, t_{36}) reaches the target region $[t_{5s} - 0.1\text{K}, t_{5s} + 0.1\text{K}] \times [t_{36s} - 0.1\text{K}, t_{36s} + 0.1\text{K}]$. The specification is to accomplish this within two hours, when starting with arbitrary measurements within $T_5 \times T_{36}$.

In our previous work [5], where we considered the start-up of the stand-alone main column, our approach was based on finding a suitable discrete abstraction of the plant model in form of a finite nondeterministic automaton whose state variable memorises the last l values of the discrete control input and measurement signals (l-complete approximation), [6, 2]. The automaton provides a conservative approximation of the plant, i.e. its behaviour is a superset of the discretised plant behaviour. On the basis of the abstraction, we then synthesised a discrete supervisory controller that enforces the specification for the abstraction, and by implication, also for the plant model. The controller synthesis procedure is a slight modification of RAMADGE and WONHAM's method [7]. While the controller synthesis procedure on its own is rather straightforward, the construction of a suitable discrete abstraction can be very demanding. The computational effort grows exponentially with the dimension of the state space of the continuous plant model. The state vector of the DAE model (18)-(22) has dimension 212. Hence, the computational effort needed to determine a sufficiently accurate abstraction would be enormously high. Therefore, there is hardly a chance to solve the control task for the coupled plant in the same way as for the stand-alone main column.

Instead, as indicated in Section 3, we use an alternative approach. We first "simplify" the plant dynamics by implementing a number of suitable low-level continuous feedback loops ("elementary manoeuvres"). Then, we approximate the resulting closed loop configuration by an automaton model. Finally, on the basis of this model, we synthesise a discrete supervisory controller that enforces the overall specification by switching between different continuous controllers. If the "elementary manoeuvres" are chosen appropriately, a sufficient level of approximation accuracy for the automaton model can be achieved with much less effort. For example, for a rectangular partition of the two-dimensional measurement space, cf. Fig. 4, four "elementary manoeuvres" corresponding to "up", "down", "right" and "left" movements reduce nondeterminism in the corresponding simple (1-complete) approximation.

Therefore, we formulate four corresponding low-level subtasks with the following specifications: subtasks 1: "increase t_5 with constant rate, keep t_{36} constant", subtask 2: "keep t_5 constant, increase t_{36} with constant rate", subtask 3: "decrease t_5 with constant rate, keep t_{36} constant", subtask 4: "keep t_5 constant, decrease t_{36} with constant rate". The rates are chosen such that the target region can be reached within two hours for every initial condition. Notice, that the specifications $\mathcal{B}_s^1, \ldots, \mathcal{B}_s^4$ of these four low-level subtasks together with the specification $\mathcal{B}_s$ of the original overall task satisfy requirement (4). For our application example, we were able to solve all four low-level subtasks by the same conventional MIMO PI controller C feeding back the measurements t_5 and t_{36} to the control inputs d_1 and v and tracking the corresponding reference signal for each subtask. In Laplace domain, C is given by

$$\begin{bmatrix} d_1(s) \\ v(s) \end{bmatrix} = 0.05 \left(1 + \frac{1}{s}\right) \begin{bmatrix} 1 & 1 \\ -10 & 10 \end{bmatrix} \begin{bmatrix} t_5(s) \\ t_{36}(s) \end{bmatrix}. \quad (23)$$

The purpose of the top level subtask is to ensure that the original overall specification is satisfied by generating an appropriate sequence of requests for "elementary manoeuvres". With "elementary manoeuvres" restricting the plant dynamics in the t_5, t_{36} output space to vertical and horizontal movements, an appropriate discrete abstraction (with behaviour $\mathcal{B}_{cla}$) can be found very easily. The map $Quant : Y \to Y_a$ partitioning the continuous set Y is shown in Fig. 4. Y_a contains five elements – hence, there are five partition cells. To avoid numerical problems with "elementary manoeuvres" on the boundaries of the partition cells, their sizes can be adjusted after switching to another "elementary manoeuvre" (recall that $Quant$ may depend on the current value of k). A suitable abstraction is a Moore-Automaton with input $k(i) \in \{1, 2, 3, 4\}$, corresponding to the four "elementary manoeuvres", output $y_a(i) \in Y_a$ and state set Y_a. It generates an obvious behaviour.

Writing down the top level specification $\mathcal{B}_{sa}$ is straightforward. Finding a suitable high-level controller is practically trivial, but can of course be formally accomplished by using RAMADGE and WONHAM's framework. Depending on the current value of y_a, the high-level controller activates a certain continuous low-level controller, which implements the corresponding "elementary manoeuvre". When the value of y_a changes, the high-level supervisor may also switch "elementary manoeuvres" until finally $y_a(i) = y_a^{(5)}$ occurs, signifying that the plant state has reached its target area. Then, start-up is considered to be complete and control over the plant is handed over to the operating point control unit, which keeps the plant close to the operating point steady state. The control action of the high-level supervisor is summarised in Table 2.

After the top level and the low-level subtasks have been solved, we also can guarantee that the original overall control problem is solved. Fig. 5 shows the results of a closed loop simula-

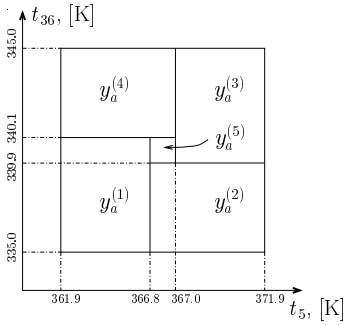

Figure 4: Quantisation of the measurement signal.

measured output	$y_a^{(1)}$	$y_a^{(2)}$	$y_a^{(3)}$	$y_a^{(4)}$	$y_a^{(5)}$
control input	1	2	3	4	other control unit

Table 2: Discrete control on top level.

tion, where the resulting two-level control strategy was applied. The concentration profiles in both columns indeed reach their operating point steady state positions within two hours. The temporal evolution of the continuous measurement variables t_5 and t_{36} as well as the concentration of propanol on tray $z = 5$ and of the concentration of methanol on tray $z = 36$ is shown in Fig. 6. After two successive "elementary manoeuvres" the measurement variables reach their operating point steady state values t_{5s} and t_{36s}. Furthermore, comparing the temperature and concentration plots in Fig. 6, one can see that the temperature control does indeed imply the desired effect on product concentrations.

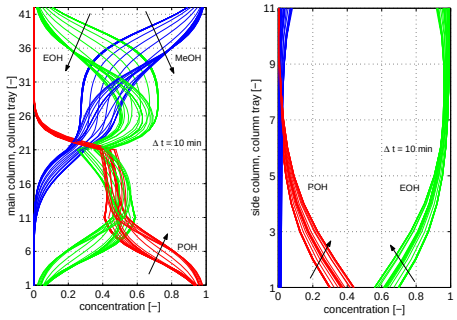

Figure 5: Third step of start-up, closed loop.

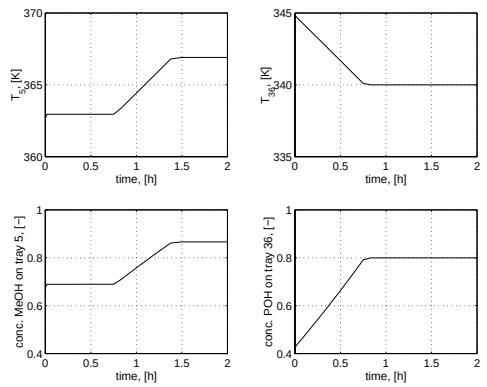

Figure 6: Measurement variables and corresponding concentrations, third step of start-up, closed loop.

5 Conclusions

In this paper we suggested an efficient hierarchical two-level hybrid control scheme for the start-up of a coupled distillation pant with multiple continuous control units at the lower level and a single coordinating discrete control unit at the top level. It is set within J.C. WILLEMS' behavioural framework. The method provides a mathematical guarantee that the resulting hierarchical control strategy indeed solves the original control problem.

References

[1] P. E. Caines and Y. J. Wei. Hierarchical hybrid control systems: a lattice theoretic formulation. *IEEE Transactions on Automatic Control*, 43(4):501–508, April 1998. Special Issue on Hybrid Systems.

[2] T. Moor and J. Raisch. Supervisory control of hybrid systems within a behavioural framework. *Special Issue on Hybrid Systems, System and Control Letters*, 38(3):157–166, 1999.

[3] T. Moor, J. Raisch, and J. M. Davoren. Computational advantages of a two-level hybrid control architecture. In *Proc. 40th IEEE Conference on Decision and Control*, pages 358–363, Orlando, USA, 2001.

[4] G. J. Pappas, G. Lafferriere, and S. Sastry. Hierarchically consistent control systems. *IEEE Transactions on Automatic Control*, 45(6):569–573, 2000.

[5] J. Raisch, A. Itigin, and T. Moor. Hierarchical control of hybrid systems. In *Proc. ADPM 2000 (Automatisation des Processus Mixtes: les Systemes Dynamiques Hybrides)*, pages 67 – 72. Shaker-Verlag, 2000.

[6] J. Raisch and S. D. O'Young. A Totally Ordered Set of Discrete Abstractions for a given hybrid or continuous system. In P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors, *Hybrid Systems IV*, volume 1273 of *LNCS*, pages 342–360. Springer Verlag, 1997.

[7] P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event processes. *SIAM Journal of Control and Optimization*, 25(1):206 – 230, 1987.

[8] R. C. Reid, J. M. Prausnitz, and B. E. Poling. *The Properties of Gases and Liquids*. McGraw-Hill, New York, 4. edition, 1987.

[9] M.G. Singh. *Dynamical Hierarchical Control*. North-Holland, Amsterdam, 1980.

[10] J. C. Willems. Models for dynamics. *Dynamics Reported*, 2:172 – 269, 1989.

[11] K. C. Wong and W. M. Wonham. Hierarchical control of timed discrete-event systems. *Discrete Event Dynamic Systems*, 6(3):274–306, July 1996.