International Journal of Innovative Research in Engineering & Management (IJIREM)
ISSN: 2350-0557, Volume-2, Issue -5, September 2015

Simulation Study of Concurrent Structural Decentralised Discrete-Event Systems: Case Study with an Industrial Sized Chemical Batch Plant

Jae-Sam Park
Department of Electronics Engineering, Incheon National University, 119 Academy Road, Songdodong, Yeonsugu, Incheon, Korea,

Sang-Heon Lee
The School of Engineering, The university of South Australia, Mawson Lakes, SA 5095 Australia, sang-

Moon-Kyu Lee
Department of Industrial and Management Engineering, Keimyung University,1095 Dalgubeoldaero Dalseo-gu, Daegu 704-701, Republic of Korea,

Ill-Soo Kim
Department of Mechanical Engineering, College of Engineering, Mokpo National University,1666, Yeongsan-ro, Muan-gun, Jeonnam, 534-729, Korea,

ABSTRACT
A structural decentralised control of discrete-event system (DES) is employed for a simulation study of a sequence control of cleaning-in-process of chemical batch plant. Due to the number of states of the whole plant model is too big, the whole plant has sub-divided into smaller sizes and in the local level, the DES models for the local plant and local specifications are established. The conditions developed for the structural decentralised DES are verified to ensure the concurrent operations of decentralised control will guarantee the global optimality for the whole plant. Since the conditions are structure-dependent (not specification-dependent like other approaches of decentralised DES), the verification needs to be done only once, and then the all future operations in the same plant will guarantee the global optimality. The simulation study demonstrate the practicability of the approach with computational savings.

Keywords
Discrete-Event System, Structural Decentralised Control, Chemical Batch Process, Sequence Control.

1. INTRODUCTION
The proceedings are the records of the conference. IJIREM hopes to give these conference by-products a single, high-quality appearance. To do this, we ask that authors follow some simple guidelines. In essence, we ask you to make your paper look exactly like this document. The easiest way to do this is simply to down-load a template from [2], and replace the content with your own material. The nature of discrete-event systems (DES) is quite different from that of conventional time driven systems, and hence several formal methods to analyse and design such systems are developed. One of such methods called supervisory control theory (SCT) [1, 2] is proposed to automatically synthesise a supervisor that allows the optimally permissible behaviours of the system restricted by the given specification. Since this method treats the open-loop and close-loop control separately, it is possible to compare and select the optimal performance of different control policies on the behaviours of the uncontrolled plant. A large class of dynamic systems in computer network, manufacturing systems, supply chain management and others can be analysed and synthesised using this method. Since the operations of chemical batch processes are naturally discrete in terms of the changes of states, SCT has been identified as an alternative approach for the generation of the optimal sequence control for efficient batch operations. Typically chemical batch processes are dealt with multi-products requiring flexible production using multi-purpose facilities, and in most cases several different products of different quantities are produced using the same equipment. Hence it is crucial to have the optimal sequence control to produce the right product with the right quantity in the most effective way. Such effective sequence control (sometimes called operation scheme) of chemical batch plants is vital for small and medium sized companies especially pharmaceutical, food processing, cosmetics and biochemical industries. As a matter of fact, the sequence control or scheduling problem in chemical batch processes is a classical problem which has attracted a significant amount of attention for the last few decades [3, 4]. The methods using mixed integer linear or nonlinear programming (MILP/MINLP) [5, 6, 7, 8, 9] are the most widely used ones in the sequence control of chemical batch processes thanks to its simplicity and rigorousness. The main problem of such approaches is in the computational complexity, which is increased exponentially with the increase of the number of components, in the worst case. Hence they often become impractical for large scale plants. To overcome this issue, more comprehensive techniques with a combination of graphical representations, intuitive skills of experts or even with some artificial intelligence have been proposed and used to some practical applications; for example, state-task network [10], genetic algorithm [11], Petri nets[12,13], timed automata [14] and many more. SCT has also been proposed as an alternative approach in the sequence control of chemical batch processes [15]. However, the application of SCT into real industrial practices is much smaller than the research activities due to its inheriting computational complexity problem. Even though the computational complexity to design a supervisor in SCT is polynomial in terms of state sizes of plants and its specifications, it will be increased exponentially with the increase of the number of components involved [16]. Over the last few decades, several schemes are suggested and used into practical

applications to overcome such problem: for example, modular or decentralised control [17, 18] and hierarchical control [19]. However, since the conditions to establish such control schemes need to be checked and they are specification dependent, even a minor change introduced in the specification requires the conditions to be re-evaluated again for the entire system. This definitely could introduce a computational burden to the synthesis process. Intuitively, it is logical that if a certain system is established with a more flexible structure, then some of future changes in specifications can be handled without further additional efforts. This is a fundamental idea behind the structural decentralised approach in SCT [20]. Basically, since the conditions to synthesis the decentralised control scheme become system structure dependent, once the scheme is established, a large set of minor changes in specifications can be automatically validated without any further analysis. The study shows that there is an exponential savings on computational efforts involved [20]. Only when a major change is introduced, the conditions need to be verified again. However, the structural decentralised approach in SCT has not been applied into a large scale industrial case study. In this paper, we are going to discuss this issue: how such structural decentralised control in SCT could be used in a large scale chemical batch plant to demonstrate the advantages of the approach. This case study is crucial for the verification of the practicability of the approach. The cleaning-in-processing (CIP) of an industrial scale chemical batch plant for small scale multi-purpose and multi-products is used in this study [21]. Generally, the CIP in many chemical processes is used to clean the equipment and their associated pipelines using water and/or detergent solutions for safety reasons and product quality. Obviously, since the CIP in multi-product chemical batch processes may take a large portion of the total production time, the operation of CIP in an efficient and optimal sequence is important to improve the productivity of the whole plant. Since the CIP is typically running in a batch-mode (so it is inherently discrete), the process can be suitably modelled as a DES. Furthermore, the CIP often requires flexible configurations. For example, the cleaning process for one part of the plant is often different from the cleaning of another part. Hence a decentralised control would be preferable. On top of it, since the cleaning procedure might need to be modified frequently depending on the materials being processed in the plant or maybe a new material being introduced for the operation, the structural decentralised approach of SCT would be a desirable choice to handle the computational complexity. The implementation process of the CIP in this paper is divided into two parts. Firstly, the formal specification expressing the desirable behaviour of the batch chemical processes under the decentralised control scheme is created, including the decentralised plant configurations. This specification is about the function describing types of products, types of equipment, time of production, the sequences of operation and other restrictions. Secondly, the supervisor is synthesised to allow the maximally possible behaviour of the system satisfying the given specification. The specification and DES are generated using the structural decentralised approach of SCT, expressed as finite state automata. The results clearly demonstrate the usefulness of this approach with a considerable amount of computational savings. The remaining of the paper is organised as follows. Section 2 will cover the background information on the SCT and structural decentralised supervisory control framework. Section 3 provides the description of CIP process and specifications. As the main section of this paper, the simulation study using CIP is presented in section 4. Finally, section 5 presents the conclusion and the future works.

2. STRUCTURAL DECENTRALISED CONTOL OF SUPERVISORY CONTROL SYSTEM

In the approaches of SCT [22], an uncontrolled DES is modelled by a 5-tuple automaton, $G = (Q, \Sigma, \delta, q_0, Qm)$, where Q is a set of states (we assume that Q is finite), Σ is a finite set of event labels, often called an alphabet, $\delta : Q \times \Sigma \rightarrow Q$ is a (partial) state transition function, $q_0 \in Q$ is the initial state, and $Q_m \subseteq Q$ is a set of marker states. Let $\Sigma^* = \Sigma^+ \cup \{\cdot\}$, where $\cdot\cdot \notin \Sigma$ represents the empty string and Σ^+ is the set of all finite strings of event labels in Σ. Among the subsets of Σ^*, known as languages over Σ, two of them are more significant than the others: the closed behaviour of G defined as $L(G) = \{s \in \Sigma^* | \delta(q_0, s) \text{ is defined}\}$, and the marked behaviour of G defined by $L_m(G) = \{s \in L(G) \mid \delta(q_0,s) \in Q_m\}$. Note that L(G) is the set of all possible finite sets of events that G can generate from q_0 and $L_m(G) \subseteq L(G)$ is a subset of L(G) that reach Q_m, possibly representing completions of a certain significant task. The control over G is introduced using a subset of events which can be disabled (prevented from occurring) and enabled (permitted to occur) by some supervisor whenever desired. These events, $\Sigma_c \subseteq \Sigma$, are called controllable events. The remaining events $\Sigma_u = \Sigma - \Sigma_c$ are uncontrollable. For a given plant G and a specification E with its behaviour representing the desired closed-loop behaviour, the supervisor S can be obtained by an algorithm presented in [23]; $L_m(S/G) = K_{L(G)}(L_m(E) \cap Lm(G))$. Abstractly, we can consider $\cdot\, k_{L(G)}$ as representing the process of synthesising a least restrictive supervisor for a plant (G) satisfying the given specification (E). Fundamentally, the supervisor S does not force G to execute a particular event; rather it simply permits some events to occur so that S can effectively force a particular event to occur by disabling all other events at a given state except the desired event. Two or more smaller DES's can be combined into a larger DES using the natural projection concept. Let Σ_1 and $\Sigma 2$ be two event sets, not necessarily disjoint, i.e., $\Sigma 1 \cap \Sigma 2 \neq \phi$. Let $\Sigma = \Sigma 1 \cup \Sigma 2$. The natural projection $p_i : \Sigma^* \rightarrow \Sigma i^*$ is defined by

$$p_i(\varepsilon) = \varepsilon$$
$$p_i(s\sigma) = \begin{cases} p_i(s)\sigma & \text{if } \sigma \in \Sigma_i \\ p_i(s) & \text{Otherwise,} \end{cases}$$

for $s \in \Sigma^*$ and $\sigma \in \Sigma$. The action of p_i on a string s is just to erase all occurrences of event $\cdot$ which do not belong to Σ_i. Using this we can combine several DES's into one DES. This process is called the synchronous composition [24]. For $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_2^*$, the synchronous composition $L_1 \| L_2 \subseteq \Sigma^*$ is defined according to

$$L_1 \| L_2 = p_1^{-1}(L_1) \cap p_2^{-1}(L_2).$$

Note that $p_i^{-1}()$ is the inverse projection of p_i. If $L(G_i) = L_i$ for i=1, 2, then one can think of G_1 and G_2 as generating the resultant DES $L_1 \| L_2$ by agreeing to synchronise common (sharing) events and the remaining uncommon events can occur whenever possible. For a decentralised system, we assume that a centralised plant can be divided into several smaller decentralised plants as described in [25]. The concurrent operations of several decentralised plants could achieve the global objectives under certain conditions [26]. The framework of decentralised control is as follows: Let $\Sigma_1, \Sigma_2, \cdots, \Sigma_n$ be the event sets of decentralised plants, $G_1, G_2, \cdots, G_n$, respectively with $\Sigma_i \cap \Sigma_j \neq \phi$, for i, j $\in$ {1, 2, ⋯ , n} and $i \neq j$. Assume that $\Sigma_i = \Sigma_{ic} \cup \Sigma_{iu}$ and the two subsystems, G_i and G_j ($i \neq j$), agree on the control status of shared events, that is, $\Sigma_{iu} \cap \Sigma_j = \Sigma_i \cap \Sigma_{ju}$. Then for the global system, G, the event set, the controllable events and the uncontrollable events are obtained, respectively, as

$$\Sigma := \bigcup_{i=1}^{n} \Sigma_i,\ \Sigma_c := \bigcup_{i=1}^{n} \Sigma_{ic},\ \text{and}\ \Sigma_u := \bigcup_{i=1}^{n} \Sigma_{iu}.$$

Let $L_{i,m}$, $L_i \subseteq \Sigma^*$ represent respectively the marked and the closed behaviours of local plant $\mathbf{G}_i$. Then the marked and the closed behaviours of the overall system **G** are, respectively

$$L_m = L_{1,m} \parallel L_{2,m} \parallel \cdots \parallel L_{n,m} = \bigcap_{i=1}^{n} p_i^{-1}(L_{i,m}),$$
$$L = L_1 \parallel L_2 \parallel \cdots \parallel L_n = \bigcap_{i=1}^{n} p_i^{-1}(L_i).$$

The specifications are given locally. Let $E_i \subseteq L_{i,m}$, a $L_{i,m}$-closed language, represent a specification on a local plant $\mathbf{G}_i$. The corresponding specification on the global plant **G** is

$$(p_i|_L)^{-1}(E_i) = p_i^{-1}(E_i) \cap L,$$

where $(p_{i|L})$ denotes the restriction of p_i on L. The overall specification for the global plant will then be the combination of all local specifications applied on the global plant:

$$E := \bigcap_{i=1}^{n} (p_i|_L)^{-1}(E_i).$$

The decentralised control aims to ensure the global supervisory control on the global plant to be the same as the concurrent actions of local supervisory controls applied in each local plant:

$$\bigcap_{i=1}^{n} (p_i|_L)^{-1}\overline{(\kappa_{L_i}(E_i))} = \overline{\kappa_L(E)},$$
$$\text{or, } \overline{(\kappa_{L_1}(E_1)) \parallel (\kappa_{L_2}(E_2)) \parallel \cdots \parallel (\kappa_{L_n}(E_n))}$$
$$= \overline{\kappa_{L_1 \parallel L_2 \parallel \cdots L_n}(E_1 \parallel E_2 \parallel \cdots \parallel E_n)}.$$

Generally, this is not always true. Hence a set of conditions in decentralised control schemes has developed to ensure the decentralised control to be the same as the centralised control [18, 25, 27]. However the conditions developed in these researches need to be verified whenever the specification is changed (even minor changes). As a consequence, the computational complexity becomes high if the specification changes required frequently. Practically, this is the case since it is common to produce multi-products using multi-production sequences in most of flexible chemical batch processes. A structural decentralised control system [20] has addressed this computational complexity issue with the introduction of conditions given in the system structure. Hence once the decentralised control scheme is established, a large set of minor changes of specifications can be automatically validated without any further analysis. This actually agrees with the intuition that if the system structures are properly established, then the operation would become easier and more flexible. These conditions are the first computationally efficient ones that systematically guarantee the optimality of decentralised control in the structure of DES rather than on certain specifications only. There are two conditions to establish a structural decentralised control in SCT:

i) Shared Marking Condition: $L_{i,m}$ marks $\Sigma_i \cap \Sigma_j$ and $L_{j,m}$ marks the same set,

ii) Mutual controllability condition: L_i and L_j are mutually controllable.

For the details, refer [20]. It seems that the verification of two conditions might be computationally expensive. However, the verification needs to be done only once for a given system structure. Hence in the long run, this cost would be paid off. In a special case that if only controllable events are shared among decentralised plants, only the shared marking conditions need to be verified and the computational complexity will be linear in the numbers of states and transitions. This paper presents how such structural decentralised control systems can be applied to a large scale chemical batch plant to demonstrate the significant computational savings in the process.

3. DESCRIPTION OF CIP IN A CHEMICAL BATH PROCESS

The CIP plant employed in this study is a small scale chemical batch process for food and pharmaceutical products. The schematic diagram of the plant is presented in figure 1 (the diagram is adopted from [21] with slight modifications).

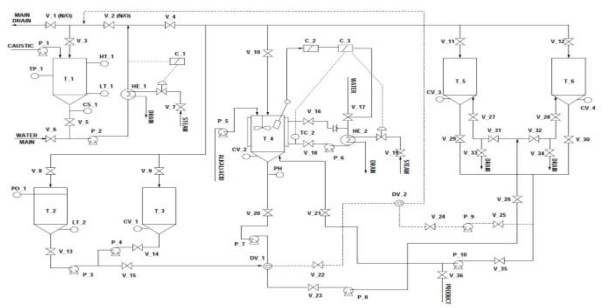


Figure 1. A schematic diagram of the whole chemical batch process

In this study, we focus on the automatic cleaning process of the feed preparation tank T_2 and its associated pipelines using caustic liquid prepared in the caustic detergent solution tank T_1. Figure 2 shows all components involved in this operation.

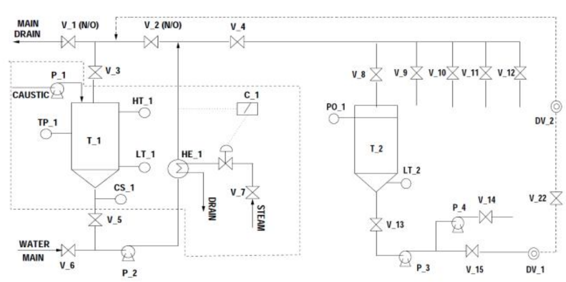


Figure 2. A schematic diagram for the components involved in CIP for tank T_2

The detailed list of components involved in this operation is presented in table 1.

Table 1. All components involved in CIP of tank T_2

Components	Labels	Function
Pumps	P_1, P_2, P_3, P_4	To supply water or detergent solution
Valves	V_1-V_{15}, V_{22}, DV_1, DV_2	To supply water or detergent solution. Note the initial state of valves V_1 and V_2 are open while all others are closed.
Temperature controller	C_1	To control the temperature in heat exchanger HE_1
Conductivity level sensor	CS_1	To monitor the caustic level in detergent solution
Lid position sensor	PO_1	To check the lid of tank T_2 (either open or shut)
Temperature probe	TP_1	To check the temperature of tank T_1
Low level sensor	LT_1	To check the low level limit of tank T_1
High level sensor	HT_1	To check the high level limit of tank T_1
Liquid level sensor	LT_2	To check the liquid level of tank T_2
Timer	TS_1	To deal with the timing requirements

The operation procedure of CIP for T_2 is as follows: firstly T_2 will be pre-rinsed for 10 minutes with high pressured water supplied from the water main. The water level in T_2 will be monitored to ensure that it will not overflow. After the water pre-rinse operation, all water will be drained out via the water drain. Then the hot caustic detergent solution is prepared in T_1. The caustic liquid is supplied by pump P_1 to T_1. The level and the temperature of the caustic liquid will be monitored during the operation. In the next step, the caustic solution prepared in T_1 will be sprayed to T_2 for 10 minutes. The level in T_2 is monitored to prevent from overflowing. Once it is done, the water post-rinse operation is conducted to ensure all caustic detergent solution is washed away. The process is the same as the water pre-rinse operation. This completes one cycle of CIP for T_2. Note that without any control actions applied, each component can work independently and asynchronously. Since the number of the elementary components is 30 (as shown in table 1) and if each component is assumed to have 2 states each, the total number of states could be more than 1.0×10^9. This is too big to analyse as a whole. Hence a decentralised control of DES is an ideal choice for the study. From the analysis of operations, the whole operations are divided into 4 sub-operations as shown in figure 3. Note that we have denoted a symbol for each operation as shown in the figure: Water pre-rinse operation ($\mathbf{G}_{wr}$), Detergent preparation operation ($\mathbf{G}_{det}$), Detergent rinse operation ($\mathbf{G}_{dr}$) and water post-rinse operation ($\mathbf{G}_{wpr}$). Due to the nature of CIP of chemical batch plants, the operation procedures are required to be modified frequently to meet the demands for production flexibility. Therefore, the structural decentralised DES framework [20] is an ideal choice to deal with the computational complexity issue.

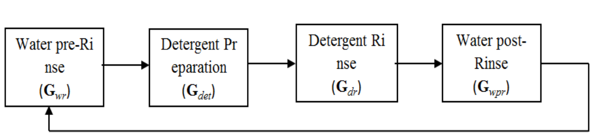


Figure 3. Sequential operations of sub-processes of CIP for tank T2

Due to the lengthy representation of the whole operation, we describe mainly the simulation study on the water pre-rinse operation in this paper. Other operations can be established in a similar way. The procedure of water pre-rinse operation is as follows:

(1) At the beginning of the operation, each elementary omponent is in its initial state, the tank T_2 is empty and the lid of T_2 is closed.

(2) The operation is started with opening the feed route from the water main to T_2 (via V_4, V_6, V_8), and the drain route from T_2 to the main drain (via V_{13}, V_{15}, DV_1, V_{22}, DV_2, and V_1).

(3) The tank T_2 is cleaned with high-pressure water spray using pump P_2 for 10 minutes. During the normal operation, the water level inside T_2 should remain between 6L to 20L (monitored by LT_2).

(4) If the lid of T_2 is opened at any time during the operation, water feed should be stopped and not be permitted to restart until the lid is shut for safety reason. When the lid is closed, water feed can be started again only after the water in the tank T_2 is fully drained.

(5) Water is drained from T_2 by pump P_3. The pump P_3 should be stopped if the water level of T_2 is less than 3L.

(6) After 10 minutes of the continuous operations, T_2 is fully drained.

(7) After the operation is completed, all components are returned to their initial states.

The number of components involved in the water-rinse operation is 23. So the size of the whole plant is still too big (more than 8,000,000 states) to analyse. From the careful observation, it is found that some components are actively involved in the process and some components are staying in its initial state during the entire operation. Hence the whole plant is further divided into two main sub-plants: a plant with the components actively participated in the operation (called $\mathbf{G}_{yp}$), and the other plant with the components remaining in its initial states (called $\mathbf{G}_{id}$). The plant **G**yp is further sub-divided into two sub-plants: the one with the components for water pre-rinse preparation ($\mathbf{G}_{pr}$), and the one with the component for water spraying and cleaning operation ($\mathbf{G}_{pa}$). Table 2 shows the list of those components. Note that there is no common component among sub-plants.

Table 2. Partitioned elementary component list for water pre-rinse operation of CIP

<table>
<tr><th colspan="3">Water pre-rinse operation ($\mathbf{G}_{wr}$)</th></tr>
<tr><td rowspan="2">Plant with components remaining in the initial state ($\mathbf{G}_{id}$)</td><td colspan="2">Plant with components actively participating in the operation ($\mathbf{G}_{yp}$)</td></tr>
<tr><td>Plant with components for water spraying and cleaning operation ($\mathbf{G}_{pa}$)</td><td>Plant with components for water pre-rinse preparation ($\mathbf{G}_{pr}$)</td></tr>
<tr><td>V_5, V_9, V_{10}, V_{11}, V_{12}, P_4, V_1, V_3, V_{14}</td><td>PO_1, LT_2, P_2, V_8, TS_1, LT_2, P_3, V_{13}</td><td>V_2, V_4, V_6, DV_1, DV_2, V_{15}, V_{22}</td></tr>
</table>

The synthesis of structural decentralised DES is progressed as follows:

Step 1: from the analysis of the operation, the whole operation is divided into a sequence of sub-operations and the components involved in each sub-plant are identified.

Step 2: within the structural decentralised control of DES framework, the synthesis of a decentralised supervisor using the specifications given in each sub-plant is conducted.

Step 3: the necessary conditions for structural decentralised control are verified. The verification will guarantee that the concurrent operation of the structural decentralised supervisory control will always be the same as the global optimal one and there is no need to check the condition again for a set of future minor changes in local specifications.

Note that all simulations in this study are carried out using DES software package XPTCT developed along with the note by Wonham [22].

4. SUPERVISOR SYNTHESIS PROCESS FOR CIP BATCH PROCESS

To synthesise a supervisor, we need to create DES models for plants and specifications, which can be represented as automata. Most of components have only two states, on and off for pumps, open and closed for valves, or similar to others. Low level sensor and the timer, LT_1 and TS_1, are exceptions with 4 states each. Figure 4 shows the DES automata model for some typical components.

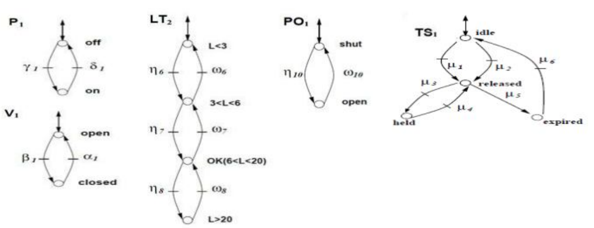


Figure 4. DES automata model for components in CIP

In those automata models, a state is represented by a circle (o) and an event is described by an arrow from an exit state to an entrance state with an event label attached. The initial state is labelled with an entering arrow (→ o), and a marker state is labelled with an exiting arrow (o →). A double arrow (o ↔) indicates that the initial state is also a marker state. The arrow with a 'tick' indicates that the event is controllable. Due to the nature of the operations (like valves and pumps), most events are controllable events except those in the lid position sensors (PO_1) (events labelled as η_{10}, ω_{10}) and one event (expired, labelled as μ_5) in the timer TS_1. Note that TS_1 can be set either 10 minutes (μ_1) or 15 minutes (μ_2). The operation of the whole CIP has divided into a sequence of 4 sub-operations as shown in Figure 3, To ensure the sequential process of these sub-operations, we introduced additional controllable shared events, λ_1, λ_2, λ_3 and λ_4**.** These synchronous shared events indicate a completion of each corresponding sub-operation and allow the next sub-operation to proceed. For example, the event λ_1 represents a completion of the water pre-rinse operation and hence the detergent preparation operation can now proceed. The events λ_2 and λ_3 can be interpreted similarly. The event λ_4 is an event shared by all local operations, representing a complete cycle of the CIP operation of T_2. In addition, from the analysis, it is found that the water pre-rinse operation can be divided into the three sub-processes: preparation for the water feed route, preparation of water drain route and water spraying and cleaning operation. Note that the water spraying and cleaning operation cannot start until water feed route and water drain route are ready. To enforce such sequential orders, three controllable shared events (σ_1, σ_2 and σ_3) are also introduced. The events σ_1 and σ_2 represent the feed route and the drain route ready, respectively while the event σ_3 represents that the water spraying and cleaning to T_2 is completed and now all components can return to their initial states. DES automata representing these sequential constraints with physical constraints of the plant have been names as **flag** and they will be a part of the DES model for the local plant. The DES model of each local operation is obtained by the synchronous composition of the elementary components involved in that particular local operation with such DES model for **flag**. In the following sections, we describe how the supervisors of local processes are synthesised.

4.1 Supervisor synthesis of DES model for water pre-rinse preparation operation ($\mathbf{G}_{pr}$)

Before the water spraying and cleaning operation of T_2 is conducted, the water feed route and the water drain route should be ready. Naturally, this operation is divided into two: water feed preparation ($\mathbf{G}_{\text{fpr}}$) and water drain preparation ($\mathbf{G}_{\text{dpr}}$). Table 3 shows the components involved in those operations with their even labels.

Table 3. Partitioned elementary component list for water pre-rinse preparation ($\mathbf{G}_{pr}$)

Components for water feed preparation ($\mathbf{G}_{fpr}$)	$V_2(\alpha_2, \beta_2)$, $V_4(\alpha_4, \beta_4)$, $V_6(\alpha_6, \beta_6)$
Components for water drain preparation ($\mathbf{G}_{dpr}$)	$V_{15}(\alpha_{15}, \beta_{15})$, $V_{22}(\alpha_{16}, \beta_{16})$, $DV_1(\alpha_{17}, \beta_{17})$, $DV_2(\alpha_{18}, \beta_{18})$

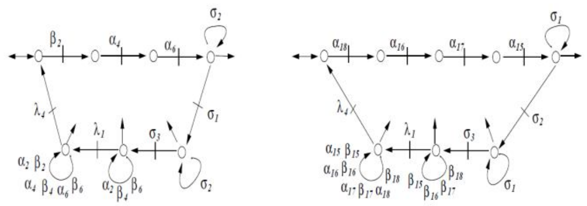
In addition, DES models for **flag** in this operation are also created to ensure the right sequential operations. Figure 5 shows the DES models of **flag**'s. The interpretation of $\mathbf{flag}_{fpr}$ for water feed preparation (shown in figure 5(a)) is as follows: firstly after V_2 is closed (event β_2), and V_4 and V_6 are opened (α_4, α_6), then signalling the water feed route is ready (σ_1). After the water spraying and cleaning for T_2 is completed (σ_3), all valves are allowed to return to their initial states (α_2, β_4, β_6). This completes the water pre-rinse operation (λ_1). Then signalling a completion of one whole cycle of CIP for T_2 is made (λ_4). The DES model $\mathbf{flag}_{dpr}$ for the water drain route (shown in figure 5(b)) can be interpreted similarly. The orders of valve operations are important to avoid unnecessary waste of water. The self-loops of events after the event λ_1 in both figures are essential for those events to be allowed to occur in the other sub-operations in order not to restrict their behaviours. The synchronous composition of DES models of all components involved and **flag**'s will generate DES models for the whole operations of $\mathbf{G}_{\text{fpr}}$ and $\mathbf{G}_{\text{dpr}}$.

Their event sets are

$$\Sigma_{fpr} = \{\alpha_2, \beta_2, \alpha_4, \beta_4, \alpha_6, \beta_6, \sigma_1, \sigma_2, \sigma_3, \lambda_1, \lambda_4\}$$
$$\Sigma_{dpr} = \{\alpha_{15}, \beta_{15}, \alpha_{16}, \beta_{16}, \alpha_{17}, \beta_{17}, \alpha_{18}, \beta_{18}, \sigma_1, \sigma_2, \sigma_3, \lambda_1, \lambda_4\},$$

and the shared events are

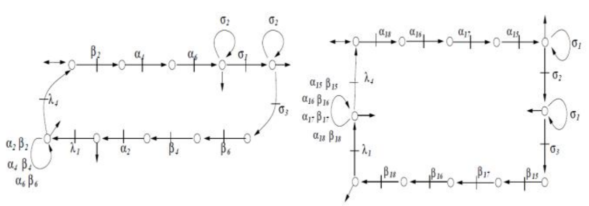
$$\Sigma_{fpr} \cap \Sigma_{dpr} = \{\sigma_1, \sigma_2, \sigma_3, \lambda_1, \lambda_4\}$$



(a)$\mathbf{flag}_{fpr}$ (b) $\mathbf{flag}_{dpr}$

Figure 5. DES models ensuring the sequential orders of $\mathbf{G}_{pr}$

The local specifications, $\mathbf{E}_{\text{fpr}}$ and $\mathbf{E}_{\text{dpr}}$, for the two local preparations, $\mathbf{G}_{\text{fpr}}$ and $\mathbf{G}_{\text{dpr}}$, respectively can be generated according to the operation procedure and restrictions. Their DES models are given in figure 6(a) and 6(b), respectively.



(a)$\mathbf{E}_{fpr}$ (b) $\mathbf{E}_{dpr}$

Figure 6. DES models for the specification of $\mathbf{G}_{pr}$

Using the DES models for the local operation and the corresponding specification given locally, the supervisor in the local operation can be obtained by the algorithm presented in [23] using the simulation software XPTCT[22]. The results show that the supervisor for $\mathbf{G}_{\text{fpr}}$ has 17 states with 36 transitions, and the supervisor for $\mathbf{G}_{\text{dpr}}$ has 27 states with 78 transitions.

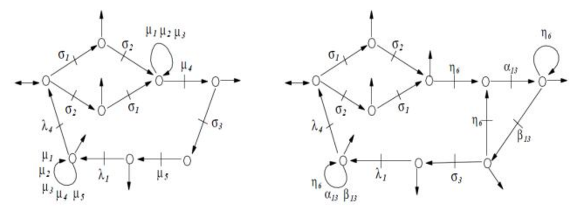
4.2 Supervisor synthesis of DES model for water spraying and cleaning operation ($\mathbf{G}_{pa}$)

After the preparation is completed, water spraying/cleaning and draining operations (denoted as $\mathbf{G}_{\text{pa}}$) can be started. The initial modelling shows that the DES model for $\mathbf{G}_{\text{pa}}$ is still very complex: it could have more than 65,000 transitions. From the careful analysis, it is found that the entire plant, $\mathbf{G}_{\text{pa}}$, can be divided into two sub-plants: $\mathbf{G}_{\text{fpa}}$ consisting the components involved in the water spraying and cleaning operations and $\mathbf{G}_{\text{dpa}}$ consisting those involved in the water drain operations (refer Table 4 for the component list and their event labels).

Table 4. Partitioned elementary component list for water spraying and cleaning operation ($\mathbf{G}_{pa}$)

Components for water spraying and cleaning operation ($\mathbf{G}_{fpa}$)	$PO_1(\eta_{10}, \omega_{10})$, $LT_2(\eta_6, \omega_6, \eta_7, \omega_7, \eta_8, \omega_8)$, $P_2(\gamma_2, \delta_2)$, $V_8(\alpha_8, \beta_8)$, $TS_1(\mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6)$
Components for water draining operation ($\mathbf{G}_{dpa}$)	$LT_2(\eta_6, \omega_6, \eta_7, \omega_7, \eta_8, \omega_8)$, $P_3(\gamma_3, \delta_3)$, $V_{13}(\alpha_{13}, \beta_{13})$

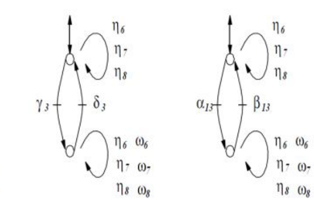
To ensure the sequential operations and synchronisation with other local operations, the two DES models $\mathbf{flag}_{\text{fpa}}$ and $\mathbf{flag}_{\text{dpa}}$ are introduced in here as well. The interpretation of $\mathbf{flag}_{\text{fpa}}$, shown in figure 7(a), is as follow: when the feed route and the drain route are ready (σ_1, σ_2), the timer TS_1 is started with 10 minutes setting (μ_1). After the timer has expired (μ_5), send a signal σ_3 to the other local operations and allow all the components to return to their initial states. After the timer has been reset (μ_6), the water pre-rinse operation is completed with the occurrence of the synchronisation event λ_1 and the detergent preparation operation can now proceed. Then, one cycle of CIP for T_2 is finished (λ_4). Figure 7(b) for $\mathbf{flag}_{\text{dpa}}$ can be interpreted similarly.



(a) $\mathbf{flag}_{fpa}$ (b) $\mathbf{flag}_{dpa}$

Figure 7. DES models for ensuring the sequential orders of ($\mathbf{G}_{pa}$)

In addition, some physical constraints, usually coming from conservation of mass, gravitational consideration and others among the components, are necessary to be considered to restrict the system behaviours by deleting physically infeasible states and transitions. For example, the level of T_2 can only be increased after the feed pump P_2 is turned on (see figure 8(a)). The synchronous composition of DES models for all components involved and the corresponding **flag** with such physical constraints generates all possible system behaviours. Note that the physical constraints in figure 8(a) and (b) are for ($\mathbf{G}_{\text{fpa}}$), while (c) and (d) are for ($\mathbf{G}_{\text{dpa}}$).



(a) (b)$_r$ (c) (d)

Figure 8. DES models for physical constraints of ($\mathbf{G}_{pa}$): a & b for $\mathbf{G}_{fpa}$ and c & d for $\mathbf{G}_{dpa}$

The synchronous composition of DES models of physical constraints and the corresponding local DES models gives the final DES models for $\mathbf{G}_{\text{fpa}}$ and $\mathbf{G}_{\text{dpa}}$. The event sets are

$$\Sigma_{fpa} = \{\alpha_8, \beta_8, \gamma_2, \delta_2, \eta_6, \omega_6, \eta_7, \omega_7, \eta_8, \omega_8, \eta_{10}, \omega_{10}, \mu_1, \mu_2, \mu_3, \mu_4, \mu_5, \mu_6, \sigma_1, \sigma_2, \sigma_3, \lambda_1, \lambda_4\}$$

$$\Sigma_{dpa} = \{\gamma_3, \delta_3, \alpha_{13}, \beta_{13}, \eta_6, \omega_6, \eta_7, \omega_7, \eta_8, \omega_8, \sigma_1, \sigma_2, \sigma_3, \lambda_1, \lambda_4\}.$$

The shared events are

$$\Sigma_{fpr} \cap \Sigma_{dpr} = \{\eta_6, \omega_6, \eta_7, \omega_7, \eta_8, \omega_8, \sigma_1, \sigma_2, \sigma_3, \lambda_1, \lambda_4\}.$$

The DES model for $\mathbf{G}_{\text{fpa}}$ has 736 states with 3922 transitions while $\mathbf{G}_{\text{dpa}}$ has 118 states with 337 transitions. Compared to the size of $\mathbf{G}_{\text{pa}}$ (with more than 65,000 transitions), we can easily observe the reduction of the size of the operation. The local specifications ($\mathbf{E}_{\text{fpa}}$) for the water spraying/cleaning operation ($\mathbf{G}_{\text{fpa}}$) are as follows

- Wait until receiving signals that indicate the feed and drain routes are ready (σ_1, σ_2).
- If the lid of T2 is opened (η_{10}), wait until the lid is shut (ω_{10}).
- Open valve V_8 first (α_8) and turn on pump P_2 (γ_2). Then release the timer TS_1 with 10 minutes setting (μ_1).
- Pump P_2 should be turned off (δ_2) before valve V_8 is closed (β_8).
- If the lid of T_2 is opened (η_{10}), hold the timer (μ_3) and turn off P_2 (δ_2). Then close V_8 (β_8).
- If the lid of T_2 is shut (ω_{10}), drain all water in T_2 (ω_6) and open V_8 (α_8). Then turn on P_2 (γ_2) and re-release the timer (μ_4).
- If the water level in T_2 is increased to more than 20L (η_8), turn off P_2 (δ_2) and then close V_8 (β_8) to stop water supply to tank T_2.
- If the water level is decreased to less than 6L (ω_7), open V_8 (α_8) and turn on P_2 (γ_2).
- After the timer expired (μ_5), turn off P_2 (δ_2) and close V_8 (β_8). Then send the signal σ_3.
- The timer TS_1 can be reset (μ_6) only after the lid of T_2 is shut (ω_{10}), P_2 is off (δ_2) and the timer is expired (μ_5).
- Finish the operation with sending the signal λ_1 and allow the signal λ_4 to happen to complete the water pre-rinse operation
- The self-loops at the state after λ_1 ensure that the local operation $\mathbf{G}_{fpa}$ does not restrict the behaviours of the other local operations.

Formally, they can be modelled as a DES automaton as shown in figure 9. Similarly, the local specifications ($\mathbf{E}_{dpa}$) for the water drain operation ($\mathbf{G}_{dpa}$) can be established (See figure 10).

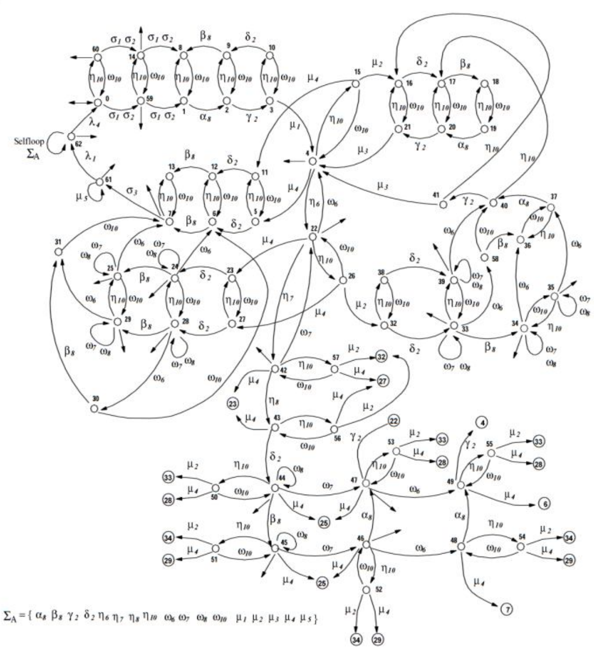


Figure 9. DES models of the local specification ($\mathbf{E}_{fpa}$) for ($\mathbf{G}_{fpa}$)

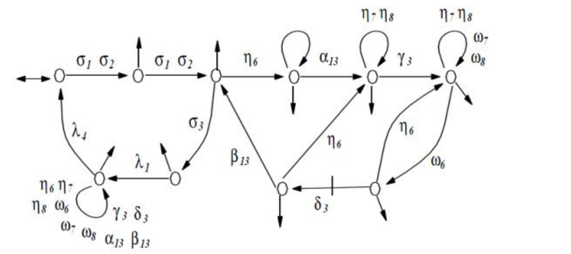


Figure 10. DES models of the local specification ($\mathbf{E}_{dpa}$) for G 엠

Supervisors satisfying these requirements are computed using XPTCT. The sizes of the supervisors are respectively 95 states with 221 transitions for $\mathbf{G}_{fpa}$ and 33 states with 75 transitions for $\mathbf{G}_{dpa}$.

4.3 Supervisor synthesis of DES model for the sub-plant with components remaining in their initial state ($\mathbf{G}_{id}$)

During the entire operation of water pre-rinse, some components should stay in its initial states (usually closed or off) to ensure the connected equipment or pipe lines to be isolated from water. This sub-plant ($\mathbf{G}_{id}$) needs to be modelled to ensuring that they stay in their initial states. Naturally, this can also be divided into two parts: $\mathbf{G}_{fid}$ (for water feed route) and $\mathbf{G}_{did}$ (for water drain route). Refer table 5 for the list of components and their event labels.

Table 5. Partitioned elementary component list for components remaining in initial state ($\mathbf{G}_{id}$)

Components in water feed part ($\mathbf{G}_{fid}$)	$V_5(\alpha_5,\beta_5)$, $V_9(\alpha_9,\beta_9)$, $V_{10}(\alpha_{10},\beta_{10})$, $V_{11}(\alpha_{11},\beta_{11})$, $V_{12}(\alpha_{12},\beta_{12})$
Components in water drain part ($\mathbf{G}_{did}$)	$P_4(\gamma_4,\delta_4)$, $V_1(\alpha_1,\beta_1)$, $V_3(\alpha_3,\beta_3)$, $V_{14}(\alpha_{14},\beta_{14})$

The DES models for shared synchronisation events to ensure the sequential operation, **flag**, can be easily established. The synchronous composition of DES models for all components and the corresponding **flag** generates a DES model for each local operation. The size of $\mathbf{G}_{fid}$ is 64 states with 384 transitions, while $\mathbf{G}_{did}$ has 32 states with 160 transitions. The event sets for $\mathbf{G}_{fid}$ and $\mathbf{G}_{did}$ are

$$\begin{aligned}\Sigma_{fid} &= \{\alpha_5,\beta_5,\alpha_9,\beta_9,\alpha_{10},\beta_{10},\alpha_{11},\beta_{11},\alpha_{12},\beta_{12},\lambda_1,\lambda_4\}\\ \Sigma_{did} &= \{\gamma_4,\delta_4,\alpha_1,\beta_1,\alpha_3,\beta_3,\alpha_{14},\beta_{14},\lambda_1,\lambda_4\}\end{aligned}$$

The shared events of Σ_{fid} and Σ_{did} are λ_1 and λ_4. For these two systems G_{fid} and G_{did}, the specifications are simple; to prohibit the occurrences of all events at their initial states except λ_1 and λ_4. The decentralised supervisors in the local operations can be obtained using XPTCT and the sizes of the supervisors are respectively 64 states with 304 transitions for $\mathbf{G}_{fid}$ and 32 states with 128 transitions for $\mathbf{G}_{did}$.

4.4 Verification of the conditions for structural decentralised control for water pre-rinse operation

In the above subsections, we have obtained the DES models of the local operations and local specifications, and then computed the corresponding local supervisors. In this section we will verify that the decentralised DES models of water pre-rinse operation satisfy the conditions for the structural decentralised control [20]. Firstly as a summary, the event sets of local operations are

$$\begin{aligned}
\Sigma_{fpr} &= \{\alpha_2, \beta_2, \alpha_4, \beta_4, \alpha_6, \beta_6, \sigma_1, \sigma_2, \sigma_3, \lambda_1, \lambda_4\} \\
\Sigma_{dpr} &= \{\alpha_{15}, \beta_{15}, \alpha_{16}, \beta_{16}, \alpha_{17}, \beta_{17}, \alpha_{18}, \beta_{18}, \sigma_1, \sigma_2, \sigma_3, \lambda_1, \lambda_4\} \\
\Sigma_{fpa} &= \{\alpha_8, \beta_8, \gamma_2, \delta_2, \eta_6, \omega_6, \eta_7, \omega_7, \eta_8, \omega_8, \eta_{10}, \omega_{10}, \mu_1, \mu_2, \mu_3, \\
&\qquad \mu_4, \mu_5, \mu_6, \sigma_1, \sigma_2, \sigma_3, \lambda_1, \lambda_4\} \\
\Sigma_{dpa} &= \{\gamma_3, \delta_3, \alpha_{13}, \beta_{13}, \eta_6, \omega_6, \eta_7, \omega_7, \eta_8, \omega_8, \sigma_1, \sigma_2, \sigma_3, \lambda_1, \lambda_4\} \\
\Sigma_{fid} &= \{\alpha_5, \beta_5, \alpha_9, \beta_9, \alpha_{10}, \beta_{10}, \alpha_{11}, \beta_{11}, \alpha_{12}, \beta_{12}, \lambda_1, \lambda_4\} \\
\Sigma_{did} &= \{\gamma_4, \delta_4, \alpha_1, \beta_1, \alpha_3, \beta_3, \alpha_{14}, \beta_{14}, \lambda_1, \lambda_4\}
\end{aligned}$$

The shared event set of each system $(\Sigma_s)_i = \Sigma_i \cap (\cup_{k \neq i}(\Sigma_i))$ are as follows

$$\begin{aligned}
(\Sigma_{fpr})_s &= \{\sigma_1, \sigma_2, \sigma_3, \lambda_1, \lambda_4\} \\
(\Sigma_{dpr})_s &= \{\sigma_1, \sigma_2, \sigma_3, \lambda_1, \lambda_4\} \\
(\Sigma_{fpa})_s &= \{\eta_6, \omega_6, \eta_7, \omega_7, \eta_8, \omega_8, \sigma_1, \sigma_2, \sigma_3, \lambda_1, \lambda_4\} \\
(\Sigma_{dpa})_s &= \{\eta_6, \omega_6, \eta_7, \omega_7, \eta_8, \omega_8, \sigma_1, \sigma_2, \sigma_3, \lambda_1, \lambda_4\} \\
(\Sigma_{fid})_s &= \{\lambda_1, \lambda_4\} \\
(\Sigma_{did})_s &= \{\lambda_1, \lambda_4\}
\end{aligned}$$

Since all the shared events are controllable, the mutual controllability conditions in the conditions for structural decentralised DES control are verified trivially. For the shared-event-marking conditions, we mark all the states before the shared events since they are significantly important states representing either signalling their corresponding readiness for the synchronisation, or warning the water level in the tank. In addition, the conditions in [20] require that local specification languages should be Li,m-closed. We verify this by checking if

$$\mathbf{E}_i = \overline{\mathbf{E}_i} \cap L_{i,m},$$

where $\mathbf{E}_i$ is the specification given on the local operation $\mathbf{G}_i$, and i is the index of the local operation. In fact, they are proved to be true using XPTCT. Therefore, all the conditions for the structural decentralised control are now satisfied. This guarantees that the concurrent actions of the structural decentralised DES supervisors achieve the same optimal behaviour as the centralised counterpart without blocking problems. Note that the supervisors for other 3 sub-operations, detergent preparation operation ($\mathbf{G}_{det}$), detergent rinse operation ($\mathbf{G}_{dr}$) and water post-rinse operation ($\mathbf{G}_{wpr}$), can also be synthesised using a similar process.

To see the advantages of structural decentralised DES control, assume a minor specification change is introduced. For example, instead of 10 minutes of setting in the timer (μ_1), the 15 minutes of setting (μ_2) is now required for hygienic reasons. For other decentralised control scheme [18, 25, 27], even with this simple specification change, the conditions to ensure the decentralised control to be same as the global optimal synthesis need to be verified again. This obviously introduces some computational burden to the synthesis process. However in the structural decentralised control scheme [20], it is not necessary to check again since such specification change is a part of structural synthesis process considered during the process to guarantee the global optimality.

5. CONCLUSION

In this paper, a simulation study of the structural decentralised DES control using the supervisory control theory is presented for a CIP process for a small scale industrial sized chemical batch plant. Since the number of states in the whole plant is too big (1 x 10^9), the centralised synthesis of supervisory control is not feasible. Hence the whole plant is subdivided into smaller local plants and in each local plant, the local specifications are established. The DES model for each local plant is synthesised by synchronous composition DES models for all components involved and the shared event model. In addition, a DES model for local specifications is obtained using the constraints given in the local level. The optimal local supervisor is obtained using supervisory control framework in each local operation. The conditions for the structural decentralised control developed by [20] are adopted for this simulation study. Unlike other studies, since those conditions are structure dependent, once the conditions are verified, the operations of decentralised control will ensure the global optimality for a set of future synthesis. The simulation study shows the computational savings, which can also be observed more clearly when all shared events are controllable like in this study. The work illustrated in this paper demonstrates the feasibilities on how the structural decentralised control can be applied to a practical large scale industrial plant.

REFERENCES

[1] Ramadge, P. J. & Wonham, W. M. 1987. Supervisory control of a class of discrete-event processes. SIAM J. Control and Optimization, 25, 1, 206–230.

[2] Wonham, W. M. 2011. Notes on control of discrete-event systems. Department of Electrical Engineering, University of Toronto, Toronto, Canada, 11, 1636-1637.

[3] Mendez, C.A., Cerda, J., Grossmann, I. E., Harjunkoski, I., and Fahl, M. 2006. State-of-the-art review of optimization methods for short-term scheduling of batch processes. Computers and chemical engineering, 30, 913-946.

[4] Floudas, C.A., Lin, X. 2004. Continuous-time versus discrete -time approaches for scheduling of chemical processes. a review, Computers & Chemical Engineering, 28, 2109-2129.

[5] Karimi, I.A., McDonald, C.M. 199. Planning and Scheduling of Parallel Semi-Continuous Processes - Part II : Short-Term Scheduling. Industrial and Engineering Chemistry Research, 36, 2701-2714.

[6] Pinto, J.M., Grossmann, I.E. 1995. A Continuous Time Mixed Integer Linear Programming Model for Short Term Scheduling of Multistage Batch Plants. Industrial and Engineering Chemistry Research, 34, 3037-3051.

[7] Sahinidis, N.V., Grossmann, I.E. 1991. MILNP model for cycle multiproduct scheduling on continuous parallel lines. Computers and Chemical Engineering, 15, 85-103.

[8] M´endez, C.A., Henning G.P., and Cerd´a J. 2001. An MILP Continuous-Time Approach to Short-Term Scheduling of Resource-Constrained Multistage Flowshop Batch Facilities. Computers and Chemical Engineering, 25, 701–711.

[9] Floudas, C.A. and Lin, X. 2005. Mixed Integer Linear Programming in Process Scheduling: Modelling, Algorithms,

and Applications. Annals of operation research, 139, 131-162.

[10] Kondili, E., Pantelides, C.C. and Sargent, R.W.H. 1993. A general algorithm for short-term scheduling of batch operations - 1. Mixed integer linear programming formulation. Computers and Chemical Engineering, 17, 211–227.

[11] Wang, K., Lohl, T., Stobbe, M., and Engell, S. 2000. A genetic algorithm for online-scheduling of multi-product polymer batch plant. Computers and Chemical Engineering, 24, 393-400.

[12] Ghaeli, M., Bahri, P.A., Lee, P. and Gu, T. 2005. Petri –net based formulation and algorithm for short-term scheduling of batch plants. Computers and Chemical Engineering, 29, 249-259.

[13] Yamalidou, E.C., Kantor, J.C. 1991. Modeling and optimal control of discrete-event chemical processes using Petri nets, Computers and Chemical Engineering, 15, 503-519.

[14] Panek, S., Engell, S., Subbiah, S. and Strusberg, O. 2008. Scheduling of multi-product batch plants based upon times automata models. Computers and Chemical Engineering, 32, 275-291.

[15] Akesson, K. and Fabian, M. 1999. Implementation of supervisory control for chemical batch processes. Proceedings of 1999 IEEE International conference on control applications, 1272-1277.

[16] Gohari, P. and Wonham, W. M. 2000. On the complexity of supervisory control design in the RW framework. IEEE transactions on systems, Man, and Cybernetics, Special Issue on DES, 30, 5, 643-652.

[17] Ramadge, P. J. and Wonham, W. M. 1988. Modular supervisory control of discrete event systems. Mathematics of Control, Signal and Systems, 1, 1,13–30.

[18] Lin, F. and Wonham, W.M. 1988. Decentralised supervisory control of discrete-event systems. Information and Science, 44, 199-224.

[19] Wong K.C. and Wonham W.M. 1996. Hierarchical control of discrete-event systems. Discrete-event dynamic systems, 6, 241-273.

[20] Lee, S., H. and Wong, K. C. 2002. Structural decentralised control of concurrent discrete-event systems. European Journal of Control, 8, 5, 477-491.

[21] Alsop, N. J. 1996. Formal technique for the procedural control of industrial processes. hD thesis, The Imperial college, London, U.K.

[22] Wonham, W. M. 2011. Design software for SCT for Window XP. Department of Electrical Engineering, University of Toronto, Canada.

[23] Ramadge, P. J. and Wonham, W. M. 1987. On the supremal controllable sublanguage of a given language. SIAM J. Control and Optimization, 25, 3, 637–659.

[24] Cassandras, C. G., Lafortune, S. and Olsder, S. 1995. Introduction to the modelling, control and optimisation of discrete-event systems. In A. Isidori, editor, Trends in Control-European perspective, Springer-Verlag, Berlin, 217–291.

[25] Willner, Y. and Heymann, M. 1991. Supervisory control of concurrent discrete-event systems. International Journal of Control, 54, 5, 1143–1169.

[26] Kozak, P. and Wonham, W. M. 1995. Fully decentralised solutions of supervisory control problems. IEEE Transactions on Automatic Control, 40, 12, 2094–2097.

[27] Yoo, T. S. and Lafortune, S. 2002. A general architecture for decentralized supervisory control of discrete-event systems. Discrete Event Dynamic Systems: Theory and Applications, 12, 335-377