We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

BOOK CITATION INDEX
CLARIVATE ANALYTICS
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Supervisory Control Systems: Theory and Industrial Applications

Hamdi Awad

Additional information is available at the end of the chapter

Abstract

Hybrid control system is an exciting field of research where it contains two distinct types of systems: one with continuous dynamics continuous variable dynamic system and the other with discrete dynamics discrete event dynamic system, that interact with each other. The research in the area of hybrid control can be categorized into two areas: one deals with the conventional control systems, and the other deals with the decision making systems. The former addresses the control functions at the low level (field level). The latter addresses the modeling, analysis, and design at the higher level found in the supervision, coordination and management levels. The study of hybrid systems is central in designing intelligent hybrid control systems with high degree of autonomy and it is essential in designing discrete event supervisory controllers for continuous systems.

Keywords: discrete event systems, supervisory control systems, petri nets, embedded systems, industrial processes

1. Introduction

In general automation systems' structure can be categorized into six levels: Sensor/Actuator Level, Machine/Controller Level, Process Automation Level, Operation Unit Level, Plant Level. Trends are making this structure possible and desirable to create streamlined three-level automation systems or even to collapse it [1]. This is due the trend to create embedded control systems, cyber physical systems, networking, and discrete hybrid control systems. The trend to reduce machine size and cost while increasing productivity using nanotechnology, requires new approaches to control systems [2]. Thanks to the increased reliability of industrial PC technology, traditional rack-based PLCs can be replaced with more powerful PC-based

control systems. While Industrial PC's provide the highest performance and control capacity, new generations of PC technology based on open embedded operating systems, combine the functions of a PLC and an operator panel in one unit which is applicable to smaller scale applications.

The development of complex man-made systems that perform complicated and interacted tasks, has been accompanied by an ever increasing demand for even more sophisticated modeling and control schemes. The need for a systematic and mathematical approach to analysis, design, and control of complex large scale systems is highly demanded. In fact, In operations research, for example, researchers have been interested for a long time in systematic methods to deal with large-scale systems [3–6]. However, control engineers have taken up this challenge in recent years to develop intelligent models for hybrid dynamical systems [7].

The study of hybrid systems is central in designing intelligent hybrid control systems with high degree of autonomy. Such systems include discrete and continuous activities [8]. The field of discrete vent systems (DES) is relatively a new research area that combines different formal-isms, methodologies and tools from Supervisory control theory, artificial intelligence (AI) and operations research (OR) [9]. The domains of DES are: manufacturing automation, communi-cation protocols, robotics, process control, nuclear reactors, space exploration systems, aircraft control systems, fault diagnosis, and refinery systems [5, 10, 11]. Historically, DES were intro-duced in the early 1980s, in the field of chemical engineering. They quickly gained popularity in modeling and supervision of hybrid systems [12, 13].

The design of supervisory controllers for discrete event systems has received considerable attention in research centers [14–16]. There are methods for designing supervisors based on automata models [17], however, they need exhaustive search over the system states that makes them impractical for systems with large number of states, as the number of states increases the state space explosion problem arises [18, 19]. One way of dealing with these problems is to model discrete event systems with Petri Nets (PNs). In this way the state explosion problem can be avoided. Some recent contributions on the Petri net based supervisory control can be found in [20, 21]. PN models are normally more compact compared with automata-based models and are better suited for the representation of discrete event systems due to its math-ematical manipulation and graphical representation [22–24]. Timed Petri nets are common used for industrial control systems [14].

The main objective of this chapter is to explore and step by step construct a supervisory control scheme in the field of DES modeling and control. It also shows how the continuous activities; temperature control, pressure control, etc. are represented by few places resided in the embed-ded PN models. These objectives can be achieved as follows.

1.1. Investigation of DES modeling and supervision

Types of events that may occur in discrete event systems are controllable, and uncontrollable events (sensors). The latter arises a severe problem when the plant works under control. It cannot be inhibited from firing by the supervisor. Embedded supervisors should be developed to deal with such problems.

1.2. Selection of the best modeling formalism

There are many tools to develop DES models e.g. finite state automata, and Petri nets. The latter has a good descriptive power compared with the former. Petri nets models are more compact than automata-based models for representing DES.

1.3. Developing supervisory control scheme

There are two types of supervisors, one is mapping supervisors, and the other is compiled supervisors. The latter has two notable features, its computational demand is small, and its structure can be reconfigured online. The developed supervisors should perform resource allocation, coordination, and deadlock avoidance tasks at the higher level of complex hybrid industrial systems. Ordinary and timed Petri nets are employed in this chapter to structure embedded supervisory control models for industrial processes.

1.4. Testing the proposed scheme

Chemical batch processes and kernel railroad crossing system were employed for testing the proposed control scheme; they have resources scarce, forbidden states, and deadlock problems. This chapter structures embedded supervisory control scheme that deals with the continuous activities at the lower level, as well as the discrete ones at the higher level in efficient manner. The continuous activities can be modeled and supervised using intelligent control schemes.

2. Supervisory control systems

This section gives an introduction to the hybrid nature of complex systems as well as their hierarchical structure. It also includes a brief overview of relative work in the area of DES supervision using finite state automata and Petri nets. A comparison between Petri nets and finite state automata as modeling formalisms for the purpose of supervisor synthesis is given in this section.

2.1. Investigation of DES modeling and supervision

Discrete event systems are useful when dealing with dynamic systems that are not fully modeled by classical models, such as differential or difference equations. It is noted that while differential and difference equation models evolve with time, a discrete event system evolves with the occurrence of events. An comprehensive literature on discrete event systems has appeared in the last 30 years, and their study continues to be an area of ongoing research. DED combines different formalisms, methodologies and tools from control theory (CT), artificial intelligence (AI), and operations research (OR) as shown in **Figure 1** [9]. These methods facilitate the modeling of continuous and discrete activities as a unified approach.
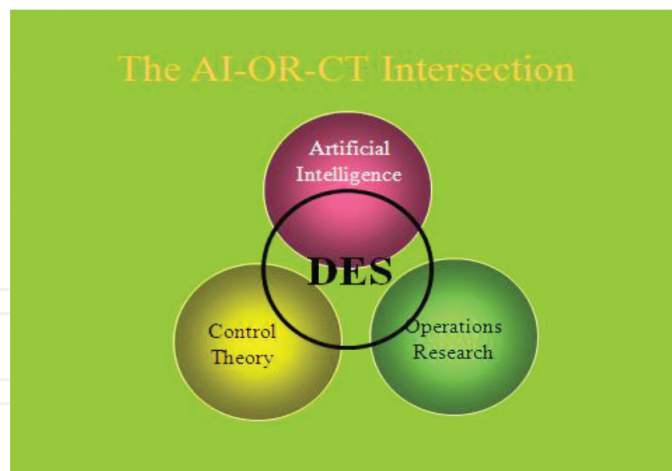
Figure 1. DES: the AI-OR-CT intersection.

System modeling is an important phase in the supervisory system synthesis procedure. There is a set of methods for designing supervisors based on automata system models [16, 17]. The disadvantage of these methods is that they need a huge search over the system states as mentioned in Section 1. One way of dealing with these problems is to model discrete event systems with Petri nets (PNs). Petri net-based solutions have several advantages over finite state automata. These advantages recommends the Petri nets to be used in this chapter. They are listed as follows.

• The states of a Petri net are represented by the possible markings and not by the places. Thus Petri nets give a more compact description.

• The plant and the specifications can be represented graphically in an easily understood format using Petri nets instead of using textual descriptions or mathematical notations, which are difficult to understand.

• Petri net models can be used for the analysis of their properties, performance evaluation and the systematic construction of discrete event supervisors [20, 22].

• The Petri net model allows for the simultaneous occurrence of multiple events.

2.2. Structuring a supervisory control scheme

This subsection shows the use of Petri nets to design a supervisor, including its synthesis techniques, methods of handling uncontrollable and unobservable transitions within the plant structure. The supervision based on place invariants method is employed in this chapter to build an embedded supervisory control systems. In this method, the control objective is to force the process to obey linear constraints in the form of linear inequalities. The ideas of developing such supervisors were borrowed from [21]. The developed supervisory control scheme can be employed to control the processes that have controllable and uncontrollable transitions [25]. Using an embedded Petri net structure, the developed scheme is easy to implement and its computational demand is relatively small. The design method has numerical properties that make it particularly appealing for large scale systems. Mathematically, the developed scheme can be detailed as follows.

A place invariant is an integer vector x that satisfies [21, 25]:

$$x^T \mu = x^T \mu_0 \tag{1}$$

for all reachable marking μ. Thus $x^T \mu$ is constant for all reachable states if x is a place invariant. Place invariants can be computed by finding solutions to Eq. (2).

$$x^T D = 0 \tag{2}$$

Based on the method of place invariants, it is possible to enforce a set of constraints on the plant state μ_p. The plant state is represented by an $n \times 1$ marking vector of non-negative integers, where each vector component is equal to the marking of the corresponding place in the Petri net model of the plant. The supervisory control goal is to restrict the reachable marking vectors of a plant μ_p as:

$$L\mu_p \leq B \tag{3}$$

where L is a $n_c \times n$ integer matrix ($L \in Z^{nc \times n}$), and B is a $n_c \times 1$ integer vector ($B \in Z^{nc}$), and n_c is the number of constraints. After adding the slack variables, constraint define in Eq. (3) becomes:

$$L\mu_p + \mu_c = B \tag{4}$$

Each place invariant defined in Eq. (6) must satisfy Eq. (2) such that:

$$LD_p + D_c = 0 \tag{5}$$

The matrix D_c contains the arcs that connect the controller places to the transitions of the process net. So, given the Petri net model of the process D_p and the constraints, the Petri net controller D_c is constructed see [25].

$$D_C = -LD_P \tag{6}$$

If the initial marking defined in Eq. (7) does not violate the given set of constraints, these constraints can be enforced by a supervisor with the incidence matrix D_c [21].

$$\mu_{C0} = B - L\mu_{p0} \tag{7}$$

where μ_{p0} is the $n \times 1$ initial plant marking vector of non-negative integers. The supervisor is a Petri net with incidence matrix D_c made up of the process net's transitions and a separate set of places. With the addition of supervisor places the overall system is given by.

$$D = \begin{bmatrix} D_p \\ D_C \end{bmatrix} \mu = \begin{bmatrix} \mu_p \\ \mu_C \end{bmatrix} \tag{8}$$

This method admits the structure of the process net as well as a set of specifications. This is because the constraints on these transitions is a subset of the specifications. Supervisors are used to insure that the behavior of the plant does not violate a set of constraints defined in Eq. (3) under a variety of operating conditions. Every single constraint is transformed to a marking

invariant that corresponds to a place invariant of the supervised system. The regulatory actions of the supervisor are based on observations of the plant state, resulting in feedback control. More details about the P-invariant-based supervisory control schemes can be found in [17, 21, 25].

2.3. Step by step developing P-invariant-based supervisory control scheme

The most common examples of hybrid systems are batch processes which are characterized by combination of discrete and continuous dynamics [17, 26]. Batch processes are currently used in the chemical and food process industries. A comprehensive model of batch systems has to include discrete event aspects as well as continuous ones. As a consequence, their automation and optimization pose difficult issues mainly because it is necessary to operate concurrently with continuous and discrete models.

Batch plants consist of many transport resources (*transporters*) like valves and pipes, and processing resources (*processors*) like mixing tanks, batch reactor vessels, and other container like units [27–29]. Transporters and processors are involved in transforming a batch from raw materials to final products.

The main problems inherent to the modeling and supervision of batch processes are:

- *The hybrid nature of the process.* State variables like the tank level or the pump speed are continuous, others like the on-off valves, are discrete-state components. Moreover, the whole process behaves as a cycle of discrete events.

- *The variety of knowledge.* Some elements of the process can be described by physical equations, e.g. the tank level, and others by experimental models, e.g. the behavior of the pump described by a transfer function.

The event driven part of a batch plant is modeled using Petri nets. The most appropriate method in batch plants modeling is the Petri net Bottom-up synthesis method [30, 31], where plants in process industries generally exhibit less flexible structure than manufacturing processes and the individual process units are well defined and standardized [29].

This section uses the bottom-up approach to Petri net modeling of the chemical batch processes. It also employs the P-invariant supervisory control scheme described in Section 2.2 to structure an embedded PN-supervisory control model of the batch process.

2.3.1. Example 1: a simple batch plant

This process used to illustrate the idea of supervisor design for the purpose of resource allocation for two process lines chemical process shown in **Figure 2**.

In this figure, two mixing tanks shared the same supply tank and only one tank can be filled at a time. Based on the receipt given in **Table 1**, using Petri net tool ver. 2.1 [33] the PN model of the two individual process lines shown in **Figure 2** is constructed as depicted in **Figure 3**. This model is structured using the bottom-up synthesis method [31].

In this sample, the places P_{m3} are corresponding to the outlet valve of the supply tank. The invariant-based supervision discussed in Section 2.2 is employed for the purpose of supervisor
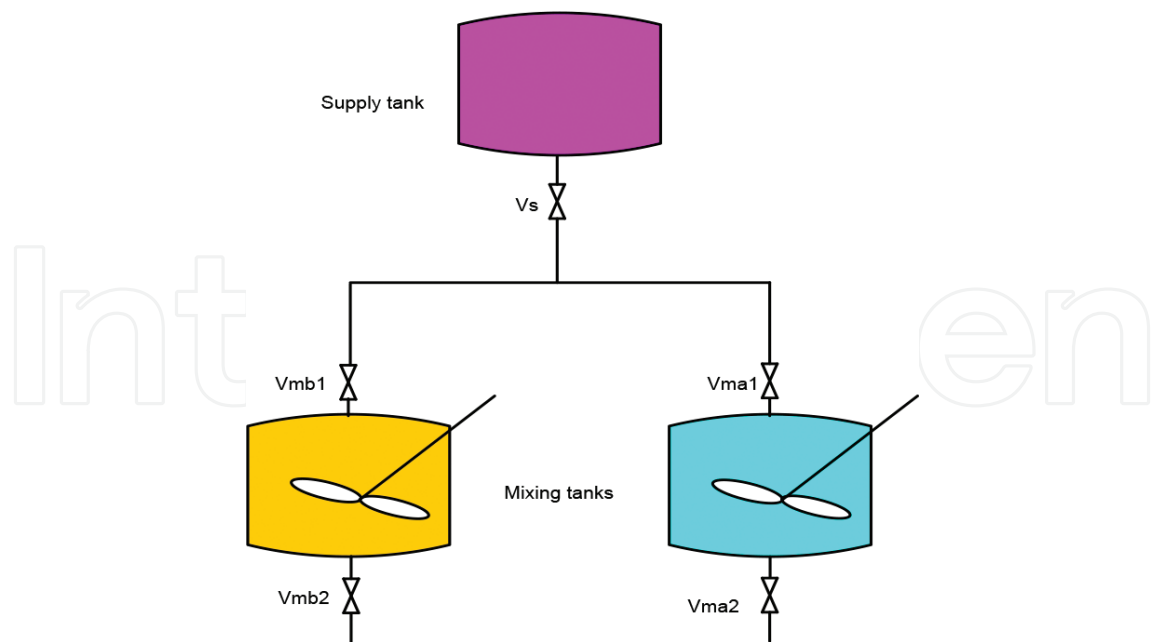
Figure 2. Batch process cell.

Place	Associated action
P_{m1}	Process ready
P_{m2}	Open the inlet valve of the mixing tank
P_{m3}	Open the valve of supply tank
P_{m4}	Stir the content of mixing tank
P_{m5}	Discharge the mixing tank (open the outlet valve)
Transition	Associated event
T_{m1}	Start a new batch
T_{m2}	Mixing tank is filled
T_{m3}	Duration of mixing operation is vanished
T_{m4}	Empty the mixing tank

Table 1. Places and transitions for each submodel of **Figure 3**.

synthesis. This is particularly interesting, because the resulting supervisory mechanism is computed efficiently. All transitions are assumed to be controllable.

It is clear that, if transitions T_{ma1} and T_{mb1} are fired, the place P_{m3} contains two tokens and therefore the Petri net model is not safe. The safeness of the place P_{m3} is required, because it represents the operation of opening and closing the outlet valve of the supply tank. The double-booking problem should be avoided in this case, otherwise, the situation is considered as a malfunction. To overcome this problem, the supervisor has to be designed to co-ordinate the two mixers in such a way that only one will be filled at a time. This requirement is written as:

$$\mu_{m3} \leq 1 \tag{9}$$

where μ_3 is the marking vector component that corresponds to place P_{m3}. The requirement can be easily transformed to the form Eq. (6) with the plant marking vector being:

$$\mu_p = \begin{bmatrix} \mu_{ma1} & \mu_{ma2} & \mu_{m3} & \mu_{ma4} & \mu_{ma5} & \mu_{mb1} & \mu_{mb2} & \mu_{mb4} & \mu_{mb5} \end{bmatrix}^T$$

$$\mu_{p0} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T, L = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \text{ and } B = 1.$$

Given the incident matrix of the PN model shown in **Figure 3**; D_p:

$$D_p = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

The supervisor can be computed by Eqs. (6) and (7) as follows:

$$D_c = -LD_p = \begin{bmatrix} -1 & 1 & 0 & 0 & -1 & 1 & 0 & 0 \end{bmatrix}, \mu_{C0} = B - L\mu_{p0} = 1 - 0 = 1.$$
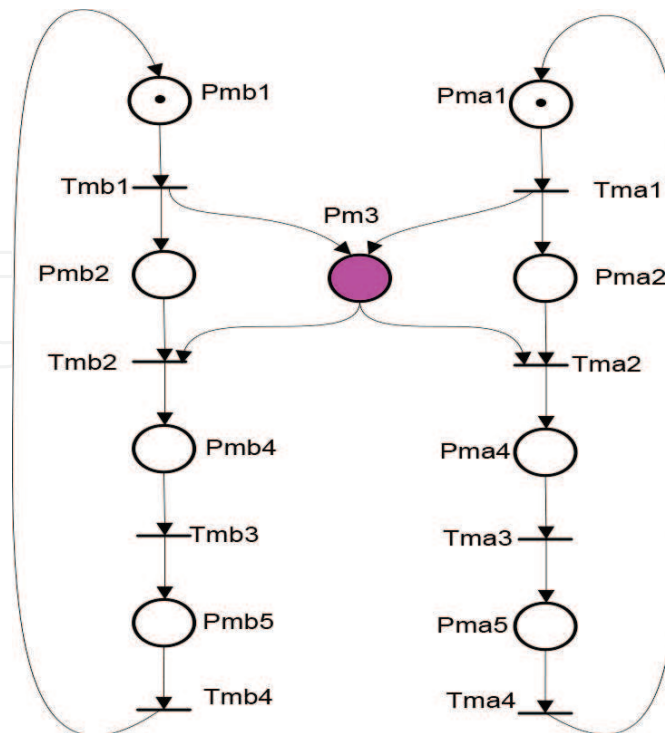

Figure 3. Petri net model of the overall system.

The supervisor consists of a single place that is connected to the plant Petri net as shown in **Figure 4**. The marking invariant that is enforced by the supervisor is:

$$\mu_{m3} + \mu_{c1} = 1 \tag{10}$$

The simulation of the unsupervised system via Reachability graph analysis method of PN indicates that the unsupervised plant has 16 reachable states, one of them indicates the absence of safety condition, this marking vector is: $\mu_4 = [0\ 1\ 2\ 0\ 0\ 0\ 1\ 0\ 0]^T$ i.e. $\mu_{m3} = 2$. On the other hand, the simulation of the supervised system indicates that the supervised plant has 15 reachable states. In this case, the supervisor eliminates the marking μ_4 which is forbidden state, and all the reachable states satisfy the safety condition. This procedure can be generalized for more complex batch processes such as coordination, deadlock avoidance, and resource allocation discussed in [17].

Quiz 1: With the help of our work in [14], can you model the chemical batch process shown in **Figure 2** using timed Petri nets?

2.3.2. Control of continuous activities as a part of hybrid systems

The main objective of this subsection is to control the continuous part of a complex batch process shown in **Figure 5**. This process comprises six input buffers, two mixing tanks and two reactor vessels. In this case, heating and cooling are continuous variables of this batch process. The preparation of the input substances takes place in two mixing tanks to which the raw materials are supplied from three supply tanks (buffers). The substance is composed from one of the two basic components (component 'a' or component 'b') that is diluted to the required concentration by component 'c'. The filling of the mixing tank is controlled by the on/off valve Vma in
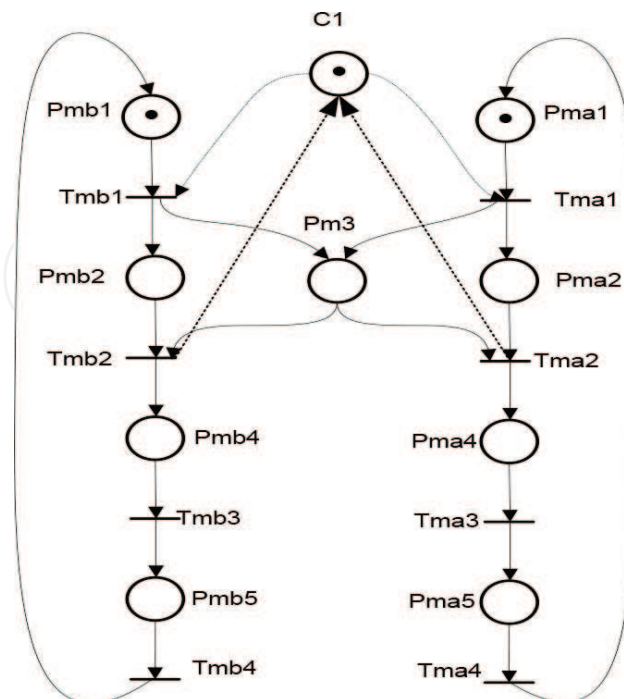

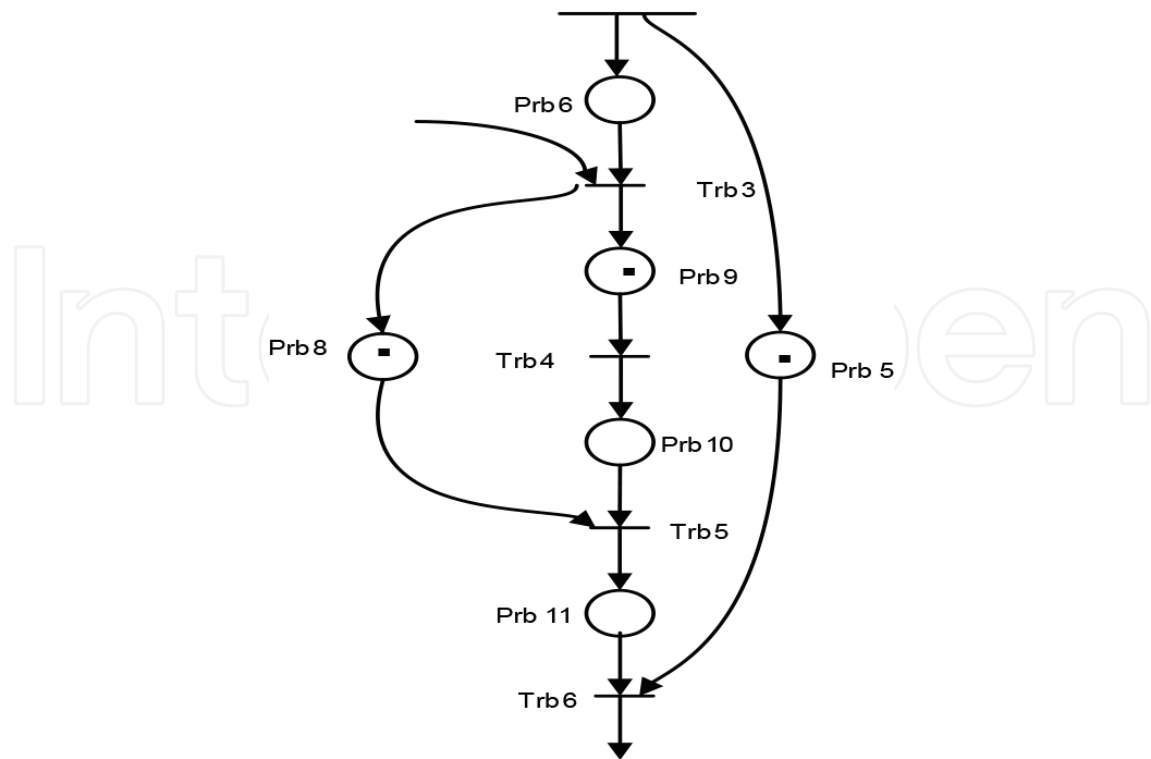**Figure 4.** Petri net model of the supervised system.

Figure 5. Batch process cell with two mixers and two reactors.

combination by one of the supply tank valves Vsa, Vsb, or Vsc. The discharging of the mixer is controlled by the on/off valve Vmb. The level in the mixing tank is measured by the level sensor; L. The required quantities of each input depend on the recipe detailed in [17].

The mathematical differential equations of the heat system of the reactor is detailed in [22]. In this case, the temperature of the reaction for each reactor is controlled using fuzzy neural systems-based local controllers [32] that are designed for each of heating and cooling phases in each process line. It is controlled by feeding hot or cold glycol through the reactor jacket, which surrounds the reactor vessel.

Using the P-invariant supervisory control method discussed in Section 2.2, the embedded PN-based supervisory control model is structured using Petri net tool ver. 2.1 [33]. A part of this embedded model is shown in **Figure 6**. The main objective of this subsection is to show how can activate the continuous activities resided inside the embedded model of the process. It also shows the control result of heating phase as a set of continuous places resided in the embedded model using fuzzy neural controller. The full simulation results can be found in our work [17]. To show the firing of the continuous activity resided in the embedded model, consider the initial marking vector, μ_0 defined below.

$$\mu_0 = [0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1]^T$$

Starting from this initial marking vector, the simulation of a part of the embedded PN-based model shown in **Figure 6** [17], can be carried out. When the transition T_{rb3} is fired, the fuzzy neural controller resided in place 8; $Prb8$, is activated as shown in **Figure 6** and the marking vector becomes, μ_{HR}.

$$\mu_{HR} = [1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,1\,0\,0\,0\,0\,0\,1\,1\,1\,1]^T$$

The response of the reaction at the desired set point 50°C is depicted in **Figure 7**.

Figure 6. A part of the embedded PN model of batch process shown in **Figure 5**.
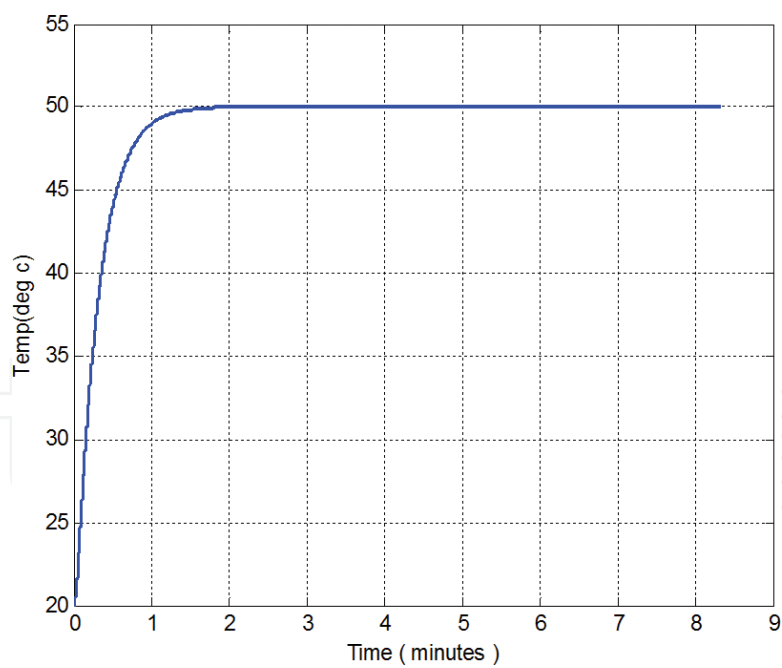

Figure 7. Reactor temperature for the heating phase.

2.3.3. Example 3: KRC modeling and supervision using timed PNs

The kernel railroad crossing (KRC) shown in **Figure 8** is a standard benchmark in real time systems [14]. When a train is sensed to approach the crossing, a signal is sent to the supervisor

that sends a command to the specified gate that is closed to prevent cars crossing that survives ourselves. Having more than one track and more than one train may enter crossing zone, leads to a complicated situation that is out of our paper scope. For simplicity let us merge the two depicted zones as one zone (region). The Petri net of the system is depicted in **Figure 9**. The train needs one time unit (t.u.) to enter the R-Y zoon and five (t.u.) to leave it for departure phase. The gate needs no time to start closing and requires two (t.u.) to be completely closed. It needs extra two (t.u.) to be completely opened after firing the transition T_5. The problem arises
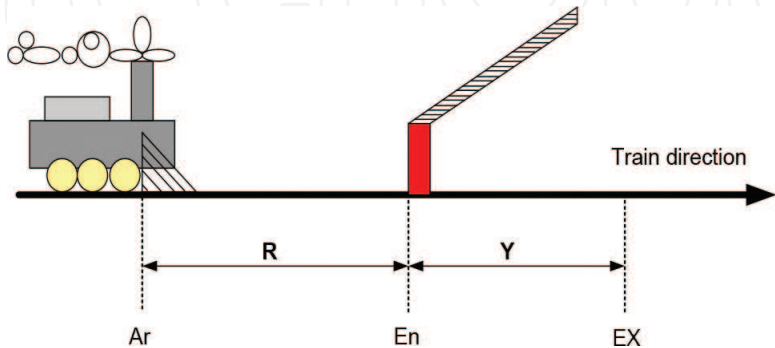
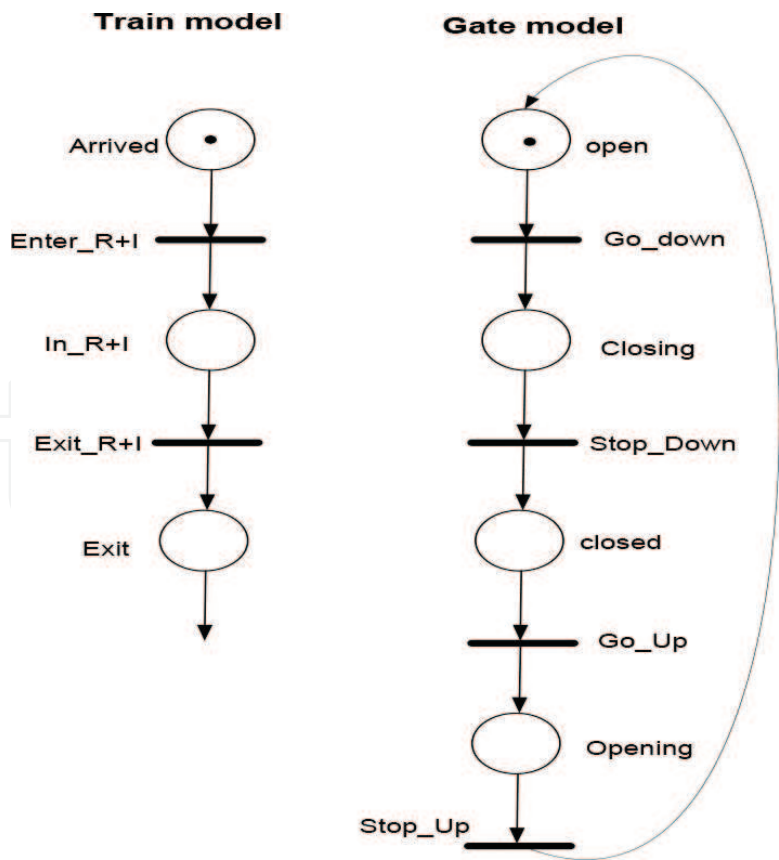

Figure 8. A kernel railroad crossing system.

Figure 9. The simplified Petri net model of **Figure 8**.

at the beginning of opening the gate that has unknown time units "[0, ∞]". This is due to the reason that no expectation for beginning the departure phase of the train; its departure depends on the passenger riding. The analysis should be performed to get the exact period time unit required to activate the transition T_5 every departure phase. This means that this transition is continuously evaluated. The system comprises two tasks, train task and gate task. In our work [14], consider that the controllable events are the beginning of each task. However, the accomplishing of the tasks is uncontrollable. Therefore, our goal was to control the beginning of the tasks in order to obtain safe arrival and departure of the train. Also, the synthesized control scheme should avoid the forbidden state (P_2, P_4). This means that the train in the R-Y zone and the gate still opened.

Using Petri net tool software ver. 2.1 [33], the system incidence matrix is of the PN model shown in **Figure 9** is:

$$D_P = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

In this simulation, there are 11 reachable states starting from the initial marking vector μ_0 to the final vector μ_{11}. The firing sequence shows that the marking vector, $\mu_1 = [0\,1\,0\,1\,0\,0\,0]^T$ is a forbidden state. It is clear that the marking vector μ_1 includes the forbidden state (P_2, P_4). Based on P-invariant, a supervisory control scheme for the KRC system is synthesized in Section 2.2. The constraints vector is $L = [0\ 1\ 0\ 1\ 0\ 0\ 0]$, the controller incidence matrix is $D_c = -LD_p = [-1\ 1\ 1\ 0\ 0\ -1]$, the initial marking of the controller is $\mu_{c0} = B - L\mu_0 = 0$, and $B = 1$. The developed supervised time Petri net of the KRC system is depicted in **Figure 10**. There are 10 reachable states starting from the initial marking vector μ_0 to the final vector μ_{10} and the supervisor eliminates the forbidden state vector $\mu_1 = [0\,1\,0\,1\,0\,0\,0]^T$.

Another issue, in distributed hybrid systems, each process line is controlled by its own logic controller and supervisory part resides in another level. The interaction among different modules is performed through synchronized transitions. In practical implementation, it is difficult to achieve such synchronization among the logic controllers of the embedded systems. Because of the communication delays, it cannot be guaranteed that transitions in different controllers fire simultaneously. One way for dealing with this problem is to define the firing order of the transitions. In the cases when the two logic controllers share the same resource and the supervisor performs the resource allocation such as indicated in **Figure 4**, the transition that reserve the resource must be fired in the supervisor first and then in the local controller. In the opposite case the communication delay would allow double booking problem of the shared resource by the two controllers or even deadlock. Due to the pages limitation, more details about this implementation problem the readers can be directed to read our work detailed in [34].
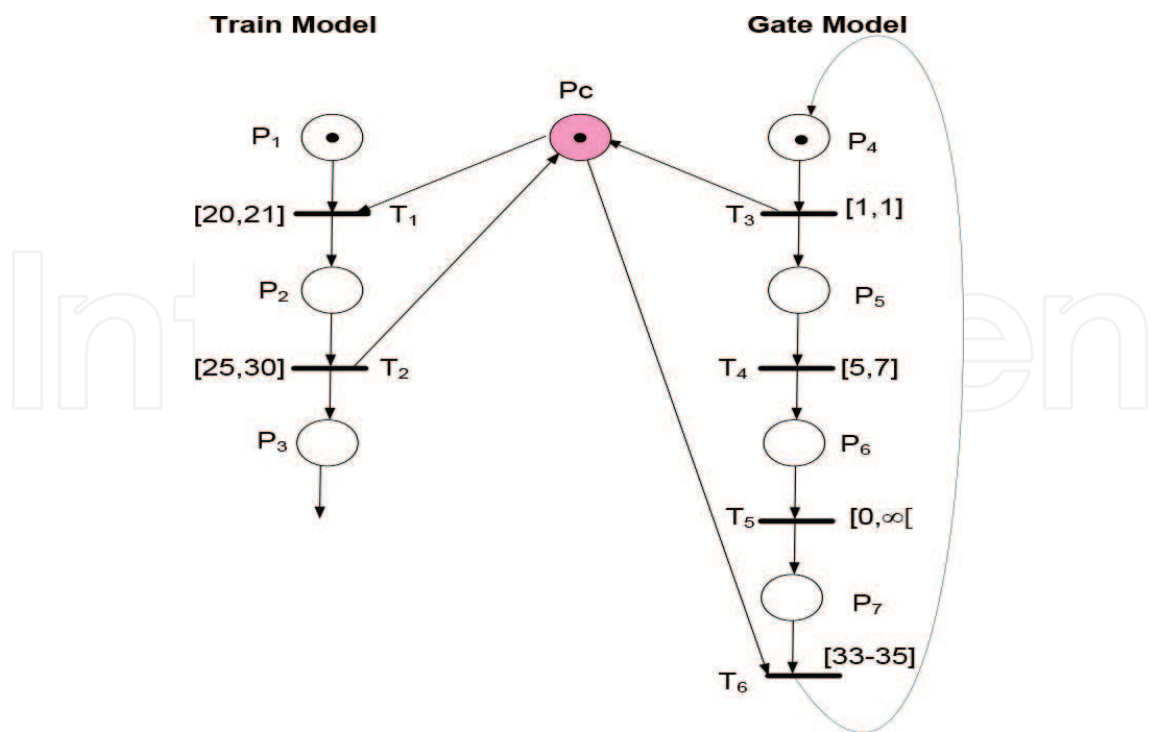
Figure 10. The supervised timed Petri net of the KRC (**Figure 8**).

Quiz 2: With the help of our work in [17], can the reader complete the missing part of the supervised embedded PN model depicted in **Figure 6** using P-invariant supervisory control discussed in Section 2.2?

Quiz 3: With the help of our work in [34], can you overcome the communication problems through the embedded PN model of Quiz 2?

3. Conclusions

Hybrid systems modeling and supervision have been used extensively in automation, robotics, and manufacturing applications. Different frameworks for dynamic supervisory controllers are used in flexible manufacturing systems and automated batch processes. The high-level system changes in hybrid systems are modeled as discrete event dynamic systems, while the low-level systems changes are modeled as continuous variable dynamic systems. The major issue in studying hybrid systems is the consistency between continuous and discrete models evolution. Petri nets possess many assets as models for DES. They provide more compact representation for larger reachable state spaces, and increase the behavioral complexity compared with automata-based models. Batch plants are common examples of hybrid systems. In this chapter Petri net embedded models were developed by abstracting the behavior of hybrid systems. As a final conclusion, the work in this chapter allows the readers to design and analysis their own supervisory control schemes using Petri net tools ver. 2.1 or higher [33]. Although the developed schemes are tested using batch chemical

processes, they are promising to control complex industrial automated processes. This chapter opens several research directions to be considered and investigated. It may extend this work to optimization of supervisory control schemes, modeling and supervision of hybrid industrial systems using timed Petri nets, and implementing the proposed schemes for large scale systems.

Author details

Hamdi Awad[1,2]

Address all correspondence to: awadhaa@yahoo.co.uk

1 College of Engineering, Shaqra University, Dawadmi, KSA

2 Faculty of Electronic Engineering, Menoufia University, Egypt

References

[1] Strasser T, Froschauer R. Autonomous application recovery in distributed intelligent automation and control systems. Intelligent Automation and Control Systems, IEEE Transactions on Systems Man and Cybernetics Part C (Applications and Reviews). 2012;**42**:1054-1070. DOI: 10.1109/TSMCC.2012.2185928

[2] Lu K. Nanoparticulate Materials: Synthesis, Characterization, and Processing. USA: John Wiley & Sons, Inc.; 2013. p. 464, ISBN: 978-1-118-29142-9. DOI: 10.1002/9781118408995

[3] Lemmon MD, He KX, Markovsky I. A tutorial introduction to supervisory hybrid systems. Technical Report of the ISIS Group at the University of Notre Dame ISIS-98-004; 1998

[4] Antsaklis PJ. Intelligent control. In: Encyclopedia of Electrical and Electronics Engineering. Vol. 10. USA: John Wiley & Sons, Inc.; 1999. pp. 493-503. DOI: 10.1002/047134608X.W1019

[5] Böhme TJ, Frank B. Hybrid Systems, Optimal Control and Hybrid Vehicles: Theory, Methods and Applications. Berlin: Springer Nature; 2017, 519 p. DOI: 10.1007/978-3-319-51317-1

[6] Jalivand A, Khanmohammadi S. Integrating of event detection and mode recognition in hybrid systems by fuzzy Petri nets. In: Proceeding of the IEEE Conference on Robotic, Automation, and Mechatronics; 1–3 December 2004; Singapore. pp. 265-270

[7] Giua A, Silva M. Modeling, analysis and control of discrete event systems: A Petri net perspective. IFAC-PapersOnLine. 2017;**50**:1772-1783. DOI: 10.1016/j.ifacol.2017.08.156

[8] Chen CH, Dai JH. Design and high-level Synthesis of hybrid controller. In: Proceedings of the IEEE International Conference on Networking, Sensing ,and Control; 21–23 March 2004; Taipei, Taiwan. pp. 433-438. DOI: 10.1109/ICNSC.2004.1297477

[9] Ho Y-C. A road map for DEDS research. IFAC Proceedings Volumes. 1993;**26**:663-669. DOI: 10.1016/S1474-6670(17)49211-1

[10] Villani E, Miyagi PE, Valette R. Landing system verification based on Petri nets and a hybrid approach. IEEE Transactions on Aerospace and Electronic Systems. 2006;**42**:1420-1436. DOI: 10.1109/TAES.2006.314582

[11] Sobh TM, Benhabib B. Discrete event and hybrid systems in robotics and automation: An overview. IEEE Robotics and Automation Magazine. 1997;**4**:16-19. DOI: 10.1109/100.591642

[12] Cohn G, Cnet PM, Quadrat JP. Linear system theory for discrete event systems. In: Proceedings of the 23rd IEEE Conference on Decision and Control, Las Vegas; 12–14 December 1984; USA. p. C1. DOI: 10.1109/CDC.1984.272058

[13] Silva M, Recalde L. On fluidification of Petri nets: From discrete hybrid and continuous models. Annual Reviews in Control. 2004;**28**:253-266. DOI: 10.1016/j.arcontrol.2004.05.002

[14] Allaa H, Awad H, Anwar A, El-Hajri E. Modelling and control of KRC systems using TPN and TA. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, USA. 2015;**4**: ISSN (Online): 2278-8875:1111-1130

[15] Hopcroft JE, Ullman JD. Introduction to Automata Theory, Languages, and Computation. 3rd ed.. Addison-Wesley series in Computer Science, ISBN 0-201-02988-X, 530 p ed1979

[16] Ramadge PJ, Wonham WM. The control of discrete event systems. Proceedings of the IEEE. 1989;**77**:81-97. DOI: 10.1109/5.21072

[17] Anwar A. Modelling and supervision of discrete event systems [thesis]. Egypt: Menoufia University; 2007

[18] Giua A, DiCesare F. Petri net structural analysis for supervisory control. IEEE Transactions on Robotics and Automation. 1994;**10**:185-195. DOI: 10.1109/70.282543

[19] Music G, Matko D. Petri net based control of a modular production system. In: Proceedings of the IEEE International Symposium on Industrial Electronics, Vol. 3; 12–16 July 1999; Bled, Slovenia. pp. 1383-1388. DOI: 10.1109/ISIE.1999.796909

[20] Wang S, Wang C, Zhou M. Design of optimal monitor-based supervisors for a class of Petri nets with uncontrollable transitions. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2013;**43**:1248-1255. DOI: 10.1109/TSMC.2012.2235427

[21] Iordache MV, Antsaklis PJ. Supervision based on place invariants: A survey. Technical Report of the ISIS Group at the University of Notre Dame ISIS-2004-003; July, 2004

[22] Murata T. Petri nets: Properties, analysis and applications. Proceedings of the IEEE. 1989;**77**:541-580. DOI: 10.1109/5.24143

[23] David R, Alla H. Petri nets for modeling of dynamic systems: A survey. Automatica. 1994;**30**:175-202. DOI: 10.1016/0005-1098(94)90024-8

[24] Cheung KS, Cheung TY, Chow KO. A Petri-net-based synthesis methodology for se-case-driven system design. Journal of Systems and Software. 2006;**79**:772-790. DOI: 10.1016/j.jss.2005.06.018

[25] Moody JO, Antsaklis PJ. Petri net supervisors for DES with uncontrollable and unobservable transitions. IEEE Transactions on Automatic Control. 2000;**45**:462-476. DOI: 10.1109/9.847725

[26] Lennartson B, Egardt B, Tittu M. Hybrid systems in process control. In: Proceedings of 33rd Conference on Decision and Control; 14–16 December 1994; Lake Buena, USA. IEEE. pp. 3587-3592. DOI. 10.1109/CDC.1994.411706

[27] Tittus M, Lennartson B. Hierarchical supervisory control for batch processes. IEEE Transactions on Control Systems Technology. 1999;**7**:542-554. DOI: 10.1109/87.784418

[28] Falkman P, Lennartson B, Tittus M. Specification of a batch plant using process algebra and Petri nets. In: Proceedings of the IEEE International Conference on Automation Science and Engineering; 1–2 August 2005; Edmonton, Canada. pp. 339-344. DOI: 10.1109/COASE.2005.1506792

[29] Music G, Matko D. Petri net based supervisory control of flexible batch plant. IFAC Proceedings Volumes. 15–17 July 1998;**31**:941-946. DOI: 10.1016/S1474-6670(17)41919-7

[30] Luo J, Zhou M. Petri-net controller synthesis for partially controllable and observable discrete event systems. IEEE Transactions on Automatic Control. 2017;**62**:1301-1313. DOI: 10.1109/TAC.2016.2586604

[31] Wang S, You D, Wang C. Optimal supervisor synthesis for Petri nets with uncontrollable transitions: A bottom-up algorithm. Information Sciences. 2016;**363**:261-273. DOI: 10.1016/j.ins.2015.11.003

[32] Awad H. Fuzzy neural networks for modelling and controlling dynamic Systems [thesis]. Wales, England: Cardiff University; 2001

[33] Petri Net Toolbox for Matlab. Department of Automatic Control and Industrial Informatics of the Technical University "Gh. Asachi" of Iasi, Romania. Available from: http://www.ac.tuiasi.ro/pntool/ [Accessed: May 2007]

[34] Gomaa M, Awad H, Anwar A. Design and implementation of supervisory control schemes in industrial automation systems. In: Proceedings of the IEEE International Conference of Computer Engineering & Systems, ICCES'08; 25–27 November 2008; Cairo, Egypt. pp. 398-404. DOI: 10.1109/ICCES.2008.4773035