

MODELLING OF INDUSTRIAL PRODUCTIVITY PROCESSES

Hamdi A. Awad

*Dept. of Industrial Electronics and Control Eng., Faculty of Electronic Eng.,
Menouf, 32952, Menoufia University, Egypt*

Awadhaa@yahoo.co.uk

(Received January 2, 2010 Accepted February 24, 2010).

Compared with fault detection and isolation (FDI) issues modeling of an industrial productivity process (IPP) is more important issue due to the fact that FDI schemes usually depend on the developed process models. IPP is always modeled by Grafcet that represents a process model with safe or discrete modules. However, hybrid systems are discrete, continuous, and actually are unsafe. Unlike Grafcet, Petri nets are mathematical approaches that combine a well defined mathematical theory with a graphical representation of the systems' dynamic behaviors. The theoretic aspect of Petri nets allow precise modeling and analysis of the system behavior, while the graphical representation of Petri nets (PNs) enable visualization of the changes of the system states. According to these reasons, modeling, analyzing, and verifying of the IPP using Petri nets are valuable issues to be addressed in this paper. Introducing Petri nets as graphical representation tool for modeling and supervision of the IPP instead of Grafcet is the main contribution of this paper. This contribution can be achieved as follows. First, the required information is collected from the expertise and the available catalogues of the IPP to be modeled. Second, the process will be modeled using Petri nets based on the collected information. Third, the PN-modules of the IPP and its net PN-model should be verified and validated using MATLAB-based PN-tool and P-invariant concepts. Simulation results show that the proposed PN formalism is promising for modeling and validating of the IPP at real time.

KEYWORDS: *Modeling formalism, Hybrid systems, Petri nets, Grafcet, IPP*

1- INTRODUCTION

We would like to zoom into industrial production processes (IPPs), and focus on their modeling issues to enhance and optimize the operation and reduce number of upsets of these plants. In such station there are three primary pumps stations which receive stabilized crude oil from different stabilizers and the output of these shipping pumps flows to the main shuttle lines. From the shuttle lines the oil can be diverted to different production pipelines. In addition, the flow control valves FCV's control the flow amount in the shuttle line. Unfortunately, the shuttle line pressure and flow can be disturbed for reasons related to operating problems, such as pump vibrations and high bearing temperature which may cause damage. In this case the Emergency shutdown system (ESD) would isolate the effected pump. Using discrete event systems concepts

and Petri net tools, modeling of this industrial process is important issue to enhance its behavior and mass production.

As the scope of control theory is being extended into the domains of manufacturing automation, robotics, computer and communication networks, and many other complex man-made technological systems, there is an increasing demand for different models, capable of describing these systems [1]. Industrial production processes are commonly classified as hybrid processes, where they contain two distinct types of systems that interact with each other: systems with continuous dynamics and systems with discrete dynamics [2-3]. Discrete event system models are useful when dealing with dynamic systems that are not fully modeled by classical models, such as differential or difference equations. Hybrid processes have control levels. The basic control level (local control) deals with processes that are usually continuous in nature. The basic control level is therefore superimposed by higher levels of the control hierarchy, ranging from supervisory control, production planning to business process management which is discrete in nature. While differential and difference equation models evolve with time, a discrete event system evolves with the occurrence of events. It is common to see discrete event systems modeled as finite automata [4]. Finite automata models are impractical for systems with large number of states. Huge states of the system to be modeled, results the state space explosion problem.

One way of dealing with these problems is to model discrete event systems using Petri nets. In this method, the state explosion problem can be avoided. Petri net models are normally more compact than similar automata based models and are better suited for the representation of discrete event systems [5-6]. Some of the common successful application area of Petri nets in discrete event modeling and control are flexible manufacturing [7], industrial automation and robotics [8], and batch chemical processes [9]. Numerous approaches to the systematic construction of Petri net models have been proposed. These approaches can be broadly classified into Bottom-up, Top-down, and Hybrid approaches.

Grafset inherits many of its features from the theory of Petri nets [10]. It is a special case of Petri nets. It represents the processes that are safe or binary in nature. However, hybrid systems are discrete, continuous, and actually are non safe. Unlike Grafset, Petri nets, are mathematical approaches. They combine a well defined mathematical theory with a graphical representation of the dynamic system behavior. The theoretic aspect of Petri nets allow precise modeling and analysis of a system behavior, while the graphical representation of Petri nets enable visualization of the changes of the system states [11]. This combination is the main reason for the great success of Petri nets that have been used to model various kinds of dynamic event-driven systems. Many Petri net-based software modules are published recently [12]. They can be summarised as: Allen Bradley, Mitsubishi, Shneider Automation Group (AG), Omron, Siemens, and ABB Automation. However, Siemens modules are common used in oil producing plants.

Introducing Petri nets instead of Grafset for modeling the IPP is the main contribution of this paper. This main contribution can be achieved by:

1. Formulating the collected information about the IPP to a suitable recipe
2. Modelling the IPP using Petri nets based on the described recipe
3. Verifying and validating the IPP modules using the MATLAB

4. Integrating the developed PN-modules into one sequential model that represent the IPP

This paper is organized as follows: In section 2, the essential background of Petri nets used in this paper are summarized. The modelling algorithm is detailed in Section 3. Section 4 describes some units of the IPP to be modeled. The developed Petri net modules of the IPP are detailed in section 5. Section 6 describes the proposed analyses, and validation methodologies of the developed modules. Section 7 gives the simulation results obtained. Section 8 concludes the topics issued through the paper and suggests points for future work.

2. THE ESSENTIAL BACKGROUND

Petri nets can be analyzed using the reachability graph or the coverability graph [13] and [14]. This paper addressed the required properties that are employed for modeling purposes. Petri nets are classified into many classes based on structural and behavioral properties involving the ways arcs connect places and transitions. These classes can be summarized as follows.

Definition-1: Petri net graph

A Petri net graph (or structure) is a weighted bipartite graph $N = (P, T, F, W)$ where:

$P = \{p_1, p_2, \dots, p_n\}$ is a finite set of places represented by circles, $|P| = n$.

$T = \{t_1, t_2, \dots, t_m\}$ is a finite set of transitions represented by bars, $|T| = m$.

$F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs from places to transitions and from transitions to places in the graph.

$W : I \cup O \rightarrow \{1, 2, 3 \dots\}$ is a weight function on the arcs.

Definition-2: Petri net marking

The marking function $M : P \rightarrow Z^+$ represents the number of tokens (depicted as dots) residing inside each place. The marking vector of a PN is usually expressed as an n-entry vector $M = \{m_{p1}, m_{p2}, \dots, m_{pn}\}$. Where Z^+ are the non negative integers.

Definition-3: Enabled transition

In a PN system, a transition $t_j \in T$ in a Petri net is said to be enabled if $m_k(p_i) \geq w(p_i, t_j)$ for all $p_i \in I(t_j)$. An enabled transition t_j can be fired reaching a new marking M_{k+1} which can be computed as (PN state equation):

$$M_{k+1} = M_k + C q_k(i) \tag{1}$$

$$y_{k+1} = \Phi M_{k+1} \tag{2}$$

where $q_k(i) = 0, i \neq j, q_k(j) = 1, q$ is the firing vector, C is the incident matrix, Φ is the output matrix, and y is the output of the PN net, and k is the time step.

Definition-4: Incidence matrix

The architecture or layout of a Petri net can be represented with an integer matrix known as the incidence matrix. The incidence matrix C of a Petri net is an $n \times m$ matrix whose (i, j) entry is of the form $d_{ij} = w(t_j, p_i) - w(p_i, t_j)$. The incidence matrix is useful in studying the reachability problem.

Definition-5: Firing sequences

A firing sequence from M_0 is a sequence of transitions $\sigma = t_1 \dots t_k$ such that $M_0[t_1]M_1[t_2]M_2 \dots [t_k]M_k$, where $M_0[\sigma]M_k$ denotes that σ may be fired at M_0 yielding M_k .

Definition-6: Pure and ordinary Petri nets

A pair of a place p and a transition t is called a self-loop if p is both an input and output place of t . A Petri net is said to be pure if it has no self-loop. A Petri net is said to be ordinary if all of its arc weights are 1's.

Definition-7: A state machine

A State Machine (SM) is a Petri net in which all transitions have one input and one output place, i.e.,

$$\left| \bullet t \right| = \left| t \bullet \right| = 1 \quad \text{for all } t \in T \quad (3)$$

The significant nodes in a state machine are the places. Each transition allows tokens to flow from one place to another, but a token in a particular place may enable multiple transitions. *The SM Petri net has no concurrency or synchronization but parallelism.*

Definition-8: A marked graph

A Marked Graph (MG) is a Petri net in which all places have a single input and a single output transition, i.e.,

$$\left| \bullet p \right| = \left| p \bullet \right| = 1 \quad \text{for all } p \in P \quad (4)$$

The significant nodes in a marked graph are the transitions. Each place receives tokens from one transition and loses tokens to another, but a single transition may have multiple input and output places. *The MG Petri net has no conflict or parallelism but synchronization*

Definition-9: A free choice net

A Free Choice (FC) net is a Petri net such that for every arc from a place p to a transition t ($p \rightarrow t$), where t is the only output transition of p (no conflict), or p is the only input place of t . *The FC Petri net allows conflict and synchronization.*

Definition-10: An extended free choice

An Extended Free Choice (EFC) net is a Petri net such that for every arc ($p \rightarrow t$) there exists an arc from all input places of t to all output transitions of p . Using the "bullet" notation, for all pairs of places, p_1 and p_2 , in an EFC net,

$$p_1 \bullet \cap p_2 \bullet \neq \emptyset \Rightarrow p_1 \bullet = p_2 \bullet \quad (\text{symmetric choice}).$$

Definition-11: An asymmetric choice

An asymmetric choice net is a Petri net such that for all pairs of places, p_1 and p_2 ,

$$p_2 \bullet \subseteq p_1 \bullet \quad \text{or} \quad p_1 \bullet \cap p_2 \bullet \neq \emptyset \Rightarrow p_1 \bullet \subseteq p_2 \bullet.$$

Definition-12: (liveness)

A Petri net (N, M_0) is said to be live if, no matter what marking has been reached from the initial marking M_0 , it is possible to ultimately fire any transition of the net N defined in definition-1 by progressing through some further firing sequence. This

means that a live Petri net guarantees deadlock-free operation, no matter what firing sequence is chosen. Liveness is an ideal property for many systems. However, it is impractical and too costly to verify this property for some systems such as batch chemical processes.

Definition-13: Reachable states

A marking μ_n is said to be reachable from a marking μ_0 if there exists a sequence of firings that transforms μ_0 to μ_n . $\sigma = t_1 \mu_0 t_2 \mu_2 \dots t_n \mu_n$: a firing or occurrence sequence or simply $\sigma = t_1 t_2 \dots t_n$. $\mu_0 [\sigma > \mu_n$: μ_n is reachable from μ_0 by σ . The set of reachable markings is denoted $R(G, M_0)$.

Definition-14: Some of the structural properties of Petri nets, properties that depend only on the topological structure; the incident matrix C of the Petri net and not on the net's initial marking, are the net invariants. Invariants are important means for analyzing Petri nets since they allow for the net's structure to be investigated independently of any dynamic process. These invariants are Transition Invariant (T-invariant) and Place Invariant (P-invariant) respectively. The latter important structural property will be used to design the supervisor of the developed PN-modules of the IPP as will be discussed in section 6.

Definition-15: (boundedness, safety)

A Petri net (N, μ_0) is said to be k -bounded or simply bounded if the number of tokens in each place does not exceed a finite number k for any marking reachable from μ_0 , i.e., $\mu(p_i) \leq k$ for every place p and every marking $\mu \in R(\mu_0)$. A Petri net is said to be safe if it is 1-bounded.

Definition-16: (The interpreted Petri net)

The interpreted Petri net (IPN) structures with $Q=(N,\Sigma,\lambda,\varphi)$ and an initial marking M_0 . Where N is a PN structure defined above, $\Sigma = \{\alpha_1,\alpha_2,\dots,\alpha_r\}$ is the alphabet of input symbols α_i , $\lambda : T \rightarrow \Sigma \cup \{\varepsilon\}$ is a labeling function of transitions with the following constraint: $\forall t_j, t_k \in T, j \neq k, \text{ if } \forall p_i I(p_i, t_j) = I(p_i, t_k) \neq 0$ and both $\lambda(t_j) \neq \varepsilon, \lambda(t_k) \neq \varepsilon$, then $\lambda(t_j) \neq \lambda(t_k)$; ε represents a system internal event, And $\varphi : R(Q, M_0) \rightarrow (Z^+)^q$ is an output function, that associates to each marking in $R(Q, M_0)$ q -entry output vector; q is the number of outputs.

3. MODELING OF INDUSTRIAL SYSTEMS USING PETRI NETS

In recent years, a large number of hybrid system modeling formalisms have been proposed [15]. These modeling formalisms can be grouped into three classes [16]. The first one comprises the extensions of continuous formalisms by the introduction of discrete variables, such as ordinary differential equations with Boolean variables. The second class comprehends discrete formalisms where new elements are introduced for representing the continuous dynamic, such as hybrid Petri net. It incorporates

continuous places and transitions in order to model the dynamics of continuous flows. The third class of formalisms combines a continuous formalism, described by differential equation systems, with discrete ones, such as Petri nets or automata. Among the approaches of these groups, the formalisms derived from Petri nets are particularly considered because of their well-known power for representing process features such as concurrency, conflict, ... etc.

3.1. Model Classification

Mainly there are two models for discrete event systems (DEDS) [17], logical discrete event model (DEM), and timed DEM. At the DEM, a common simplifying assumption is to ignore the times of occurrence of the events and consider only the order in which they occur. This simplification is justified when the model is to be used to study properties of the event dynamics that are independent of specific timing assumptions. The timed DEM is intended for the study of properties explicitly dependent on intervened timing. These models can be further classified as: Deterministic: if the timing is priori known, and stochastic: if the timing is not priori known due to random delays or random occurrences of events. In this paper, DEM is employed for modeling IPP and Petri nets [6] as discrete transition system descriptions are used.

3.2. The modeling methodology

The modeling methodology proposed in [13] is borrowed to model a logical DEM of the IPP in this paper. The methodology follows a modular bottom-up strategy. After identifying the system components, a set of state variables is assigned to every component, each state variable behavior is modeled by an interpreted Petri net model defined in section2- (Definition-16), herein named module. Then the set of modules are merged into a single model according to the appropriate relationships achieved through two module composition operations. This methodology builds binary interpreted-PN (IPN) modules to represent the behavior of each component of the identified discrete event system (DES) and the relationships between them. The model captures the normal and faulty behavior of the individual components of the system. The methodology can be detailed as follows.

System components: A system component (SC) is identified **first**. A finite set of System Components can be defined as: $SC = \{sc_1^i, sc_2^i, \dots, sc_m^i\}$. The system components could be a pump, a valve, a transmitter, etc.

State variables: For each system component, the different variables needed to represent its behaviour must be chosen. The finite set of state variables (SV) can be defined as: $SV^i = \{sv_1^i, sv_2^i, \dots, sv_m^i\}$ associated to the system component $sc^i \in SC$ (for instance sc^i a valve position) must be built. There exists at least one state variable for each system component.

Set of values: For each state variable, $sv_j^i \in SV^i$, the set $Val_{sv_j^i} = \{val_1^{ij}, val_2^{ij}, \dots, val_p^{ij}\}$ of possible values of sv_j^i must be stated. Necessary faulty values

should also be considered in this set. For instance, the valve position may take four values: "Open", "Closed", "Error On Open" and "Error On Closed".

Codification: The values in each set $Val_{sv_j^i}$ must be represented in terms of PN markings. This can be easily achieved if binary places are used. Thus, for each $sv_j^i \in SV^i$ a set $P_{sv_j^i} = \{P_1^{ij}, P_2^{ij}, \dots, P_P^{ij}\}$ of places such that $Val_{sv_j^i} = P_{sv_j^i}$ must be created. The marking of these places is binary and mutually exclusive. Then, $M(P_z^{ij})=1$ means that the variable sv_j^i takes the value Val_z^{ij} . Because of the existence of faulty values, the set of places can be partitioned into the subsets $P_{sv_j^i}^F$ and $P_{sv_j^i}^N$, representing the faulty and normal values respectively.

Event modelling: For each pair of values val_n^{ij} and val_m^{ij} such that the state variable sv_j^i could change from val_n^{ij} to val_m^{ij} a transition T_{nm}^{ij} must be created. Then one arc going from place p_n^{ij} to transition T_{nm}^{ij} and one arc going from T_{nm}^{ij} to place P_m^{ij} must be created.

Initial marking: The initial marking is defined as: $M_0(P_m^{ij})=1$ if the initial variable sv_j^i is val_n^{ij} and $M_0(P_m^{ij})=0$ otherwise.

Output: The output of this algorithm is a set of isolated IPN modules, each one modelling the behaviour of a state variable sv_j^i .

The IPN global model: This paper performs synchronous composition and permissive among IPN modules developed to obtain the global IPN model system. The basic recipe of the IPP and its IPN modules are detailed in section 4 and 5 respectively.

4. THE COLLECTED INFORMATION OF THE IPP

This paper models an industrial production process that comprises three pump stations (IPP -1, 2 and 3) to pump oil from this plant to different location as depicted in Fig. 1. The pump stations IPP-1 and IPP-2 are steam pumps with pump speed control, while the pump station IPP-3 is a motor driven pumps with fixed speed. The latter station is selected in this paper to be controlled due to its complexity. The IPP-3 comprises five units, the tank farm unit #2 (TFU-2), the booster pump (BP-3), the shipper pump (SP-3), the flow control valve (FCV) unit, and the pressure reducing unit (PRU). The BP-3 consists of a motor driven booster pump with 1500 hp, and a motor driven shipper pump (SP-3) with 10000 hp. The former is piped to take the suction from TFU-2 common header and discharge to the latter at pressure of 100 psi. The SP-3 takes suction from the BP-3 at TFU-2. It discharges into two shuttle lines headers at 650 psi. For pages reduction reasons, the recipes collected from the expertise and the IPP-3 catalogues of the TFU-2, and PRU units as selected parts of the IPP-3 are discussed in this paper as follows.

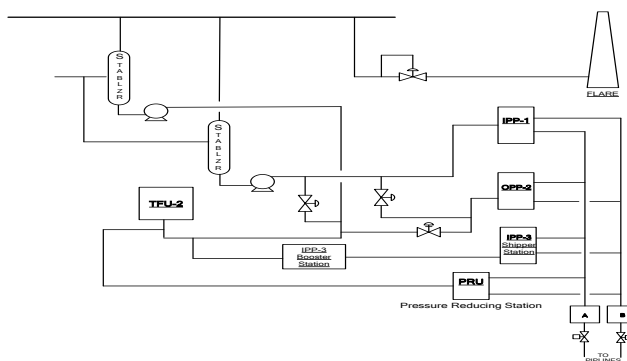


Fig.1. The schematic diagram of the IPP

4.1. The tank farm unit

Figure 2 shows the tank farm unit, TFU-2. It comprises two tanks which provide storage capacity for stabilized crude from stabilizers. Stabilized crude oil from the stabilizer is routed to the tanks. The oil level in these tanks is normally maintained within the range of certain two values, however during emergency conditions the oil level increases at a higher range. All tanks are utilized for oil storage to feed pump units. The TFU-2 performs the following two functions:

- Open the outlet-block-valves on the tanks in TFU-2,
 - Tank-2020: Valve MOV-0057
 - Tank-2021: Valve MOV-0061
- Open the air operating valves (AOV-152 & AOV-163), these valves are the output from the rundown line and input to the booster suction valve

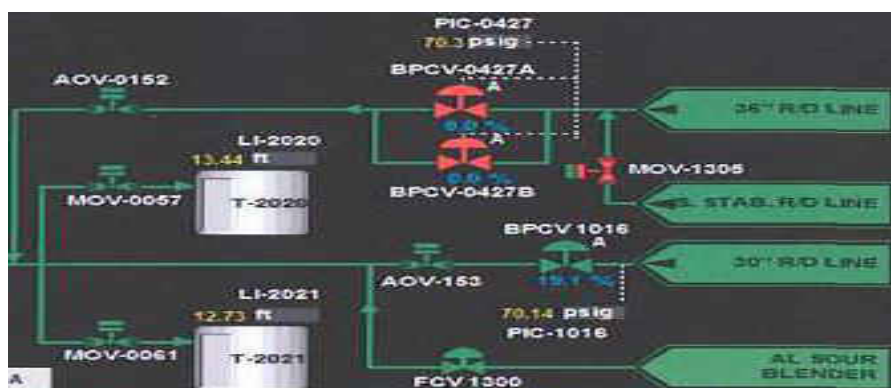


Fig. 2. The TFU-2 unit

4.2. The pressure reducing unit

The pressure reducing unit shown in Fig. 3 consists of four pressure control valves (PCV-952-1, PCV-952-2, PCV-952-3 & PCV-452). These control valves are sensed by the four-shuttle line transmitters. These transmitters' are:

Inputs: The shuttle line-B pressure transmitters are PT-1025 and PT-1405 respectively. The shuttle line-A pressure transmitters are PT-1024 and PT-1406 respectively.

Output: The control valves are PCV-452-1, PCV-952-2, PCV-952-3, and PCV-452.

The high selected reading (pressure) from these four transmitters will be used as PV for the PRU control.

- Get the four transmitters readings
- Select the highest reading out of the four transmitters
- Compare the highest reading (PV) with the set points
- If the PV > 610 psi then use the control valve PCV-452 only to reduce pressure.
- If the PV > 620 psi then use the control valves PCV-452, and PCV-952-1 respectively.
- If the PV > 630 psi then use the control valve PCV-452, PCV-952-1, and PCV-952-2 respectively.
- If the PV > 640 psi then use the control valve PCV-452, PCV -952-1, PCV-952-2, and PCV-952-3 respectively.
- Vice versa when pressure goes down
- Alarm will be set when the pressure goes higher than 700 psi and ESD system maybe takes place.

Note that the PV is always the high selected value from the four transmitters only when the pressure reading is within the operating range. If any transmitter is faulty, below or high its range, or put on maintenance by DCS operators, then it will be de-selected from the control and the next highest reading will be selected as an input (PV) to the PRU controllers.

These staggered set points will adjust the pressure release to TFU-2 and will provide smoother control response for shuttle line's pressure. For example, when the shuttle line pressure rises to 610 psi, only PCV-452 will act to release the pressure to TFU-2. If the shuttle line pressure continues to increase then PCV-952-1&2 & 3 will open to release the pressure as described above. The following conditions should be considered in this case.

- Normal pressure = 550 psig
- Very High pressure = 700 psig (all PRS valves should go widely open)
- Low pressure = 480 psig

5. DEVELOPING THE PETRI NETS MODULES

Based on the information collected and described in subsection 4.1 about the IPP-3, this section forms the recipe of the TFU-2, and PRU and develops their corresponding Petri net modules that are linked to the rest developed modules to create the sequential PN-model of the plant.

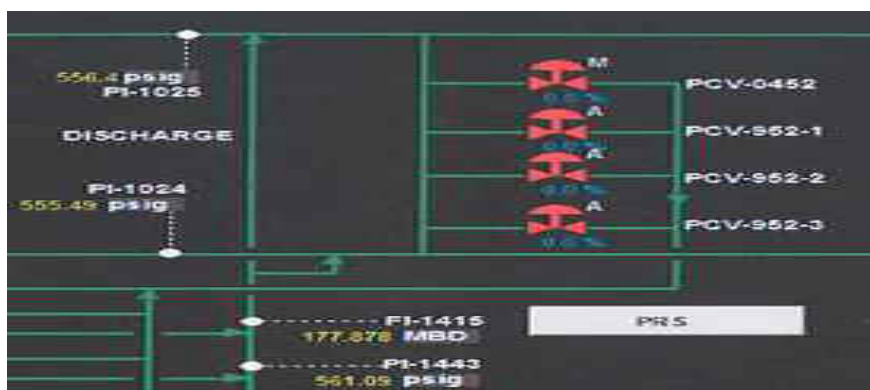


Fig. 3. The PRU unit at the shuttle lines A and B

5.1. Developing the PN-based TFU-2 module

The basic recipe and the developed PN-model of the tank farm unit, TFU-2 can be described in Table-1, and Fig. 4 respectively.

Table-1: The TFU-2 basic recipe

Pls.	Associated actions	Trs.	Associated events
P _{T1}	Initialization	t _{T1}	Start Operation
P _{T2}	Open the outlet-block-valve- MOV-0057	t _{T2}	The outlet-block-valve-54 open timing
P _{T3}	The outlet-block-valve- MOV-0057 is completely opened	t _{T3}	The outlet-block-valve-58 open timing
P _{T4}	Open the outlet-block-valve- MOV-0061	t _{T4}	Unconditional transition
P _{T5}	The outlet-block-valve- MOV-0061 is completely opened	t _{T5}	Intermediate transition
P _{T6}	Intermediate place	t _{T6}	Intermediate transition
P _{T7}	Intermediate place	t _{T7}	The valve AOV-0152 open timing
P _{T8}	Open valve AOV-0152	t _{T8}	The valve AOV-0163 open timing
P _{T9}	The valve AOV-0152 is completely opened	t _{T9}	Unconditional transition
P _{T10}	Open valve AOV-0163	t _{T10}	Intermediate transition
P _{T11}	The valve AOV-0163 is completely opened	t _{T11}	Intermediate transition
P _{T12}	Intermediate place		
P _{T13}	Intermediate place		
P _{Tc1}	The condition of the transition t _{T4} that it is pressed one time at each start		
P _{Tc2}	The condition of the transition, t _{T9} , that is pressed one time after opening the valves valve-54 and valve-58.		
P _{B1}	Initialize the IPP-3 booster pump		

5.2. Developing the PN-based PRU module

This paper formulates the information collected in subsection 4.2 and develops also the PN-module of the pressure reducing unit, PRU. This is depicted in Table-2, and Fig. 5 respectively.

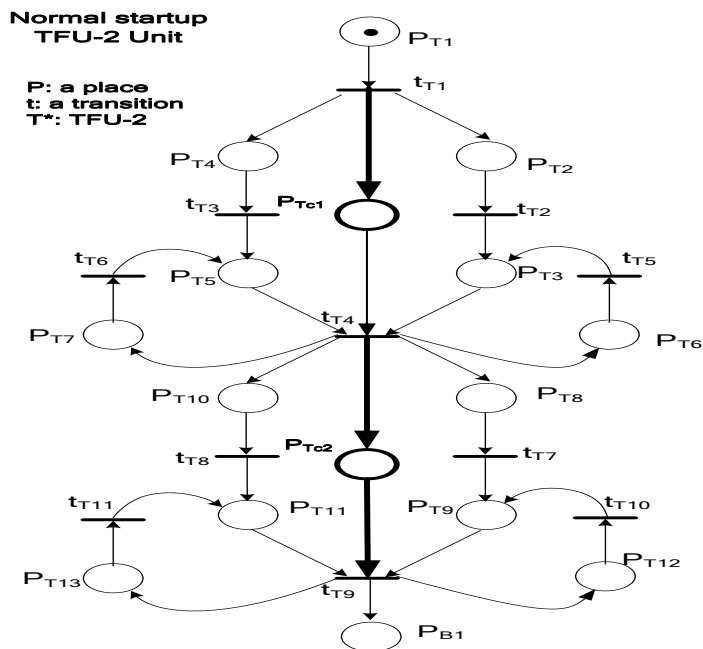


Fig. 4. The developed TFU-2 PN-module

Table-2: The PRS basic recipe

Pls.	Description	Trs.	Description
P _{P1}	The four transmitters readings	T _{P1}	Read the four transmitters values (PI) every 10 min.
P _{P2}	The highest reading is selected	T _{P2}	Select the highest reading
P _{P3}	V-452 is opened	T _{P3}	610 ≤ PV < 620
P _{P4}	V-952-1 is opened	T _{P4}	620 ≤ PV < 630
P _{P5}	V-952-2 is opened	T _{P5}	630 ≤ PV < 640
P _{P6}	V-952-3 is opened	T _{P6}	640 ≤ PV < 700
P _{P7}	V-952 is closed	T _{P7}	PV < 610
P _{P8}	V-952-1 is closed	T _{P8}	PV < 620
P _{P9}	V-952-2 is closed	T _{P9}	PV < 630
P _{P10}	V-952-3 is closed	T _{P10}	PV < 640
P _{P11}	Intermediate place	T _{P11}	Intermediate transition
P _{P12}	Intermediate place	T _{P12}	Intermediate transition
P _{P13}	Intermediate place	T _{P13}	Intermediate transition

6. THE PROPOSED ANALYSIS AND VERIFICATION METHOD

This paper proposes a method for analysis of the developed PN-based IPP-3 modules and their sequential merged model. It depends on the P-invariant property of the Petri nets. The method also uses the MATLAB and special PN-tool software for analysis and verification reasons. In general, there are three analysis methods for Petri nets. They are listed as follows, the reachability graph method, incidence matrix and state equation, and simple reduction rules and decomposition techniques.

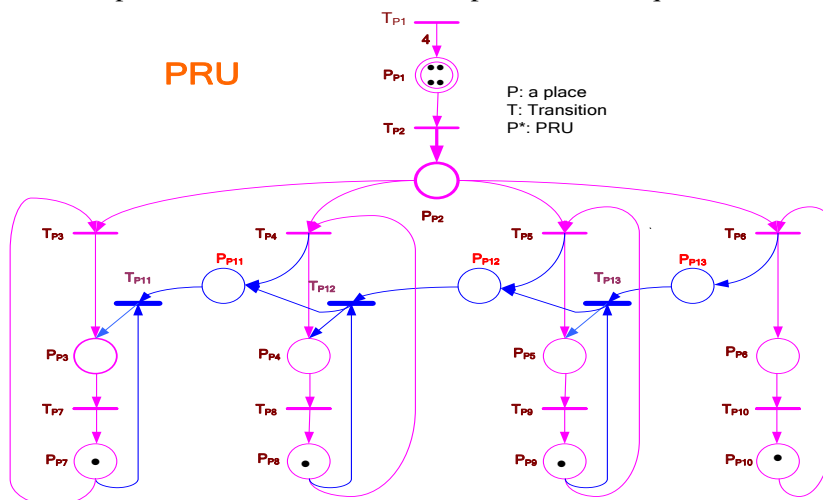


Fig. 5. The developed PRU PN-module

The reachability graph method involves essentially the enumeration of all reachable markings or their coverable markings [14]. It is fundamental to analyze the qualitative properties of the systems using a Petri net. However, it suffers from the state explosion [18-20]. It should be able to apply to all classes of nets, but is limited to "small" nets due to the complexity of the state-space explosion. On the other hand, the matrix equation method is very powerful, but it may lose the information when the net is impure, i.e. there is a self-loop in the net. So that, they are applicable only to special subclasses of Petri nets or special situations. The reduction method itself is not a completed method. It translates the Petri net to a simple one and then either using the reachability method or the matrix equation method to perform the analysis. It is clear that no method is enough for analyzing a PN, so using any analyzing methods is application dependent. Among these three methods, this paper employs the first method to analyze the developed modules of the IPP. The analysis and verification of the modules also depend on the P-invariant PN properties that can be described as follows.

6.1. P-Invariant Method

P-invariant property is a set of places whose weighted token count remains constant for all possible marking vectors. They are represented by n -dimensional integer vectors x , where n is the number of places of the Petri net; non-zero entries correspond to the

places that belong to the particular invariant. A place invariant is defined as every integer vector x that satisfies

$$x^T \mu_0 = x^T \mu \tag{5}$$

Equation (5) means that the weighted sum of the tokens in the places of the invariant remains constant for all reachable markings. This sum is not depending on the initial marking of the Petri net. It depends on the structural of the Petri net. The place invariants of a net can be computed by finding integer solutions to

$$x^T C^n = 0 \tag{6}$$

where C^n is the $n \times m$ incidence matrix of the Petri net. Given any firing vector q ,

$$\begin{aligned} x^T \mu(k+1) &= x^T (\mu(k) + C^n q(k)) \\ &= x^T \mu(k) \text{ iff } x^T D = 0 \end{aligned} \tag{7}$$

Equation (7) means that the weighted sum of the tokens in the places of the invariant is constant for all reachable markings. It is very important to check the boundness property of a Petri net. Accordingly, this P-invariant method is detailed in this paper for verifying the developed PN-modules. It can be briefly described as follows.

Referring to (7), where at least one entry of the vector x^T is nonzero. For simplicity use C instead of C^n that is an incidence matrix of PN; the case when $x = [0 \ 0 \ \dots \ 0]^T$ is trivial and hence it is excluded by Equation (10) that is in fact a system of homogeneous linear algebraic equations [21].

$$\begin{aligned} c_{11}x_1 + c_{12}x_2 + \dots + c_{1n}x_n &= 0 \\ c_{21}x_1 + c_{22}x_2 + \dots + c_{2n}x_n &= 0 \\ \dots & \\ c_{m1}x_1 + c_{m2}x_2 + \dots + c_{mn}x_n &= 0 \end{aligned} \tag{8}$$

Obviously, its zero solution always exists. That is,

$$x_1 = x_2 = \dots = x_n = 0 \tag{9}$$

For a system of linear algebraic equations it is well known that when the rank r of its coefficient matrix is equal to the number of equations $r = n$, exactly one solution exists. For the case of the homogeneous system at Equation (8) there is the solution at Equation (9). Coefficients c_{ij} in the system are integers as it follows from the construction of the C-incidence matrix. In this case, the condition $r < n$ is necessary but not sufficient for the existence of solutions. In general the weight associated with a place may be any integer, but usually one is mainly interested by P-invariants whose weights are positives.

The set of places having a weight not nil in such an invariant is a conservative component. *This means that the weighted number of tokens in this set of places is constant.* In otherwords, $x^T \mu_0 = x^T \mu$.

Using the above P-invariant method described, the developed PN-modules can be tested and modified using the MATLAB-based PN-tool software. The developed model should be P-invariant model to assure that the number of tokens in the model

not exceed certain value in a set of places. The value in each place in our application should be one to assure that the developed modules is one-bounded and consequently safe. To clarify the boundness and safeness PN-properties, this paper employs the developed module of the TFU-2 as follows.

The obtained results for the TFU-2 module can be listed as follows. Based on the given recipe shown in table-1 the weight vector is $X^T=[x_1, x_2]$; where $x_1=[1 1 1 0 0 1 0 0 0 0 0 0 0]^T$ and $x_2=[1 0 0 1 1 0 1 0 0 0 0 0 0]^T$. This means that the number of tokens in the set of places $\{P_1, P_2, P_3, P_6\}$ and places $\{P_1, P_4, P_5, P_7\}$ should be constant, however, this is not the case and there is no guarantee to say that this module is bounded based on the P-invariant test. After reconfiguring the developed model by adding conditioned places $\{P_{c1}, P_{c2}\}$, the developed module becomes bounded based on the same test. This can be clarified as follow. The obtained weight vector using the PN-tool is $X^T=[x_1, x_2, x_3, x_4, x_5]$, where $x_3=[1 0 0 0 0 0 0 1 1 0 0 1 0 0 1 0]^T$, $x_4=[1 0 0 0 0 0 0 0 1 1 0 1 0 1 0]^T$, and $x_5=[1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1]^T$. This means that the number of tokens in the set of places $\{P_1, P_2, P_3, P_6\}, \{P_1, P_4, P_5, P_7\}, \{P_1, P_8, P_9, P_{12}, P_{c2}\}, \{P_1, P_{10}, P_{11}, P_{13}, P_{c2}\}$, and $\{P_1, P_{c1}, P_{c2}, P_{14}\}$ are constant. This guarantees that the developed net is bounded but there is no guarantee for the safeness property. For safeness property described in Deffention-15, of these set of places should equal five tokens and each place in the net should less or equal one. In this TFU-2 module, these five tokens can be described as follows. The outlet-block-valve-54, outlet-block-valve-58, AOV-0152, and AOV-0163 are completely opened. These states need four tokens and the rest token initializes the IPP-3 booster pump. However, this is not always true.

In short, the developed module of the TFU-2 based on the given recipes is not P-invariant. So, the predesigned modules based on the given recipes are not only unbounded but also not safe. This paper reconfigured the developed model using the MATLAB-based PN-tool software and the P-invariant method described above to obtain the P-invariant modules using the conditioned places shown in table-1, $\{P_{c1}, P_{c2}\}$. However, although the redesigned model is bounded, but there is no guarantee for safeness property. This the main reason for needing of the validation method.

6.2. The validation of the developed PN-modules

To assure the safeness property and the soundness of the developed module, it should be validated as follows. Using MATLAB, this paper proposes a method to verify the developed PN-modules. The proposed method comprises two main blocks, one is the developed PN-module of a unit of the IPP-3, and the other is the state and the output equation defined in (1) and (2) respectively. Firing the developed PN-module and its mathematical model defined by these two equations, results two marking vectors, the real marking vector (M_{k+1}), and the estimated marking vector (M_{k+1}^s). Comparing these two vectors may result error values which indicate that the developed model should be redesign. This operation continues until the error values equal zero.

7. SIMULATION RESULTS

In this paper, using the PN-tool supported by MATLAB, the developed modules described in section 5 are analyzed and their problems were overcome. Each analysis is

performed in the sense of topology analysis (net class, traps, and siphons), behavioral analysis (coverability, and liveness), structural analysis (boundness, safeness, conservativeness, repetitiveness, and consistency), and P-invariant analysis described in sections 2 and 6 respectively. Among these features, this analysis assures the most important ones and tries to enhance the PN-models to develop reliable PN-modules. In this paper, the analyzed PN-modules are also verified using the proposed MATLAB-based validation method described in subsection 6.2. These analysis results are detailed as follows.

7.1. Analysis of the TFU-2 module

7.1.1. Verification of the TFU-2 module

Using the P-invariant method and the verification method described in section 6, the TFU-2 module can be analyzed and verified. Illustration of places and transitions of the TFU-2 unit is summarized in Table 1. This paper analysis this model using the Petri net tool software, and obtains the following results:

- 1- it is free choice net
- 2- it has 398 reachable states
- 3- it has 372 unsafe states
- 4- it is not-bounded; $M_{16} = [0,0,0,0,0,1,1,\omega,0,\omega,0,0,0,0]^T$

The state vector of the developed PN-model of the TFU-2 based on the given recipe is $M=[P_{T1},P_{T2}, \dots, P_{T13},P_{B-1}]^T$. Based on the P-invariant concept described in subsection 6.1 the developed module is not P-invariant. The MATLAB-based Petri net tool software has the facility to enhance the model graphically. So, adding two places enhances the PN-model and the state vector of the enhanced PN-model of the TFS-2 becomes, $M=[P_{T1},P_{T2}, \dots, P_{T13},P_{Tc1},P_{Tc2}, P_{B-1}]^T$. The obtained results of the modified model are:

- 1- it is free choice net
- 2- it has 36 reachable states
- 3- all states are safe
- 4- it is 1-bounded

7.1.2. Validation of the TFU-2 module

Using the proposed validation method described in subsection 6.2, the TFU-2 module can be validated using the PN-tool as follows. The marking vector is $M=[P_{T1},P_{T2}, \dots, P_{T13},P_{Tc1},P_{Tc2},P_{B1}]^T$, and the input or firing vector is $q=[t_{T1},t_{T2}, \dots, t_{T11}]^T$. Let us starting from firing t_{T1} ; $q=[1,0,0,0,0,0,0,0,0,0,0]^T$, and the initial marking vector $M_0=[1,0,0,0,0,0,0,0,0,0,0,0,0,0]^T$, the new state after firing the transition t_{T1} is: $M_1=[0,1,0,1,0,0,0,0,0,0,0,0,1,0]^T$. According to the recipe shown in table 1, the new states is start opening the outlet-block-valve-54 (P_{T2}), start opening the outlet-block-valve-58, (P_{T4}), and the condition of the transition t_{T4} is activated to make it is fired one time at each start of the TFU-2. Also, firing the transition t_{T2} ; $q=[0,1,0,0,0,0,0,0,0,0,0]^T$ leads to the new state, $M_2=[0,0,1,1,0,0,0,0,0,0,0,0,1,0]^T$. This means the outlet-block-valve-54 is completely opened (PT3). This firing sequence is carried out until firing t_{TF9} ; $q=[0,0,0,0,0,0,0,0, 1,1,1]^T$. This leads to the final state of this TFU-2 as, $M_F=[0,0,1,0,1,0,0,0,1,0,1,,0,0,0,0,1]^T$. This final state means that the

outlet-block-valve-54 & 58 (P_{T3}, P_{T5}) and the air operating valves AOV-0152 & 0163 (P_{T9}, P_{T11}) are completely permanent opened and the next unit BP-3 (P_{B1}) is initialized. It is clear that the developed module is safe $M(.) \leq 1$, and is identical to the given recipe depicted in table-1.

7.2. Analysis of the PRU module

7.2.1. Verification of the PRU module

Illustration of places and transitions of the PRS is summarized in table 2. This paper cannot analysis this model using the Petri net tool software because it is not ordinary PN. This is due to the arc weight between the source transition and the first place is four. However, the author designed this module to be, 4-bounded, safe, and live.

7.2.2. Validation of the PRU module

Using the proposed validation method described in subsection 6.2, the model can be validated using the PN-tool as follows. The marking vector is $M=[P_{P1}, P_{P2}, \dots, P_{P13}]^T$, and the input or firing vector is $q=[t_{P1}, t_{P2}, \dots, t_{P13}]^T$. Let us starting from firing t_{P1} ; $q=[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$, and the initial marking vector $M_0=[0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0]^T$, the new state after firing the transition t_{P1} is: $M_1=[4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$. According to the recipe shown in table 5, the new states means that the four transmitters readings are available (P_{P2}). Also, firing the transition t_{P2} ; $q=[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$ leads to the new state, $M_2=[3, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$. This means the highest reading of the transmitters is selected (PP3). This firing sequence has conditioned transitions as depicted in table 3. Supposed t_{P5} is fired. This leads to the state, $M=[3, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1]^T$. This state means that the valves V-452, V-952-1, and V-952 are automatically opened in a sequence to reduce the pressure from approximately 640 psi to less than 610 psi. If the pressure is 625 psi, the valve V-952-2 should be closed and V-452 and V-952-1 remain open. This operation should be automatically continued, until the pressure is ok that leads to the flow request management. This intelligent module adapts itself to reconfigure its valves. This is very important for industrial processes to avoid any unplanned shut down.

7.3. The two Modules Composition Operations

As mentioned in subsection 3.2, the system components maybe valves, pumps, ... etc. For example, a valve can be represented by a state variable that is associated with a set of values. These values, open or closed, can be represented by two places and one transition that represent the valve module. Through the two-module composition operations [13], the set of modules developed in sections 5 are merged into a single model according to the appropriate relationships achieved. To clarify this point, suppose that the four modules of the TFU-2 are the valves MOV-0057, MOV-0061, AOV-152, and AOV-163 respectively. Theses modules should be linked to represent concurrently operation as depicted in Fig. 6. The transition number four was appeared two transitions; one for each branch. Based on the two modules composition operations, this transition becomes one to link the two modules such that the output places of the valves MOV-0057, and MOV-0061 are linked with the input places of the

valves AOV-152, and AOV-163 respectively. This structure represents concurrent operation that is an attribute of system interactions. This philosophy is used to link all modules developed for the TFU-2, the PRU, and all IPP units.

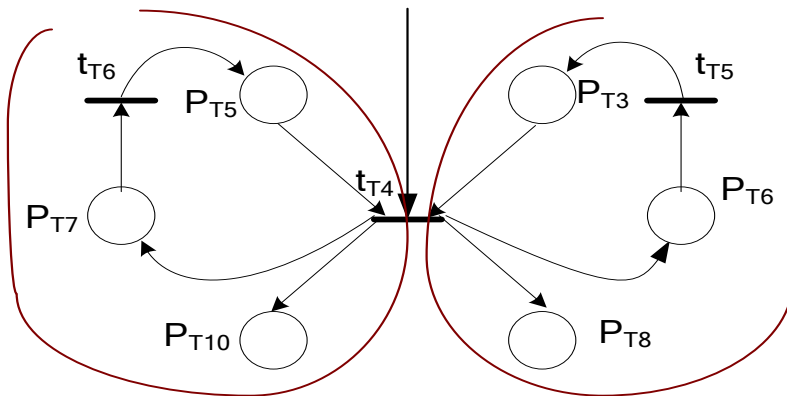


Fig. 6. A part of Fig. 4

7.4. The developed net sequential model

Linking the developed validated modules detailed above with the other developed modules that have not appeared in this paper due to their long analyses, structures the whole PN-model of the IPP-3. This link is obtained from the whole recipes collected and formulated in this paper. It can be achieved by the set of places and transition, $\{P_{B1}, P_{S1}, P_{F-1}, \text{ and } t_{SP-5}\}$, where P_{B1} is initialization place from TFU-2 unit to booster pump (BP-3) unit, P_{S1} is initialization place from BP-3 unit to Shipper pump (SP-3) unit, P_{F-1} is initialization place from SP-3 unit to flow control valve (FCV) unit. The transition t_{SP-5} links a booster pump place, P_{B3} , and the shipper pump, SP-3 unit. Where P_{B3} means that the valve V-1783 should be opened The emergency shut down (ESD) signal can shutdown the process and reset the developed model. This is designed by linking all the activated places of the TFU-2, BP-3, and SP-3 units and the signals of the FCV (very low flow) and the PRU (very high pressure) to a transition named reset transition. Simulation results of the sequential model are too long to written in this paper. However, the main features of the developed sequential PN-model are: 1- bounded, safe, reachable, live, and simple to follow its sequence.

8. CONCLUSIONS AND FUTURE WORK

Modeling of discrete event systems was addressed in this paper that focused on the IPP. This paper developed the modules of the IPP-3 that was integrated to develop the sequential PN-model of the station. Analysis of the developed PN-modules and the sequential PN-model of the IPP-3 were performed in this paper using MATLAB-based PN-tool and simulation MATLAB programs. Introducing Petri nets for modeling purposes and control instead of Grafcet is the main contribution of this paper. This contribution was achieved by formulating the collected information from the expertise and the available catalogues, modelling the IPP using Petri nets, and verifying and

validating the IPP modules using the MATLAB-based Petri-net tool software and MATLAB tools. The TFU-2 and PRU were selected as parts of the IPP-3 to highlight this contribution. Simulation results reflect that using Petri nets for modelling industrial processes instead Grafcet is an added value for industry. These promising results encourage the author to verify the developed modules and the PN-model at real time.

9. REFERENCES

1. Pepyne D.L., and Cassandras C.G. "Optimal control of hybrid systems in manufacturing", *Proceedings of the IEEE*, vol. 88, pp. 1108-1123, July 2000.
2. Chen C.H., and Dai J.H. "Design and high-level synthesis of hybrid controller", *Proceedings of the IEEE International Conference on Networking, Sensing, and Control*, Taipei, Taiwan, pp. 433-438, March 2004.
3. Morari M. "Hybrid Systems: Theory, Computation and Applications", *IEEE International Conference on Control and Automation (ICCA 2005)*, Budapest, Hungary, pp.10, 2005.
4. Demongodin I., and Koussoulas N.T. "Differential Petri net models for industrial automation and supervisory control", *IEEE Transactions on Systems, Man, And Cybernetics-Part c: Applications and Reviews*, vol.36, pp.543-553, July 2006.
5. Murata T. "Petri Nets: Properties, analysis and applications", *Proceedings of IEEE*, vol. 77, pp.541-580,1989.
6. David R., and Alla H. *Petri Nets and grafcet tools for modeling discrete event systems*, Prentice Hall Ed., London,1998.
7. Caradec M., and Prunet F. "A new modeling tool for hybrid flexible systems," *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, pp. 2246 – 2251, 1997.
8. Freedman P. "Time, Petri nets, and robotics", *IEEE Transactions on Control Systems Technology*, vol. 7, pp. 417-433, 1991.
9. Ragab A. *Modeling and supervision of discrete event systems*, Ph.D. thesis, Faculty of Electronic Eng., Menoufia University, Egypt, 2007.
10. David R. "Grafcet: A powerful tool for specification of logic controllers", *IEEE Transactions on control systems technology*, vol. 3, pp. 253-268, 1995.
11. Russo M.F. "Modeling, analysis, simulation, and control of laboratory automation systems using Petri nets analysis and control", *The Association for Laboratory Automation, JALA*, vol. 13, pp. 103-115, 2008.
12. Yao A. W-L, Liao H-T, Chi S-C J., and Peng S-S. " A Petri net based offline simulation and online diagnostic platform for manufacturing systems", *Journal of the Chinese Institute of Industrial Engineers*, vol. 22, pp. 64-75, 2005.
13. Arámburo J., and Ramírez-Treviño A. *Advances in Robotics, Automation and Control*, ISBN 78-953-7619-16-9, I-Tech, Vienna, Austria, 2008.
14. Wang F.-Y., Gao Y., Zhou M.C., "A modified reachability tree approach to analysis of unbounded Petri nets", *IEEE Transactions on Systems, Man, and Cybernetics: Part B*, vol 34, no. 1, 303–308, 2004.
15. Lennartson B., Egardt B., and Tittus M. "Hybrid Systems in Process Control", *In Proceedings of 33rd IEEE Conference on Decision and Control*, Lake Buena, USA. pp. 3587-3592, 1994.

16. Villani E., Miyagi P.E., and Valette R. *Advances in Industrial Control: Modeling and analysis of hybrid supervisory systems*, Springer-Verlag London Limited, 2007.
17. David R., and Alla H. " Petri Nets for Modeling of Dynamic Systems :A Survey", *Automatica*, vol. 30, pp. 175-202, 1994.
18. David R., and Alla H. *Discrete, Continuous, and Hybrid Petri Nets*, ISBN: 3-540-22480-7 ,Springer-Verlag Berlin Heidelberg 2005
19. Bourdeaud'huy T., Hanafi S., Yim P. "Mathematical programming approach to the Petri nets reachability problem", *European Journal of Operational Research*, vol. 177, pp. 176–197, 2007.
20. Kordic V., Lazinica A., and Merdan M., *Manufacturing the Future Concepts, Technologies & Visions*, ISBN 3-86611-198-3, 2006.
21. Hruz B., and Zhou M.C. *Advanced Textbooks in Control and Signal Processing: Modeling and control of discrete-event dynamic systems*, ISBN-13: 9781846288722, Springer-Verlag London Limited 2007.

نمذجة العمليات الصناعية الإنتاجية

مقارنة بقضايا اكتشاف الأعطال وعزلها تعتبر قضية نمذجة العمليات الصناعية الأكثر أهمية حيث يرجع ذلك إلى إن تقنيات اكتشاف الأعطال تعتمد بدرجة كبيرة علي النموذج المستخدم. عادة يتم نمذجة العمليات الصناعية الإنتاجية (IPP) باستخدام جرافست (Grafset) والذي يتعامل مع تلك العمليات بصورة متقطعة إما واحد أو صفر أو بمعنى آخر آمنة - أي لايسمح بالعمليات المتعددة في آن واحد. النظم الصناعية هجين من التماثلية والمتقطعة و غالبا غير آمنة. علي عكس Grafset فإن شبكات بتري والتي تجمع بين التمثيل الرياضي والتمثيل المرئي لديناميكيات النظم - تمتلك قدرة فاعلة للتعامل مع النظم الصناعية المهجنة. ولهذه الأسباب فإن نمذجة وتحليل وتوكيد نماذج IPP تعد قضايا رئيسة يجدر التطرق إليها ومعالجتها في هذا البحث. أن الإسهام الأساسي لهذا البحث هو إدخال شبكات بتري لنمذجة IPP بدلا من Grafset المستخدم حاليا. ولتحقيق هذا الهدف تم وضع المعلومات - التي تم تجميعها من الخبرات الموجودة في حقل تشغيل المحطة والكاتالوجات الخاصة ب IPP - في صورة وصفية علمية دقيقة. وكذلك تم بناء نموذج لهذه العملية باستخدام شبكات بتري وتم تحليله والتحقق منه باستخدام برامج خاصة Petri Net Tools معتمدة علي MATLAB ومفهوم خاصية P-invariant. وضحت النتائج المحاكية قدرة شبكات بتري علي التعامل مع مثل هذه الأنظمة المعقدة وشجعت هذه النتائج علي الاستمرار في تحسين أداء هذه النظم الصناعية وصولا إلي التحقق عند الزمن الحقيقي من تلك النماذج التي تم بناؤها في هذا البحث.

Dr. Hamdy Ali Ahmed Awad received the B.Sc. degree in Industrial Electronics, and the M.Sc. degree in Adaptive Control Systems from Faculty of Electronic Engineering, Minuf, Minufiya University, Egypt in 1988 and 1994 respectively.



Dr. Awad received the Ph.D. degree in Artificial Intelligent Systems in 2001 from School of Engineering, Cardiff University, Cardiff-Wales, England.

Dr. Awad has got Associative Prof. degree in Computer Sciences and Control Systems in 2007. Since August 2007, he has been with the Faculty of Electronic Engineering, Minufiya University, Egypt.

Dr. Awad interests in intelligent control systems and their applications in control systems, Mechatronics, Petri nets, Discrete event systems, Hybrid systems, Fault detection & isolation, and Medical engineering.

Manuscript received January 2, 2010