Generalized State Equation for Petri Nets

CEM BAŞKOCAGİL, SALMAN KURTULAN
Faculty of Electrical and Electronics, Department of Control Engineering
Istanbul Technical University
34469, Maslak, Istanbul
TURKEY
cem@uekae.tubitak.gov.tr, kurtulans@itu.edu.tr

Abstract: - Petri nets (PN) are becoming more common as a tool for analysis and design of industrial automation systems (IAS). As an important advantage, PN is able to represent the systems both mathematically and graphically. We can utilize mathematical features of PN to carry out modeling, simulation, testing and verification phases of IAS. Thus, the entire analysis and design process of large-scale systems is greatly simplified and system modeling can be performed accurately and efficiently. However, there is no state equation available defined for PN in the literature, when dealing with the inhibitor and enabling arcs. In addition, although resetting mechanism is frequently used for IAS, there is no arc type directly representing such a mechanism in the literature.

In this study, the state equation of PN is revisited and novel mathematical descriptions of both inhibitor and enabling arcs are introduced. Additionally, a new type of arc, called "nullifier arc", and its corresponding modified state equation are proposed for modeling of the resetting mechanism. The state equation of all three arc types are combined and expressed as "generalized state equation". New software is developed and utilized to demonstrate the application of generalized state equation on an IAS. Finally, it is revealed that this novel generalized state equation is capable of representing the special arcs, which facilitates analysis of large scale systems.

Key-Words: - discrete event systems, modeling, analysis, simulation, petri nets, incidence matrix, state equation, inhibitor arc, enabling arc, nullifier arc

1 Introduction

Ever since Petri nets (PN) were proposed by Carl Adam Petri for system modeling in 1962 [1], a number of researchers have been working on PN [2], [3] for application oriented tasks. This makes PN an attractive tool for the challenging problems of today [3]. PN is frequently used for discrete event control system design [4], [5], [6], and [7].

PN is both a graphical and a mathematical tool used for system modeling [8]. Being a graphical tool, it greatly simplifies the investigation of system behavior. At the same time, one can exploit its mathematical representation to efficiently perform the analysis of systems. This makes it possible to investigate the system behavior and to foresee possible problems in a more detailed manner.

Despite the mentioned features which make PN powerful for industrial applications, there have been a few problems noted over time for the use of PN. There was no time representation defined in PN, furthermore, modeling complex systems by PN was hard. Also the structure of PN made it impossible to distinguish any given process in a large-scale system. In order to overcome such shortfalls and to improve modeling capabilities, some enhancements were made to ordinary PN. Hierarchical PN was proposed to model complex systems in a modular fashion. Colored PN was invented to easily distinguish specific events within the system. Timed PN was created to model time-based systems. All of these are called "high-level PN" [9], [10], and [11].

Although there were many enhancements made to PN, there were still some mechanisms required for the modeling of certain systems. There was a need to introduce prioritization into PN. Formally, when two or more transitions are connected to only one common place and we need a specific transition to take precedence over the others, there was no way to represent this situation graphically in PN. In addition, it was not possible to define negative conditions which prevent a particular transition from triggering. Inhibitor arcs were proposed to resolve such problems. When an inhibitor arc is connected to a transition and the input place which is connected to this arc has enough token(s), we can block the related transition [4], [5], [12], and [13]. Inhibitor arcs make it possible to represent the negative conditions. Inhibitor arcs are addressed in more detail in Section 3.

Under certain conditions, it was required to fire a transition without removing any tokens from the input places. For such a purpose, enabling arcs were introduced into PN. An enabling arc is very similar to an ordinary arc. The only difference is the fact that there is no token removal from the input place(s) in case of triggering of transition(s). The rest of the firing mechanism is identical to that of an ordinary arc [12], [14]. Enabling arcs are mentioned in more detail in Section 3.

In the literature, we observe that there is no formal way to implement the resetting functionality in PN. We need such a functionality to remove all of the tokens from a particular place when the corresponding transition triggers. In this study, we propose a new type of arc called "nullifier arc" to perform this functionality in PN. The nullifier arc is explained thoroughly in Section 4.

We can investigate simple systems graphically in PN. But, as the systems get complex, they yield huge graphical representations. This is a great challenge in analyzing such systems by graphical means. Therefore, we opt for mathematical methods for the same purpose. In the literature, there is a matrix representation which is known as the state equation [3], [15]. It is used to model and analyze systems mathematically. However, we do not have any formalized state equations for the inhibitor, enabling and the proposed nullifier arcs. Therefore, we cannot analyze the systems mathematically when these types of arcs are involved. In this study, the state equation for PN is revisited and modified to represent the inhibitor, enabling and nullifier arcs mathematically. We accomplish this by incorporating the behavior of the respective arcs into the state equation. The formalized methodology is given in Sections 3 and 4.

The rest of this paper is organized as follows: Section 2 gives some preliminaries and notation used for the PN. In Section 2, the state transition mechanism and the state equation are revisited. In Section 3, we mention about the inhibitor and the enabling arcs and propose a methodology to modify the state equation to represent the behavior of these arcs. Section 4 mentions about the newly introduced nullifier arc in detail and proposes a new incidence matrix to model this type of arc mathematically. Finally, conclusions are discussed in Section 5.

2 Preliminaries and Notation

2.1 Basic Components

PN is defined as a bipartite weighted directed graph. It is composed of places, transitions, directed arcs and tokens. Places represent the conditions and transitions represent the events. Places and transitions are connected by directed arcs. These arcs represent the flow of events within the system. Tokens simulate the system dynamics. All of these components are illustrated in Fig.1 [15]:

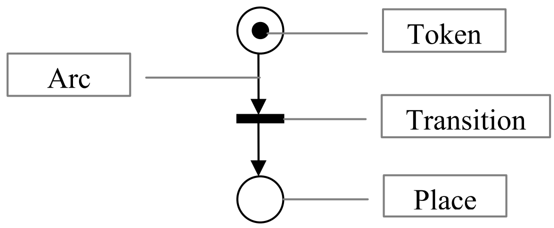

Fig.1 Basic components of PN

Places are used to specify the conditions, whereas transitions are utilized to represent the triggering of events. Directed arcs connect the places and transitions together. It must be noted that same types of nodes cannot be connected. Namely, two places or two transitions cannot be connected together. Only different types of nodes can be connected [15], [16].

Tokens are contained in places. The current state of a place is determined by the number of token(s) in this place. If a place has enough token(s), it satisfies the required conditions. Otherwise, if a place does not have enough tokens, it does not satisfy the conditions. Instantaneous distribution of tokens in all of the places determines the overall state of the system [15]. At any time, a place may contain one or more tokens as well as no tokens. This depends on the behavior of the underlying system. Given below are the definitions for the basic components of PN:

Definition 2.1: The net is defined by the triple NET = (P, T, F), where $P = \{p_1, p_2, \ldots, p_n\}$ is a finite non-empty set of places, $T = \{t_1, t_2, \ldots, t_m\}$ is a finite non-empty set of transitions, and P and T are the disjunctive sets, i.e., $P \cap T = \varnothing$ (empty set). F is the union of two binary relations F_1 and F_2: $F = F_1 \cup F_2$. F_1 is a binary relation from P to T: $F_1 \subseteq P \times T$. Analogously $F_2 \subseteq T \times P$ is the binary relation from T to P. F is the set of ordered pairs consisting of a place (transition) at the first position and a transition (place) at the second one. F is called a flow relation [17].

Definition 2.2: A Petri net is defined by the triple PN = (NET, W, M_0), where NET is a net by definition 2.1 such that PN = (P, T, F, W, M_0).

W is the weight function given as $W:F \rightarrow N^+$ where N^+ is a set of positive integers. $M_0:P \rightarrow N$ is a function called the initial marking whose element $M_0(p)$ is the number of tokens initially in place p where N is a set of non-negative integers.

The numbers to which the pairs of F are mapped are called weights. Obviously, the weights are positive integers. The initial marking is non-negative integers [17].

Definition 2.3: The function $M:P \rightarrow N$ is called the marking of a Petri net. $M(p_i)$ represents the number of tokens in place p_i at marking M.

The initial marking is specifically given in the definition of a Petri net. Similarly as in finite automata, it is reasonable to include the initial state in Petri net model definition because any real system begins its activity at an initial state. The different functions $M:P \rightarrow N$ correspond to the different markings [17].

Definition 2.4: The set of input places of transition t_j is denoted by $I(t_j)$ and called the pre-set of t_j. Similarly the set of output places belonging to transition t_j is denoted by $O(t_j)$ and called the post-set of t_j. These definitions are given as follows [15], [17]:

$$I(t_j) = \{p_i \in P \ : \ (p_i, t_j) \in F\} \tag{1}$$

$$O(t_j) = \{p_i \in P \ : \ (t_j, p_i) \in F\} \tag{2}$$

Definition 2.5: The notation $W(p_i, t_j) = k$ states that there are k arcs connecting place p_i and transition t_j, which means there is a unique k-weighted arc connecting these nodes. In the same way, $W(t_j, p_i) = k$ notation states that there are k arcs connecting transition t_j and place p_i [15].

2.2 State Transition Mechanism

The state transition mechanism in PN is realized by moving tokens between places. That is how state changes occur in PN [8]. Transitions are utilized to move tokens around. When the necessary conditions for a transition are satisfied, this transition can trigger. And when a transition triggers, tokens are removed from input places and added to output places. By this way, PN changes its current state and settles in another state. Actually, it is the triggering of transitions that causes the state changes in a PN. For a transition to trigger and to change the system state, the required conditions for that transition should be met [15]. The following definitions are given regarding the state transition mechanism in PN:

Definition 2.6: The following condition must hold to enable any transition $t_j \in T$ in PN [15]:

$$M(p_i) \geq W(p_i, t_j), \quad \forall \ p_i \in I(t_j) \tag{3}$$

In other words, a transition t_j is called "enabled" if and only if, for each pre-place of t_j, the marking of this place is equal to or greater than the weight of the arc connecting it to t_j, or t_j has no pre-place [17].

Definition 2.7: After a particular transition t_j triggers, the next state $M'(p_i)$ of the PN is defined as follows [15], where n is the total number of places:

$$M'(p_i) = M(p_i) - W(p_i, t_j) + W(t_j, p_i), \quad i = 1, \ldots, n \tag{4}$$

According to (4), if p_i is an input place of transition t_j, after t_j triggers, tokens are removed from this place. And the number of tokens removed from place p_i equals to the weight of the arc connecting this place to transition t_j. On the other side, if p_i is an output place of transition t_j, after t_j triggers, tokens are added to this place. And the number of tokens added to place p_i equals to the weight of the arc connecting transition t_j to this place [4], [5], and [15]. Consequently, after a transition triggers, input places lose tokens, whereas output places gain tokens.

2.3 The State Equation

It is vital to revisit the state equation of PN for understanding the methodology that will be proposed later. PN is known to represent the dynamic behavior of a system graphically. Therefore, the graphical features of PN can be exploited to inspect the dynamics of a system. This approach is suitable for simple systems. However, graphical methods are not efficient when large and complex systems are involved. In this case, we need a mathematical method for the inspection of such systems. The state equation is used to resolve this problem for PN [15]. When a transition triggers and there is a state change in the system, we can obtain the final state by evaluating the state equation of the system. This is a comfortable way to analyze the system behavior.

Before the definition of the state equation, we give some auxiliary definitions as in the following:

Definition 2.8: An m-dimensional firing vector u is defined as follows [15], where m is the total number of transitions:

$$u = [0, \ldots, 0, 1, 0, \ldots, 0] \tag{5}$$

In this vector, all of the elements assume a value of zero except the j^{th} element. A value of one for the j^{th} element implies that only the j^{th} transition will trigger.

Definition 2.9: The incidence matrix D is an m×n matrix whose entries are given as follows [15], where n is the total number of places and m is the total number of transitions:

$$d_{ji} = W(t_j, p_i) - W(p_i, t_j) \quad (6)$$

The incidence matrix identifies the behavior of the system. Therefore, the incidence matrix can be assumed to represent the system itself.

Definition 2.10: By using (5) and (6), we can give the state equation for the PN as follows, where M' is the final state vector, M is the initial state vector, u is the firing vector and D is the incidence matrix [15]:

$$M' = M + uD \quad (7)$$

Equation (7) is known as the state equation of the PN. The state equation mathematically represents the dynamic behavior of a system. When a particular transition triggers, we can obtain the next state using the state equation.

3 Formalizing the State Equation for Inhibitor and Enabling Arcs

3.1 Inhibitor Arc

We already know that a transition triggers when all of the required conditions are satisfied. However, there may also be other conditions that prevent the triggering of a transition. Inhibitor arcs are used to simulate such negative conditions. Inhibitor arc is an extension to ordinary PN. It is used to prevent a particular transition from triggering [12], [16]. This can also be used to prioritize particular transitions. For more information on inhibitor arcs, refer to [8], [12]. In this section, a methodology is formalized to represent the behavior of the inhibitor arcs in a matrix form.

During our research, we have noticed that there is no formalized state equation for inhibitor arcs in the literature. In order to study inhibitor arcs in a computerized environment, we need to represent them mathematically. This can be accomplished by incorporating the behavior of inhibitor arcs into the state equation.

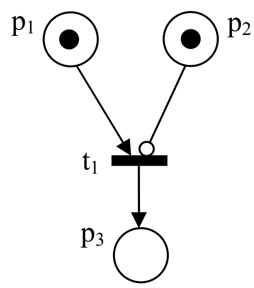

Fig.2 An illustration of inhibitor arc

In Fig.2, we have an inhibitor arc between place p_2 and transition t_1. If we disregard this inhibitor arc, transition t_1 can trigger since place p_1 contains a token. However, since place p_2 has at least one token; transition t_1 is prevented from triggering. Now, let us construct the state equation for this particular PN. When writing down the equations, we treat the inhibitor arc just like an ordinary arc:

$$M' = M + uD$$

$$M' = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 1 \end{bmatrix}\begin{bmatrix} -1 & -1 & 1 \end{bmatrix}$$

$$M' = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Eventually, since the inhibitor arc blocks the transition, there would be no token transfer and the final state would remain unchanged. However, according to (8), t_j triggers and p_1, p_2 lose tokens whereas p_3 gains a token. Obviously, this is not the expected result, because it does not take into account the effect of inhibitor arc. Therefore, the state equation cannot directly be used to model the behavior of the inhibitor arc. So, we need to modify it accordingly. In order to incorporate the effect of inhibitor arcs into the state equation, we propose a new m×m matrix called “inhibitor matrix” which will be denoted by $\tilde{H}(M)$. In this notation, M is the current marking of the PN. The definition of inhibitor matrix $\tilde{H}(M)$ is given as follows:

Definition 3.1: The inhibitor matrix is an m×m diagonal matrix whose off-diagonal elements are either 0 or 1. It is given as in the following:

$$\tilde{H}(M) = \begin{cases} \tilde{H}_{ij}(M) = 0, \quad i = 1, \dots, m, \; j = 1, \dots, m, & \text{if } i \neq j, \\ \tilde{H}_{jj}(M) = 1, \quad j = 1, \dots, m, & \text{if } t_j \text{ is not blocked by an inhibitor arc,} \\ \tilde{H}_{jj}(M) = 0, \quad j = 1, \dots, m, & \text{otherwise.} \end{cases} \quad (9)$$

In (9), m denotes the total number of transitions. The inhibitor matrix is variable and depends on the current system state. It identifies whether or not any given transition is blocked.

The inhibitor matrix itself is not enough to model the behavior of the inhibitor arc. When a transition triggers, no tokens are removed from the place which is connected to this transition by an inhibitor arc. In order to realize such behavior, we need to make a slight modification on the incidence matrix as well. The new incidence matrix D' is defined as follows:

Definition 3.2: The incidence matrix for inhibitor arcs is given as in the following:

$$D' = \begin{cases} d'_{ji} = 0, \quad i = 1, \dots, n, \; j = 1, \dots, m, & \text{if there is an inhibitor arc between } p_i \text{ and } t_j, \\ d'_{ji} = d_{ji}, \quad i = 1, \dots, n, \; j = 1, \dots, m, & \text{otherwise.} \end{cases} \quad (10)$$

In (10), n denotes the total number of places and m denotes the total number of transitions. The proposed incidence matrix revises the token transfer mechanism and makes it suitable for the inhibitor arcs. When ordinary arcs are involved, the new incidence matrix behaves just like the standard incidence matrix.

So far we have defined the inhibitor matrix and the modified incidence matrix. Now it is time to integrate them into the state equation. The new state equation can be defined as in the following:

Definition 3.3: Using (9) and (10), we define the state equation for inhibitor arcs as follows:

$$M' = M + u\,\tilde{H}(M)\,D' \quad (11)$$

Equation (11) can be used to model the inhibitor arcs. Let us readdress the previous PN example given in Fig.2. First of all, we need to figure out the inhibitor matrix $\tilde{H}(M)$. Since place p_2 has at least one token, transition t_1 is blocked. Using (9), we obtain the inhibitor matrix $\tilde{H}(M)$ as in the following:

$$\tilde{H}(M) = [0] \quad (12)$$

The new incidence matrix D' can be figured out according to (10). There is an inhibitor arc between place p_2 and transition t_1. Thus, evaluating the element d'_{12} yields a value of zero. Since there are not any other inhibitor arcs in the PN, the other elements of the new incidence matrix are identical to those of the standard incidence matrix. In this case, the new incidence matrix is given as follows:

$$D' = [-1 \quad 0 \quad 1] \quad (13)$$

Finally, substituting the inhibitor matrix (12) and the new incidence matrix (13) into the new state equation (11), we obtain the following:

$$M' = M + u\,[0]\,[-1 \quad 0 \quad 1] \quad (14)$$

Note that the value of the inhibitor matrix is zero in (14). Therefore, when we evaluate this new state equation, the final state equals to the initial state of the system. That means there is no change in the system state. This is the expected result. Because of the inhibitor arc blocking transition t_1, this transition cannot trigger and system state does not change.

Now let us consider another scenario for the same example. In this scenario, we suppose that place p_2 has no tokens. So, the inhibitor arc does not block transition t_1 and this transition can trigger. In this case, inhibitor matrix is obtained as follows:

$$\tilde{H}(M) = [1] \quad (15)$$

We have already obtained the new incidence matrix, so there is no need to recalculate it. After we substitute the inhibitor matrix (15) and the new incidence matrix (13) into the new state equation and evaluate it, we obtain the following:

$$M' = [1 \quad 0 \quad 0] + [1]\,[1]\,[-1 \quad 0 \quad 1]$$

$$M' = [1 \quad 0 \quad 0] + [1]\,[-1 \quad 0 \quad 1]$$

$$M' = [0 \quad 0 \quad 1] \quad (16)$$

This is the expected result. Since place p_2 does not have any tokens, transition t_1 is not blocked and it triggers. When transition t_1 triggers, one token is removed from place p_1 and one token is added to place p_3. The final state obtained in (16) reveals this situation.

In this section, we have proposed a modified state equation that can represent the behavior of inhibitor arcs. Using this new state equation, the inhibitor arcs can be modeled mathematically. This enables us to analyze and verify PN in computer systems when inhibitor arcs are involved.

3.2 Enabling Arc

Another extension to ordinary PN is the enabling arc. An enabling arc is simply used to enable a particular transition. Its behavior is very similar to that of an ordinary arc. The only difference is in the token transfer mechanism. Let us assume an enabling arc between place p_i and transition t_j. In this case, transition t_j can trigger if there is at least one token in place p_i. However, when this transition triggers, no token is removed from the input place p_i. So, the triggering of a transition does not change the marking in the places which are connected to enabling arcs. An enabling arc is illustrated by an arc with an empty arrow ending [12], [14]. For more information about enabling arcs, refer to [8], [12].

In the literature, we observed that there is no corresponding state equation for the enabling arcs. To leverage mathematical methods when studying enabling arcs, it is also required to represent them mathematically. For such a purpose, we use the fact that an enabling arc can be graphically represented using ordinary arcs. This situation is illustrated in Fig.3 as in the following:

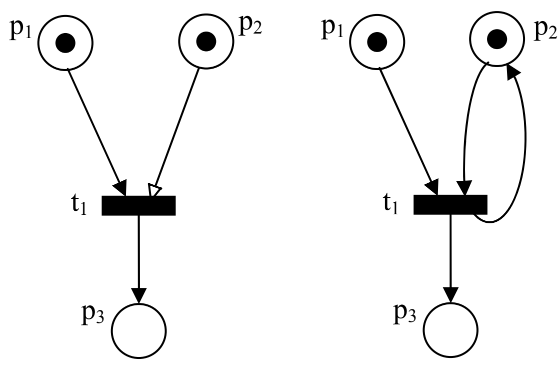

Fig.3 The enabling arc (a), and its counterpart (b)

Fig.3a illustrates a PN which includes an enabling arc and Fig.3b gives a representation of this enabling arc using ordinary arcs. We observe that the enabling arc can be represented utilizing two ordinary arcs. One of these arcs is drawn from place p_2 to transition t_1. And the other one is drawn from transition t_1 back to place p_2. By studying following conditions, it can be verified whether or not this representation exactly corresponds to an enabling arc:

Condition 1: For transition t_1 to trigger, there should be at least one token in both places p_1 and p_2.

Condition 2: When the required conditions are satisfied and transition t_1 triggers, the marking of place p_2 should not change.

In Fig.3b, it is obvious that transition t_1 can trigger if and only if the places p_1 and p_2 contain at least one token. In this case, “Condition 1” is already satisfied. When transition t_1 triggers, the token in place p_2 is removed due to the arc directed to transition t_1. At the same time, another token is added to place p_2 due to the arc directed back to place p_2. Thus, the number of tokens in place p_2 is conserved and does not change. This means the marking of place p_2 does not change. Therefore, “Condition 2” is also satisfied. Since we have all of the necessary conditions satisfied, we conclude that an enabling arc can be represented using two ordinary arcs, as illustrated in Fig.3b.

Since an enabling arc can be represented using ordinary arcs, incorporating its behavior into the state equation is straightforward. We know that the triggering of a transition does not change the marking of the place that is connected to the enabling arc. This feature only requires a slight modification of the standard incidence matrix. Like inhibitor arcs, when constructing the new incidence matrix, d_{ji} is taken to be zero for any enabling arc that is drawn from place p_i to transition t_j. Otherwise, the other elements of this incidence matrix remain the same as those of the standard incidence matrix. So, we observe that the new incidence matrix D' given in (10) can also be used for representing the enabling arcs. Other than that, no further modification is required. We give the modified state equation in the following definition:

Definition 3.4: The state equation for enabling arcs is given as in the following:

$$M' = M + u\,D' \tag{17}$$

We can use the new state equation (17) for the PN given in Fig.3a. The new incidence matrix is obtained same as (13). When we substitute this matrix into this state equation and evaluate it, we obtain the following:

$$M' = \begin{bmatrix}1 & 1 & 0\end{bmatrix} + \begin{bmatrix}1\end{bmatrix}\begin{bmatrix}-1 & 0 & 1\end{bmatrix}$$

$$M' = \begin{bmatrix}0 & 1 & 1\end{bmatrix} \tag{18}$$

When transition t_1 triggers, one token is removed from place p_1 and one token is added to place p_3. However, since we have an enabling arc between place p_2 and transition t_1, no token is removed from place p_2. The corresponding final state is demonstrated in (18). This is the expected result.

In this section, we have modified the state equation to represent the behavior of enabling arcs. We conclude that the proposed state equation can be used to model the enabling arcs mathematically. This helps us to analyze and verify PN in computer systems when enabling arcs are involved.

4 Introducing the Nullifier Arc

4.1 Nullifier Arc

We discussed both the inhibitor and the enabling arcs so far. These arcs are introduced into PN for modeling complex systems. In the previous section, we demonstrated a methodology to represent the behavior of the inhibitor and the enabling arcs mathematically. For this purpose, the state equation is modified for each of these arcs accordingly. In this section, we propose a new type of arc and follow the same methodology to represent it mathematically.

For certain applications, when a transition triggers, we require a mechanism to remove all of the tokens from the corresponding input place with no dependence on the number of tokens. This is necessary to implement the resetting functionality. For such a purpose, we should previously know the number of tokens in this place and adjust the weight value of the related arc accordingly. However, since the weight value is static, we cannot modify it dynamically. Therefore, in ordinary PN, it is not possible to implement such functionality. We introduce "nullifier arc" to overcome this problem. The nullifier arc modifies the state transition mechanism defined in (3) and (4) as in the following:

Definition 4.1: The following condition must hold to enable any transition $t_j \in T$ in PN [15]:

$$M(p_i) \geq 0\,, \quad \forall\ p_i \in I(t_j) \tag{19}$$

When nullifier arcs are involved, a transition t_j is called "enabled" if and only if, for each pre-place of t_j, the marking of this place is greater than or equal to zero. Note that there is no dependency on the weight values.

Definition 4.2: After a particular transition t_j triggers, the next state $M'(p_i)$ of the PN is defined as follows [15]:

$$M'(p_i) = 0\,, \quad i = 1, \ldots, n \tag{20}$$

According to (20), if p_i is an input place of transition t_j, after t_j triggers, all of the tokens are removed from this place. Thus, the nullifier arc gives us the opportunity to remove all of the tokens from an input place with no dependence on the number of tokens. In order to give a mathematical description of the nullifier arc, the standard incidence matrix is redefined as given below:

Definition 4.3: The incidence matrix for nullifier arcs is given as follows:

$$D''(M) = \begin{cases} d''_{ji} = -M(p_i),\ i = 1,\ldots,n,\quad j = 1,\ldots,m, & \text{if there is a nullifier arc between } p_i \text{ and } t_j, \\ d''_{ji} = d'_{ji}\ ,\qquad i = 1,\ldots,n,\quad j = 1,\ldots,m, & \text{otherwise.} \end{cases} \tag{21}$$

In (21), n denotes the total number of places and m denotes the total number of transitions. The proposed incidence matrix depends on the current system state. When nullifier arcs are involved, the corresponding elements of this matrix indicate that all of the tokens will be removed from the related input places. Otherwise, this incidence matrix behaves just like the one defined in (10).

Definition 4.4: Using (21), we define the state equation for nullifier arcs as follows:

$$M' = M + uD''(M) \tag{22}$$

We can use (22) to represent the behavior of nullifier arcs mathematically. Nullifier arc is depicted in Fig.4:

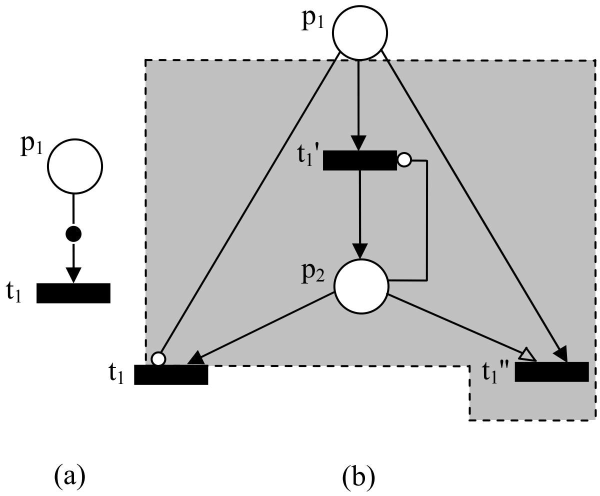

Fig.4 The nullifier arc (a), and its counterpart (b)

In Fig.4a, the arc with a solid disc at the center denotes the nullifier arc. The nullifier arc provides an intuitive way to implement the resetting mechanism for PN. It performs an immediate resetting functionality. Therefore, when transition t_1 triggers, all of the tokens are removed from place p_1 at once. We can realize the similar functionality using the illustration given in Fig.4b. However, in this illustration, the resetting functionality takes as many steps as the number of tokens in place p_1, whereas nullifier arc performs this functionality at a single step. For clarity, we will call this illustration as the counterpart of the nullifier arc. In the next section, we demonstrate that both the nullifier arc and its counterpart behave the same. For this purpose, we show the equivalence of their state equations.

5 Construction of the Generalized State Equation for PN

So far, we defined the inhibitor matrix only for inhibitor arcs and the modified incidence matrix for inhibitor, enabling and nullifier arcs individually. However, a unique state equation is required when dealing with all of these arcs at the same time. Therefore, we combine the inhibitor matrix and the modified incidence matrix to obtain the generalized state equation.

Definition 5.1: Using (9), (10) and (21) we define the generalized state equation for PN as follows:

$$M' = M + u\,\widetilde{H}(M)\,D''(M) \tag{23}$$

Equation (23) is called the generalized state equation and it can mathematically represent the behavior of all of the arcs.

Now, we use the generalized state equation to demonstrate that both the nullifier arc (Fig.4a) and its counterpart (Fig.4b) can perform the resetting mechanism. We inspect whether or not their state equations match with each other. At first, let us write down the generalized state equation for the counterpart of the nullifier arc in Fig.4b:

$$M' = M + u\,\tilde{H}(M)\,D''(M)$$

$$M' = \begin{bmatrix} M(p_1) & M(p_2) \end{bmatrix} + u\,\tilde{H}(M)\begin{bmatrix} -1 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \quad (24)$$

Note that place p_2 has no tokens initially. If we suppose that place p_1 has at least one token, equation (24) becomes:

$$M' = \begin{bmatrix} M(p_1) & 0 \end{bmatrix} + u\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} -1 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \quad (25)$$

According to (25), only transition t_1' can trigger. Substituting the appropriate firing vector u into (25), we obtain the next state as in the following:

$$M' = \begin{bmatrix} M(p_1) & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} -1 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$M' = \begin{bmatrix} (M(p_1)-1) & 1 \end{bmatrix} \quad (26)$$

When the transition t_1' triggers, one token is removed from place p_1 and one token is added to place p_2. The next state obtained in (26) reveals this situation. In this case, transition t_1' is blocked because of the inhibitor arc between place p_2 and this transition. Under this condition, we can discuss two distinct cases:

Case 1: If place p_1 has only one token initially, after transition t_1' triggers, this place will not have any tokens. Therefore, transition t_1'' cannot trigger. However, since place p_1 has no tokens and place p_2 has one token, transition t_1 can trigger. The evaluation of the state equation yields the following next state:

$$M'' = \begin{bmatrix} 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} -1 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$M'' = \begin{bmatrix} 0 & 0 \end{bmatrix} \quad (27)$$

When transition t_1 triggers, the token in place p_2 is removed. As shown in (27), there are no tokens left in both places p_1 and p_2. Thus, the resetting mechanism is realized.

Case 2: If the initial number of tokens in place p_1 is more than one, after transition t_1' triggers, this place will have at least one token. In this case, transition t_1 is blocked because of the inhibitor arc between place p_1 and this transition. However, since both places p_1 and p_2 have at least one token, transition t_1'' can trigger. After transition t_1'' triggers, the next state is obtained as in the following:

$$M'' = \begin{bmatrix} (M(p_1)-1) & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} -1 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$M'' = \begin{bmatrix} (M(p_1)-2) & 1 \end{bmatrix} \quad (28)$$

(28) implies that one token is removed from place p_1 when transition t_1'' triggers. As long as place p_1 has at least one token, transition t_1'' triggers and one token is removed from place p_1. After all of the tokens are removed from place p_1, "Case 1" becomes valid. Thus, the token in place p_2 is removed as well. This leaves no more tokens in both places p_1 and p_2. Finally, the resetting mechanism is realized.

This exhibits that the counterpart of the nullifier arc can realize the resetting mechanism. It is also verified that the generalized state equation works perfectly for both inhibitor and enabling arcs.

We follow the same methodology for the nullifier arc. Again, we use the generalized state equation to obtain the next state for the nullifier arc in Fig.4a:

$$M' = M + u\,\tilde{H}(M)\,D''(M)$$

$$M' = [M(p_1)] + [1][1][-M(p_1)] = [0] \quad (29)$$

As can be seen in (29), all of the tokens in place p_1 are removed when transition t_1 triggers. This verifies the functionality of the nullifier arc. We conclude that the generalized state equation can be used to mathematically model inhibitor, enabling and nullifier arcs.

6 Illustrative Example

6.1 The Mixer System

In this example, a mixer system is introduced. The corresponding PN model of this system includes inhibitor, enabling and nullifier arcs. We use the generalized state equation to figure out the states of this system. In addition, we developed software to implement our methodology. We use this software to obtain the system states as well. In order to verify our proposed methodology, we calculate the states of the system both using our methodology and graphically and compare the results.

The mixer system has two inputs and two outputs (Fig.6). One of the inputs starts and the other input stops the system. In order to perform the mixing mechanism, we control a motor which can be instructed to spin in either forward or reverse direction according to the system output. The system also has four timers to control the amount of time during which the motor spins in a specific direction and the motor stops between spins. When the motor completes spinning in both directions consecutively, the system is said to operate once. A counter is used to record the number of times the system operates. The system stops after it operates a predefined number of times. A nullifier arc is used to reset the counter. The corresponding PN is given in Fig.5:

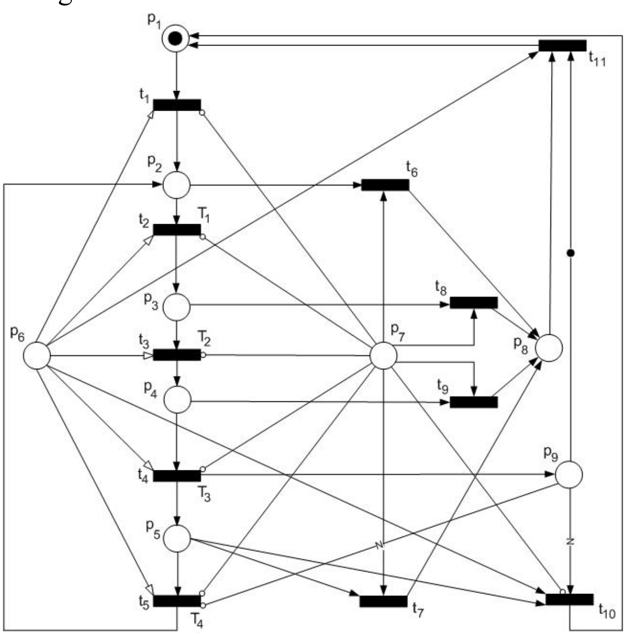

Fig.5 PN of the mixer system

Note that there is a nullifier arc between p_9 and t_{11} and $W(p_9, t_5) = W(p_9, t_{10}) = N$. The description of the places is given in the following table:

Table 1 Place descriptions

Place	Description
p_1	Initial state
p_2	Motor spins in forward direction
p_3	Motor stops
p_4	Motor spins in reverse direction
p_5	Motor stops
p_6	System is online
p_7	System will be stopped
p_8	System is about to stop
p_9	The number of times system operates

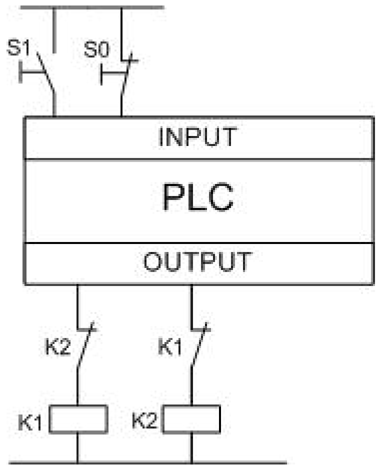

Fig.6 System setup

PN of the system given in Fig.5 includes all types of arcs mentioned in this paper. In this PN, the use of nullifier arc is also given. We can use the generalized state equation to obtain the states of this system and to verify our methodology. Let us consider the following cases:

Case 1: $M = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 2 \end{bmatrix}$

Since place p_7 contains a token, transition t_3 is blocked. Place p_3 has a token. Thus, transition t_8 can trigger. When this transition triggers, place p_3 and place p_7 lose token and place p_8 gains one token. Other places are not affected. In this case, the next state of the system is obtained as follows:

$$M' = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 2 \end{bmatrix} \tag{30}$$

We use the generalized state equation to obtain the next state mathematically. First, we figure out the inhibitor matrix according to (9). Since only transitions t_6, t_7, t_8, t_9 and t_{11} are not blocked, the corresponding diagonal elements have a value of one, whereas all other elements become zero:

$$\widetilde{H}(M) = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1 \end{bmatrix} \tag{31}$$

We calculate the modified incidence matrix according to (10) and (21):

$$D''(M)=\begin{bmatrix}-1&1&0&0&0&0&0&0&0\\0&-1&1&0&0&0&0&0&0\\0&0&-1&1&0&0&0&0&0\\0&0&0&-1&1&0&0&0&1\\0&1&0&0&-1&0&0&0&0\\0&-1&0&0&0&0&-1&1&0\\0&0&0&0&-1&0&-1&1&0\\0&0&-1&0&0&0&-1&1&0\\0&0&0&-1&0&0&-1&1&0\\1&0&0&0&-1&-1&0&0&-N\\1&0&0&0&0&-1&0&-1&-2\end{bmatrix} \quad (32)$$

We substitute (31) and (32) into the generalized state equation (23):

$$M' = M + [0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0]\,\tilde{H}(M)\,D''(M)$$

$$M' = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 1\ \ 2] \quad (33)$$

The results obtained graphically (30) and mathematically (33) are identical. We verify the result using our software. The software also computes the modified incidence matrix and the inhibitor matrix (N=3):

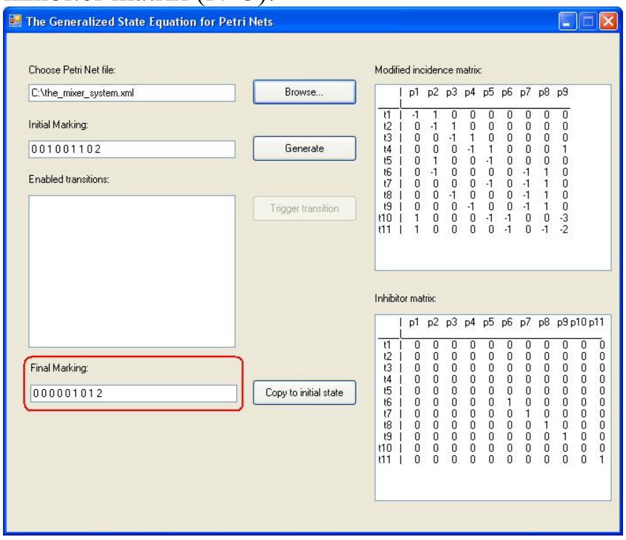

Fig.7 The next state as computed by the software (case 1)

Case 2: $M = [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 0\ \ 1\ \ 2]$

Since place p_6 and place p_8 contain one token and place p_9 contains two tokens, transition t_{11} can trigger. When this transition triggers, place p_6 and place p_8 lose one token, and all tokens in place p_9 are spent by the nullifier arc. Place p_1 gains one token. Other places are not affected. In this case, the next state of the system is obtained as follows:

$$M' = [1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0] \quad (34)$$

We use the generalized state equation to obtain the next state mathematically. First, we figure out the inhibitor matrix according to (9). Since no transitions are blocked, all diagonal elements have a value of one. All other elements are already zero. The inhibitor matrix becomes 11×11 identity matrix:

$$\tilde{H}(M)=\begin{bmatrix}1&0&0&0&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0&0&0\\0&0&0&0&0&1&0&0&0&0&0\\0&0&0&0&0&0&1&0&0&0&0\\0&0&0&0&0&0&0&1&0&0&0\\0&0&0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&0&0&0&1\end{bmatrix} \quad (35)$$

We calculate the modified incidence matrix according to (10) and (21):

$$D''(M)=\begin{bmatrix}-1&1&0&0&0&0&0&0&0\\0&-1&1&0&0&0&0&0&0\\0&0&-1&1&0&0&0&0&0\\0&0&0&-1&1&0&0&0&1\\0&1&0&0&-1&0&0&0&0\\0&-1&0&0&0&0&-1&1&0\\0&0&0&0&-1&0&-1&1&0\\0&0&-1&0&0&0&-1&1&0\\0&0&0&-1&0&0&-1&1&0\\1&0&0&0&-1&-1&0&0&-N\\1&0&0&0&0&-1&0&-1&-2\end{bmatrix} \quad (36)$$

We substitute (35) and (36) into the generalized state equation (23):

$$M' = M + [0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1]\,\tilde{H}(M)\,D''(M)$$

$$M' = [1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0] \quad (37)$$

The results obtained graphically (34) and mathematically (37) are identical. We verify the result using our software. The software also computes the modified incidence matrix and the inhibitor matrix (N = 3):

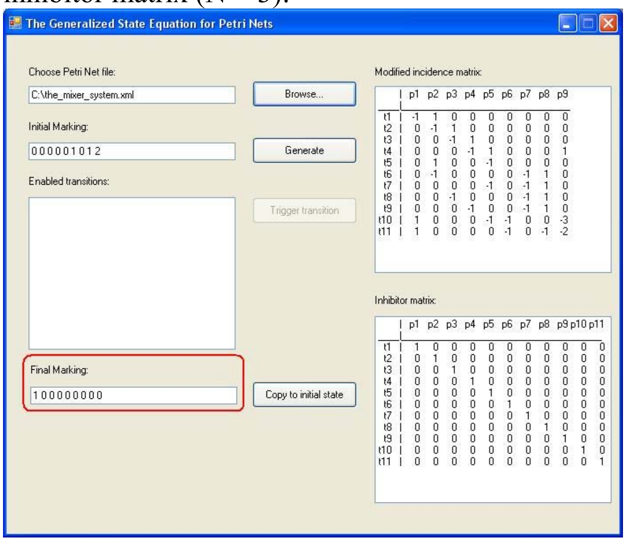

Fig.8 The next state as computed by the software (case 2)

7 Conclusion

In this paper, a new type of arc called "nullifier arc" is proposed to implement the resetting mechanism and a novel methodology is formalized to represent inhibitor, enabling and nullifier arcs mathematically. For this purpose, we modified the incidence matrix for all of these arcs and constructed a new matrix called the "inhibitor matrix" only for the inhibitor arcs. Combining the inhibitor matrix and the modified incidence matrix, we obtained a generalized state equation. Following this, we demonstrated the usage of this generalized state equation to investigate the states of a case study: industrial automation system. In this system, we used nullifier arc to reset a counter. We also developed software which implements our methodology. This software is utilized to obtain the states of the system. Finally, we verified that the generalized state equation can be used for mathematical modeling of all types of the arcs.

References:

[1] Petri C. A., *Kommunikation mit Automaten*, Ph.D. Thesis, Schriften des IIM Nr.2, University of Bonn, 1962.

[2] Reisig W., *Petri Nets: An Introduction*, Springer-Verlag, 1985.

[3] Peterson J. L., *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, 1981.

[4] Uzam M., Jones A.H., Discrete Event Control System Design Using Automation Petri Nets and Their Ladder Diagram Implementation, *The International Journal of Advanced Manufacturing Technology*, Vol.14, No.10, 1998, pp. 716-728.

[5] Uzam M., Jones A.H., Yücel İ., Using a Petri-Net-Based Approach for the Real-Time Supervisory Control of an Experimental Manufacturing System, *The International Journal of Advanced Manufacturing Technology*, Vol.16, 2000, pp. 498-515.

[6] Uzam M., Synthesis of Feedback Control Elements for Discrete Event Systems Using Petri Net Models and Theory of Regions, *The International Journal of Advanced Manufacturing Technology*, Vol.24, Numbers 1-2, 2004, pp. 48-69.

[7] Uzam M., Koç İ. B., Gelen G., Aksebzeci B. H., Asynchronous Implementation of Discrete Event Controllers Based on Safe Automation Petri Nets, *The International Journal of Advanced Manufacturing Technology*, Vol.41, Numbers 5-6, 2009, pp. 595-612.

[8] Murata T., Petri Nets: Properties, Analysis and Applications, *Proceedings of the IEEE*, Vol.77, No.4, 1989, pp. 541-580.

[9] Jensen K., *Colored Petri Nets: Basic Concepts, Analysis Methods and Practical Use (Vol.1–3)*, Springer-Verlag, 1997.

[10] Wang J., *Timed Petri Nets: Theory and Application*, Kluwer Academic Publishers, 1998.

[11] Koniewski R., Dzielinski A., Amborski K., *Use of Petri Nets and Business Processes Management Notation in Modeling and Simulation of Multimodal Logistics Chains*, Proceedings 20th European Conference on Modeling and Simulation, Institute of Control and Industrial Electronics, Warsaw University of Technology, 2006.

[12] Uzam M., *Petri-Net-Based Supervisory Control of Discrete Event Systems and Their Ladder Logic Diagram Implementations*, Ph.D. Thesis, University of Salford, 1998.

[13] Nauber W., *Design and Analysis with Petri Nets*, Lecture Notes, Technische Universtät Dresden, 2010.

[14] Wu W., Su H., Chu J., Supervisory Control of Discrete Event Systems Using Enabling Arc Petri Nets, *Proceedings of the IEEE*, Vol.2, 2002, pp. 1913-1918.

[15] Cassandras C.G., Lafortune S., *Introduction to Discrete Event Systems*, Kluwer Academic Publishers, 1999.

[16] Moody J.O., Antsaklis P.J., *Supervisory Control of Discrete Event Systems Using Petri Nets*, Kluwer Academic Publishers, 1998.

[17] Hruz B., Zhou M.C., *Modeling and Control of Discrete-Event Dynamic Systems: With Petri Nets and Other Tools*, Springer, 2007.