2019 CAA Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS)
July 05-07, 2019, Xiamen, China

Design of the Safety Control Logic for Railway Stations Based on Petri Nets

Yike Li, Yin Tong*, Jin Guo
School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, P. R. China
E-mail: yikeli@my.swjtu.edu.cn, yintong@swjtu.edu.cn, jguo_scce@home.swjtu.edu.cn

Abstract: The control logic in the traditional interlocking depends on human's experience, lacking a unified generation and verification method. In this paper, we use Petri nets to model railway stations and then obtain the mathematically optimal control logic to ensure safety and liveness. First, we use a modular method based on its devices including track segments, signals and switches. The interlocking condition is formulated as a set of linear constraints. According to the supervisory control theory, monitor places can be calculated to enforce the constraints. Finally the control logic derived from the monitor place is proven maximally permissive. Finally we take the simulation to investigate the performance of the control logic.

Key Words: Railway Station Control, Petri Nets, Modeling, Supervisory Control Theory

1 Introduction

The station is a hub of the railway networks, as well as the main place for passenger and freight operations [1]. Hence ensuring safety and raising efficiency of stations become particularly important [2]. At present, the size of railway network is growing rapidly, and the operate condition of stations has got far more complicated. During the early phases of the interlocking software development [3], an appropriate modeling tool and effective methods to analyze the control logic of stations can improve safety and reliability of the software.

The train operation in a station can be seen as a process in which trains share tracks under the premise of ensuring safety. There exist lots of parallel and concurrent events. The control logic is an abstraction of control methods that implemented on those events. As an efficient discrete event system analysis tool [4], Petri nets can graphically model elementary devices in a station [5], so that a model can be established for a station based on its actual structure. In addition, a series of analysis and simulation methods on Petri nets with strict mathematical basis can be applied to solve controlling problems.

According to the norm and operational requirements, under the premise of ensuring safety and liveness, the generalized mutual exclusion constraint (GMEC) is designed for the station model. Then based on the supervisory control theory in discrete event systems, the control logic of stations is obtained [6]. Note that the controller is the maximum permissive control policy obtained by mathematical calculation, which makes the final logic guarantees the operating efficiency to the greatest extent while satisfying the constraints.

1.1 A Short Introduction on Petri Nets

A Petri net is a four-tuple structure $N = (P, T, Pre, Post)$, where $P = \{p_1, p_2, \cdots, p_m\}$ is a set of places, represented by circles; $T = \{t_1, t_2, \cdots, t_n\}$ is a set of transitions, represented by bars; $Pre : P \times T \to \mathbb{N}$ is the pre-incidence function that specifies the weight of the arcs directed from places to transitions; $Post : P \times T \to \mathbb{N}$ is the post-incidence function that specifies the weight of the arcs directed from transitions to places. The incident matrix of a net is defined as $C = Post - Pre$. When the weight of the arc is 1, it can be omitted in the graph.

To introduce a dynamic behavior to a net, a function called marking: $M : P \to \mathbb{N}$ associates to each place a non negative number of tokens. Tokens are represented as black dots within places. The initial marking is denoted M_0. A net N with the initial marking M_0 is called a Petri net system or net system $\langle N, M_0 \rangle$. Place i is said to be marked when $M(i)$ (also denoted as m_i) $\neq 0$.

The evolution of the net system is driven by the firing of transitions. Once a transition t becomes able to fire, we call it is enabled. Transition t is enabled if and only if, for every input place of t, the marking is greater than or equal to the weight of the corresponding arc directed to t. The firing of the transition indicates an event occurs, and the state of the system changes. The transition firing will remove tokens from its input places, simultaneously, move tokens to output places, and the number of tokens equals the weight of corresponding arc. As the Petri net shown in Fig.1(a), transition t_1 is enabled. After t_1 fires, the net will reach a new marking as shown in Fig.1(b).

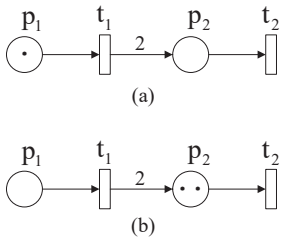


Fig. 1: The firing of transition t_1 in a Petri net system

For some cases, in order to facilitate the analysis of the system that has time order or time variable, the time factor is introduced [7]. If a non-negative random value is assigned to each transition in the net as a delay, such Petri net is called a timed Petri net.

*Corresponding Author.

This work was supported by the National Natural Science Foundation of China under Grant No. 61803317, the Fundamental Research Funds for the Central Universities under Grant No. 2682018CX24, and the Sichuan Provincial Science and Technology Innovation Project under Grant No. 2018027.

2 Modeling Railway Station With Petri Nets

By analyzing the functions and characteristics of elementary devices, Petri net models of the track, signal and switch can be established separately. The integral station model can be regarded as the composition of a certain number of those elementary modular models.

2.1 The Track Model

Consider a track of the station in Fig. 2. This track can be divided into 5 sections σ_i $(i = 0, \cdots, 4)$, and sections σ_0 and σ_4 are interstation lines that don't belong to the station. Arrival signals X and S set respectively at the beginning of the arrive-departure tracks σ_1 and σ_3, while starting signals SI and XI protect the upward throat and downward throat respectively. The two opposite black arrows indicate a long, switchless section in the middle of the station, which is called yard in following sections.

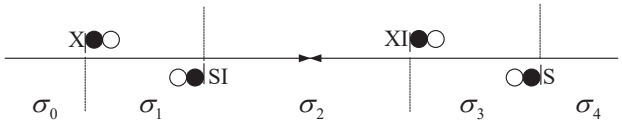


Fig. 2: A railway track

The movement of a train on the track can be regarded as the transfer of tokens among places. The model of the track is shown in Fig. 3, where the place $p_1 \sim p_3$ and $\tilde{p}_1 \sim \tilde{p}_3$ denotes the section $\sigma_1 \sim \sigma_3$ respectively for opposite direction. For instance, the marking of p_3 (with respect to, $\tilde{p}_3$) denotes the occupation of a train directing downward (with respect to, upward). Obviously, p_3 and $\tilde{p}_3$ couldn't be marked simultaneously. The firing of the transitions denotes the process of trains' entering the next section under permission (the signal is clear), therefore the signal can be modeled as a transition, particularly controllable. For the section without a protective signal, the corresponding transition is modeled as an uncontrollable one. In the next section, the monitor place can't have an arc to the uncontrollable transition, but to the controllable only. To distinguish them graphically, an uncontrollable transition shows as a bar with stripes.

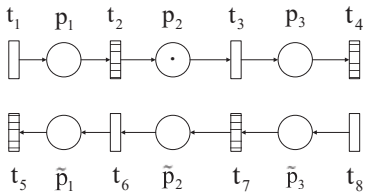


Fig. 3: The Petri net model of a track

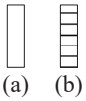


Fig. 4: The controllable and uncontrollable transitions

For the downward train, the enabling of t_1 denotes that signal X is at clear, namely allowing the train to enter the protected section. After t_1 fires, p_1 is marked, which denotes σ_1 is occupied, i.e., there is a train in σ_1. During the simulation, a delay will be assigned to each transition, which denotes the time used for the train to run through the previous section to the next.

2.2 The Switch Model

A switch has two movable point rails [8], providing two positions that can be connected. Thus its actual function is to select the path. Based on this, we can model the switch as a 2-to-1 selector.

The Petri net model of a switch is illustrated in Fig. 5. It includes 3 places (p_1, p_2, p_f) and 4 transitions ($t_{o,1}$, $t_{o,2}$, $t_{f,1}$, $t_{f,2}$). When $p_i(i = 1, 2)$ is marked, it implies that the switch connects to the track i, namely the train can enter the track i through the switch; if p_f is marked, it implies that the switch is being converted or under maintenance. Under both circumstances, the train cannot pass the switch nor enter any track.

Transitions in the switch model are all controllable because external manipulations can be carried on the switch to change its working conditions.

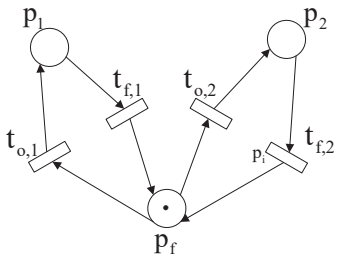


Fig. 5: The Petri net model of a switch

2.3 The Station Model

For illustration, a two-track station shown in Fig. 6 is modeled using the method described in Section 2.1 and 2.2. Its corresponding Petri net model is shown in Fig. 7. Notice that the bidirectional arc represents a loop, that is, two arcs of two different directions. The two subnets at the bottom left and right represent switch 1 and switch 2 respectively.

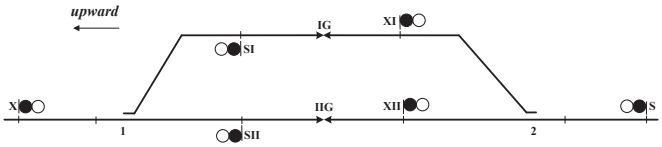


Fig. 6: A railway station with 2 yards

3 GMEC and the Control Logic Design

The control logic is a limitation on the state of the system. All states that the system can reach, namely the reachable markings, may contain some states against safety or other specifications, which are called the illegal markings. In the process of supervising and controlling, it is necessary to specify some rules to meet the requirements of the system operation. In other words, the reachable markings of the controlled system are all legal markings. In this paper, the rules are formulated by a set of linear inequalities that are

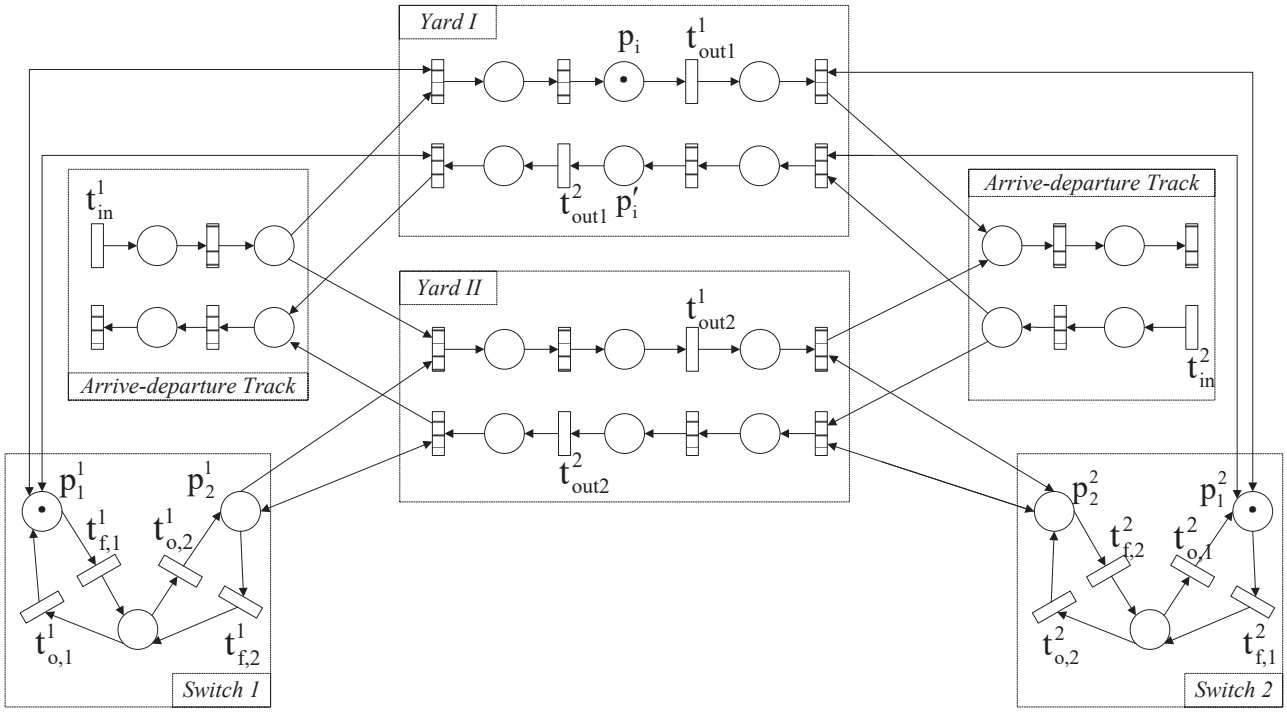


Fig. 7: The Petri net model of the station

called the generalized mutual exclusion constraint(GMEC, for short). In the supervisory control theory, a GMEC may be enforced by adding to the net a control structure that consists of a new place called monitor place. Each of the constraint corresponds to an additional place.

As shown in Fig. 8, the process of designing the control logic basically is to calculate the monitor place by matrix operations. Finally, a new incidence matrix is obtained and the structure of the monitor places are specified by the last row of the incidence matrix. With the monitor places, the behavior of the system is controlled such that all its reachable markings are legal. Due to limited space, we refer the reader [6] for more details of computing monitor places and herein the *PNmonit* toolbox is used to compute the monitor places automatically.

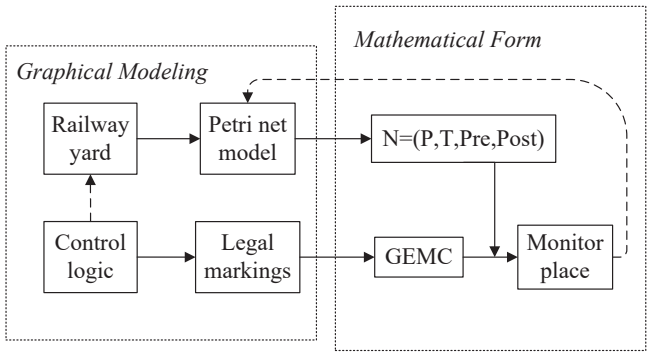


Fig. 8: The process of designing the control logic

3.1 Design and Formulation of Constraints

To take into account both liveness and safety, in the following discussion, a modified model is proposed as shown in Fig. 10 (without the monitor place p_s^1 and its arcs), which is more practical.

Compared to the previous model, the main modifications are: (1) Merging two places denoting the same track section in different directions at the middle switchless track into one place; (2) Merging two places denoting the successive track sections at the arrive-departure track into one place; (3) Adding two places p_{13} and p_{14} to represent the upward and downward interstation line respectively. First, the middle track of the station is used to park the trains and no switch is placed there. It is regarded as a whole during the control process for its major length can be ignored. Second, in the general sense, the arrive-departure track contains one train at most. Last but not least, interstation lines of two directions should be taken into considered in order to investigate the throughput of the station.

According to the interlocking condition that need to be checked when selecting and locking the route in the working scene (including the route selection requisition, route locking requisition and signal clearing requisition) [1], we formulate the following two constraints to ensure safety and liveness of the station in Fig. 7:

(a) No more than two trains is in the station and approach/departure sections simultaneously.

(b) If a switch is connected to the occupied track, another train cannot enter the arrive-departure track (i.e., put the arrival signal at stop).

Constraints above can be converted into a set of GMECs:

$$m_1 + m_2 + m_3 + m_4 \leq 2 \quad (1)$$

$$m_2 + m_5 + m_{13} \leq 2 \quad (2)$$

$$m_3 + m_6 + m_{13} \leq 2 \quad (3)$$

$$m_2 + m_8 + m_{14} \leq 2 \quad (4)$$

$$m_3 + m_7 + m_{14} \leq 2 \quad (5)$$

More specifically, GMEC in Eq. (1) is formed by constraint (a); GMEC in Eq. (2) to (5) are formed by constraint (b).

Using the *PNmonit* toolkit loaded in MATLAB, five monitor places are obtained after inputting the structure of the net and GMECs. They are shown in Fig. 9.

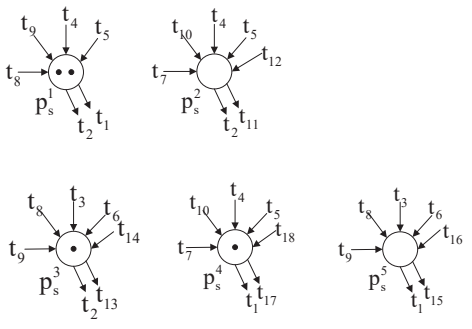


Fig. 9: The designed five monitor places

3.2 The Control Logic Design

Before discussing how to obtain detailed control logic from the monitor place, the following points are noteworthy: (1) The monitor place has no actual physical meaning similar to the device subnet in the net (track sections, signals, switches, etc.), yet an executor of the control logic of the net system; (2) A monitor place calculated through GMEC is determined only by the structure of the net and has nothing to do with the initial state. As long as the initial marking satisfies this GMEC, then we can calculate the monitor place by matrix operations; (3) Marking of a monitor place has no actual meaning. There exist token flows between a monitor place and the controllable transitions of the original net, so the control logic may be obtained by analyzing the input and output arcs of the monitor place.

As aforementioned in Section 2, the controllable transition denotes the signal device including the signal and the switch. Combined with the firing rule of transition, the control logic reflected by a monitor place is: (1) acquiring the state of the signal device corresponding to its input transition (being enabled or not, for example, the signal is at clear or at stop); (2) controlling the operation of the signal device corresponding to its output transition (to control the signal indication and convert the switch).

In the following discuss, we consider a scenario where the monitor place p_s^1 plays a role in system safety control.

Assume that: (1) there exist no train in the station ($m_1 = m_2 = m_3 = m_4 = 0$), (2) there are $n_1 + n_2$ trains in interstation lines prepare to enter the station successively ($m_{13} = n_1$, $m_{14} = n_2$, $n_1, n_2 \in \{1, 2\}$). (3) switch 1 connects to section IG, and switch 2 connects to IIG (p_5, p_7 are marked). Such a system is shown in Fig. 10, and we have $m_s^1 = 2$.

Now there are two trains entering the station from both sides (t_1 and t_2 fires, $m_{13} = n_1 - 1$, $m_{14} = n_2 - 1$, $m_s^1 = 0$), and run into IG and IIG respectively (t_3 and t_{10} fires, $m_2 = m_3 = 1$, $m_1 = m_4 = 0$), the current state of the net is shown in Fig. 11.

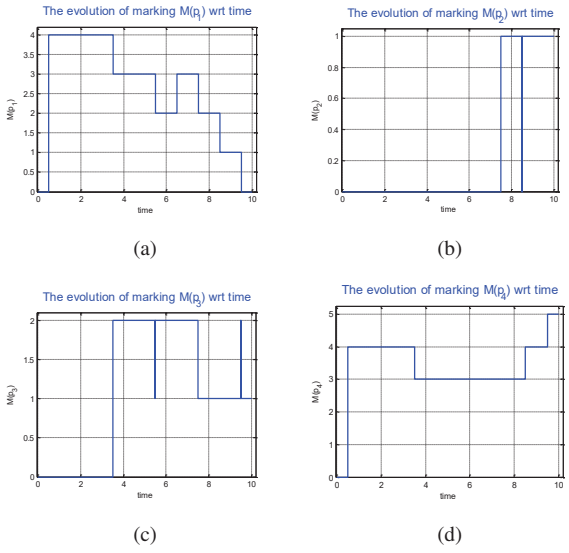
Since $m_s^1 = 0$, t_1 and t_2 are not able to fire, the constraint (a) is realized: No more than two trains should be in the station and approach/departure sections simultaneously. Without the monitor place p_s^1, t_1 and t_2 can fire at random, which will violate safety and put trains at risk. Monitor places p_s^2, p_s^3, p_s^4 and p_s^5 implement the specifications on switches and signals, corresponding to GMEC in Eq. (2) to (5) that enforce constraint (b).

4 Simulation and Analyze

In this section, to evaluate the performance of the control logic as well as the carrying capacity of a station, we simulate the evolution of the controlled system and test the effect that the monitor place generates. A toolkit loaded in MATLAB called *HYPENS* will be used to carry out the random Petri net simulation. We slightly modify the previous model in Section 3 for the simulation. Two interstation line places are merged into p_{13} (respectively, in the controlled net with extra five monitor places, p_{20}) equivalent to the abstraction of all external lines. By analyzing m_{13} (respectively, m_{20}) during the simulation we can see the carrying capacity of a station.

4.1 Safety Analysis

To verify whether the controlled net meets safety requirements, separately analysis of each GMEC is needed. Because of limited space, we only introduce the simulation verification of GMEC in Eq. (1) in the following. Assign a fixed delay to each transition, and let $m_{13} = 8$ (respectively, $m_{20} = 8$). The evolution of markings of $p_1 \sim p_4$ in the original net and controlled net is shown in following figures.



(a) (b) (c) (d)

Fig. 12: Evolution graphs of tokens in $p_1 \sim p_4$ in the original net

It is clear that the controlled net satisfies GMEC in Eq. (1). Notice that *HYPENS* can also inspect some specificities such as the average marking, and the probability distribution of markings, etc. Therefore results that can be observed directly are diversiform. It means that we may achieve the analysis result through another angle.

4.2 Efficiency Analysis

Assign an exponentially distributed delay to each transition. The evolution of tokens in p_{13} and p_{20} is shown in

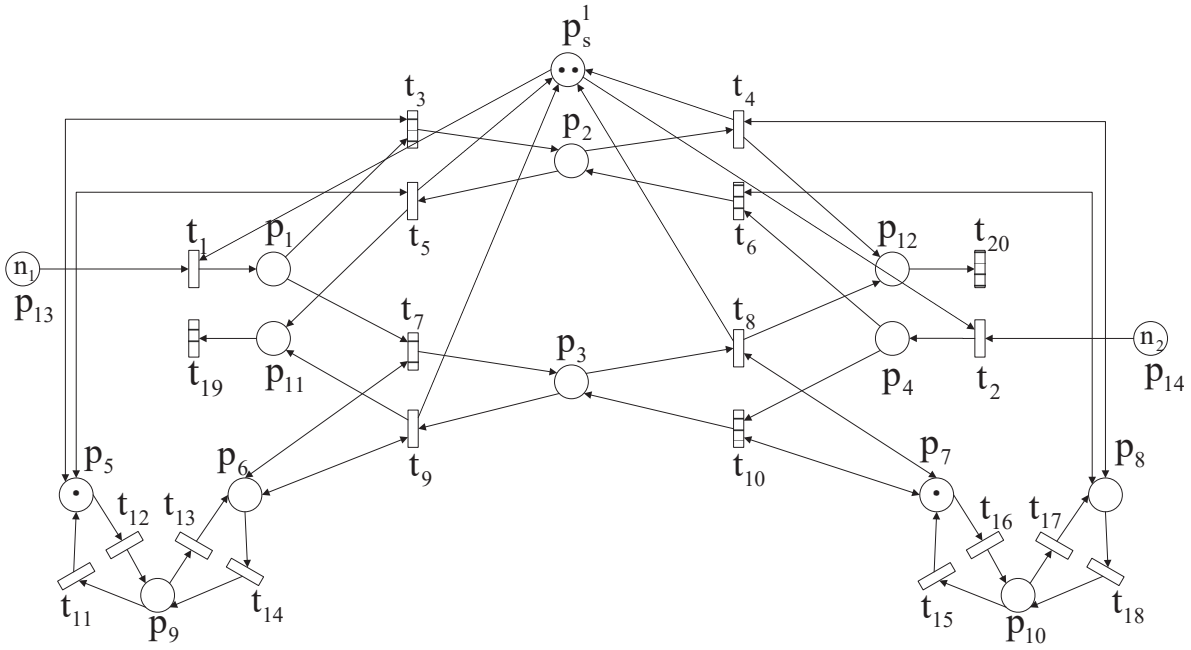


Fig. 10: an example of p_s^1 controlling the net(a)

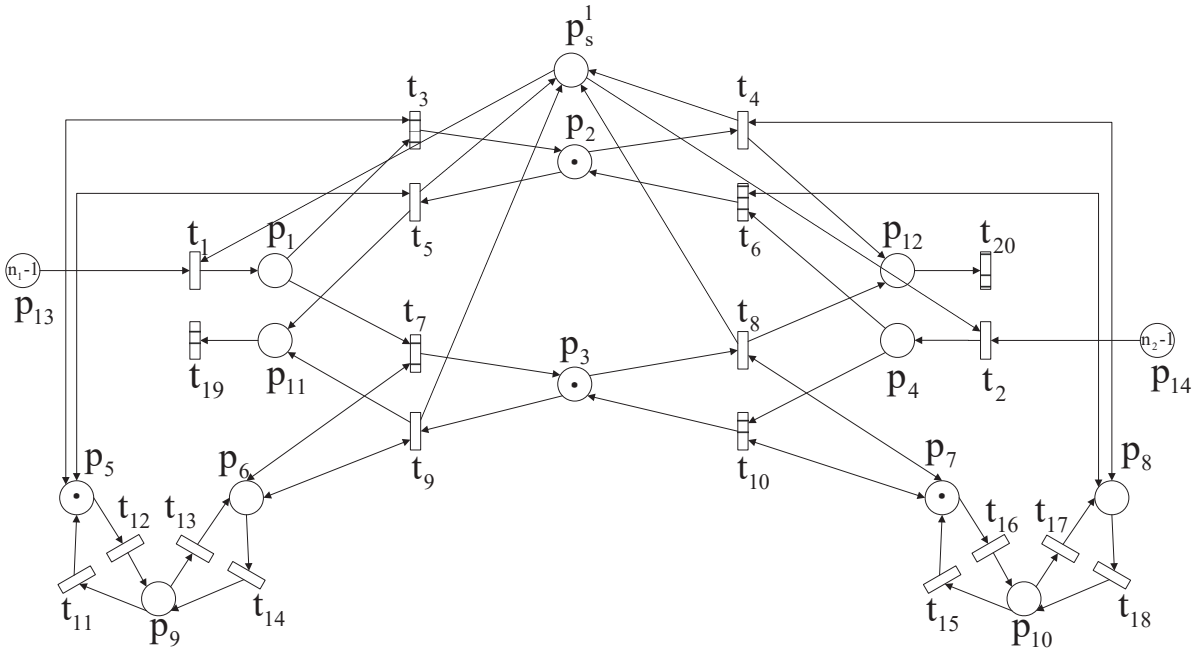


Fig. 11: an example of p_s^1 controlling the net(b)

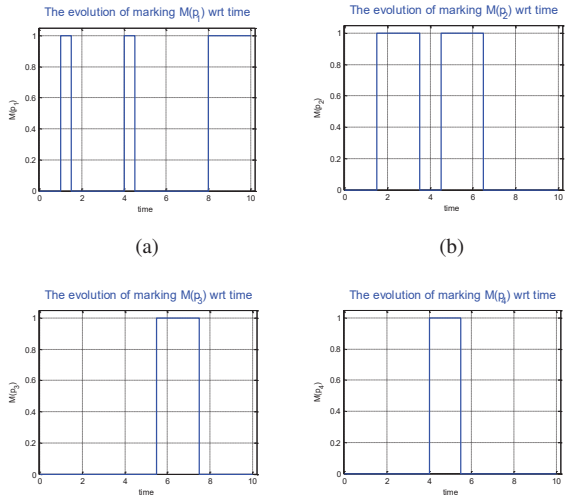
Fig. 13.

By observing the marking evolution cycle time of the two places that transfer from 2 to 2, we can find that the flow efficiency of tokens in interstation line places doesn't reduce after adding monitor places, which means that the control logic works meanwhile ensuring the efficiency of the system.

5 Conclusion

The modular modeling method is applicable to a general station. It means that we can quickly establish the model from the actual structure. Once the Petri net model is determined and the specifications are given, its sufficient to design its monitors. During the formulation of specifications in the form of GEMCs, we abstract interlocking conditions in term of model's features. Hence the superiority of interlocking in engineering is inherited. The control logic finally obtained in this paper is more concise than the fundamental rules in the traditional interlocking, while retaining the constraint effect of them.

This series of methods is of high reusability and realizes the parameterization and generalization. In the future research, we consider: (1) to model and design control logic for the station with more complex device compositions; (2) to take other technical operations scenario and its requirements into account; (3) to develop softwares that complete more indepth analysis on Petri nets.



(a) (b)

(c) (d)

Fig. 13: Evolution graphs of tokens in $p_1 \sim p_4$ in the controlled net

References

[1] Y. Yang, L. Liu and X. Chen, Station Signal Control System, Chengdu: Southwest Jiaotong University Press, 1–101, 2012.

[2] B. Chen, F. Wu, Research on formal modeling of railway signal interlocking logic, in *Journal of the China Railway Society*, (6): 50–54, 2002.

[3] F. Han, Formal modeling and analysis of interlocking software based on time-colored colored Petri nets, Tongji University, 2007.

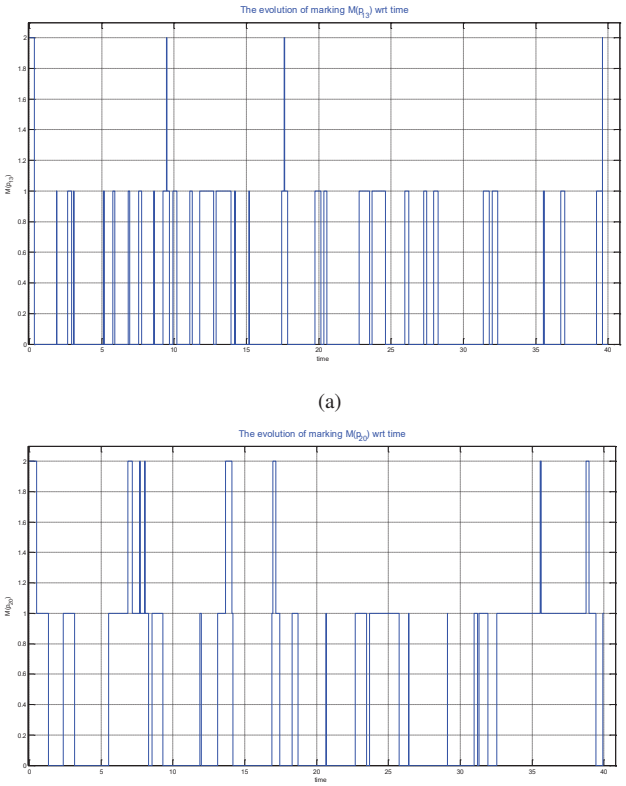
[4] A. Giua, C. Seatzu, A Systems Theory View of Petri Nets, in *Control Theory and Applications*, Springer Berlin Heidelberg, 99–127, 2007.

[5] A. Giua, C. Seatzu, Modeling and Supervisory Control of Railway Networks Using Petri Nets, *IEEE Transactions on Automation Science and Engineering*, 5(3): 1545–1555, 2008.

[6] A. Giua, F. DiCesare, and M. Silva, Generalized Mutual Exclusion Concstraints on Nets with Uncontrollable Transitions, in *IEEE International Conference Systems, Man, Cybernetics*, 974–979, 1992.

[7] C. Ramchandani, Analysis of Asynchronous Concurrent Systems by Petri Nets, in *Cambridge*, 15(4): 5–13, 1973.

[8] J. Guo, Y. Wei, and L. Liu, Railway Signal Infrastructure Equipment, Chengdu: Southwest Jiaotong University Press, 53–190, 2008.



(a)

(b)

Fig. 14: Evolution graphs of tokens in p_{13} and p_{20}