ISSN:2231-0711

Mandira Banik et al | IJCSET |July 2013 | Vol 3, Issue 7,249-255

Railway Network Modelling Using Petri Nets

Mandira Banik, Sudeep Ghosh

Guru Nanak Institute of Technology,
West Bengal,India

Abstract - This paper deals with the use of Petri nets in modelling railway network and designing appropriate control logic for it to avoid collision. Here, the whole railway network is presented as a combination of the elementary models – tracks, stations and points (switch) within the station including sensors and semaphores. We use generalized mutual exclusion constraints and constraints containing the firing vector to ensure safeness of the railway network. In this research work, we have actually introduced constraints at the points within the station. These constraints ensure that when a track is occupied, we control the switch so that another train will not enter into the same track and thus avoid collision.

Keywords: Petri nets, safeness constraints, firing vectors, asynchronous systems.

INTRODUCTION

Modelling of complex systems for better understanding is a very wide-spread research activity and researchers all over the world are trying to model more and more complex systems. Several tools have also been developed for this purpose and Petri Net [1] is one of such tools used for quite some time to model various asynchronous systems. Railway network is considered as a very complex system and appropriate modelling of it to avoid collision is of very high importance. In [3], Giua and Seatzu have used Petri Net to model railway network and developed some expression of constraints to avoid collision. However the paper does not include the situation when there is a train already in a track and another train is in the input line. In our research work, Giua and Seatzu model has been augmented to avoid collision in such cases.

BRIEF OVERVIEW OF PETRI NETS

1.1 Definition of Petri nets

Petri net is a formal modelling technique and consists of places, transitions and arcs directed from either places to transitions or transitions to places, representing flow relations. Pictorially, places are drawn as circles and transitions as boxes or bars (Figure 1). Arcs are labelled with weights. Labels for unity weights are generally not given. A place from which a directed arc goes to a transition is called input place of that transition. A place, to which there is a directed arc from a transition, is called output place of that transition. A Petri net is given a state by marking its places with tokens. A marking M is a function [9] that assigns to each place a non negative integer representing number of tokens at that place. In graphical representation, black dots in circles denote tokens in places. Petri Nets may formally be defined as [1]

A Petri net is a 5-tuple - (P, T, F, W, M_0)

where :

$P = \{ p_1, p_2, \ldots, p_m \}$ is a finite set of places,
$T = \{ t_1, t_2, \ldots, t_n \}$ is a finite set of transitions,
$F \subseteq (P \times T) \cup (T \times P) \}$ is a set of arcs (flow relations),
$W: F \rightarrow \{1, 2, 3, \ldots\}$ is a weight function,
$M_0: P \rightarrow \{0, 1, 2, 3, \ldots\}$ is the initial marking,
$P \cap T = \varnothing$ and $P \cup T \neq \varnothing$.

where, $(P \times T)$ and $(T \times P)$ denotes the ordered pair of sets P and T.

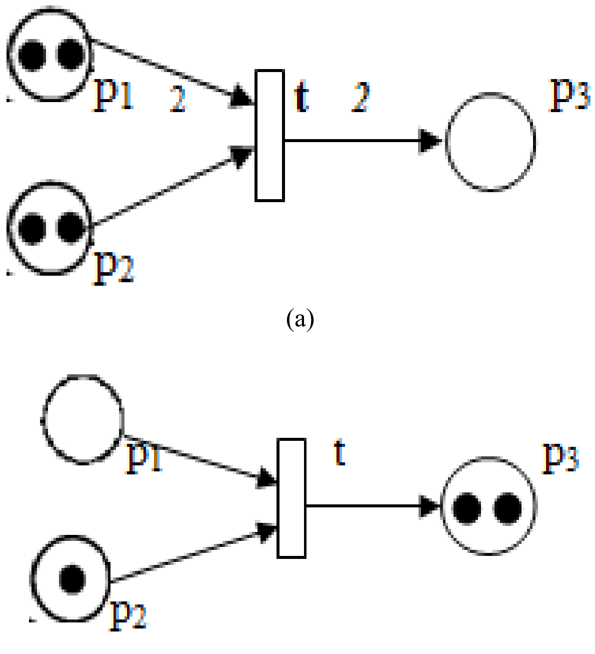
By changing distribution of tokens on places the occurrence of events (transitions) may be reflected. The flow of tokens in Petri net are governed by the following rules [2]

- A transition t is said to be enabled if each input place p of t contains at least the number of tokens equal to the weight of the directed arc connecting p to t.
- A firing of an enabled transition t removes from each input place p the number of tokens equal to the weight of the directed arc connecting *p* to *t.* It also deposits in each output place *p* the number of tokens equal to the weight of the directed arc connecting *t* to p – giving a new marking.

There are also some high-level Petri nets – timed Petri net [6][7], coloured Petri net [8] etc.

1.2 Applications of Petri nets

Petri nets are used for a very wide variety of applications. Specially they are well-suited for systems those are concurrent, asynchronous, distributed, parallel and nondeterministic [1]. In [6], the author presents how timed Petri net is used to model the GPRS charging system and to analyze its performance when the system works in the normal status and how it handles the maximum supportable busy hour call attempts of the GPRS network. [7] also depicts application of timed Petri net to model traffic signal control where two separate subnets are designed for signal indications (green, yellow, and red) and the transitions between indications (one light becomes red before another becomes green). Besides these, Petri nets have been successfully applied in modelling and performance analysis of communication protocols, flexible manufacturing systems, sequence controllers, distributed-software systems, distributed-database systems, multiprocessor systems, fault-tolerant systems, programmable logic and VLSI arrays [1] etc. Using Petri nets dynamic behaviour of the systems can also be studied [2].



(b)

Figure 1: The marking (a)before firing of enabled transition t (b)after firing of t

1.3 Generalized Mutual Exclusion Constraints (GMECs)

(Un)controllable and/or (un)observable transitions may be used in many applications, for example, in the railway network problem. Uncontrollable transitions are transitions that cannot be disabled by any control action and unobservable transitions are transitions whose firing cannot be directly detected or measured [3]. In [3], generalized mutual exclusion constraints (GMECs) has been utilized for expressing collision avoidance constraints for designing controllers for Petri nets with (un)controllable and/or (un)observable transitions. A generalized mutual exclusion constraint [5] is a linear constraint that limits the weighted sum of tokens in some places of a Petri net. GMECs are used for expressing constraints that states which system state i.e., what marking will be allowed to achieve, for example, safeness in the railway model. A GMEC may be enforced adding to the net a single control structure consisting in a new place called monitor place.

1.4 Constraints containing Firing Vector

Constraints may contain both marking and firing vector elements. These constraints represent that an event (modelled by a transition) can occur at a system's state governed by the marking vector elements, if that system's state allows it to occur [4]. Firing vector q denotes which transition is the next to fire when the Petri net is going from one marking to another marking. It is an nx1 column vector of 0's and an only one, where n is the number of transitions. Now depending on whether a system's state allows a transition t_j to occur, jth element of firing vector

$q_j = 1$ indicating transition t_j is enabled and

$q_j = 0$ indicating transition t_j is not enabled.

Railway network problem

The whole railway network is presented as a combination of the elementary models - tracks, stations and points within the station [3].

1.5 Track model

A railway track is shown in Figure 2. Here the track is divided into three segments – α1, α2, α3. Trains can go on this single track in left or right direction. The track includes sensors and semaphores. Sensors A and B detect the passage of train in both directions [3]. The passage of a train directed rightward (from segment α1 to segment α2) can be stopped and also can be detected by semaphore A. Semaphore B does the same for the train directed leftward (from segment α3 to segment α2).

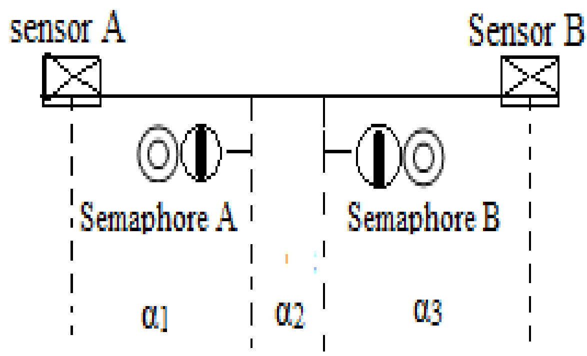


Figure 2: A Railway Track

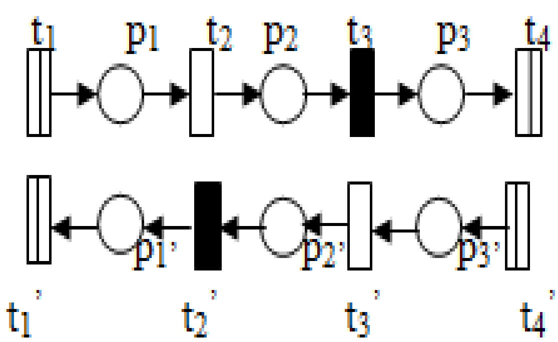


Figure 3: Petri net model of the track of Figure 2

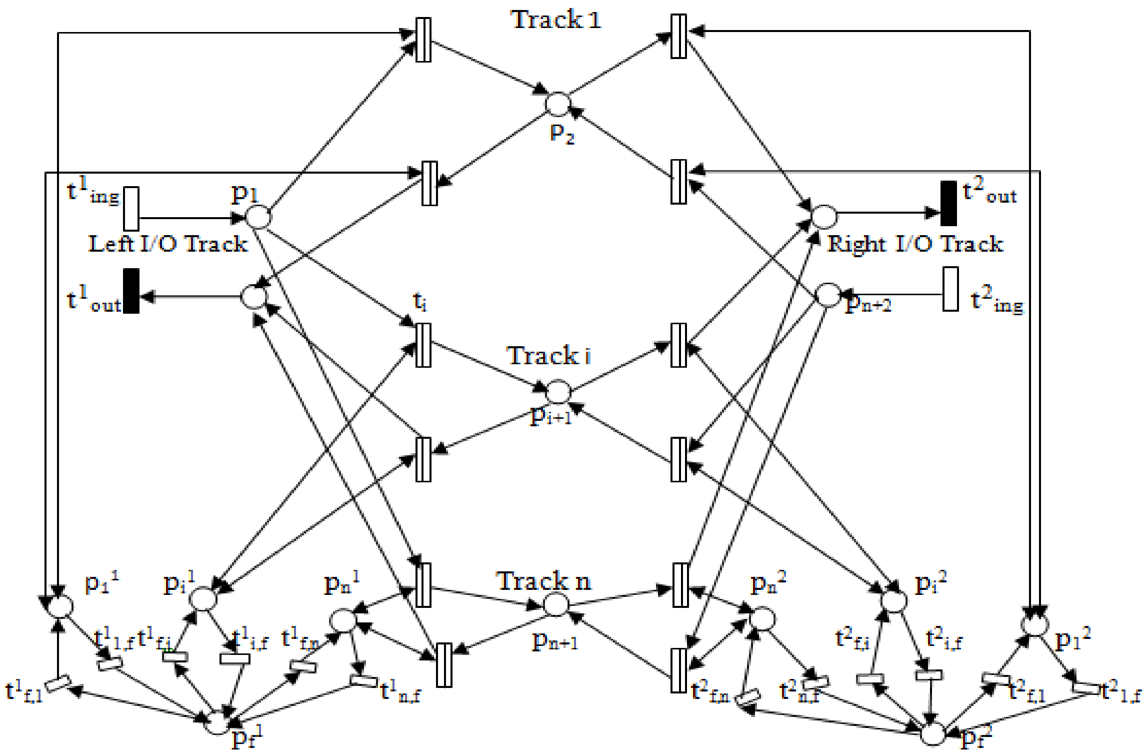


Figure 4: The simplified Petri net model of an n-tracks railway station

In the Petri net model of the track shown in Figure 3, there are two sets of places (p_1, p_2 ,p_3 and $p_{1'}$,$p_{2'}$, $p_{3'}$) and transitions(t_1, . . . , t_4 and t_1', . . ,t_4') , representing flow of trains in right and left directions respectively. Each couple of places p_i , p_i' are used to represent segment α_i of the track i.e., the marking of p_i or p_i' indicates the presence of a train directed rightward or leftward in segment α_i. Transitions denote the passage of a train from one segment to another in a certain direction.

In the Petri net model, transitions may be (un)controllable and/or (un)observable so as to represent sensors and semaphores. A both controllable and observable transition represents a semaphore [3]. For example, transitions t_2 and t_3' in Figure 3 correspond to semaphore A and semaphore B respectively: transitions t_2 and t_3' are controllable and observable to denote that at those points of the net the presence of trains can be detected and controlled (their transit can be forbidden) by the corresponding semaphores. Transitions t_1, t_4, t_1' and t_4' represent sensors as a sensor can only detect the passage of train. Transitions t_2' and t_3 are uncontrollable and unobservable.

1.6 Railway station model

Figure 4 can be referred for the Petri net model of an n-tracks railway station. The station [3] consists of n+2 different stretches among which there are n inner tracks (tracks 1, . .,i, . ,n) and two input/output (I/O) tracks on the left and right side. The controllable and observable transitions t^1_{ing} and t^2_{ing} fires when a train enters into the station, while the uncontrollable and unobservable transitions t^1_{out} and t^2_{out} fires to denote the exit of a train from the station.

Now depending on the position of the points a certain track in the station is enabled and trains may be routed to that track. The Petri net models of the points (switch) of the n-tracks station are given by the two subnets on bottom left and on bottom right of Figure 4. The superscripts 1 and 2 are used to indicate places and transitions relative to the left and the right points, respectively. The models represent points (switch) with n possible tracks. Consider for left points. When place p_i^1 (i =1, . . ,n) is marked, left points is connected to track i and trains on the left I/O track may be directed to that track. Actually, transition t_i fires (train enters in track i) if both of its input places p_1 and p_i^1 are marked. But, if place p_f^1 is marked, no train can cross the points. Points position is changed among points places p_i^1 (i=1, . . ,n) by firing transition $t^1_{f,i}$, to enable different paths. Same is for the right points. Place p_f^1 or p_f^2 is marked when the enabled path is being changed or points are switched off.

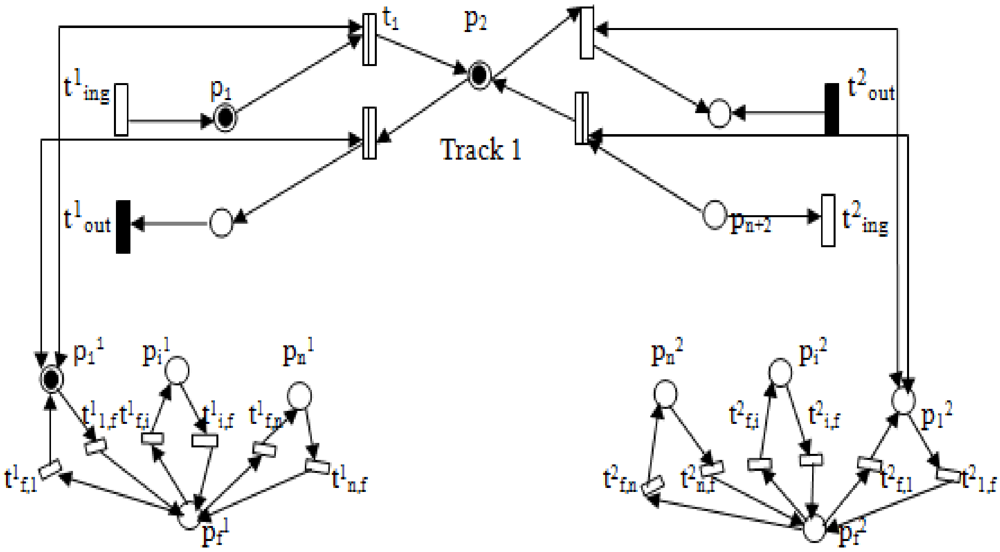


Figure 5: Petri net model of a part of the n-tracks railway station

1.7 Control logic for Tracks and Stations

1.7.1 Safeness of tracks

The collision avoidance constraints can be written as GMECs to ensure that each couple of places p_i and p'_i in Figure 3, representing the same segment of a single-track cannot contain tokens simultaneously, and moreover each place never has more than one token. The GMEC is [3] -

$$m_i + m_i' <=1$$

relative to a segment of a track. Here, m_i denotes the marking of place p_i. Same constraints can be applied for the tracks of the station also.

Because of the presence of the uncontrollable and/or unobservable transitions in the railway network model, a more restrictive constraint has been developed in [3]. That constraint ensures safeness, but imposes too restriction [3] if applied to the places corresponding to tracks within a station.

So a better solution is to develop a new set of station admission rules or constraints, some of which also include the firing vectors.

1.7.2 Station admission rules

A simplified Petri net model of the station is given in Figure 4. The *station admission rules* can be described as [3]

- No more than n trains should be simultaneously present in the station including the I/O tracks. It can be written as GMEC -

$$\sum_{i=1}^{n+2} m_i <= n$$

- No train may arrive from outside entering the left (or right) I/O track if one inner track is nonempty and the left (or right) points is enabling the flow of trains towards the nonempty track.

$$q^1_{ing} + m_{i+1} + m_i^1 <= 2, \quad i= 1, . . . ,n \quad \text{(for left side)}$$
$$q^2_{ing} + m_{i+1} + m_i^2 <= 2, \quad i= 1, . . . ,n \quad \text{(for right side)}$$

For example, if a train exists in inner track 1(place p2 is marked) and the left points is connected to track 1(p_1^1 is marked) then t1ing cannot fire, otherwise an accident occurs. It is applicable for right side also. This case is generalized for n-tracks in the above equations that control the entrance of trains in the station.

Limitation of the railway network model

The Petri net model described in section 3 takes care for the selection of vacant tracks among various available tracks. However it does not include successive incoming of trains in the presence of train in station as described here. For example, consider the case in Figure 5. A train exists in inner track 1(place p_2 is marked) and a train also is in the left I/O track (place p_1 is marked) and the left points is connected to track 1(p_1^1 is marked). Now t_1 fires as both of its input places (p_1 and p_1^1) are marked, entering the train in the left I/O track in track 1 and a collision occurs. Same thing will happen for the right side also. Thus the given station admission rules in section 3.3 regulate only the input of trains in the station, not the points within it and that may result in an accident.

Also consider these cases. Assume that n-1 inner tracks contain trains and the left I/O track contains a train directed rightward. If a further train arrives from outside entering the right I/O track, a collision will eventually occur. Same thing will happen if the right I/O track contains a train directed leftward and a new train arrives from outside entering the left I/O track, while there exist trains on n-1 inner tracks.

Proposed model for revised station admission rules

The limitation is addressed here. This accident can be avoided if the left points is not connected to track 1 when trains exist on both track 1 and left I/O track. Now token can be deposited on place p_1^1 only by firing transition $t^1_{f,1}$ of left points model (see Figure 5). So to make p_1^1 not to be marked, we impose constraints on firing of $t^1_{f,1}$. The constraint is $t^1_{f,1}$ cannot fire if both places p_1 and p_2 are marked simultaneously. This is written as following inequality –

$$q^1_{f,1} + m_1 + m_2 <= 2$$

Thus the points can be regulated. We generalize this constraint for the n-tracks railway station to regulate the points within station to ensure safeness. We develop the following generalized constraints –

$$q^1_{f,i} + m_1 + m_{i+1} <= 2, \quad i = 1, \ldots, n \quad \text{(for left side)}$$
$$q^2_{f,i} + m_{n+2} + m_{i+1} <= 2, \quad i = 1, \ldots, n \quad \text{(for right side)}$$

which imply that if both places p_1(or p_{n+2}) and p_{i+1} are marked, $t^1_{f,i}$ (or $t^2_{f,i}$) cannot fire to make point place p_i^1(or p_i^2) (i= 1, . . . ,n) marked (see Figure 4). That is the left (or right) points corresponding a nonempty inner track cannot be enabled if train exists in the left (or right) I/O track.

Case Study

The railway model that we have discussed till now, considers only one input /output track on each of the left and right sides of the railway station. But in most cases in real life, we find more than one input/output track on each side of the railway station. In this section, an effort has been made to extrapolate the previously discussed railway station model into such a railway station model with more than one I/O track on each side of the station.

In this model, each I/O track is not connected with all inner tracks of the station as in the previous model. Here, the first I/O track on the left side of the station in Figure 6 is connected with only inner track 1 and inner track 2. The second I/O track on the left side of the station is connected with three inner tracks of the station. Similarly on the right side, three inner tracks are connected with the first I/O track and so on. The Petri net model of the railway station scheme shown in Figure 6 is given in Figure 7. In that portion of the Petri net model where the first I/O track on the left side is connected with two inner tracks, a simplified points model [7] is used instead of the previously discussed points model. Same points model is used where inner track 5 and inner track 6 are connected with the last I/O track on the right side. The cycles $p_{u,1}$, $t_{d,1}$, $p_{d,1}$, $t_{u,1}$ and $p_{u,2}$, $t_{d,2}$, $p_{d,2}$, $t_{u,2}$ model the points. When places $p_{u,1}$ and $p_{u,2}$ are marked, trains are directed to the up track or may leave the up track. On the other hand, when places $p_{d,1}$ and $p_{d,2}$ are marked, trains are directed to the down-track or may leave the down-track.

But in the case where I/O tracks connect with three inner tracks, the previous points model has been used with n=3. Now for this model the safeness constraints are as follows. For track 1 and track 2 if trains come from left side then for that case the safeness constraints will be –

$$q_{ing,1} + m_{u,1} + m_1 <= 2$$
$$q_{ing,1} + m_{d,1} + m_2 <= 2$$
$$m_7 + m_1 + m_{u,1} <= 2$$
$$m_7 + m_2 + m_{d,1} <= 2$$

For track 1, track 2 and track 3 if trains come from left side then for that case the safeness constraints will be –

$$q_{ing,2} + m_3 + m_3^1 <= 2$$
$$q_{ing,2} + m_4 + m_4^1 <= 2$$
$$q_{ing,2} + m_5 + m_5^1 <= 2$$
$$q^1_{f,3} + m_8 + m_3 <= 2$$
$$q^1_{f,4} + m_8 + m_4 <= 2$$
$$q^1_{f,5} + m_8 + m_5 <= 2$$

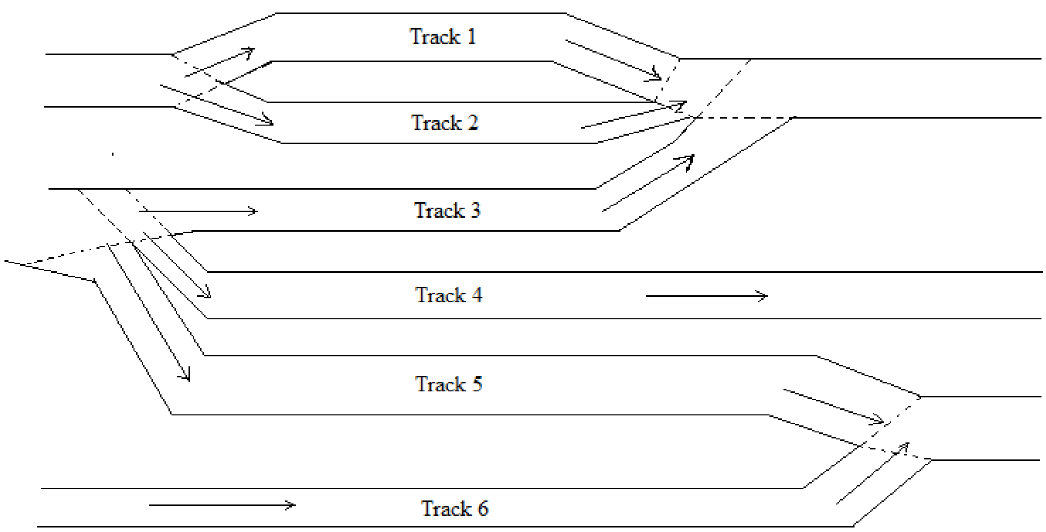


Figure 6 : A railway station scheme with more than one I/O track on each side

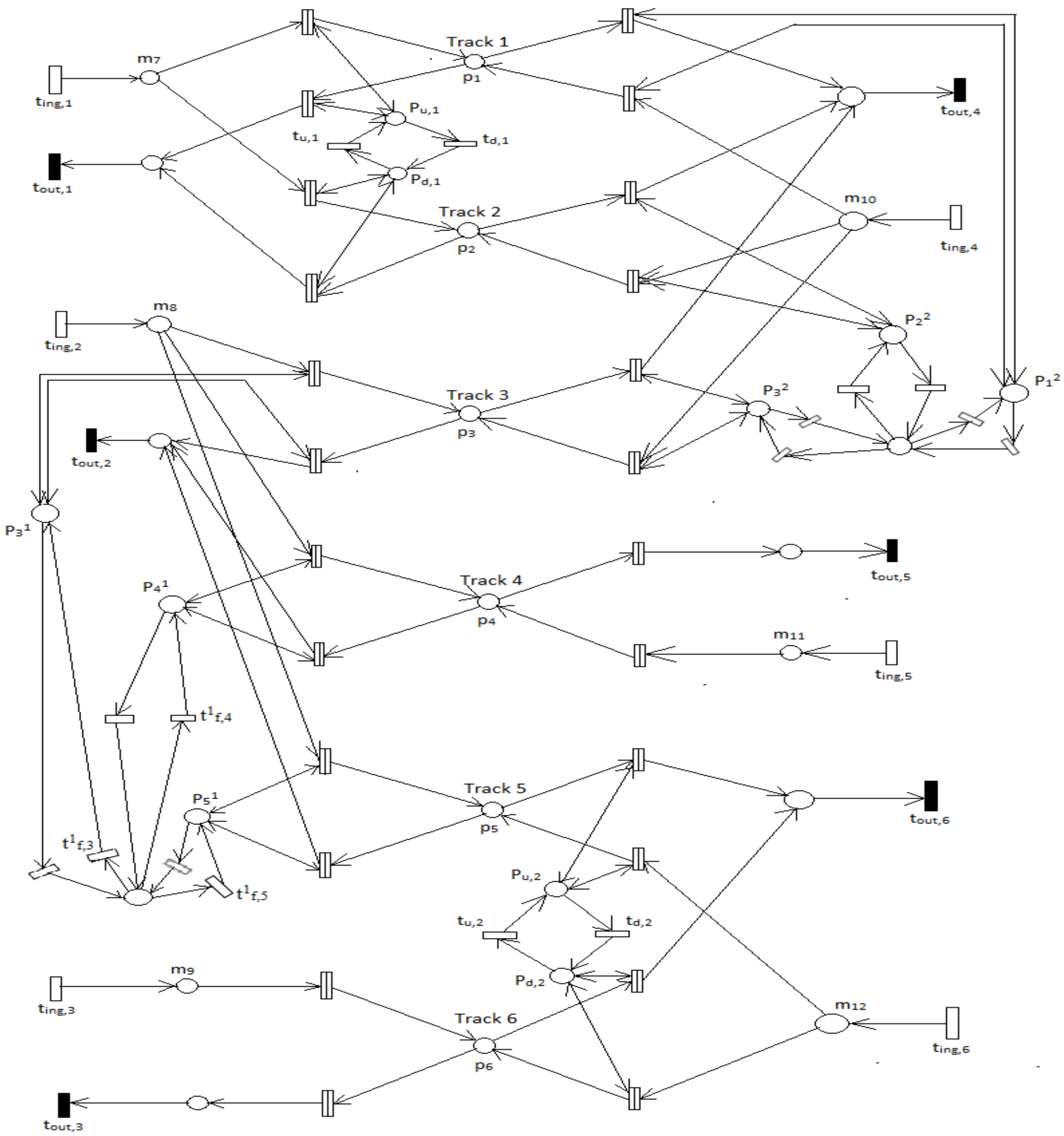


Figure 7: Petri net model of railway station scheme of figure 6

Conclusion

Modelling with Petri net is being considered as one of the very helpful tools to detect collision in a railway network problem. Following the design model of Giua and Seatzu in [3], in this research work we have augmented the model by introducing constraints at the points (switch). This ensures that when a track is occupied, we control the switch so that another train will not enter into the same track and thus avoids collision.

References

[1] Tadao Murata, "Petri Nets: Properties, Analysis and Applications", Proceeding of the IEEE, Vol 77, No. 4, April 1989.

[2] Richard Zurawski and Meng Chu Zhou, "Petri Nets and Industrial Applications: A Tutorial", IEEE Transactions on industrial electronics, vol. 41, no. 6, December 1994.

[3] Alessandro Giua and Carla Seatzu, "Modeling and Supervisory Control of Railway Networks Using Petri Nets", IEEE Transactions on automation science and engineering, vol. 5, no. 3, July 2008.

[4] J. O. Moody, K. Yamalidou, M. D. Lemmon, and P. J. Antsaklis, "Feedback Control of Petri Nets Based on Place Invariants", *Automatica*, vol. 32, no. 1, pp. 15–28, Jan. 1996.

[5] Maria Pia Fanti, Alessandro Giua and Carla Seatzu, “Generalized Mutual Exclusion Constraints and Monitors for Colored Petri Nets”, IEEE 2003.

[6] Jiacun Wang, “Charging Information Collection Modeling and Analysis of GPRS Networks”, IEEE Transactions on systems, man, and cybernetics—part c: applications and reviews, vol. 37, no. 4, July 2007.

[7] George F. List and Mecit Cetin, “Modeling Traffic Signal Control Using Petri Nets”, IEEE Transactions on intelligent transportation systems, vol. 5, no. 3, September 2004.

[8] K. Jensen, “Coloured Petri Nets: A High-level Language for System Design and Analysis”, In:G. Rozenberg (ed.): Advances in Petri Nets 1990, Lecture Notes in Computer Science Vol. 483, Springer-Verlag 1991, 342-416.

[9] C Ghezzi, M Jazayeri and D Mandrioli, “Fundamentals of Software Engineering”, PHI