Modeling and supervisory control of railway networks using Petri nets*

Alessandro Giua, and Carla Seatzu

Department of Electrical and Electronic Engineering, University of Cagliari, Italy
e-mail: {giua,seatzu}@diee.unica.it.

Abstract

In this paper we deal with the problem of modeling railway networks with Petri nets so as to apply the theory of supervisory control for discrete event systems to automatically design the system controller. We provide a modular representation of railway networks in terms of stations and tracks including sensors and semaphores. We first ensure safeness and local liveness imposing both Generalized Mutual Exclusion Constraints and constraints also involving the firing vector. The detailed model used in this first step can be abstracted, considering a higher-level description of a railway network that belongs to the class of ES^{2}PR (Extended Simple Sequential Process with Resources) nets and show that global liveness may be enforced by adding appropriate monitor places designed using siphon analysis. In our approach this can be done without an exhaustive computation of all siphons and we can characterize the cases in which the procedure can be recursively applied, giving a simple test for closed loop net to remain an ES^{2}PR net.

Note to Pratictioners

The automation of railway systems (and in general of large scale distributed discrete event systems) is a complex problem that can be decomposed in several subproblems. It is necessary to be able to build a model of the plant, to simulate it, to design appropriate control logic to ensure that abnormal situations, such as collisions or deadlocks, never occur. In this paper we present an approach based on a graphical model that is familiar to the automation community: Petri nets. Our approach presents two main advantages. Firstly, we are able to address all these subproblems within a single formalism, thus making it worthwhile for the practitioner the effort of investing some valuable time to learn more about Petri nets. Secondly, the mathematical approaches used to determine the solution in all these steps are based on simple matrix computation and standard operation research tools, that are easily accessible to practitioners with a background in computer and systems engineering. Another interesting problem that we also plan to solve using Petri nets — but that is not addressed in this paper — is that of scheduling the railway system so as to optimize its efficiency.

*Published as: A. Giua, C. Seatzu, "Modeling and supervisory control of railway networks using Petri nets," *IEEE Trans. on Automation Science and Engineering*, Vol. 5, No. 3, pp. 431-445, July 2008.

1 Introduction

Railways form one of the most important part of transportation systems and their constantly improving safety record, makes them a very attractive option compared with other modes of transport [20, 22, 29, 30, 37]. As a result, the overall complexity of railway systems increases, and hence greater demands are placed on the control logic of these systems [22].

Consider, to mention just one example, the case of the European Union. As the EU is opening to counties of Eastern Europe, it is also making substantial investments to revitalize the railways and plans to achieve the following objectives by 2020 [7]:

- for rail to increase its market share of passenger traffic from 6% to 10% and of goods traffic from 8% to 15%;
- a trebling of manpower productivity on the railways;
- a 50% gain in energy efficiency;
- a 50% reduction in emissions of pollutants;
- an increase in infrastructure capacity commensurate with traffic targets.

The specification, analysis and implementation of railway control logic is an important activity because its failure can lead to railway accidents and loss of human life [31]. At present time, this activity is even more important because railway networks are often large, the speed of trains and traffic density is increasing, and activities within networks are taking place concurrently and at geographically different locations [22, 29].

Note that the control of a railway network can be divided into two parts: *logical control* and *performance control*. The first one deals with structural problems, and imposes the satisfaction of a series of safeness constraints (collision avoidance) and liveness constraints (deadlock freeness). The second one, is related to the operation of the network and is concerned with problems such as scheduling both the departures and the stops, so as to optimize the efficiency of the net. In this paper the attention is uniquely devoted to the design of control logic for logical control.

Very different approaches have been used to design efficient controllers for railway networks. For a detailed treatment of the subject, the interested reader may consult the literature [20, 22, 29, 30, 37] and follow links provided on internet sites [5, 6].

In this paper, we focus our attention on the modeling and control of railway networks with Petri nets [28], that provide a powerful framework for the analysis and control of distributed and concurrent systems. Some of the advantages of Petri nets as models for discrete event control include [19]: graphical representation, solid foundations based in mathematics, the existence of simulation and formal analysis techniques, and the existence of computer tools for simulation, analysis and control.

The literature on modeling and analyzing railway systems using Petri nets is not extensive and a good survey is given by Janczura in [22]. The idea of applying Petri nets theory goes back to Genrich [15], then it was revisited in [4, 23] and in [22] where coloured Petri nets have been used. In [32] Ren and Zhou presented Petri net models for the tactical scheduling of railroad operation. In particular, they provide a criterion to establish whether a given target schedule is feasible or not. Other significant contributions

in this field are due to Decknatel and Schnieder [9], to Di Febbraro *et al.* [11] who used hybrid Petri nets to model transportation systems, and to Moen Hagalisletto and Yu [25]. The control approach proposed in [25], even if it uses Petri nets, is essentially based on graphical specification language, and appropriate composition rules that require a certain amount of algebra. The notions of controllable and observable transitions, as well as that of monitor places, are not used here.

In a recent work [14] we have also discussed the control of train networks using colored Petri nets. Unlike the approach presented in this paper, however, we only considered the case of systems with controllable and observable transitions which makes the control problem much simpler.

In this paper the use of Petri nets as a modelling formalism allows us to use within a single framework several different approaches such as supervisory control, monitor design, siphon and deadlock analysis that are necessary at different stages of the control logic design. The original contribution of our paper concerns three aspects described in the following.

1.1 Modeling

We first provide a modular representation of railway networks in terms of stations and tracks including sensors and semaphores. The overall model is a place/transition (P/T) net whose transitions may be (un)controllable and/or (un)observable, following the paradigm of supervisory control [38]. As an example, a controllable and observable transition is associated to the crossing of a section controlled by a semaphore, where a traffic signal that may stop a train and a sensor that detects the passing of the train, is placed. The possibility offered by supervisory control to handle such primitives as uncontrollable and unobservable events leads to a very simple model that can be directly exploited in the subsequent phase of control synthesis. The use of Petri nets allows a modular representation of railway networks where each of the composed subnets corresponds to a station or a track.

It is also important to mention that at different levels of the control logic design, different models may be suitable. In this paper we use two main models: at a lower level we use a detailed model to solve the safeness problem; at a higher-level we use an abstracted one, that we call *skeleton net*, to solve the liveness problem.

1.2 Control

There exist several techniques for automatically designing controllers for P/T nets with uncontrollable and/or unobservable transitions [19]. In particular, we show how collision avoidance constraints can be expressed as *Generalized Mutual Exclusion Constraints* (GMECs) [16] and how the corresponding controller takes the form of a set of monitor places that can be computed using Moody's parametrization [27].

However, it is well known that in general a monitor–based solution to a GMEC may not be maximally permissive. We show that this is the case for constraints related to the arrival and departure of a train from a station, where the designed monitor controller is too restrictive and leads to a blocking condition when two trains going in opposite directions meet at a station. We call such a situation a *local deadlock* because it does not depend on the global state of the network (i.e., on the presence of other trains)

but only on the local configuration of the two trains. The maximally permissive control policy for this configuration corresponds to a set of legal markings that is not convex and thus cannot be enforced by a monitor place. However, we are able to solve this problem designing a control structure that is still very simple and takes the form of a "monitor with self-loops".

A nice feature of the overall approach is that the whole control problem can be divided into a certain number of local sub–problems, thus making the proposed control procedure suitable even for large dimensions cases.

1.3 Deadlock avoidance

Although the controller designed in the previous phase is locally deadlock free, as the number of trains admitted into the network increases different blocking conditions may occur depending on the exact train distribution. This is what we call a *global deadlock*.

There exists a rich literature on the design of deadlock avoidance controllers for discrete event systems in general and for Petri nets in particular (see for instance [13, 36, 39, 43] and the references therein) and we are well aware that some of the approaches already presented in the literature may also be applied to the case at hand.

Most of the results on deadlock control have been proposed within the framework of Automated Guided Vehicles (AGV) systems. As an example, a significant contribution in this area is due to Wu and Zhou [43] who presented a Colored Resource-Oriented Petri Net (CROPN) modeling method to deal with conflict and deadlock arising in AGV systems. A control policy suitable for real-time implementation is also presented here.

Other important procedures of deadlock prevention have been proposed within the framework of flexible manufacturing systems. Among these we mention in particular the work by Li and Zhou [24]. This paper is based on siphon analysis, and its main contribution consists in exploring ways to miminize the addition of new places to prevent siphons to be empty, while achieving the same control purpose.

In this paper, however, we adopt a simple approach that is computationally viable and is tailored to the problem at hand. These are the main features of the procedure we propose.

- We simplify the model of railway system considering an abstracted net (that we call *skeleton net*) and we show that it belongs to the class of ES^{2}PR (Extended Simple Sequential Process with Resources) nets. In simple words, an ES^{2}PR net is obtained by a strongly connected state machine where all circuits contain a common place p_0, adding a set of places representing shared resources (see Definitions 2 and 3 for a formal statement).

- For the class of ES^{2}PR nets deadlock–freeness ensures liveness and can thus be characterized by siphon analysis. In particular, it is well known that for ordinary nets deadlock freeness may sometimes be enforced adding new monitors that control the net siphons to prevent them from becoming empty: see [21] as an example of recent development in this area.

One original feature of our approach, which was firstly presented in [17] and that is also used in this paper, consists in the fact that to compute the liveness enforcing monitors, we use a very efficient linear algebraic technique that does not require the exhaustive enumeration of all siphons, whose number may

be too large even for small nets such as the one we consider. In fact, we are able to compute a liveness enforcing monitor solving an integer programming problem (IPP). Siphon based techniques were also used in [1, 2, 21] and other approaches based on IPP can be found in [8, 35, 42].

- We propose to add monitors to the net following an iterative procedure as the number of trains that are admitted into the network increases. We initially assume that only $k = 2$ trains may enter the net, i.e., the skeleton net contains only two tokens. We determine if from the initial marking there exists a reachable marking such that a siphon is empty: if such is the case, we add a monitor place to prevent it from becoming empty.

If a live net has been obtained for k tokens we consider an initial marking with $k+1$ tokens. We continue until we reach a value $k = K$ with K sufficiently large to cover all cases of practical interest (the number of trains that can be admitted within a train network is upper bounded by the number of available vehicles).

In general the addition of such a monitor may give rise to some problems as discussed in [12, 21].

Problem 1: The closed loop net may not be an ES^{2}PR net and we cannot carry on with our iterative procedure. One of the main contributions of this paper is the derivation of a necessary and sufficient condition to verify if the addition of a monitor to an ES^{2}PR net still produces an ES^{2}PR net. Note that this result is also useful to characterize the class of ES^{2}PR nets.

Problem 2: The monitor may create new siphons that require to be controlled as well, i.e., new deadlocks involving the newly added monitors may occur and the procedure needs to be reapplied. We cannot always ensure that the procedure will eventually converge to a live net[1].

This means that we may be obliged to halt this iterative procedure at a lower level of $k < K$ because either at a given step the addition of a monitor generates a net that is not an ES^{2}PR anymore (Problem 1), or because it does not converge to a live net (Problem 2).

A general solution to Problem 1 was given by Park and Reveliotis. In [35] they defined a class broader than ES^{2}PR and showed that for these nets it is possible to compute the liveness enforcing monitors solving, as we do in this paper, an IPP. The class of nets they consider is closed under the addition of a monitor and thus Problem 1 may never occur.

A general solution to Problem 2 requires different deadlock avoidance policies [13, 36] that are not monitor based.

However, the approach we propose here is practically useful in many cases, because the linear characterization we derive requires the solution of an IPP with a reduced computational complexity (in terms of integer variables) with respect to the more general approaches just mentioned. Thus we suggest that our procedure should be initially used and only if it stops at a level of k that is deemed too small, should these more general procedures be invoked during the successive steps.

[1]There is an intuitive explanation for this. The set of legal markings enforced by a monitor is a convex set. On the contrary, the set of markings of a net that is deadlock-free may not have this special structure.

2 Background

2.1 A short introduction to Petri nets

In this section we recall the formalism used in the paper. For more details on Petri nets we address to [28].

A *Place/Transition net* (P/T net) is a structure $N = (P, T, \boldsymbol{Pre}, \boldsymbol{Post})$, where P is a set of m places; T is a set of n transitions; $\boldsymbol{Pre} : P \times T \rightarrow \mathbb{N}$ and $\boldsymbol{Post} : P \times T \rightarrow \mathbb{N}$ are the *pre–* and *post–* incidence functions that specify the arcs; $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ is the incidence matrix.

A *marking* is a vector $\boldsymbol{m} : P \rightarrow \mathbb{N}$ that assigns to each place of a P/T net a non–negative integer number of tokens, represented by black dots. In the following we denote as m_i the marking of place p_i. A P/T *system* or *net system* $\langle N, \boldsymbol{m}_0 \rangle$ is a net N with an initial marking $\boldsymbol{m}_0$ and its set of reachable markings is denoted $R(N, \boldsymbol{m}_0)$.

We denote $PR(N, \boldsymbol{m}_0)$ the *potentially reachable set*, i.e., the set of all markings $\boldsymbol{m} \in \mathbb{N}^m$ for which there exists a vector $\boldsymbol{\sigma} \in \mathbb{N}^n$ that satisfies the *state equation* $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C\sigma}$, i.e., $PR(N, \boldsymbol{m}_0) = \{\boldsymbol{m} \in \mathbb{N}^m \mid \exists \boldsymbol{\sigma} \in \mathbb{N}^n : \boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C\sigma}\}$.

A non-null vector $\boldsymbol{x} \in \mathbb{N}^m$ such that $\boldsymbol{x}^T \boldsymbol{C} = \boldsymbol{0}$ is called a *P–semiflow* (or *P–invariant*) of the net N. The *support* $\|\boldsymbol{x}\|$ of a P–semiflow is the set of places p_i such that $x_i > 0$. Let $\boldsymbol{X}$ be a matrix where each column is a P–semiflow of N, and denote the set of *invariant markings* $\mathcal{I}_X(N, \boldsymbol{m}_0) = \{\boldsymbol{m} \in \mathbb{N}^m \mid \boldsymbol{X}^T \boldsymbol{m} = \boldsymbol{X}^T \boldsymbol{m}_0\}$.

The following result holds: $R(N, \boldsymbol{m}_0) \subseteq PR(N, \boldsymbol{m}_0) \subseteq \mathcal{I}_X(N, \boldsymbol{m}_0)$, i.e., the potentially reachable set and the invariant set are outer approximations of the reachability set.

A P/T net is called *ordinary* when all of its arc weights are 1's. A *state machine* is an ordinary Petri net such that each transition t has exactly one input place and exactly one output place. A net is *strongly connected* if there exists a directed path from any node in $P \cup T$ to every other node.

A *siphon* of an ordinary net is a set of places $\mathcal{S} \subseteq P$ such that: $\bigcup_{p \in \mathcal{S}} {}^\bullet p \subseteq \bigcup_{p \in \mathcal{S}} p^\bullet$. A siphon is *minimal* if it is not the superset of any other siphon. The number of tokens assigned to the siphon $\mathcal{S}$ by a marking $\boldsymbol{m}$ is $\boldsymbol{m}(\mathcal{S}) = \sum_{p_i \in \mathcal{S}} m_i$. A siphon can also be described by its *characteristic vector* $\boldsymbol{s} \in \{0,1\}^m$ such that $s_i = 1$ if $p_i \in \mathcal{S}$, else $s_i = 0$; thus $\boldsymbol{m}(\mathcal{S}) = \boldsymbol{s}^T \boldsymbol{m}$.

2.2 GMECs, monitors and controllability

The development of this subsection is kept very concise for sake of brevity. Please, refer to [27] for a more complete discussion of this topic.

Assume that we are given a set of legal markings $\mathcal{L} \subseteq \mathbb{N}^m$, and consider the basic control problem of designing a supervisor that restricts the reachability set of the plant in closed loop to $\mathcal{L} \cap R(N, \boldsymbol{m}_0)$. Of particular interest are those PN state–based control problems where the set of legal markings $\mathcal{L}$ is expressed by a set of n_c linear inequality constraints called *Generalized Mutual Exclusion Constraints* (GMECs).

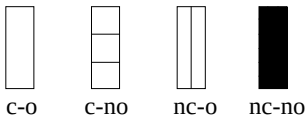

Figure 1: Convention on transitions: *c* (controllable), *o* (observable), *nc* (uncontrollable), *no* (unobservable).

Each GMEC is a couple $(\boldsymbol{w}, k)$ where $\boldsymbol{w} : P \to \mathbb{Z}$ is a $m \times 1$ weight vector and $k \in \mathbb{Z}$. Given the net system $\langle N, \boldsymbol{m}_0 \rangle$, a GMEC defines a set of markings that will be called *legal markings*: $\mathcal{M}(\boldsymbol{w}, k) = \{\boldsymbol{m} \in \mathbb{N}^m \mid \boldsymbol{w}^T \boldsymbol{m} \leq k\}$. The markings that are not legal are called *forbidden markings*. A controlling agent, called supervisor, must ensure that the forbidden markings will be not reached. So the set of legal markings under control is $\mathcal{M}_c(\boldsymbol{w}, k) = \mathcal{M}(\boldsymbol{w}, k) \cap R(N, \boldsymbol{m}_0)$.

In the presence of multiple constraints, all constraints can be grouped and written in matrix form as

$$\boldsymbol{W}^T \boldsymbol{m} \leq \boldsymbol{k} \tag{1}$$

where $\boldsymbol{W} \in \mathbb{Z}^{m \times n_c}$ and $\boldsymbol{k} \in \mathbb{Z}^{n_c}$. The set of legal markings is $\mathcal{M}(\boldsymbol{W}, \boldsymbol{k}) = \{\boldsymbol{m} \in \mathbb{N}^m \mid \boldsymbol{W}^T \boldsymbol{m} \leq \boldsymbol{k}\}$.

A GMEC may be enforced adding to the net a single control structure consisting in a new place, called *monitor place*. In the case of n_c constraints we have n_c monitors and to each of them it corresponds an additional row in the incidence matrix of the closed loop system. In particular, let $\boldsymbol{C}_c$ be the matrix that contains the arcs connecting the monitor places to the transitions of the plant, and $(\boldsymbol{m}_{c0})$ $\boldsymbol{m}_c$ the (initial) marking of the monitors. The incidence matrix $\boldsymbol{C} \in \mathbb{Z}^{(m+n_c) \times n}$ of the closed loop system is

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C}_p \\ \boldsymbol{C}_c \end{bmatrix} \tag{2}$$

and the marking vector $\boldsymbol{m} \in \mathbb{Z}^{m+n_c}$ and initial marking $\boldsymbol{m}_0$ are

$$\boldsymbol{m} = \begin{bmatrix} \boldsymbol{m}_p \\ \boldsymbol{m}_c \end{bmatrix}, \quad \boldsymbol{m}_0 = \begin{bmatrix} \boldsymbol{m}_{p0} \\ \boldsymbol{m}_{c0} \end{bmatrix}, \tag{3}$$

where the subscript p has been used to denote the variables of the plant.

In the case of controllable and observable transitions, Giua *et al.* provided the following theorem.

Theorem 1 ([16]). *If $\boldsymbol{k} - \boldsymbol{W}^T \boldsymbol{m}_0 \geq \boldsymbol{0}$ then a Petri net controller with incidence matrix $\boldsymbol{C}_c = -\boldsymbol{W}^T \boldsymbol{C}_p$ and initial marking $\boldsymbol{m}_{c0} = \boldsymbol{k} - \boldsymbol{W}^T \boldsymbol{m}_{p0}$ enforces constraint (1) when included in the closed loop system (2) with marking (3).*

The controller so constructed is maximally permissive, i.e. it prevents only transitions firings that yield forbidden markings. The controller net has n_c monitor places and no transition is added.

It often occurs that certain transitions can not be disabled by any control action (*uncontrollable transitions*) or their firing can not be directly detected or measured (*unobservable transitions*). We adopt the convention reported in Figure 1 to distinguish among controllable and/or uncontrollable, observable and/or unobservable transitions.

An admissible monitor must satisfy two structural conditions [26, 27] when uncontrollable or unobservable transitions exist. No arcs is allowed from a monitor to an uncontrollable transition t, so that t can never

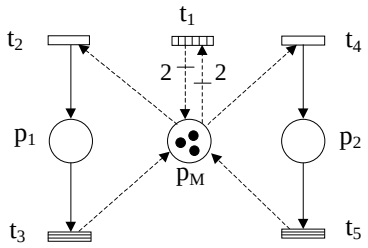

Figure 2: A monitor with self-loop.

be disabled by the controller. An unobservable transition must have the same number of input and output arcs to/from a monitor — i.e. its only admissible connection to a monitor is given by self–loops — so that its firing does not change the state of the controller and thus can never be detected.

If the monitor constructed applying the previous theorem does not satisfy these structural conditions, an appropriate set of transformed constraints (more restrictive than the original ones) needs to be determined so as to construct a Petri net controller. A general technique to do this with a simple procedure that requires little more than the integer triangularization of a suitable matrix was presented in [26, 27]. An example of constraint transformation is given in Section 4.

2.3 Constraints involving the firing vector

Certain control goals may involve the firing vector of a Petri net as well as the tokens content of places [27]. A constraint of this kind takes the form:

$$\boldsymbol{w}^T\boldsymbol{m} + v_j q_j \leq k \tag{4}$$

where $v_j \in \mathbb{N}$, and $q_j \in \{0, 1\}$ is such that $q_j = 1$ if t_j is control enabled, otherwise $q_j = 0$.

Thus, constraint (4) implies that t_j should be control enabled at marking $\boldsymbol{m}$ if and only if the following two conditions are simultaneously verified:

(a) $\boldsymbol{w}^T[\boldsymbol{m} + \boldsymbol{C}(\cdot, t)] \leq k$,

(b) $k - \boldsymbol{w}^T\boldsymbol{m} \geq v_i$.

The corresponding control structure takes the form of a monitor place with a self-loop. As an example, in Figure 2[2] we have shown the monitor with self-loop p_M that enforces the constraint $m_1 + m_2 + 2q_1 \leq 3$.

Note that transition t_1 must be controllable, transitions t_2 and t_4 must be controllable and observable, transitions t_3 and t_5 must be observable.

2.4 The class of ES^{2}PR models and its properties

[2]Using the standard notation of Petri nets an arc of multiplicity grater than one is labeled with its multiplicity. Such is the case for the arcs from p_M to t_1 and viz, each of which has multiplicity 2.

In this subsection we first recall the definition of two important classes of Petri nets, namely the S^2P and ES^2PR nets, firstly introduced by Tricas *et al.* in [12, 40, 41]. These classes of nets have been identified because they frequently appear in the framework of manufacturing systems, and for the ES^2PR class deadlock and liveness problems may be easily characterized by analyzing siphons. In the rest of the paper we shall see that a reduced model of a railway network, that we call "skeleton net", belongs to this class and the liveness problem may be solved using an important property of this model.

A *Simple Sequential Process* (S^2P) is a strongly connected state machine where all circuits contain a common place p_0, denoted as the *idle place*. From a modeling point of view, a S^2P represents the set of different sequences that a unit of the process can follow across the system.

Definition 2 ([40]). *A* Simple Sequential Process, *S^2P, is an ordinary Petri net $N = (P_S \cup \{p_0\}, T, \boldsymbol{Pre}, \boldsymbol{Post})$ where:*

1. *$P_S \neq \emptyset$, $p_0 \notin P_S$.*
2. *N is a strongly connected state machine.*
3. *All the circuits in N contain the place p_0.*

An *Extended Simple Sequential Process with Resources* (ES^2PR) is defined as a S^2P that uses resources in the states of the system that are not the idle one. In this class of nets a process state can need the use of several resources simultaneously.

Structurally, an ES^2PR is obtained from a S^2P net adding a set of places representing available resources, denoted as P_R. These places must satisfy some constraints as stated in the following definition.
Definition 3 ([40]). *An* Extended Simple Sequential Process with Resources, *ES^2PR, is a generalized self–loop free Petri net $N = (P_S \cup \{p_0\} \cup P_R, T, \boldsymbol{Pre}, \boldsymbol{Post})$, such that:*

1. *the subnet generated by the set $X = P_S \cup \{p_0\} \cup T$ is a S^2P,*
2. *$(P_S \cup \{p_0\}) \cap P_R = \emptyset$,*
3. *$\forall\ t \in T$, $\forall\ p \in {}^\bullet t$, $Pre(p,t) = 1$,*
4. *$\forall\ r \in P_R$, $\exists$ a unique minimal P–semiflow $\boldsymbol{x}_r$ such that $\{r\} = \|\boldsymbol{x}_r\| \cap P_R$, $p_0 \notin \|\boldsymbol{x}_r\|$, $P_S \cap \|\boldsymbol{x}_r\| \neq \emptyset$ and $x_r(r) = 1$.*

An example of an ES^2PR net is given in Figure 12, where $P_S = \{p_1, p_4, p_7, p_{10}, p_{14}, p_{17}, p_{20}\} \cup \{p_3, p_6, p_9, p_{13}, p_{16}, p_{19}, p_{22}\}$ and $P_R = \{p_2, p_5, p_8, p_{11}, p_{13}, p_{15}, p_{18}, p_{21}\}$.

The following important result has been proven in [40].
Proposition 4 ([40]). *Let $\langle N, \boldsymbol{m}\rangle$ be a marked ES^2PR net. If a transition $t \in T$ is dead for a reachable marking $\boldsymbol{m}$, then there exists a reachable marking $\boldsymbol{m}'$ and siphon $\mathcal{S} \neq \emptyset$ such that $\boldsymbol{m}'(\mathcal{S}) = 0$, i.e., all places in the siphon $\mathcal{S}$ are empty.*

Note that the result given in Proposition 4 was proven in [40] not only for ES^2PR nets, but for an even larger class of Petri nets.

Now, we present an important result that is useful when studying liveness problems, and in particular when applying an iterative procedure for deadlock–avoidance that will be presented in Section 5. More

precisely, let us consider an ES^{2}PR net N with K resource places. Let $(\boldsymbol{w}, k)$ be a positive and minimal–support GMEC[3] and let r_{K+1} be the corresponding monitor place. We prove that the addition of r_{K+1} to N produces a closed–loop net N' that is still an ES^{2}PR net, if and only if two conditions are verified, namely, the GMEC should only involve places in P_S and the corresponding monitor place should only have ordinary output arcs.

To do this we first recall two lemma whose proof can be found in [17]. Note that in the following we denote by $I_{min}(N)$ the set of minimal P–semiflows of N.

Lemma 5 ([17]). *Let $N = (P, T, \boldsymbol{Pre}, \boldsymbol{Post})$ be a Petri net. Let $(\boldsymbol{w}, k)$ be a positive and minimal–support GMEC and r be the corresponding monitor place. It holds that*

$$\left\{ \boldsymbol{y} = \begin{bmatrix} \boldsymbol{x} \\ 0 \end{bmatrix} \;\middle|\; \boldsymbol{x} \in I_{min}(N) \right\} \cup \left\{ \begin{bmatrix} \boldsymbol{w} \\ 1 \end{bmatrix} \right\} \subseteq I_{min}(N') \tag{5}$$

where $N' = (P \cup \{r\}, T, \boldsymbol{Pre}', \boldsymbol{Post}')$ is the closed–loop net.

The above lemma implies that after the addition of a monitor place to a generic Petri net N, the set of minimal P-semiflows of the resulting net includes all the minimal P-semiflows of the net before the addition of the monitor, plus the minimal P-semiflow induced by the monitor. Moreover, the addition of the monitor place may also originate other P-semiflows.

Lemma 6 ([17]). *Let $N = (P_S \cup \{p_0\} \cup P_R, T, \boldsymbol{Pre}, \boldsymbol{Post})$ be an ES^{2}PR net, where $P_S = \{p_1, \cdots, p_m\}$ and $P_R = \{r_1, \cdots, r_K\}$. Let $(\boldsymbol{w}, k)$ be a positive and minimal–support GMEC only involving places in P_S and r_{K+1} be the corresponding monitor place.*

Let $N' = (P_S \cup \{p_0\} \cup P'_R, T, \boldsymbol{Pre}', \boldsymbol{Post}')$, where $P'_R = P_R \cup \{r_{K+1}\}$, be the closed loop net.

It holds that

$$\left\{ \boldsymbol{y} = \begin{bmatrix} \boldsymbol{x} \\ 0 \end{bmatrix} \;\middle|\; \boldsymbol{x} \in I_{min}(N) \right\} \cup \left\{ \begin{bmatrix} \boldsymbol{w} \\ 1 \end{bmatrix} \right\} \supseteq I_{min}(N'). \tag{6}$$

The above lemma implies that if a monitor is added to an ES^{2}PR net, and the corresponding GMEC only involves places in P_S, then the set of minimal P-semiflows of the resulting net is a subset of the set of P-semiflows of the original net plus the P-semiflow corresponding to the monitor place r_{K+1}.

Theorem 7. *Let $N = (P_S \cup \{p_0\} \cup P_R, T, \boldsymbol{Pre}, \boldsymbol{Post})$ be an ES^{2}PR net, where $P_R = \{r_1, \cdots, r_K\}$.*

Let r_{K+1} be the monitor place corresponding to the minimal–support and positive GMEC $(\boldsymbol{w}, k)$. The closed loop net $N' = (P_S \cup \{p_0\} \cup P'_R, T, \boldsymbol{Pre}', \boldsymbol{Post}')$, where $P'_R = P_R \cup \{r_{K+1}\}$, is an ES^{2}PR net if and only if it holds that:

(a) $\forall\, t \in T,\ Pre'(r_{K+1}, t) = 1$,

(b) $(\{p_0\} \cup P_R) \cap \|\boldsymbol{w}\| = \emptyset$, *i.e.,* $\|\boldsymbol{w}\| \subseteq P_S$.

Proof. (if) We need to prove that the closed-loop net N' satisfies the four conditions of Definition 3. The first two conditions are trivially verified. The third is verified by assumption (a). Finally, assumption

[3] A GMEC $(\boldsymbol{w}, k)$ is called *positive* if $\boldsymbol{w} \geq \vec{0}_m$, $k > 0$, and is *minimal–support* if there exists no P-semiflow $\boldsymbol{x}$ such that $\|\boldsymbol{x}\| \subseteq \|\boldsymbol{w}\|$, i.e., $\|\boldsymbol{w}\|$ does not contain the support of any P–semiflow.

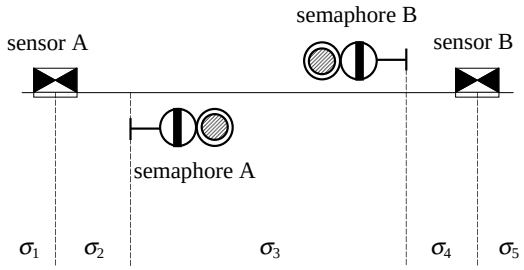

Figure 3: Scheme of a railway track.

(b), together with the Lemma 5 and Lemma 6 implies that

$$I_{min}(N') = \left\{ \begin{bmatrix} \boldsymbol{x} \\ 0 \end{bmatrix} \mid \boldsymbol{x} \in I_{min}(N) \right\} \cup \left\{ \begin{bmatrix} \boldsymbol{w} \\ 1 \end{bmatrix} \right\},$$

hence the fourth condition is verified.

(only if) If assumption (a) is violated then the third condition in Definition 3 is not satisfied, while if assumption (b) is violated then the fourth condition is not satisfied. □

3 Modeling railway networks with Petri nets

In this section we show how Petri nets can be efficiently used as a modeling tool for railway networks. In particular, we show that the whole network can be seen as the composition of a certain number of elementary modules, namely tracks, points, and stations.

3.1 The track model

Consider the railway track scheme shown in Figure 3. This track can be divided into the five segments σ_i's ($i = 1, \ldots, 5$) shown in the figure. Sensors A and B [4] can detect the passage of train regardless of its direction (leftward or rightward). Semaphore A can stop a train directed leftward (from segment σ_3 to segment σ_2) and it is also able to detect the passage of a train directed leftward. Semaphore B can stop a train directed rightward (from segment σ_3 to segment σ_4) and it is also able to detect the passage of a train directed rightward.

A Petri net model for such a track is shown in Figure 4. Each couple of places p_i, p'_i represents segment σ_i: the marking of p_i (resp., p'_i) denotes the presence of a train directed rightward (resp., leftward) in segment σ_i. Transitions $t_1, \cdots, t_4$ (resp., $t'_1, \cdots, t'_4$) represent the passage of a train directed rightward (resp., leftward) from one segment to another.

Transitions may be (un)controllable and/or (un)observable. In this setting, a transition that is both controllable and observable corresponds to a semaphore. As an example transition t'_2 (resp., t_3) in

[4] This kind of sensors only detect the passage of a train but they also count the number of axles of the train, i.e., the number of cars passing through that point. They are used to make sure that no car has been left within a section. In our exposition we are slightly simplifying the problem.

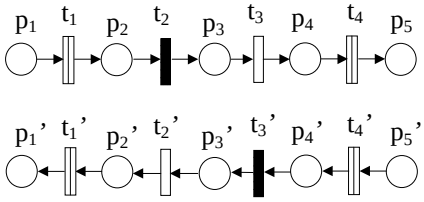

Figure 4: The Petri net model of a track.

Figure 4 corresponds to semaphore A (resp., semaphore B): it is controllable and observable to denote that presence of a train directed leftward (resp. rightward) can be detected and its transit can be forbidden. In all real situations a semaphore is placed at the exit of a track, or equivalently at the entrance of a station. A transition that is observable but not controllable (see transitions t_1, t_4, t'_1 and t'_4), represents a sensor.

Note that the same model can represent two different cases.

- A single track , i.e., a track that can be crossed in two directions (leftward or rightward). Thus, places p_i and p'_i, are used to represent the same segment of the single track. Obviously, the two places must contain at most one token and cannot be marked at the same time: this will be enforced by a suitable control logic.
- A double track, i.e., two tracks one of which is always crossed rightward while the other one is always crossed leftward. In the case of a double track, the two lines are independent and places p_i and p'_i correspond to parallel segments. The two places must contain at most one token each but both can be marked at the same time.

The number of places used to represent the track depends on the required precision. On one hand, we assume that the Petri net is safe (such a condition will be imposed by the addition of appropriate monitor places), thus the number of places is mainly limited by the required safeness distance, i.e., we assume that the length of each segment is such that no more than one train can be contained within it at any given time instant. On the other hand, we must take into account the presence of sensors and semaphores (that are modeled by appropriate transitions as discussed above).

3.2 The points model

Consider the points (switch) sketched in Figure 5.a. Depending on the points position trains may follow a different path.

The Petri net model of a points with n possible paths is reported in Figure 5.b: it contains $n+1$ places (namely, $p_1, \cdots, p_i, \cdots, p_n, p_f$) and 2 n transitions (namely, $t_{o,1}, \cdots, t_{o,i}, \cdots, t_{o,n}, t_{f,1}, \cdots, t_{f,i}, \cdots, t_{f,n}$). When place p_i ($i = 1, \cdots, n$) is marked, trains may be routed to track i. On the contrary, if place p_f is marked, then no train may cross the points. This is the case when either the enabled path is being changed or the points is under maintenance.

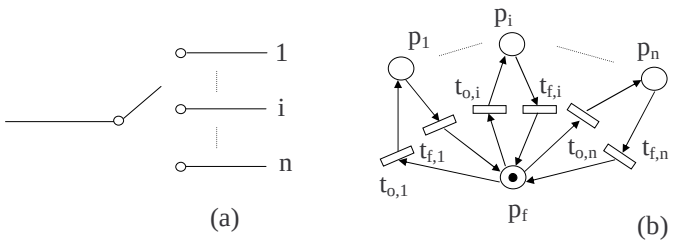

Figure 5: The scheme of a points (a) and its Petri net model (b).

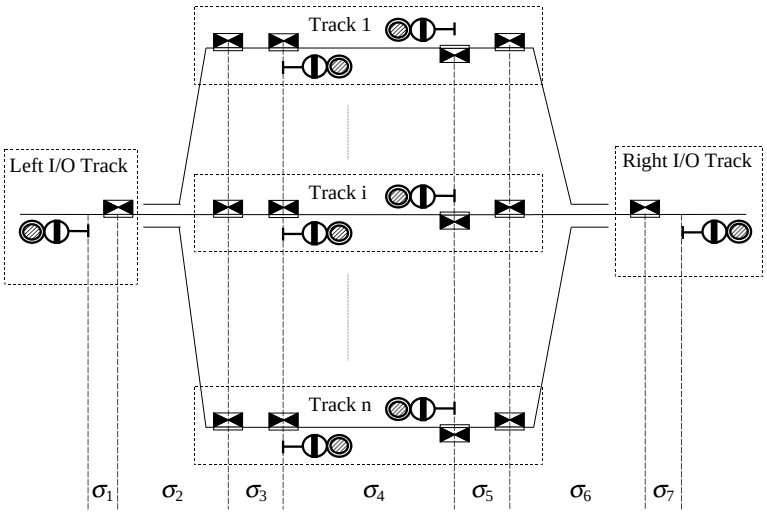

Figure 6: Scheme of a n-tracks railway station.

3.3 The railway station model

Consider the n-tracks railway station whose scheme is shown in Figure 6. The station is composed of $n+2$ different stretches: the n inner tracks within the station (tracks $1, \cdots, i, \cdots, n$) and the input/output (I/O) tracks on the left and right side.

We can identify the seven different segments shown in the figure. Segments σ_1 and σ_7 represent the actual I/O tracks while segments σ_2 and σ_6 represent the tracks controlled by the points. Segments σ_3, σ_4 and σ_5 are the leftmost, central and rightmost segments of the three inner tracks. The sensors and semaphores shown in the figure have already been described in the previous subsection.

The Petri net model of this station is sketched in Figure 7 where double arrows have been used to denote self–loops.

The firing of controllable and observable transitions t^1_{ing} and t^2_{ing} represent the entrance of a train into the station, while the firing of uncontrollable and unobservable transitions t^1_{out} and t^2_{out} represent the exit of a train from the station. Note that, as in the case of the track model, a controllable and observable transition is used to model a semaphore, while an observable but uncontrollable transition is used to model a sensor.

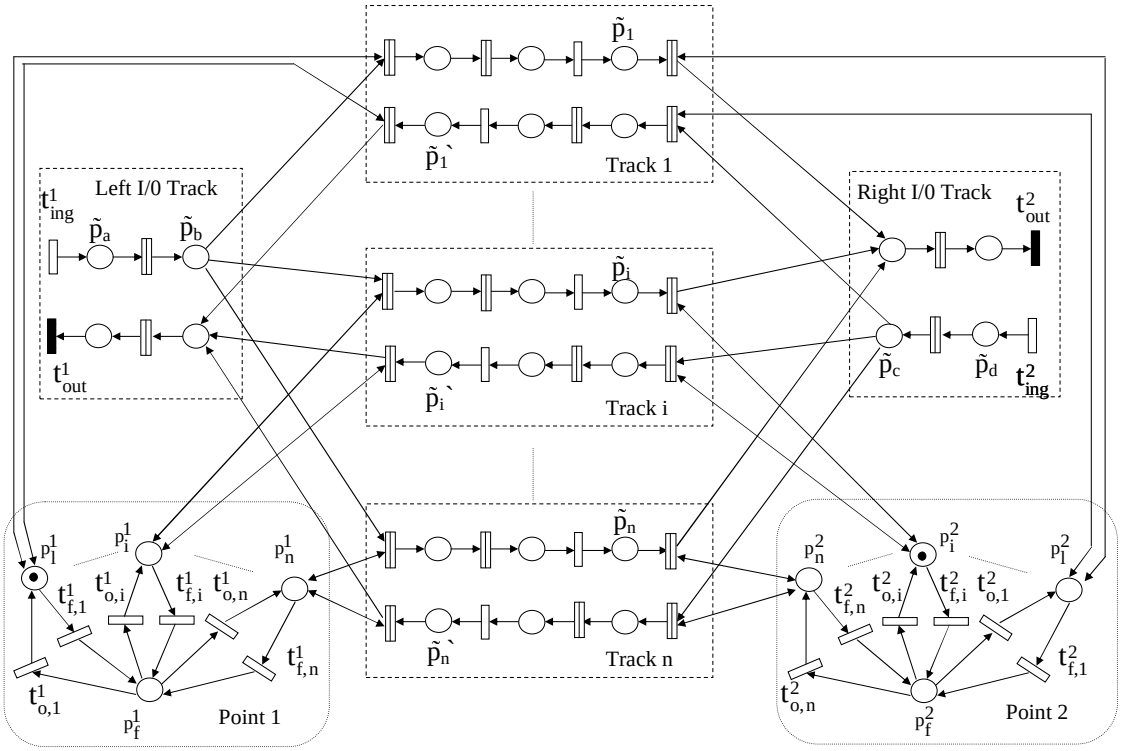

Figure 7: *The Petri net model of a n-tracks railway station.*

The two subnets on the bottom right and on the bottom left of Figure 7 model the points. The superscripts 1 and 2 have been used to denote places and transitions relative to the left and the right points, respectively.

Remark 8. Let us consider in Figure 7 the self-loops between the points places p_i^1, p_i^2, for $i = 1, \ldots, n$, and the transitions limiting the intermediate tracks. These transitions have been defined as observable but uncontrollable because they model the presence of sensors. Therefore, it may appear a contradiction that we have arcs going from the above mentioned places and such transitions. In fact, this implies that the points control uncontrollable transitions.

In effect, what is controllable is the position of the points, that establishes which is the route that is set up. Depending on the position of the movable parts in the points a certain route is enabled (or no route is enabled) and the flow of the train occurs accordingly.

A situation in which, say place $\tilde{p}_i$ is marked (i.e., a train leaving track i is going right) while place p_i^2 is not marked, is an anomalous state in which the train goes off the lines. To make sure that no situation of this type is reached the safe operation of the points must be ensured by the controller. This will be done in Subsection 4.2. ■

4 The controller design for tracks and stations

In this section we deal with the problem of designing a Petri net supervisor for a railway network so as to ensure safeness and local liveness. In other words, the goal of the supervisory controller is that of

guaranteeing that two trains may flow through the net in the same direction or in opposite directions without colliding, while prohibiting that blocking conditions may occur.

To do this we first consider single modules and derive a controller for tracks and stations separately. In particular, we observe that GMECs may be satisfactorily applied when controlling tracks. On the contrary, we will see that this kind of constraints is too restrictive when controlling the admission to stations and may lead to a deadlock. We show in detail that in this latter case, safeness may be ensured by imposing appropriate logical constraints that also ensure local liveness.

We will also discuss the constraints that should be kept into account to regulate the orderly movements of the points.

4.1 Safeness GMECs on tracks

GMECs have been firstly imposed so as to ensure safeness, i.e., to ensure that each couple of places corresponding to the same segment of a single–track (that may also belong to a station) are not marked simultaneously, and each place never contains more than one token at a time.

In accordance to the supervisory control theory briefly summarized in Subsection 2.2 each constraint requires the introduction of a monitor place. Moreover, in the case of uncontrollable and/or unobservable transitions, constraints need to be appropriately transformed.
Example 9. Let us consider a two-tracks station whose Petri net model is reported in Figure 8.a, apart from place $p_{M,i}$ and all connected arcs. Let us consider places p_i and p'_i relative to a given segment of Track 1 within the station. The GMEC that ensures that places p_i and p'_i are not marked simultaneously and each place never contains more than one token, takes the form $m_i + m'_i \leq 1$. If all transitions were controllable and observable, the monitor place ensuring the satisfaction of such a constraint would have been that in Figure 8.a.

Nevertheless, this monitor is not admissible because it has two input arcs going to uncontrollable transitions. Applying the procedure proposed by Moody in [27], that is discussed in Subsection 2.2, the more restrictive but admissible monitor, denoted $p'_{M,i}$ in Figure 8.b, is obtained; this monitor imposes the more restrictive constraint $m_{i-3} + m_{i-2} + m_{i-1} + m_i + m'_i + m'_{i+1} + m'_{i+2} + m'_{i+3} \leq 1$. ■

4.2 Safe operation of the points

In this subsection we provide a set of constraints, that we call *safe operation constraints on points*, that are necessary to guarantee that a train never goes off the lines. In practice such constraints define the admissible positions of the movable parts of the points and can be easily expressed in terms of GMECs.

In particular, for each intermediate track in the station we need two constraints regulating the input of trains in the track from the left and the right I/O tracks, and two constraints regulating the output of trains from the track, after a semaphore has been passed.

Now, if we consider the Petri net model of the n-tracks railway station in Fig. 7, the safe operation constraints on points can be written as follows:

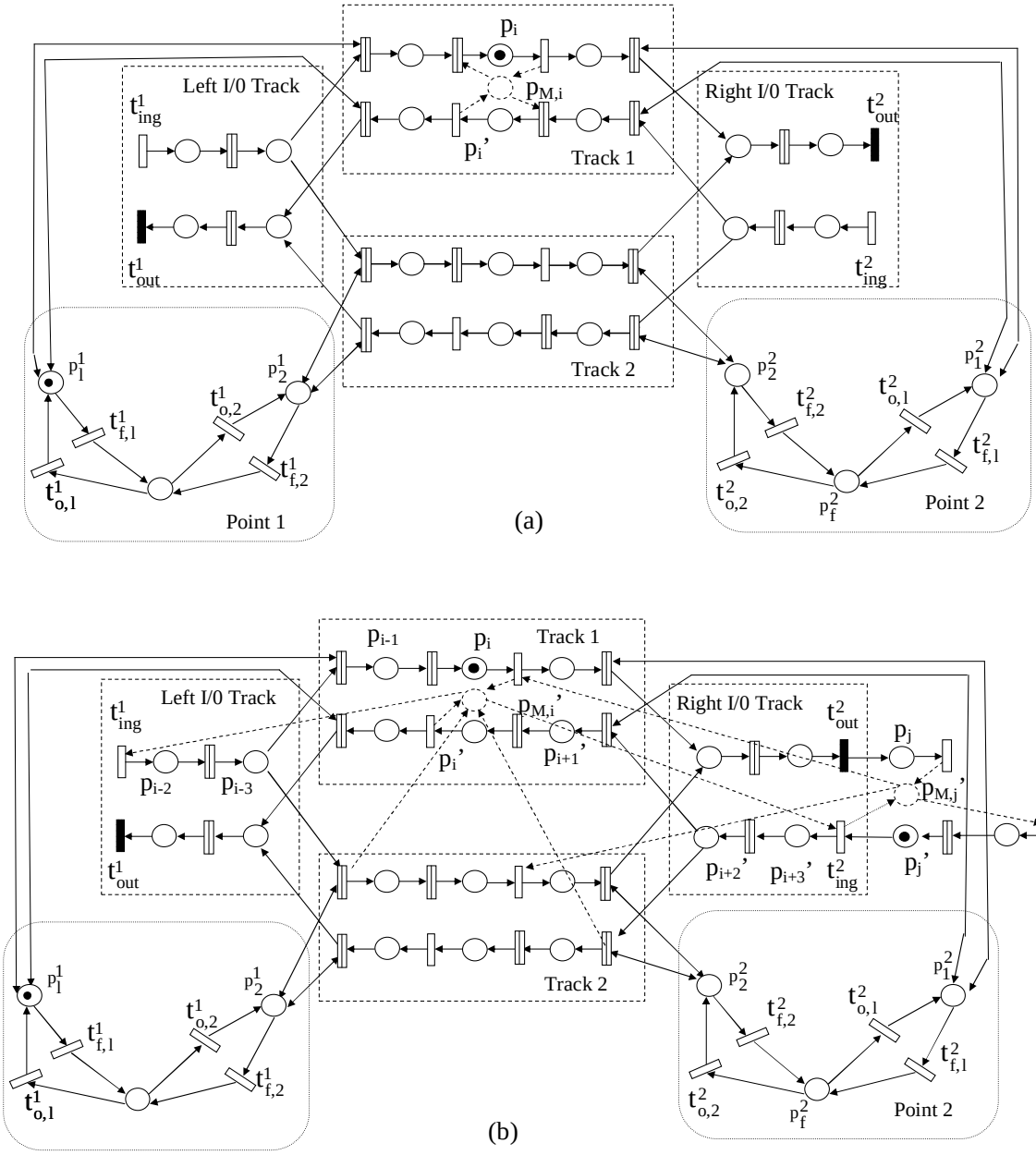

Figure 8: (a) The Petri net model of a two–tracks railway station. The monitor place $p_{M,i}$ is relative to the constraint $m_i + m'_i \leq 1$ under the assumption that all transitions are controllable and observable.

(b) The Petri net model of a two–tracks railway station and of a segment of an adjacent track on the right. The monitor places $p'_{M,i}$ and $p'_{M,j}$ enforce the GMEC $m_i + m'_i \leq 1$ and $m_j + m'_j \leq 1$, respectively. These monitors have been designed taking into account the uncontrollability and unobservability of transitions, i.e., using Moody's transformation [27].

$$\begin{cases} \tilde{m}_a + \tilde{m}_b + m_f^1 \leq 1 & (a) \\ \tilde{m}_c + \tilde{m}_d + m_f^2 \leq 1 & (b) \\ \tilde{m}_i + \sum_{p \in \mathcal{P}_2 \setminus \{p_i^2\}} m(p) \leq 1 \quad i = 1, \ldots, n & (c) \\ \tilde{m}'_i + \sum_{p \in \mathcal{P}_1 \setminus \{p_i^1\}} m(p) \leq 1 \quad i = 1, \ldots, n & (d) \end{cases} \tag{7}$$

where $\mathcal{P}_1$ and $\mathcal{P}_2$ denote the set of places in Point 1 and 2, respectively.

Constraints (a) and (b) refer to the entrance of the train in the station, while constraints (c) and (d) refer to the exit of trains from the station.

In particular, constraint (a), resp. (b), ensures that, once a train has entered the left, resp. right, I/O track, the entrance to one of the intermediate track is enabled, namely if either $\tilde{p}_a$ or $\tilde{p}_b$ is marked, then p_f^1, resp. p_f^2, is not marked.

Finally, constraint (c), resp. (d), ensures that, once a train in the i-th intermediate track, has passed a semaphore the position of the point enables its exit. Thus, if m_i, resp. m'_i, is marked, then p_i^2, resp. p_i^1, is marked.

It is easy to verify that the monitors forcing constraints (7) are admissible, thus maximal permissiveness is guaranteed.

4.3 Local deadlock

It can be observed that the addition of monitors enforcing the safe GMECs on tracks may induce local deadlocks.

Example 10. Let us consider the monitor places $p'_{M,i}$ and $p'_{M,j}$ in Figure 8.b, that have been introduced to enforce the GMECs $m_i + m'_i \leq 1$ and $m_j + m'_j \leq 1$, respectively, and applying Moody's transformation to make them controllable and observable. By looking at Figure 8.b we can immediately observe that, whenever place p_i is marked, no train can enter the station since transition t_{ing}^2 is not enabled. And this occurs regardless of the marking of places in the other stretch within the station. Thus consider the case in Figure 8.b, where a train going rightward is within the station (place p_i marked) while a train going leftward is approaching the station (place p'_j marked). The train in the station cannot leave and the train approaching cannot enter, thus a deadlock occurs. Note however, that it is perfectly safe to let the approaching train enter, provided it is directed to the empty track. ■

This example shows that GMECs do not guarantee a satisfactory policy for the admission of trains into a station. We provide a better solution, by formalizing a set of station admission rules and converting them into a new set of constraints, some of whom also involve the firing vector of the transitions that regulate the input of trains in the stations, and the points within them. The corresponding Petri net control structure takes the form of monitors with self-loops.

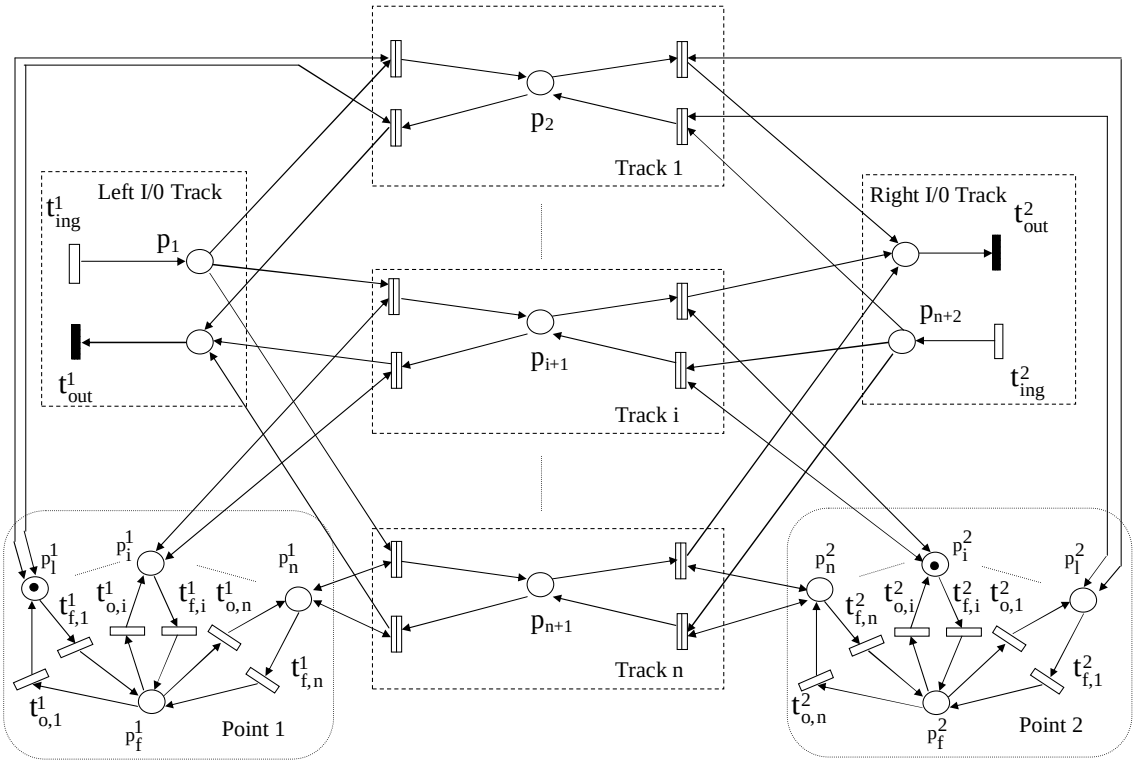

Figure 9: The reduced Petri net model of an n–tracks railway station.

4.4 Station admission rules

To make our discussion easier, we use a slightly simplified model of the railway station reported in Figure 9, that is obtained from the detailed model by simply grouping together some places.

The *station admission rules* preventing the net from deadlock may be briefly summarized as follows.

(1) No more than n trains should be simultaneously present in the station including the I/O tracks. Assume in fact that n trains are in the inner tracks (places p_2, $\cdots$ p_{i+1}, $\cdots$, p_{n+1} are marked): if a further train arrives from outside entering an I/O track (e.g., t^2_{ing} fires marking place p_{n+2}) the only way we have to forbid its entrance into an inner track, thus avoiding a collision, is that of switching off the right points. In fact the n transitions outputting place p_{n+2} are not controllable and we can disable them only acting on the right points. However, in such a way no train may cross over the right points and we reach a deadlock.

Analogously, assume that only $n-1$ inner tracks contain a train but that the left (resp., right) I/O track contains a train directed rightward (resp., leftward): if a further train arrives from outside entering the right (resp., left) I/O track, a collision will eventually occur.

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

These informal rules can be converted into a set of formal constraints:

$$\left\{\begin{array}{ll} \left.\sum_{i=1}^{n+2} m_i \leq n\right\} & \text{rule 1} \\ \left.\begin{array}{ll} q^1_{ing} + m_{i+1} + m^1_i \leq 2, & i = 1, \cdots, n \\ q^2_{ing} + m_{i+1} + m^2_i \leq 2, & i = 1, \cdots, n \end{array}\right\} & \text{rule 2} \end{array}\right. \tag{8}$$

where, apart from the constraint due to the first rule that is a GMEC, all the other constraints also involve the firing vector and the corresponding monitors can be computed using the theory reported in Subsection 2.3.

The control places corresponding to constraints (8) should be added to the net in place of the capacity monitors relative to all tracks within the station (e.g., those corresponding to places p_i, p'_i, p_{i-1}, p'_{i-1}, etc. in Figure 8.b).

5 Liveness constraints

In this section we focus our attention on the problem of *global deadlock* avoidance. In fact, it is easy to verify that although the control logic designed in the previous section avoids local deadlocks, when too many trains are admitted within a network several blocking conditions may occur.

5.1 Skeleton net

For this type of analysis, as we have already mentioned in the Introduction, it is not convenient to use the detailed model of the open loop plant (the railway systems) with the additional control structure (monitors and monitors with self-loop) designed in the previous section.

In particular, we abstract the previous model representing each station and each track by means of two places, whose token contents represent the number of trains directed leftward and rightward. The net structure consisting of the control places designed in the previous section is also abstracted as monitors that limit the capacity of the places representing tracks and stations.

The detailed models of a single track, of a double track, of a station of capacity n, and of a deviation module in the skeleton net are reported in Figure 10. Here the monitors (dotted places) limit the number of trains within tracks and stations according to each track or station capacity. Note that in the deviation module, representing a branch of the net, we assume for simplicity that each track is a single track, and only two choices are possible. The generalization of such a module is trivial and is not discussed here for sake of brevity.

We also observe that at this level of abstraction all transitions can be considered as controllable and observable.

The skeleton net associated to a given railway system is obtained as the composition of the above elementary modules plus an addition place p_0 that limits the total number of trains in the station. Note that since the entrance to each module is controllable and observable, we can assume that all transitions in the skeleton net are controllable and observable. The following example clarifies this.

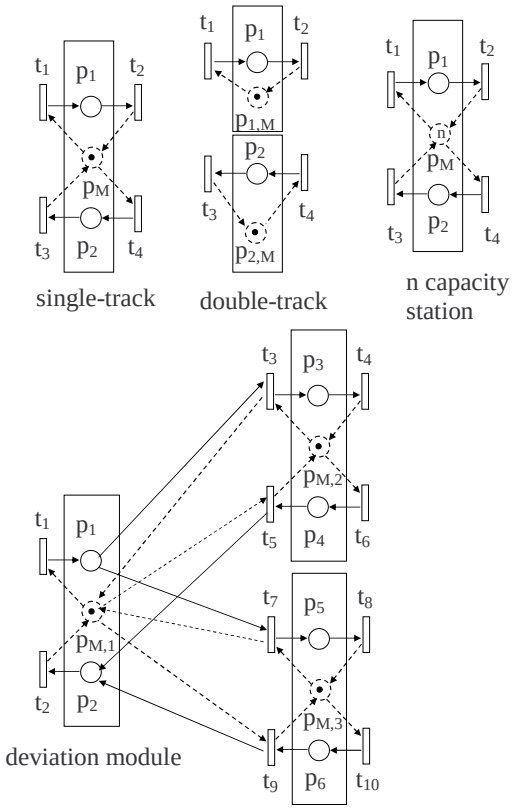

Figure 10: The elementary modules of the skeleton net.

Example 11. Consider, the railway system sketched in Figure 11 [10, 17], that represents a short segment between the stations of Chilivani and Olbia, in Sardinia, Italy. It consists of four stations, where the first one is a three–tracks station while the others are two–tracks stations. All intermediate tracks are single tracks, apart from the second one where two trains may travel in opposite directions simultaneously.

The skeleton Petri net model of the network is shown in Figure 12, where the monitor place p_0 contains the maximum number k of trains that may be allowed into the network.

It is easy to verify using this skeleton model that when different modules are put together several blocking conditions may occur. Consider the case in which $k = 3$ and two trains are in the station β directed towards station α (place p_9 contains two tokens) and one train has already left station α and is moving towards station β (place p_4 contains one token). When such a marking is reached places p_5 and p_8 are empty and the net reaches a partial deadlock. Note that the set of places $\{p_5, p_6, p_7, p_8\}$ is an empty deadlock. ■

We now prove the following result.

Theorem 12. The skeleton net associated to a given railway network is an ES^{2}PR net.

Proof. Firstly we observe that the Petri net obtained from the skeleton net removing all monitor places is a S^{2}P net. In fact, it is a strongly connected state machine with set of places P_S, and where all circuits contain the idle place p_0.

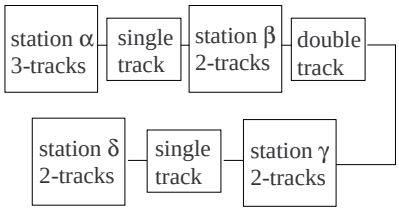

Figure 11: Scheme of the railway network.

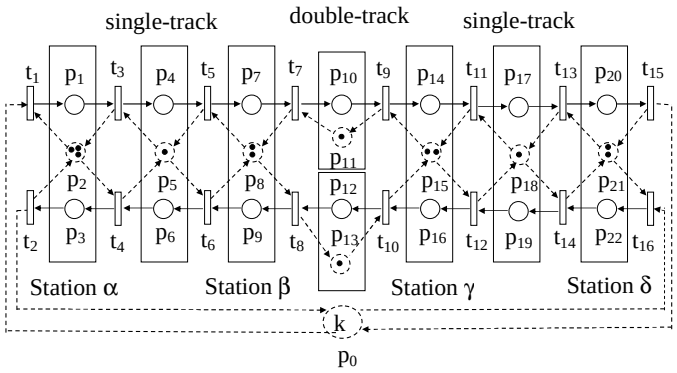

Figure 12: The skeleton Petri net model of the railway network in Figure 11.

Secondly, in the skeleton net all monitors are relative to minimal–support positive GMECs only involving places representing a track or a station (places in P_S) and have only ordinary output arcs. □

Example 13. Let us consider again the skeleton net in Figure 12. The net obtained removing from the skeleton net all monitor places is an ordinary and strictly connected state machine with two circuits — both containing p_0 — and $P_S = \{p_1, p_4, p_7, p_{10}, p_{14}, p_{17}, p_{20}\} \cup \{p_3, p_6, p_9, p_{13}, p_{16}, p_{19}, p_{22}\}$. Moreover, in the case of the skeleton net reported in Figure 12 we have 8 resource places, i.e., $P_R = \{p_2, p_5, p_8, p_{11}, p_{13}, p_{15}, p_{18}, p_{21}\}$ corresponding to the minimal–support positive GMECs: $m_1 + m_3 \leq 3$, $m_4 + m_6 \leq 1$, $m_7 + m_9 \leq 8$, $m_{12} \leq 1$, $m_{20} + m_{22} \leq 2$, $m_{14} + m_{16} \leq 2$, $m_{17} + m_{+19} \leq 1 m_{10} \leq 1$. ■

5.2 Liveness–enforcing monitor computation

In this subsection we present a technique, based on the analysis of the skeleton net, to determine a maximally permissive liveness enforcing control policy.

To ensure liveness of the model, we refer to Proposition 4 [40], stating that, given a marked ES^{2}PR net $\langle N, \boldsymbol{m} \rangle$, if a transition $t \in T$ is dead for a reachable marking $\boldsymbol{m}$, then there exists a reachable marking $\boldsymbol{m}' \in R(N, \boldsymbol{m})$ and a siphon $\mathcal{S}$ whose places are empty at $\boldsymbol{m}'$.

We then adopt a standard technique [40] to enforce liveness: we determine if there are siphons in the net that can become empty and if so add a monitor to control them and prevent this. In general cases there are two problems with this technique: first of all the new monitors may create new siphons that may need to be controlled as well; secondly, the addition of new monitors may lead to a net that is not an ES^{2}PR net any more. However, Theorem 7 provides an efficient and immediate test to verify when

the iterative procedure may be efficiently continued. In fact, we only need to verify that each additional constraint only involves places in P_S and each additional monitor has only ordinary output arcs.

5.2.1 A linear algebraic characterization of empty siphons

Now, before going on detail of the iterative procedure used to compute the liveness-enforcing monitors, we first present a linear algebraic technique to compute empty siphons. The proposed procedure is based on integer programming and does not require the exhaustive enumeration of all siphons, whose number is too large even for small nets. Although solving a linear integer optimization problem is still an NP complete problem (as is siphon enumeration) we observed that in practice the integer programming approach is much more efficient. This technique is inspired by other linear algebraic approaches appeared in the literature, in particular by the results of [8].

First of all we observe that in the case of an ES^{2}PR net, such as the skeleton net we are examining, there are $|P_R| + 1$ P–semiflows corresponding to the monitor places $P_R \cup \{p_0\}$. Thus the reachable set of the net can be approximated as

$$R(N, \boldsymbol{m}_0) \subseteq \mathcal{I}_X(N, \boldsymbol{m}_0) = \{\boldsymbol{m} \in \mathbb{N}^m \mid \boldsymbol{X}^T \boldsymbol{m} = \boldsymbol{k}\}$$

where each column of the $|P| \times (|P_R|+1)$ matrix $\boldsymbol{X}$ contains a P-semiflow and $\boldsymbol{k} = \boldsymbol{X}^T \boldsymbol{m}_0$ is a $(|P_R|+1) \times 1$ vector whose components represent the token content of each semiflow.

In the case of the net in Figure 12, there are 9 P–semiflows corresponding to the monitor places $\{p_0\} \cup P_R$ shown as dashed circles. The places in the support of each semiflow are shown within a rectangle, except for the semiflow corresponding to p_0 whose support contains all places in the net.

Although we cannot formally prove that $R(N, \boldsymbol{m}_0) = \mathcal{I}_X(N, \boldsymbol{m}_0)$, if we can enforce that no deadlock marking $\boldsymbol{m} \in \mathcal{I}_X(N, \boldsymbol{m}_0)$ is reachable, then no reachable marking may be a deadlock. Thus in the following we use the previous equation as a larger approximation of the marking reachability. Note that it may also be possible to give a better approximation of the reachability set using the potentially reachable set as mentioned in Subsection 2.1: this however requires introducing an additional integer unknown vector (the firing count vector).

To determine if there are siphons that need to be controlled, one may use the following non-linear integer program:

$$\begin{cases} \min & \boldsymbol{s}^T \boldsymbol{m} \\ \text{subject to} & \operatorname{sgn}(\boldsymbol{Pre}^T \boldsymbol{s}) \geq \operatorname{sgn}(\boldsymbol{Post}^T \boldsymbol{s}) \\ & \boldsymbol{X}^T \boldsymbol{m} = \boldsymbol{k} \\ & \mathbf{1}^T \boldsymbol{s} \geq 1 \end{cases} \tag{9}$$

where $\boldsymbol{s} \in \{0,1\}^m$ and $\boldsymbol{m} \in \mathbb{N}^m$ are the unknowns, and $\operatorname{sgn}(\boldsymbol{x})$ is a vector whose i-th component is 1 (resp., 0, -1) if the i-th component of $\boldsymbol{x}$ is positive (resp., null, negative). The equation $\operatorname{sgn}(\boldsymbol{Pre}^T \boldsymbol{s}) \geq \operatorname{sgn}(\boldsymbol{Post}^T \boldsymbol{s})$ ensures that $\boldsymbol{s}$ is the characteristic vector of a siphon $\mathcal{S}$, the second equation ensures that $\boldsymbol{m}$ belongs to the set $\mathcal{I}_X(N, \boldsymbol{m}_0)$, and the equation $\mathbf{1}^T \boldsymbol{s} \geq 1$ ensures that $\mathcal{S}$ is not the empty set. Thus a solution $(\boldsymbol{m}, \boldsymbol{s})$ of this program with optimal value $\boldsymbol{s}^T \boldsymbol{m} = 0$ corresponds to a reachable marking $\boldsymbol{m}$ such that the siphon $\mathcal{S}$ with characteristic vector $\boldsymbol{s}$ is empty.

The non-linearity of the previous program is an undesirable feature, that makes solving it a hard task.

We convert it to an equivalent (linear) integer program

$$\begin{cases} \min & \mathbf{1}^T \boldsymbol{s} \\ \text{subject to} & K_1 \boldsymbol{Pre}^T \boldsymbol{s} \geq \boldsymbol{Post}^T \boldsymbol{s} \\ & \boldsymbol{X}^T \boldsymbol{m} = \boldsymbol{k} \\ & K_2 \boldsymbol{s} + \boldsymbol{m} \leq K_2 \mathbf{1} \\ & \mathbf{1}^T \boldsymbol{s} \geq 1 \end{cases} \tag{10}$$

where $K_1 = \max\{\mathbf{1}^T \boldsymbol{Post}(\cdot, t) \mid t \in T\}$ and $K_2 = \max\{m(p) \mid p \in P, \boldsymbol{m} \in R(N, \boldsymbol{m}_0)\}$ (for the net in Figure 12 $K_1 = 2$ and $K_2 = B$). We claim (a formal proof can be found in [3]) that the program (9) has an optimal solution $(\boldsymbol{m}, \boldsymbol{s})$ such that $\boldsymbol{s}^T \boldsymbol{m} = 0$ if and only if the program (10) has an admissible solution. In fact, the first constraint in (10) is perfectly equivalent to the first constraint in (9), while the new constraint in (10) (the third one) ensures that for all $p_i \in P$, $K_2 s_i + m_i \leq K_2$ holds, i.e., either $s_i = 1$ and $m_i = 0$ or (*exclusive or*) $s_i = 0$ and $m_i > 0$. The objective function chosen for the program (10) ensures that only minimal siphons are computed.

5.2.2 Monitors computation

Now, let us present in detail the iterative procedure that enables us to compute the liveness-enforcing monitors.

Algorithm 14 (Liveness-enforcing monitors computation)**.**

1. Let $m_0(p_0) = 2$.

2. Let $\boldsymbol{X}$ be the matrix whose columns are the P–semiflows

of the ES^{2}PR net

$$N = (P_S \cup \{p_0\} \cup P_R, T, \boldsymbol{Pre}, \boldsymbol{Post})$$

corresponding to the monitor places $\{p_0\} \cup P_R$.

3. Let $\boldsymbol{k} = \boldsymbol{X}^T \boldsymbol{m}_0$.

4. Solve an IPP of the form (10).

5. If (10) has no admissible solution **then**,

begin

let $m_0(p_0) = m_0(p_0) + 1$,

let $\boldsymbol{k} = \boldsymbol{X}^T \boldsymbol{m}_0$,

goto 4

end

else begin

let $\mathcal{S}$ be the solution of (10);

let $(\overline{\boldsymbol{w}}, \overline{\boldsymbol{k}})$ be the GMEC corresponding to the

constraint

$$\boldsymbol{m}(\mathcal{S}) = \sum_{p_i \in \mathcal{S}} m_i \geq 1 \tag{11}$$

or equivalently,

$$-\boldsymbol{m}(\mathcal{S}) = -\sum_{p_i \in \mathcal{S}} m_i \leq -1; \tag{12}$$

(this GMEC prevents $\mathcal{S}$ from becoming empty).

for each place $p \in \{p_0\} \cup P_R$ such that $p \in \mathcal{S}$,

replace its marking with the marking

of its complementary places in P_S;

if the resulting GMEC $(\overline{\boldsymbol{w}}, \overline{\boldsymbol{k}})$ contains places

in $P_R \cup \{p_0\}$, **then stop**

(the closed loop net is not ES^{2}PR)

else begin

compute the incidence matrix of its

monitor place $\boldsymbol{C}_c = -\overline{\boldsymbol{w}}^T \boldsymbol{C}$;

if $\forall t \in T$ such that $\boldsymbol{C}_c(t) < 0$, it holds

that $\boldsymbol{C}_c(t) = -1$, **then**

begin

let p_M be the monitor place

forcing the GMEC $(\overline{\boldsymbol{w}}, \overline{\boldsymbol{k}})$;

let $P_R = P_R \cup \{p_M\}$;

let $\boldsymbol{Pre} = \begin{bmatrix} \boldsymbol{Pre} \\ \boldsymbol{Pre}_c \end{bmatrix}$,

$\boldsymbol{Post} = \begin{bmatrix} \boldsymbol{Post} \\ \boldsymbol{Post}_c \end{bmatrix}$;

let $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X} \\ \overline{\boldsymbol{w}} \end{bmatrix}$,

$\boldsymbol{m}_0 = \begin{bmatrix} \boldsymbol{m}_0 \\ \boldsymbol{k} - \overline{\boldsymbol{w}}^T \boldsymbol{m}_0 \end{bmatrix}$,

$\boldsymbol{k} = \boldsymbol{X}^T \boldsymbol{m}_0$;

goto 4

end

end

else stop

end

else stop ■

At each step Algorithm 14 requires matrix multiplication, but at step 4 it requires the solution of an IPP.

Therefore, we start with a value of $m_0(p_0) = 2$. We evaluate if there are siphons that may become empty by solving an integer linear programming problem of the form (10).

For any siphon $\mathcal{S}$, we compute the resulting GMEC $(\overline{\boldsymbol{w}}, \overline{k})$ preventing it to be empty. If $(\overline{\boldsymbol{w}}, \overline{k})$ only contains places in P_S, we compute the incidence matrix of its monitor place $\boldsymbol{C}_c = -\boldsymbol{w}^T\boldsymbol{C}$. If the monitor place only has ordinary output arcs, then both statements (a) and (b) of Theorem 7 hold, and the closed loop net belongs to the ES^{2}PR class. In such a case, we add a new monitor p_M to the net to prevent the siphon from becoming empty. After a few steps the procedure converges to a live net. We increase the value of $m_0(p_0)$ of one token and continue the procedure.

Note that program (10) gives only sufficient conditions for liveness (and not necessary) due to the approximation of the reachability set with the larger set of invariant markings. However, if a solution is found this solution is maximally permissive: if the siphon controlled by the monitor never gets empty the monitor is behavioral redundant.

Proposition 15. The liveness enforcing policy computed by Algorithm 14 is maximally permissive.

Proof. The monitors added by Algorithm 14 prevent the system from reaching a marking $\boldsymbol{m}$ belonging to the invariant set and such that a siphon becomes empty. Note that since the skeleton net contains only controllable and observable transitions, this monitor is also maximally permissive, i.e., it only blocks transitions that yields $\boldsymbol{m}$. Now, if $\boldsymbol{m}$ is not reachable, then the monitor is redundant and does not modify the behaviour of the net. On the contrary, if $\boldsymbol{m}$ is reachable then the monitor effectively prevents reaching a deadlock condition in a maximally permissive way. □

Example 16. We apply the proposed procedure to the application of interest here. We found out that the procedure could be successfully applied up to the case in which p_0 is initially marked with $k = 7$ tokens, in the sense that by adding new monitors the net is always an ES^{2}PR net and after a finite number of steps the net converges to a structure where no siphon may become empty.

In Table 1 we have reported the siphons computed for $k = m_0(p_0)$ varying from 3 to 7 and the corresponding GMECs. Note that when $k = 2$, no siphon is determined being the net live when no more than two trains are contemporary contained in it. When $k = 8$, the procedure finds out a siphon that cannot be controlled by a monitor with ordinary output arcs, thus we have to stop because assumption (a) of Theorem 7 is violated. More precisely, when $k = 8$ we determine $\mathcal{S} = \{p_8, p_9, p_{11}, p_{17}, p_{24}, p_{26}, p_{31}\}$ and the corresponding GMEC is $2m_7 + 2m_{10} + m_{12} + 2m_{14} + m_{16} + m_{19} + 2m_{22} \leq 13$ whose monitor has non–ordinary output arcs. ■

5.3 Discussion

As already mentioned in the Introduction, a deadlock avoidance scheme similar to our approach has been recently proposed by Park and Reveliotis in [35]. The procedure in [35] is more general than ours, but requires solving an IPP with a larger number of binary variables, that are those that significantly increase the computational complexity of the procedure. In particular, while in our approach, the number of binary variables is $|P|$, in the approach by Park and Reveliotis the number of binary variables is equal to $|P| + |T| + |Pre|$. Thus we suggest that the two procedures may be used in conjunction. So we first start with our procedure, and whenever a monitor place with only ordinary Pre arcs and satisfying the

k	Siphons	GMECs
3	$\{p_5, p_6, p_7, p_8\}$	$m_4 + m_9 \leq 2$
3	$\{p_{18}, p_{19}, p_{20}, p_{21}\}$	$m_{17} + m_{22} \leq 2$
3	$\{p_{15}, p_{16}, p_{17}, p_{18}\}$	$m_{14} + m_{19} \leq 2$
4	$\{p_{17}, p_{19}, p_{24}, p_{25}\}$	$m_{14} + m_{22} \leq 3$
4	$\{p_2, p_3, p_4, p_5\}$	$m_1 + m_6 \leq 3$
5	$\{p_4, p_6, p_{23}, p_{27}\}$	$m_1 + m_9 \leq 4$
6	$\{p_8, p_9, p_{11}, p_{13}, p_{14}, p_{15}\}$	$m_7 + m_{10} + m_{12} + m_{16} \leq 5$
7	$\{p_5, p_6, p_8, p_{11}, p_{13}, p_{14}, p_{15}\}$	$m_4 + m_7 + m_9 + m_{10} + m_{12} + m_{16} \leq 6$
7	$\{p_8, p_9, p_{11}, p_{13}, p_{15}, p_{17}, p_{18}\}$	$m_7 + m_{10} + m_{12} + m_{14} + m_{16} + m_{19} \leq 6$

Table 1: Results of the liveness enforcing procedure.

necessary and sufficient conditions (NSC) is derived, then we go on with it. On the contrary, if at a certain step we find out that a monitor with non-ordinary Pre arcs or not satisfying the NSC should be added, we switch to the approach proposed by Park and Reveliotis.

In the example discussed above we do not apply the procedure by Park and Reveliotis because using our procedure we are able to ensure liveness of the model for a number of trains up to 7 and it can be easily seen via numerical simulations, that it is desirable to allow no more than 5 trains in the network to bound the time it takes a train to go from one end station to the other one.

5.4 Numerical simulations

In this subsection we present the results of some numerical simulation performed via the software SIRPHYCO [18]. During numerical simulations we associate a time delay to each transition, corresponding to the time a train requires to cross over a given segment, or equivalently, in the case of the points models, it denotes the time required to change the enabled track. More precisely, we assume stochastic transitions, with an exponentially distribute law, thus the chosen time delays represent average values.

We assume that one train starts moving from station α to station δ while k trains (with $k \geq 0$) are moving from station δ to α. We compute the traversal time of first train, i.e., the time it spends within the net before reaching station δ and leaving the net. This traversal time grows with k, as shown in figure 13. In particular, we observe that the traversal time significantly increases for $k > 4$, i.e., we may conclude that $B = k + 1 = 5$ is the maximum number of trains the considered net can effectively manage.

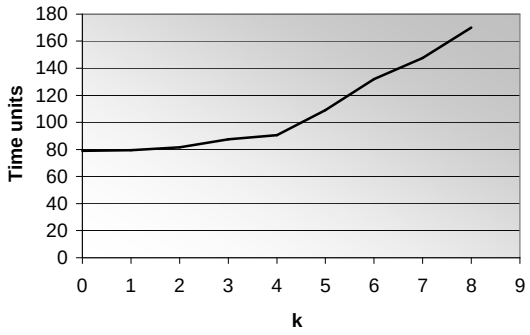

Figure 13: The results of the numerical simulation presented in Subsection 5.4.

6 Conclusions

In this paper we have studied the problem of modeling railway networks with Petri nets so as to apply supervisory control to automatically design a controller that both ensures safeness and liveness. Transitions have been assumed either (un)controllable or (un)observable so as to represent sensors and semaphores.

The proposed procedure is based on local computations applied to a distributed net, thus making it easily extensible to even large dimensions problems. We used both generalized mutual exclusion constraints and constraints involving the firing vector, and the corresponding control structures take the form of monitor places.

Then, we provide a high–level description of a railway network using a skeleton net that belongs to a particular class of Petri nets, the ES^{2}PR nets. The main feature of this class is that liveness properties can be easily characterized and verified by structural analysis. One of the main contributions of this work consists in the derivation of the necessary and sufficient condition that guarantee that a closed loop net, constructed adding a monitor place to an ES^{2}PR net, still belongs to this class. This characterization provides a useful test when enforcing liveness by applying an original recursive procedure based on siphon analysis.

Our future research in this area will be twofold.

Firstly, we plan to focus on the optimization of the net operations. In particular, we will consider the problem of scheduling the departures and the stops of the trains so as to minimize the time spent to run along certain roads, also taking into account how popular the different ways are. Since Petri nets are an efficient tool for the solution of scheduling problems [32], this would provide within a unified framework the solution to the most relevant problems that occur in the railway management, e.g., safeness, liveness, and scheduling.

Secondly, we plan to make a detailed comparison among our approach and other control approaches that we can be derived considering other classes of Petri nets, such as S^{3}PR [12], ES^{3}PR [40], S^{2}LSPR nets [34], etc.

Appendix A: Basic terms of railway operation

In this Appendix, following [33], we briefly summarize the definition of the main railway concepts used in this paper. For more detailed definitions and illustrations we address to [33].

Axle counter. A track clear detection system consisting of counting points at both ends of a section and a counter connected to the counting points. The occupancy of a section is detected by comparing the number of axles which enter the section with the number of axles which leave the section.

Points. The movable parts of a turnout that are operated to set up different routes.

Semaphore. A device that generates signals that give the aspect by the position of movable arms or discs, or by a light signal that displays the aspects by the color and the position of the lights.

Station. A station is a set of interlocked tracks between entry and exit points.

Tracks. Tracks are the roadways of a railway system.

Acknowledgements. The authors would like to thank Asma Ghaffari and Xiaolan Xie for their useful comments and valuable discussions, and Fabrizio Diana who collaborated in the initial phase of this work.

References

[1] K. Barkaoui, I. ben Abdallah, "A deadlock prevention method for a class of FMS," *1995 IEEE Int. Conf. on Systems, Man and Cybernetics*, pp. 4119-4124, Vauncouver, Canada, 1995.

[2] K. Barkaoui, A. Chaoui, B. Zouari, "Supervisory control of discrete event systems based on structure theory of Petri nets ," *1997 IEEE Int. Conf. on Systems, Man and Cybernetics*, pp. 3750-3755, Orlando, USA, 1997.

[3] F. Basile, P. Chiacchio, A. Giua, C. Seatzu "Deadlock recovery of controlled Petri net models using observers," *8th IEEE Int. Conf. on Emerging Technologies and Factory Automation*, pp. 441-449, Antibes, France, October 2001.

[4] J. Billington, "Many–sorted high–level nets," *Proc. of the 3rd Int. Work. on Petri Nets and Performance Models*, Kyoto, Japan, pp. 166-179, 1989.

[5] I. Britton, Links galore – railroad/railway links. http://www.britton2000.com/links/railroad.htm, 2000.

[6] D. Bromage, Railpage: Information on Australian Railways. http://www.railpage.org.au, 2002.

[7] Commission of the European Communities, White Paper on "European transport policy for 2010: time to decide," *COM (2001) 370*, 12/09/2001.

[8] F. Chu, X. Xie, "Deadlock analysis of Petri nets using siphons and mathematical programming," *IEEE Trans. on Robotics and Automation*, Vol. 13, No. 6, pp. 793-804, 1997.

[9] G. Decknatel, and E. Schnieder, "Modelling railway systems with hybrid Petri nets," *Proc. 3rd Int. Conf. on Automation of Mixed Processes*, Reims, France, March 1998.

[10] F. Diana, A. Giua, C. Seatzu "Safeness-enforcing supervisory control for railway networks," *2001 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, pp. 99-104, Como, Italy, July 2001.

[11] A. Di Febbraro, and A. Ferrara, "A new two–level model for multiclass freeway traffic," *IEEE Trans. on Vehicular Technology*, pp. 189–200, 1996.

[12] J. Ezpeleta, J.M. Colom, J. Martínez, "A Petri net based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. on Robotics and Automation*, Vol. 11, No. 2, pp. 173–184, April 1995.

[13] M.P. Fanti, M. Zhou, "Deadlock Control Methods in Automated Manufacturing Systems," *IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans*, Vol. 34, No. 1, pp. 5–22, January 2004.

[14] M.P. Fanti, A. Giua, C. Seatzu, "Monitor design for colored Petri nets: an application to deadlock prevention in railway networks," *Control Engineering Practice*, Vol. 14, No. 10, pp. 1231–1247, October 2006.

[15] H.J. Genrich, "Predicate/Transition nets," In W. Brauer, W. Reisig, and G. Rozenberg (eds), *Advances in Petri nets*, Lecture Notes in Computer Science, Vols. 254 and 255, Springer Verlag, 1987.

[16] A. Giua, F. DiCesare, M. Silva, "Generalized mutual exclusion constraints for nets with uncontrollable transitions", *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics*, Chicago, USA, pp. 974–979, October 1992.

[17] A. Giua, C. Seatzu "Liveness enforcing supervisors for railway networks using ES^{2}PR Petri nets," *Proc. WODES02: 6th Int. Work. on Discrete Event Systems*,Zaragoza, Spain, October 2002.

[18] G. Guerre-Chaley, "Conception et réalisation d'un simulateur de réseaux de Petri continus et hybrides pour l'evaluation de performances des systmes discrets et/ou continus," *CNAM dissertation, Conservatoire National des Arts et Métiers*, Grenoble, France, April 1997.

[19] L. E. Holloway, B. H. Krogh, A. Giua, "A survey of Petri net methods for controlled discrete event systems", *Discrete Event Systems*, Vol. 7, pp. 151-190, 1997.

[20] P.G. Howlett and P.J. Pudney, *Energy efficient train control*, Advances in Industrial Control, Springer Verlag, 1995.

[21] M. V. Iordache, J. O. Moody and P. J. Antsaklis, "Synthesis of deadlock prevention supervisors using Petri nets", *IEEE Trans. on Robotics and Automation*, Vol. 18, No. 1, pp. 59–68, February 2002.

[22] C.W. Janczura, "Modelling and analysis of railway network control logic using coloured Petri nets," *Ph.D. Thesis*, University of South Australia, August 1998.

[23] N.G. Levson and J.L. Stolzy, "Safety analysis using Petri nets," *IEEE Trans. on Software Engineering*, Vol. 13, N. 3, pp. 386–397, 1987.

[24] Z. Li and M.C. Zhou, "Elementary Siphons of Petri Nets and Their Applications to Deadlock Prevention in Flexible Manufacturing Systems," IEEE Trans. on Systems, Man, and Cybernetics, Vol. 34, No. 1, pp. 38–51, 2004.

[25] A. Moen Hagalisletto, I.C. Yu, "Large scale construction of railroad models from specifications," *IEEE Int. Conf. on Systems, Man, and Cybernetics*, Den Haag, Holland, pp. 6212 - 6219, 2004.

[26] J.O. Moody, K. Yamalidou, M.D. Lemmon and P.J. Antsaklis, "Feedback control of Petri nets based on place invariants," *Automatica*, Vol. 32, N. 1, pp. 15–28, January 1996.

[27] J.O. Moody, P.J. Antsaklis, "Supervisory control of discrete event systems using Petri nets," Kluwer Academic Publishers, 1998.

[28] T. Murata, "Petri nets: properties, analysis and applications," *Proc. of the IEEE*, Vol. 77, N. 4, pp. 541–580, April 1989.

[29] T.K.S. Murthy, L.S. Laurence, and R.E. Rivier, editors, *Computers in Railways Management.* Springer Verlag, 1987.

[30] T.K.S. Murthy, F.E. Young, S. Lehmann, and W.R. Smith, editors, *Computers in Railways Installations, Track and Signalling.* Springer Verlag, 1987.

[31] O.S. Nock, *Hystoric Railway Disasters.* Ian Allan, 1967.

[32] X. Ren, and M.C. Zhou, "Tactical scheduling of rail operations: a Petri net approach," *IEEE Int. Conf. Systems, Man and Cybernetics*, Vancouver BC, pp. 3087–3092, 1995.

[33] J. Pachl, "Railway operation and control," VTD Rail Publishing, 2002.

[34] J. Park, S.A. Reveliotis, "Algebraic sinthesis of efficient deadlock avoidance policies for sequential resource allocation systems," *IEEE Trans. on Robotics and Automation*, Vol. 16, No. 2, pp. 190–195, 2000.

[35] J. Park, S.A. Reveliotis, "Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings," *IEEE Trans. on Automatic Control*, Vol. 46, No. 10, pp. 1572–1583, 2001.

[36] J. Park and S. A. Reveliotis, "Liveness-Enforcing Supervision for Resource Allocation Systems with Uncontrollable Behavior and Forbidden States," *IEEE Trans. on Robotics & Automation*, Vol. 18, No. 2, pp. 234–240, 2002.

[37] V.A. Profillidis, *Railway Engineering.* Avebury Technical, 1995.

[38] P.J. Ramadge and W.M. Wonham, "Control of discrete-event systems", *Proc. of the IEEE*, Vol. 77, N. 1, pp. 81-98, January 1989.

[39] S.A. Reveliotis, "Conflict resolution in AGV systems", *IIE Trans.*, Vol. 32, No. 7, pp. 647-659, 2000.

[40] F. Tricas, F. García-Vallés, J.M. Colom, and J. Ezpeleta, "A structural approach to the problem of deadlock prevention in processes with resources," *Proc. WODES98: 4th Int. Work. on Discrete Event Systems* , (Cagliari, Italy), pp. 273–278, August, 1998.

[41] F. Tricas, F. García-Vallés, J.M. Colom, and J. Ezpeleta, "A partial approach to the problem of deadlocks in processes with resources," *Tech. Rep. GISI–RR-97-05*, Dpto. de Informática e Ingegnería de Sistemas – Univ. de Zaragoza, September 1997.

[42] F. Tricas, J.M. Colom, and J. Ezpeleta, "A solution to the problem of deadlocks in concurrent systems using Petri nets and integer linear programming," *Proc. of the 11th European Simulation Symp.* (Erlangen, Germany), pp. 542–546, October 1999.

[43] N.Q. Wu, M.C. Zhou, "Deadlock modeling and control of automated guided vehicle systems," *IEEE/ASME Trans. on Mechatronics*, Vol. 9, No. 1, pp. 50-57, 2004.