Liveness enforcing supervisors for railway networks using ES^{2}PR Petri nets

Alessandro Giua, Carla Seatzu
Department of Electrical and Electronic Engineering, University of Cagliari,
Piazza d'Armi — 09123 Cagliari, Italy
{giua,seatzu}@diee.unica.it

Abstract

In this paper we consider a high-level description of a railway network using a skeleton net that belongs to the class of ES^{2}PR nets. The resource places of this model correspond to the action of a safeness enforcing supervisor. Liveness constraints may also be enforced for this class by adding appropriate monitor places designed using siphon analysis. We show how this can be done without an exhaustive computation of all siphons and characterize the cases in which this procedure can be recursively applied, giving a simple test for the closed loop net to remain an ES^{2}PR net.

1 Introduction

In this paper we deal with the problem of designing a liveness enforcing supervisory controller for railway networks modeled by a place/transition net. The literature on modelling and analyzing railway systems using Petri nets (PN) is not extensive and a good survey is given by Janczura in [11].

In our approach, presented in [5, 8], we give a PN modular representation of railway networks in terms of stations and tracks including sensors and semaphores. In [5] we also showed how the safe operation[1] of such a net can be expressed by a set of Generalized Mutual Exclusion Constraints (GMECs) [7]. Thus the corresponding *safeness enforcing controller* takes the form of a set of monitor places that can be computed using Moody's parametrization [12].

In [8] we also addressed the problem of *global deadlock* avoidance. In fact, when a safeness enforcing supervisor has been designed, it may well be the case that the closed loop net is not live. A solution to this problem consists in additionally restricting its behavior so that blocking states are never reached.

To solve this problem we propose to apply siphon analysis to a simplified net (that we call skeleton). This net can be thought as a simple state machine Petri net representing the uncontrolled railway network with the addition of monitor places that abstract the behavior of the safeness enforcing supervisor. The overall net thus belongs to the class of ES^{2}PR net [6, 15], a class for which deadlock freeness ensures liveness.

Now, 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 [10] as an example of recent development in this area. One original feature of the approach we firstly presented in [8] 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 a mixed integer linear programming problem (MILPP). Similar techniques were also used in [1, 2, 4, 14].

In our approach monitors are added 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 initial marking of the ES^{2}PR net contains $k = 2$ tokens in the idle place p_0. We determine if from this 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.

In general the addition of such a monitor may give rise to some problems as discussed in [10].

Problem 1: the closed loop net may not be an ES^{2}PR net and we cannot carry on with our iterative procedure. The main contribution 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. 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 may occur and the procedure need to be reapplied. We cannot always ensure that the procedure will eventually converge to a live net.

If a live net has been obtained for k tokens we consider an initial marking with $k+1$ tokens in the idle place p_0. We continue until we reach a value $k = K$ where we have to stop because either the procedure does not converge or the value K is sufficiently large to cover all cases of practical interest.

In the example we present in this paper, the approach can be successfully applied. However, as we mentioned before, this iterative procedure may fail 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 converges to a live net (Problem 2).

[1] By safe operation we mean that collisions are avoided.

Proceedings of the Sixth International Workshop on Discrete Event Systems (WODES'02)
0-7695-1683-1/02 $17.00 © 2002 IEEE

A general solution to Problem 1 was given by Park and Reveliotis. In [14] 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, a MILPP. The class of nets they consider is closed under the addition of a monitor and thus Problem 1 may never occur.

We were not aware of the results of Park and Reveliotis when this paper was written and in fact these results were brought to our attention by an anonymous referee whose help is gratefully acknowledged. We agree that the more general results of [14] reduce the contribution of the present paper. However, we also believe that the approach we propose may still be practically useful in many cases, because the linear characterization we derive requires the solution of a MILPP with a reduced computational complexity (in terms of integer variables) with respect to the one used in [14]. Thus we suggest that our procedure should be initially used and only if it fails because of Problem 1 should the procedure of [14] be invoked during the successive steps.

2 Background

Generalities on Petri nets: In the following we recall the formalism used in the paper. For more details on Petri nets we address to [13].

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 $R(N, \boldsymbol{m}_0)$.

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

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 $\boldsymbol{m}(\mathcal{S}) = \boldsymbol{s}^T\boldsymbol{m}$.

GMECs and monitors: The development of this subsection is kept very concise for sake of brevity. Please, refer to [12] for a more complete discussion of this topic.

Assume we are given a set of legal markings $\mathcal{L} \subseteq \mathbb{N}^m$, expressed by a set of n_c linear inequality constraints called *Generalized Mutual Exclusion Constraints* (GMECs). Each GMEC is a couple $(\boldsymbol{w}, k)$ where $\boldsymbol{w} : P \rightarrow \mathbb{Z}$ 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 $\mathcal{M}(\boldsymbol{W}, \boldsymbol{k}) = \{\boldsymbol{m} \in \mathbb{N}^m \mid \boldsymbol{W}^T\boldsymbol{m} \leq \boldsymbol{k}\}$.

Each constraint requires the introduction of a new place (denoted as *monitor place*) thus the controller net has n_c monitor places and no transition is added. To each monitor place, 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 $\boldsymbol{m}_{c0}$ the initial marking of the monitors. The incidence matrix of the closed loop system is $\boldsymbol{C}' = \left[\boldsymbol{C}^T\ \boldsymbol{C}_c^T\right]^T \in \mathbb{Z}^{(m+n_c) \times n}$ while its initial marking $\boldsymbol{m}'_0$ is $\boldsymbol{m}'_0 = \left[\boldsymbol{m}_0^T\ \boldsymbol{m}_{c0}^T\right]^T$.

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

Theorem 1 ([7]). *If* $\boldsymbol{k} - \boldsymbol{W}^T\boldsymbol{m}'_0 \geq \boldsymbol{0}$ *then a Petri net controller with incidence matrix* $\boldsymbol{C}_c = -\boldsymbol{W}^T\boldsymbol{C}$ *and initial marking* $\boldsymbol{m}_{c0} = \boldsymbol{k} - \boldsymbol{W}^T\boldsymbol{m}_0$ *enforces constraint (1) when included in the closed loop system.*

The controller so constructed is maximally permissive, i.e. it prevents only transitions firings that yield forbidden markings.

3 The class of ES^{2}PR nets

In this section we first recall the definition of two important classes of Petri nets, namely the S^{2}P and ES^{2}PR nets, firstly introduced by Tricas *et al.* in [6, 15]. These classes of nets have been identified because they frequently appear in the framework of manufacturing systems, and for the ES^{2}PR class deadlock and liveness problems may be related to structural elements of the Petri net model — namely, siphons — as discussed in detail in [6, 15]. 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^{2}P) 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^{2}P represents the set of different sequences that a unit of the process can follow across the system. An *Extended Simple Sequential Process with Resources* (ES^{2}PR) is defined as a S^{2}P 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 [6].

Definition 2 ([6]). *A* Simple Sequential Process, *S^{2}P, 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.*

Definition 3 ([6]). *An* Extended Simple Sequential Process with Resources, *ES^{2}PR, 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^{2}P,*
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 important result was proved in [15].

Proposition 4. *Let $\langle N, \boldsymbol{m} \rangle$ be a marked ES^{2}PR 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 above result has been proved [15] for a wider class of generalized Petri nets, the ES^3PR nets, that are a superclass of the ES^{2}PR nets. In this paper however, results are referred to the ES^{2}PR model because it is that of interest here.

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 4. More precisely, let us consider an ES^{2}PR net N with K resource places. Let $(\boldsymbol{w}_{K+1}, k_{K+1})$ be a positive and minimal–support GMEC[2] 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, in the next two lemma we present two intermediate results. Note that in the following we denote by $I_{min}(N)$ the set of minimal P–semiflows of N.

Lemma 5. *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*

$$\left\{ \boldsymbol{y} = \left[\begin{array}{c} \boldsymbol{x} \\ 0 \end{array} \right] \ \mid \ \boldsymbol{x} \in I_{min}(N) \right\} \cup \left\{ \left[\begin{array}{c} \boldsymbol{w} \\ 1 \end{array} \right] \right\} \subseteq I_{min}(N') \tag{2}$$

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

Proof. Proof is carried out in two different steps.

(i) We first prove that:

$$\left\{ \boldsymbol{y} = \left[\begin{array}{c} \boldsymbol{x} \\ 0 \end{array} \right], \ \boldsymbol{x} \in I_{min}(N) \right\} \subseteq I_{min}(N'). \tag{3}$$

Let $\boldsymbol{C}$ ($\boldsymbol{C}'$) be the incidence matrix of N (N'). Being $\boldsymbol{x} \in I_{min}(N)$, it holds that $\boldsymbol{x}^T \cdot \boldsymbol{C} = \boldsymbol{0}$. Thus, $\boldsymbol{y}^T = \left[\boldsymbol{x}^T \ \ 0\right] \cdot \boldsymbol{C}' = \boldsymbol{x}^T \cdot \boldsymbol{C} = \boldsymbol{0}$, i.e., $\boldsymbol{y} = \left[\boldsymbol{x}^T \ \ 0\right]^T$ is a P–semiflow of N'.

We prove by contradiction that $\boldsymbol{y}$ is also minimal. Let us assume that there exists a positive vector $\overline{\boldsymbol{y}} = \left[\overline{\boldsymbol{x}}^T \ \ 0\right]^T$, with $\overline{\boldsymbol{x}} \lneq \boldsymbol{x}$, that is a P–semiflow of N'. This would imply that $\boldsymbol{x}$ is not a minimal P–semiflow of N, that is a contradiction.

(ii) Now, let us prove that

$$\boldsymbol{y} = \left[\begin{array}{c} \boldsymbol{w} \\ 1 \end{array} \right] \subseteq I_{min}(N'). \tag{4}$$

Clearly, being $\boldsymbol{y}^T \cdot \boldsymbol{C}' = \left[\boldsymbol{w}^T \ \ 1\right] \cdot \boldsymbol{C}' = \boldsymbol{w}^T \cdot C - \boldsymbol{w}^T \cdot C = \boldsymbol{0}$, we may be sure that $\boldsymbol{y}$ is a P–semiflow of N'. We may prove by contradiction that it is also minimal. Let us assume that there exists another P–semiflow of N' such that $\overline{\boldsymbol{y}} \leq \boldsymbol{y}$.

Two cases may occur.

— $\overline{\boldsymbol{y}} = \left[\overline{\boldsymbol{x}}^T \ \ 0\right]^T$. This would imply that $\boldsymbol{w} \gneq \overline{\boldsymbol{x}}$ where $\overline{\boldsymbol{x}}$ is a minimal P–semiflow of N. But this leads to a contradiction being by assumption $(\boldsymbol{w}, k)$ a minimal–support GMEC.

— $\overline{\boldsymbol{y}} = \left[\overline{\boldsymbol{x}}^T \ \ 1\right]^T$. In such a case $\tilde{\boldsymbol{y}} = \boldsymbol{y} - \overline{\boldsymbol{y}} = \left[\tilde{\boldsymbol{x}}^T \ \ 0\right]^T$ is a P–semiflow of N' and $\boldsymbol{w} \geq \tilde{\boldsymbol{x}}$, that leads again to a contradiction. □

Lemma 6. *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 $(\boldsymbol{w}_{K+1}, k_{K+1})$ be a positive and minimal–support GMEC only involving places in P_S and r_{K+1} be the corresponding monitor place. Let $N' = (P_S \cup \{p_0\} \cup P_R \cup \{r_{K+1}\}, T, \boldsymbol{Pre}', \boldsymbol{Post}')$ be the closed loop net. It holds that*

$$\mathcal{A} = \left\{ \boldsymbol{y} = \left[\begin{array}{c} \boldsymbol{x} \\ 0 \end{array} \right] \ \mid \ \boldsymbol{x} \in I_{min}(N) \right\} \cup \left\{ \left[\begin{array}{c} \boldsymbol{w}_{K+1} \\ 1 \end{array} \right] \right\} \supseteq I_{min}(N'). \tag{5}$$

Proof. We prove by contradiction that there exists no vector $\boldsymbol{v} \notin \mathcal{A}$ that is a minimal and positive P-semiflow of N'.

Let us first consider table 1. If we neglect the last column, the first $(K+1)$'s row vectors $\boldsymbol{y}_i$, $i = 0, \cdots, K$, represent the P–semiflows of N; vector $\boldsymbol{y}_{K+1}$ (when also the last term is taken into account) is the P–semiflow of the closed–loop net N' that originates from the introduction of the monitor place r_{K+1}.

In the most general case, the hypothetical vector $\boldsymbol{v} \notin \mathcal{A}$ may involve places in $P_S \cup \{p_0\} \cup \{r_1, \cdots, r_K, r_{K+1}\}$. It has been included in the last row of table 1. Clearly, we are interested in the case where the component of the P–semiflow relative to the $(K+1)$–th resource place r_{K+1}

[2] 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.

	p_0	p_1 … p_m	r_1	…	…	…	r_K	r_{K+1}
$\mathbf{y}_0^T$	1	$\mathbf{1}^T$	0	…	0	…	0	0
$\mathbf{y}_1^T$	0	$\mathbf{w}_1^T$	1	…	0	…	0	0
	$\vdots$	$\vdots$	$\vdots$	$\ddots$			$\vdots$	$\vdots$
	$\vdots$	$\vdots$	0		1		0	0
	$\vdots$	$\vdots$	$\vdots$			$\ddots$	$\vdots$	$\vdots$
$\mathbf{y}_K^T$	0	$\mathbf{w}_K^T$	0	…	0	…	1	0
$\mathbf{y}_{K+1}^T$	0	$\mathbf{w}_{K+1}^T$	0	…	0	…	0	1
$\mathbf{v}^T$	v_0	$\mathbf{v}_S^T$			$\mathbf{v}_R^T$			v_{m+K+1}

Table 1: P–semiflows considered in the proof of lemma 6.

is strictly greater than zero, i.e., $v_{m+K+1} > 0$. In fact, if $v_{m+K+1} = 0$, then $[v_0\ \boldsymbol{v}_S^T\ \boldsymbol{v}_R^T]^T$ is a P-semiflow of N.

Now, let us define $\boldsymbol{z} = v_{m+K+1} \cdot \boldsymbol{y}_{K+1} - \boldsymbol{v}$. Different cases may occur.

1. $\boldsymbol{z} = \mathbf{0}$. In such a case $\boldsymbol{v} = v_{m+K+1} \cdot \boldsymbol{y}_{K+1}$ is a non minimal P–semiflow of N' because it only differs from $\boldsymbol{y}_{K+1}$ for the positive constant v_{m+K+1}.

2. $\boldsymbol{z} \geq \mathbf{0}$. In this case $\boldsymbol{z}$ is a P–semiflow (being a non negative vector written as a linear combination of two semiflows) and may be written as $\boldsymbol{z} = \begin{bmatrix}\boldsymbol{x}^T & 0\end{bmatrix}^T$ where $\boldsymbol{x}$ is a P-semiflow of N. Being $\boldsymbol{v} = v_{m+K+1} \cdot \boldsymbol{y}_{K+1} - \boldsymbol{z} \geq \mathbf{0}$, this implies that $v_{m+K+1} \cdot \boldsymbol{w}_{K+1} \geq \boldsymbol{x}$, i.e., $||\boldsymbol{w}_{K+1}|| \supseteq ||\boldsymbol{x}||$. But this leads to a contradiction being by assumption $(\boldsymbol{w}_{K+1}, k_{K+1})$ a minimal–support positive GMEC.

3. $\boldsymbol{z} \lneq \mathbf{0}$. In this case $\tilde{\boldsymbol{z}} = -\boldsymbol{z} \in I_{min}(N')$ and $\boldsymbol{v} = \tilde{\boldsymbol{z}} + v_{m+K+1} \cdot \boldsymbol{y}_{K+1}$ is not minimal.

4. Now, let us assume that $\boldsymbol{z}$ has both positive and negative components. We can write: $\boldsymbol{z} = [\boldsymbol{z}^T\ \boldsymbol{z}_R^T\ z_{m+K+1}]^T$ where $\boldsymbol{z} \in \mathbb{Z}^{m+1}$, $\boldsymbol{z}_R \in \mathbb{Z}_-^K$. Note that $\boldsymbol{z}_R \leq \mathbf{0}$ by construction, given the structure of $\boldsymbol{y}_{K+1}$ and the assumption that $\boldsymbol{v} \geq \mathbf{0}$ (see table 1). Thus we have that $\boldsymbol{z} + \sum_{i=1}^K v_{R,i}\boldsymbol{y}_i = \begin{bmatrix} \boldsymbol{r}^T & \mathbf{0}^T & 0 \end{bmatrix}^T$ where $\boldsymbol{r} \in \mathbb{Z}^{m+1}$.

Now, let $\delta \in \mathbb{N}$ be the smallest non–negative constant such that $\boldsymbol{z} + \sum_{i=1}^K v_{R,i}\boldsymbol{y}_i + \delta\boldsymbol{y}_0 = \begin{bmatrix} \tilde{\boldsymbol{r}}^T & \mathbf{0}^T & 0 \end{bmatrix}^T \geq \mathbf{0}$. Note that $\tilde{\boldsymbol{r}}$ is a non–null left annuler of $\boldsymbol{C}_{E^2S}$, where $\boldsymbol{C}_{E^2S}$ is the incidence matrix of the E^{2}S net, i.e., $\tilde{\boldsymbol{r}}^T\boldsymbol{C}_{E^2S} = \mathbf{0}$. Since this net has a single minimal P–semiflow $\mathbf{1}$, two different cases may occur.

(a) $\tilde{\boldsymbol{r}} = \mathbf{0}$. In such a case, $\boldsymbol{v} = v_{m+K+1} \cdot \boldsymbol{y}_{K+1} + \sum_{i=1}^K v_{R,i} \cdot \boldsymbol{y}_i + \delta\boldsymbol{y}_0$ and cannot be a minimal P–semiflow, thus leading to a contradiction.

(b) $\tilde{\boldsymbol{r}} = \mathbf{1}$. In such a case $\tilde{r}(p_0) = 1 = z(p_0) = v_{m+K+1}\, y_{K+1}(p_0) - v(p_0) = -v(p_0)$, being $y_{K+1}(p_0) = 0$ (see table 1). But this leads to a contradiction, being by assumption $\boldsymbol{v} \geq \mathbf{0}$.

In this case $\begin{bmatrix}\tilde{\boldsymbol{r}}^T & \mathbf{0}^T & 0\end{bmatrix} = \boldsymbol{y}_0$ and by definition of δ we have $\delta = 0$. In this case, $v_0 = -y_0(p_0) = -1$ and this leads to a contradiction. □

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 $(\boldsymbol{w}_{K+1}, k_{K+1})$ only involving places in P_S. The closed loop net $N' = (P_S \cup \{p_0\} \cup P_R \cup \{r_{K+1}\}, T, \boldsymbol{Pre}', \boldsymbol{Post}')$ 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) $\{p_0\} \cup P_R \cap ||\boldsymbol{w}_{K+1}|| = \emptyset$, *i.e.,* $||\boldsymbol{w}_{K+1}|| \subseteq P_S$.

Proof. *(if)* We first observe that the addition of the monitor place r_{K+1} relative to the positive and minimal–support GMEC $(\boldsymbol{w}_{K+1}, k_{K+1})$ produces a new P–semiflow $\begin{bmatrix}\boldsymbol{w}_{K+1}^T & 1\end{bmatrix}^T$. Now, the *if* statement is trivially verified because, by lemma 5 and lemma 6, it holds that:

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

(only if) If (a) is violated then condition 3 in definition 3 is not satisfied; if (b) is violated, then condition 4 of definition 3 is not satisfied. □

4 Enforcing liveness constraints

In a previous work [5] we have studied in detail the problem of modeling and controlling railway networks with Petri nets. Now we focus our attention to the problem of *global deadlock* avoidance that has already been partially considered in [8].

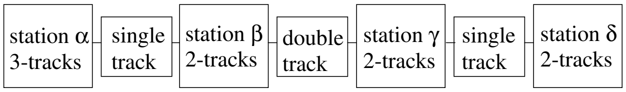

Figure 1: Scheme of the railway network.

Consider, as an example, the railway system sketched in figure 1 [5, 8], 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.

Once the procedure of [5] has been applied and the safeness enforcing supervisory controller has been designed, a skeleton Petri net model of the supervised network (at this level of abstraction all transitions can be considered as controllable and observable) can be easily constructed. For the railway system in figure 1 the skeleton net is shown in figure 2; here the monitors inside rectangles limit the number of trains within stations and tracks according to each station or track capacity. The monitor place p_0 contains the maximum number B of trains that may be allowed into the network.

It is easy to verify using this skeleton model that several blocking conditions may occur. Consider the case in which $B = 3$ 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 deadlock. Note that the set of places $\{p_5, p_6, p_7, p_8\}$ is an empty deadlock.

To determine a maximally permissive liveness enforcing control policy, we first observe that the reduced model of the railway network obtained removing from the skeleton net all monitor places, apart from p_0, is a S^{2}P net. In fact, it is an ordinary and strictly connected state machine with two circuits — both containing p_0 — and $P_S = \{p_1, p_3, p_4, p_6, p_7, p_9, p_{10}, p_{12}, p_{14}, p_{16}, p_{17}, p_{19}, p_{20}, p_{22}\}$.

Moreover, by theorem 7 the whole skeleton net reported in figure 2 is an ES^{2}PR net, because all monitors are relative to minimal–support positive GMECs only involving places in P_S and have only ordinary output arcs. In particular, 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:

$$\left\{\begin{array}{lll} m_1 + m_3 \leq 3 & m_4 + m_6 \leq 1 & m_7 + m_9 \leq 8 \\ m_{12} \leq 1 & m_{20} + m_{22} \leq 2 & m_{14} + m_{16} \leq 2 \\ m_{17} + m{+}_{19} \leq 1 & m_{10} \leq 1. & \end{array}\right.$$

To ensure liveness of the model, we use proposition 4 as in [15]. 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 the addition of a new monitor may yield 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.

We compute the liveness enforcing monitors, using a linear algebraic technique based on integer programming that does not require the exhaustive enumeration of all siphons, whose number is too large even for a small net such as the one we consider. Although solving a linear integer optmization 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 [4].

First of all we observe that the net in figure 2 has 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 place p_0 whose support is P_S. 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 23 × 9 matrix $\boldsymbol{X}$ contains a P-semiflow and $\boldsymbol{k} = \boldsymbol{X}^T \boldsymbol{m}_0$ is a 9 × 1 vector whose components represent the token content of each semiflow. Although we cannot formally prove that $R(N, \boldsymbol{m}_0) = \mathcal{I}_X(N, \boldsymbol{m}_0)$, if we can ensure that no deadlock marking $\boldsymbol{m} \in \mathcal{I}_X(N, \boldsymbol{m}_0)$ is reachable, then no reachable marking may be a deadlock.

To determine if there are siphons that need to be controlled in a structurally bounded ordinary net one can use, as shown in [3], the following mixed integer linear program:

$$\left\{\begin{array}{ll} \min & \boldsymbol{1}^T \boldsymbol{s} \\ s.t. & K_1 \boldsymbol{Pre}^T \boldsymbol{s} \geq \boldsymbol{Post}^T \boldsymbol{s} \\ & \boldsymbol{X}^T \boldsymbol{m} = \boldsymbol{k} \\ & K_2 \boldsymbol{s} + \boldsymbol{m} \leq K_2 \boldsymbol{1} \\ & \boldsymbol{1}^T \boldsymbol{s} \geq 1 \end{array}\right. \tag{6}$$

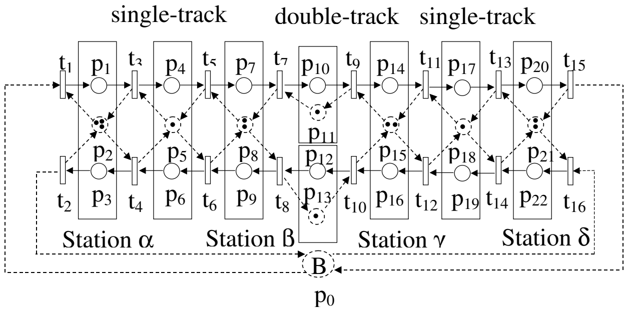

Figure 2: The skeleton Petri net model of the railway network in figure 1.

where $\boldsymbol{s} \in \{0,1\}^m$ and $\boldsymbol{m} \in \mathbb{N}^m$ are the unknowns. Here the two constants K_1 and K_2 are defined as: $K_1 = \max\{\boldsymbol{1}^T \boldsymbol{Post}(\cdot, t) \mid t \in T\}$ and $K_2 = \max\{m(p) \mid p \in P, \boldsymbol{m} \in R(N, \boldsymbol{m}_0)\}$ (for the net in figure 2 $K_1 = 2$ and $K_2 = B$).

We claim (a formal proof can be found in [3]) that program (6) has an admissible solution $(\boldsymbol{m}, \boldsymbol{s})$ if there exists a reachable marking $\boldsymbol{m}$ such that the siphon $\mathcal{S}$ with characteristic vector $\boldsymbol{s}$ is empty. The objective function chosen for the program (6) ensures that only minimal siphons are computed.

We started with a value of $B = 2$ and applied the previously described approach to determine siphons to be controlled. As such a siphon is found, we add a new monitor 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 B of one token and continue the procedure.

Note that program (6) gives only sufficient conditions for liveness (and not necessary) due to the approximation of the reachability set with the larger potentially reachable set. However, if a solution is found, as in the example we study in the paper, this solution is maximally permissive: if the siphon controlled by the monitor never gets empty the monitor is behaviorally redundant.

Now, let us discuss in detail the procedure we adopted to compute the positive GMEC preventing a given siphon from becoming empty. Let $\mathcal{S}$ be the siphon obtained by solving an integer linear programming problem of the form (6). For this siphon it should be $\boldsymbol{m}(\mathcal{S}) = \sum_{p_i \in \mathcal{S}} m_i \geq 1$ or equivalently, $-\boldsymbol{m}(\mathcal{S}) = -\sum_{p_i \in \mathcal{S}} m_i \leq -1$. First of all, for each place $p \in \{p_0\} \cup P_R$ such that $p \in \mathcal{S}$, we replace its marking with the marking of its complementary places. Then, if the resulting GMEC $(\boldsymbol{w}, k)$ only contains places in P_S, we compute the incidence matrix of its monitor place $\boldsymbol{C}_c = -\boldsymbol{w}^T \boldsymbol{C}$. If $\forall\, t \in T$ such that $C_c(t) < 0$, it holds that $C_c(t) = -1$, i.e., 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 table 2 we have reported the siphons computed for B varying from 3 to 7 and the corresponding GMECs preventing them from becoming empty. Note that when $B = 2$, no siphon is determined being the net live when no more than two trains are contained in it.

It is easy to prove that all the above positive GMECs are also minimal–support and the addition of the correspond-

B	Siphons	GMECs	Monitors
3	$\{p_5, p_6, p_7, p_8\}$	$m_4 + m_9 \leq 2$	p_{23}
3	$\{p_{18}, p_{19}, p_{20}, p_{21}\}$	$m_{17} + m_{22} \leq 2$	p_{24}
3	$\{p_{15}, p_{16}, p_{17}, p_{18}\}$	$m_{14} + m_{19} \leq 2$	p_{25}
4	$\{p_{17}, p_{19}, p_{24}, p_{25}\}$	$m_{14} + m_{22} \leq 3$	p_{26}
4	$\{p_2, p_3, p_4, p_5\}$	$m_1 + m_6 \leq 3$	p_{27}
5	$\{p_4, p_6, p_{23}, p_{27}\}$	$m_1 + m_9 \leq 4$	p_{28}
6	$\{p_8, p_9, p_{11}, p_{13}, p_{14}, p_{15}\}$	$m_7 + m_{10} + m_{12} + m_{16} \leq 5$	p_{29}
7	$\{p_5, p_6, p_8, p_{11}, p_{13}, p_{14}, p_{15}\}$	$m_4 + m_7 + m_9 + m_{10} + m_{12} + m_{16} \leq 6$	p_{30}
7	$\{p_8, p_9, p_{11}, p_{13}, p_{15}, p_{17}, p_{18}\}$	$m_7 + m_{10} + m_{12} + m_{14} + m_{16} + m_{19} \leq 6$	p_{31}

Table 2: Results of the liveness enforcing procedure.

ing monitor places does not destroy the structure of the net that still belongs to the ES^{2}PR class. This may be immediately verified by virtue of theorem 7. On the contrary, when $B = 8$, 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 $B = 8$ 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.

Finally let us observe that, as already mentioned in the introduction, a similar approach has been recently proposed by Park and Reveliotis in [14]. The procedure in [14] is more general than ours, but requires solving a MILPP 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 necessary and sufficient conditions (NSC) we 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 actual case, although using our procedure we are able to ensure liveness of the model for a number of trains up to 7, we do not apply the procedure of Park and Reveliotis because, as shown in [8], 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 Conclusions

In this paper we provided 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 contribution of this work consisted in the derivation of the necessary and sufficient condition that assures 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 a recursive procedure that consists in the addition of appropriate monitor places designed using siphon analysis.

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 using 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] 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.

[5] F. Diana, A. Giua, C. Seatzu “Safeness enforcing supervisory control for railway networks,” *2001 IEEE/ASME Int. Conf. on Adv. Intell. Mechatronics* (Como, Italy), pp. 99-104, July 2001.

[6] 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.

[7] A. Giua, F. DiCesare, M. Silva, “Generalized Mutual Exclusion Constraints for Nets with Uncontrollable Transitions”, *Proc. IEEE Int. Conf. on Systems, Man & Cybernetics* (Chicago, USA), pp. 974–979, October 1992.

[8] A. Giua, C. Seatzu “Supervisory control of railway networks with Petri nets,” *40th IEEE Conf. on Decision and Control* (Orlando, USA), pp. 5004-5009, December 2001.

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

[10] M. V. Iordache, J. O. Moody and P. J. Antsaklis, “Automated synthesis of deadlock prevention supervisors using Petri nets”, *ISIS Techinical Report ISIS 2000-003*, May 2000.

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

[12] J.O. Moody, P.J. Antsaklis, “Supervisory control of discrete event systems using Petri nets,” Kluwer Academic Publ., 1998.

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

[14] 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.

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