ISSN 0005–1144

Danko Kezić, Nedjeljko Perić, Ivan Petrović

An Algorithm for Deadlock Prevention Based on Iterative Siphon Control of Petri Net

UDK 004.822:681.515
629.5:004.822
IFAC 3.2.1;5.7.4

Original scientific paper

This paper presents a formal calculation method of a deadlock prevention supervisor by the use of Petri nets. The proposed algorithm uses reachability tree to detect deadlock state and iterative siphon control method to synthesize the deadlock prevention supervisor. Such supervisor is maximally permissive and consists of minimal number of control places. The algorithm is intended for reversible or partially reversible P-T Petri net, but it can also be applied to Ordinary Petri nets. The calculation of the supervisor is illustrated by two examples. The first example shows the synthesis of deadlock prevention supervisor in a manufacturing system consisting of three conveyors and three robots, where the deadlock can occur due to concurrent requests of the conveyors for the robot engagements and unpredictable duration of those engagements. The second example shows the synthesis of deadlock prevention supervisor in a marine traffic system, where dangerous vessel deadlock situations may occur in case of vessels' irregular motion through the system. To avoid this, the vessel traffic is supervised and controlled by traffic lights using the deadlock prevention supervisor, which is responsible for vessels' stopping only in the case of dangerous situation and until this situation elapses.

Key words: deadlock prevention supervisor, Petri net, siphon control

1 INTRODUCTION

Large number of various technical systems falls naturally in the class of Discrete Event Systems (DES), and can be viewed as a sequence of discrete events. The main characteristics of these systems are concurrency or parallelism, asynchronous operations, and event driven operation. In a DES many operations take place simultaneously and the evolution of system events is aperiodic. The completion of one operation may initiate more than one new operation. As a result of these dynamic characteristics there are two situations that can occur: conflict and deadlock. The conflict may occur when two or more processes require a common resource at the same time. For example, two workstations might share a common transport system or need access to the same storage. One simple way to resolve the conflict is to assign a priority level to each of the processes. The deadlock can happen with the sharing of two resources between the two processes. In this case, a state can be reached where none of the processes can continue. Note that one of the processes may proceed if the conflict can be resolved while in the deadlock case nothing can be done to get the system going again. This situation is unacceptable and is usually the result of the system design [2], and must be solved by using supervisory control.

Many authors have also tried to solve the deadlock problem by Petri nets. Barkaoui developed the method of deadlock prevention by control places [1]. Also Ezpelta developed an algorithm for deadlock prevention for the ordinary and conservative S^3PR class of Petri nets [2]. Iordache presents a deadlock prevention algorithm uses iterative siphon control method [3]. Lautenbach investigated the algorithm for finding the minimal siphon inside the net, as well as the algorithm for deadlock prevention by control places for ordinary Petri nets, which do not contain source places [4]. Further, Lewis developed an efficient algorithm for deadlock prevention in the MRF_1 class of Petri nets – the class that describes flexible manufacturing systems [8].

This paper deals with the calculating of deadlock prevention supervisor based on Petri nets. The proposed supervisor is applicable to DES, but it also applicable to the continuous-variable dynamic systems which can be viewed as DES at higher level of abstraction. The paper presents a deadlock prevention algorithm which uses siphon control method. We are using an iterative algorithm for generating deadlock-free Petri net from the Petri net model of the system (Process Petri net). With this algorithm it is possible to synthesize a maximally permissive deadlock prevention supervisor with fewer numbers of control places than algorithm presented in [3].

The algorithm is usable only for reversible or partially reversible P-T Petri net. However, it is possible to transform an ordinary Petri net into a P-T Petri net [3], and proposed algorithm can also be applied to ordinary Petri nets. The deadlock prevention algorithm first calculates the Petri net reachability tree. Then the reachability tree is analyzed in order to find the present deadlock states. If at least one deadlock is found to the Process Petri net, the synthesis of a deadlock-free Petri net must be done by finding all critical minimal siphons of the net and adding control places which is connected to the transitions of the Process Petri net. The control places ensure at least one token in the critical minimal siphons of the net for every reachable marking, which is the necessary condition for deadlock prevention. We illustrate our approach using realistic models of two different systems which can be described as DES: a manufacturing system (a flexible assembly cell with three robots and conveyor), and a marine traffic system.

The paper is organized as follows: Section 2 reviews basics of deadlock prevention supervisor and Petri net properties and describes notations which are used throughout the paper. Section 3 defines the deadlock prevention algorithm which uses iterative siphon control method. Flexible assembly cell example with three robots and conveyors is given in the section 4. Section 5 shows an example of calculating the deadlock prevention supervisor for marine traffic system. The supervisor is verified by a computer simulation of vessels moving through the marine traffic system.

2 DEADLOCK PREVENTION SUPERVISOR BASED ON PETRI NETS

Some of the systems can be viewed as discrete event process G with the deadlock states (if they exist). To prevent occurrence of deadlock states, the process must be supervised by the deadlock prevention supervisor C, which is connected with process in closed loop (Figure 1). The process generates a sequence of discrete events s and sends it to a supervisor C. The supervisor C then generates set of allowable events γ that is allowed to happen in the process G in the next step. The set γ depends on the sequence s and does not consist of the events that can lead a process to deadlock states in the next step.

In this paper the process G and supervisor C are modeled by the Petri net, a useful tool for describing discrete event systems.

A Petri net (Place/Transition net) is a 6-tuple [6]:

$$Q = (P, T, I, O, \Phi, \mathbf{m}_0), \tag{1}$$

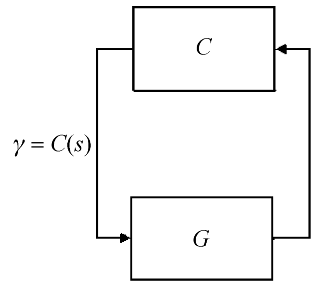

Fig. 1 Closed loop of the process and supervisor

where:

$P = \{P_1, P_2, \ldots, P_8\}$ – set of places,
$T = \{t_1, t_2, \ldots, t_n\}$ – set of transitions,
$P \cap T = \varnothing$
$I : P \times T\} \to \{0,1\}$ – input function,
$O : T \times P\} \to \{0,1\}$ – output function,
$\Phi : (I,O) \to \{1,2,3,\ldots\}$ – weight function,
$\mathbf{m}_0 : P \to \{0,1,2,\ldots\}$ – initial marking vector.

Places and transitions denote states and events in the discrete event system. A place can be empty (without any token) or contains one or more tokens $(m(p) \geq 0)$. A token can, for example, represent the availability of the resource. A transition $t \in T$ is enabled at a marking $\mathbf{m}$ $iff\_\forall p \in \bullet t, m(p) > 0$ ($\bullet t$ is a set of input places to transition t). A transition t that meets the enabled condition is free to fire. When a transition t fires, all of its input places lose a number of tokens, and all of its output places gain a number of tokens. The state of a Petri net changes from state $\mathbf{m}$ to state $\mathbf{m}'$. This fact will be denoted as $\mathbf{m}[t > \mathbf{m}'$ and is described by:

$$\mathbf{m}' = \mathbf{m} + W^T\sigma, \tag{2}$$

where:

$W = O - I$ – incidence matrix,
σ – firing vector.

A reachability tree displays every possible state that can occur in the Petri net after firing all transitions [6]. The reachability tree of the Petri net in Figure 2a) is shown in the Figure 2b). From the reachability tree it is possible to see the main characteristics of the Petri nets, such as reachability, boundness, liveliness, deadlock and reversibility. *Deadlock* is the state when no firing in the Petri net is possible. The Petri net in Figure 2a) is safe because the maximum number of tokens in the places is 1, and is partially reversible because it is possible to reach initial state $\mathbf{m}_0$ after firing transitions $\{t_1, t_3, t_5\}$. The state $\mathbf{m}_4$ in Figure 2b) is »forbidden deadlock state«, because there are no arcs from this state to the other states, and the firing of t_4 from the state $\mathbf{m}_3$ causes deadlock state.

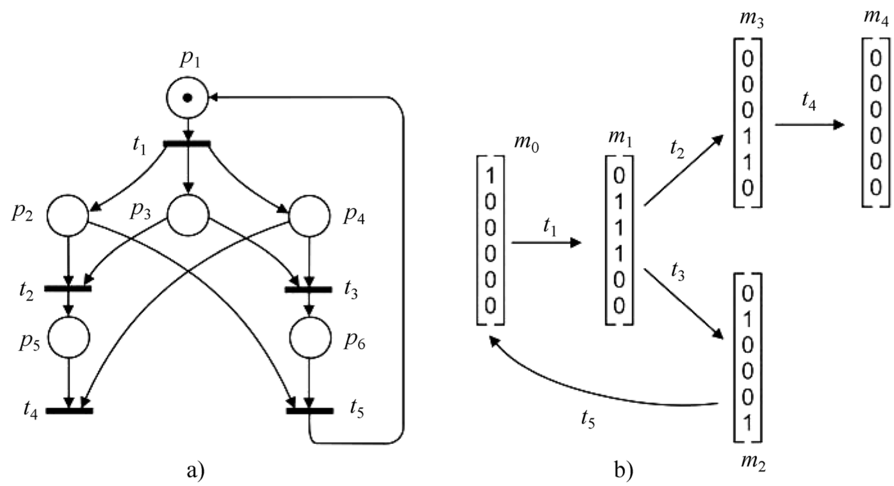

Fig. 2 Petri net a) and its reachability tree b)

Structural characteristics of the Petri net are P and T invariants, siphons and traps. P – invariant corresponds to set of places whose weighted token count remains a constant for all possible markings. Siphon S is the set of the Petri net places for which it is true that each transition having an output arc from the set also has an input arc into the set ($\bullet S \subseteq S\bullet$). Trap T is the set of places for which it is true that each transition having an input arc into the set also has an output arc from the set ($T\bullet \subseteq \bullet T$). A minimal siphon does not contain any new siphons. Once the trap becomes marked, it will always be marked for all future reachable markings. Once the siphon becomes empty, it will always remain empty.

3 ITERATIVE SIPHON CONTROL ALGORITHM FOR DEADLOCK PREVENTION

For P-T Petri nets the arcs weights that connect the places to the transition $w(p,t)$ equal 1 precisely, which can be mathematically expressed as follows [3]:

$$\forall p \in P,\ \forall t \in T,\ \text{if } (p,t) \in \Phi \rightarrow w(p,t) = 1. \quad (3)$$

In a P-T Petri net, which happens to be in a deadlock state, there must be at least one empty siphon (the opposite does not stand). Therefore, in order to prevent deadlock in the P-T Petri net, it is necessary to add one control place to each minimal siphon in the net, which will guarantee that the number of tokens in the minimal siphon is greater or equal to 1 (in other words the minimal siphon must never be emptied, which is mathematically expressed as follows):

$$\sum_{p_i \in S} m(p_i) \geq 1, \quad (4)$$

where :

p_i – places which belong to siphon S.

Formulas for calculating of control place c for a specific siphon are as follows:

$$c\bullet = \left\{ t \in S\bullet : \sum_{p_i \in \bullet t \cap S} w(p_i,t) > \sum_{p_j \in t\bullet \cap S} w(t,p_j) \right\}, \quad (5)$$

$$\bullet c = \left\{ t \in \bullet S : \sum_{p_i \in \bullet t \cap S} w(p_i,t) < \sum_{p_j \in t\bullet \cap S} w(t,p_j) \right\}, \quad (6)$$

$$w(c,t) = \sum_{p_i \in \bullet t \cap S} w(p_i,t) - \sum_{p_j \in t\bullet \cap S} w(t,p_j), \quad (7)$$

$$w(t,c) = \sum_{p_j \in t\bullet \cap S} w(t,p_j) - \sum_{p_i \in \bullet t \cap S} w(p_i,t), \quad (8)$$

$$m_0(c) = \sum_{p_i \in S} m_0(p_i) - 1, \quad (9)$$

where:

$c\bullet$ – represents to where the arcs go from the control place c to the set of siphon output transitions,

$\bullet c$ – represents from where the arcs come to the control place c from the set of siphon input transition,

S – set of places belonging to siphon,

$w(t,c)$ – weight of the arcs from transition t to the place c,

$w(c,t)$ – weight of the arcs from the place c to the transition t,

$m_0(c,t)$ – the initial marking in the control place c,

$m_0(p_i)$ – the initial marking of the siphon.

After calculating control place c for a particular minimal siphon S it is possible to distinguish four cases:

case a) $c\bullet \not\subseteq \bullet S,$ (10)

case b) $c\bullet \subset \bullet S,$ (11)

case c) $\bullet c = \emptyset,\ c\bullet \neq \emptyset,$ (12)

case d) $\bullet c = \emptyset,\ \bullet c = \emptyset.$ (13)

If the case a) and b) are satisfied, the siphon S is uncontrolled (it can be emptied). The control place c must be added and connected to the Petri net. The control place c and siphon S form a P – invariant, and siphon S thus cannot be emptied (controlled siphon). Adding a control place c, all reachable markings of a Petri net must satisfy constraints (4). These siphons are called critical minimal siphons.

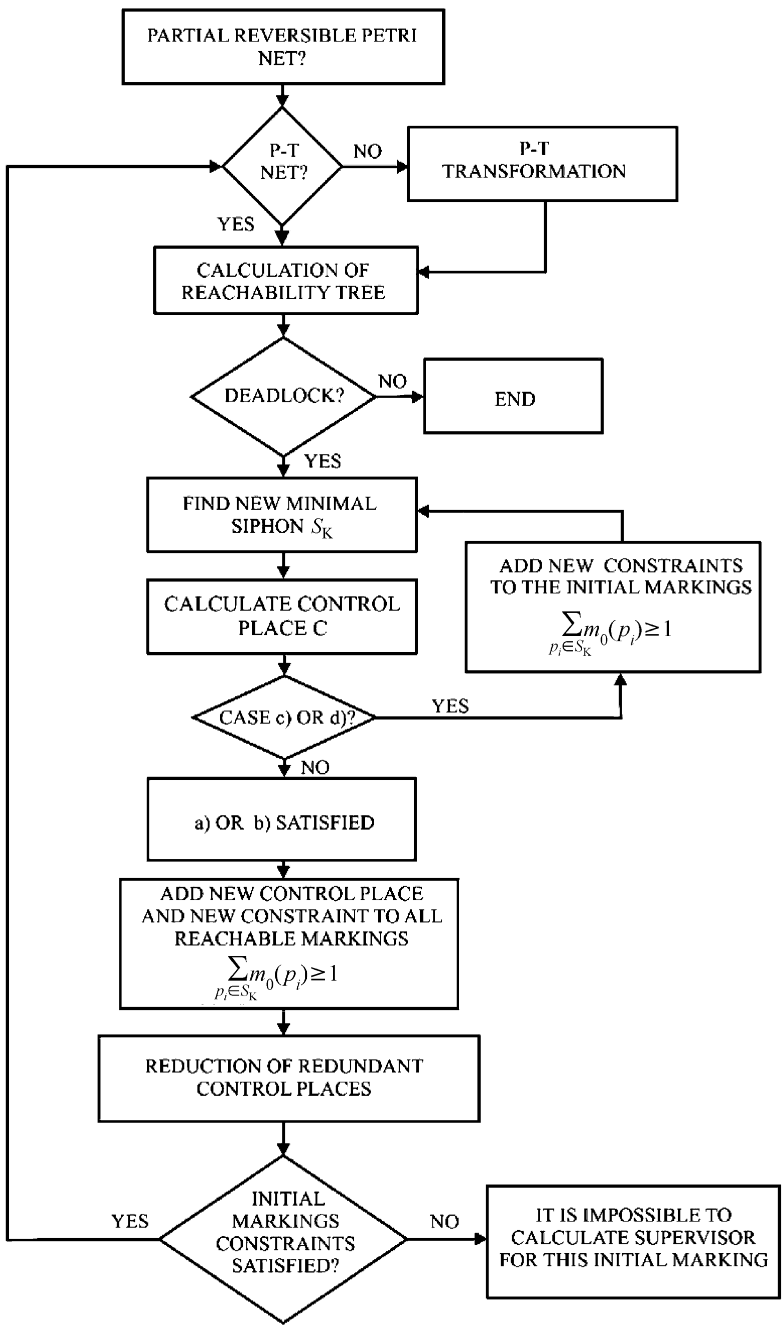

Fig. 3 Algorithm for generating deadlock – free Petri net

If the case c) is satisfied, the siphon S is uncontrolled (it can be emptied, but the control place c cannot control the siphon) [7]. If the case d) is satisfied, the control place c is isolated from the Petri net. Such siphon can never be emptied if the siphon is initially marked as it contains the P – invariant. Such a siphon should not be controlled and they do not create any threat for deadlock occurrence in a Petri net. In this case initial markings of a Petri net must satisfy:

$$\sum_{p_i \in S} m_0(p_i) \geq 1. \tag{14}$$

Iterative algorithm for generating a deadlock-free Petri net is shown in Figure 3 [8]. As it was said, the algorithm is usable for reversible or partially reversible P-T Petri net. But, it is possible to transform an ordinary Petri net into a P-T Petri net [3], and algorithm can be also applied to ordinary Petri net. The next step of the algorithm (Figure 3) is calculation of the Petri net reachability tree. Then the reachability tree is analyzed in order to find the present deadlock states [8]. If there are no deadlocks, the Petri net is deadlock-free, but if at least one deadlock is found, the synthesis of a deadlock-free Petri net must be done. At this point it is necessary to find the minimal siphons and calculate the control places for them. This will ensure that the siphons are never left without tokens. New control place is added to the Petri net only for minimal siphon that satisfies (10) and (11). The new control place restricts reachable marking of the Petri net, so the new control place puts the constraint (4) to the Petri net. If the siphon satisfies (12) and (13), there is no need to add control place to the net. In that case the siphon needs to be initially marked, and the Petri net must satisfy initial marking constraints (14). Adding a new control place to the net may produce a new uncontrolled minimal siphon and new deadlock state. So we need to calculate the reachability tree again and repeat the whole algorithm. The algorithm ends when it is not possible to find a new deadlock states in the Petri net.

4 A CASE STUDY I: DEADLOCK PREVENTION IN A FLEXIBLE ASSEMBLY CELL

In this section, a flexible assembly cell with three robots and three conveyors (Figure 4) is considered as the case study for the proposed deadlock prevention algorithm [7]. A conveyor and two neighboring robots are needed to carry an assembly task. Conveyor (C_1) requests the left robot (R_1) first, and after acquiring it, requests the right one (R_2). Similarly, conveyors (C_2/C_3) request their left robot (R_2/R_3) first, and after acquiring it, they request

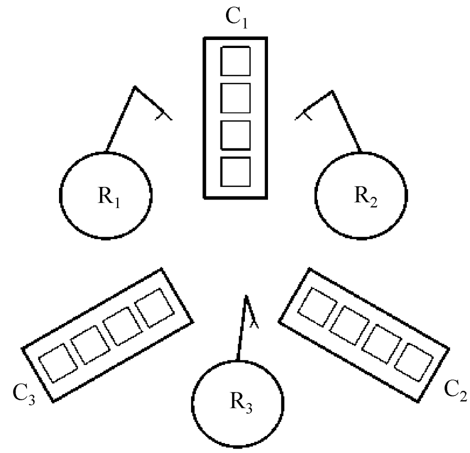

Fig. 4 A flexible assembly cell

the right one (R_3/R_1). Then the assembly operation starts. When the task is completed, the conveyor releases both robots.

Requesting and releasing a robot are two events that can occur concurrently and asynchronously. For example, the three conveyors could simultaneously request their left robots (concurrency). Also, the release of two allocated robots occurs when the assembly task is over, but the time of this event cannot be accurately predicted due to possible delays or errors (asynchronous). The Petri net model must capture the concurrent and asynchronous behavior of the system (Figure 5).

For the Petri net model from Figure 5, the following places can be assigned (Table 1).

Places p_1 to p_9 represent the operations in the system, and places p_{10} to p_{12} represent availability of the robot (Figure 5).

Events in the system are represented by transitions (Table 2).

At the start of the assembly task, all conveyors and robots are free, and the three conveyors are concurrently requesting their left robots. The initial marking of the net is

$$\mathbf{m}_0 = [100100100111]^T.$$

From the initial marking the transitions t_1, t_4 and t_7 are enabled. If these transitions are allowed to fire concurrently, the new marking will be

$$\mathbf{m}' = [010010010000]^T.$$

This means that three conveyors have acquired their left robots and are now waiting for their right one. However, all three robots are already committed and so the process is deadlocked; it cannot proceed because $\mathbf{m}'$ enables no transitions.

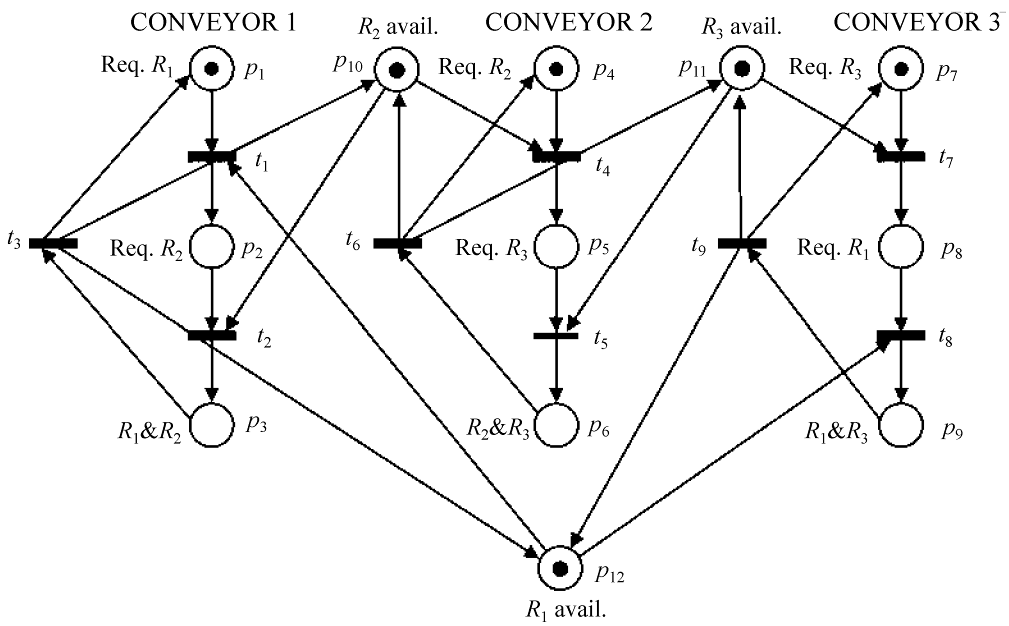

Fig. 5 Petri net model of the flexible assembly cell

Table 1. **Places assigned to Petri net model of the assembly cells**

p_1	conveyor C_1 requesting its left robot R_1
p_4	conveyor C_2 requesting its left robot R_2
p_7	conveyor C_3 requesting its left robot R_3
p_2	conveyor C_1 requesting its right robot R_2
p_5	conveyor C_2 requesting its right robot R_3
p_8	conveyor C_3 requesting its right robot R_1
p_3	conveyor C_1 and its two robots R_1 and R_2 are in use
p_6	conveyor C_2 and its two robots R_2 and R_3 are in use
p_9	conveyor C_3 and its two robots R_3 and R_1 are in use
p_{10}	robot R_2 available
p_{11}	robot R_3 available
p_{12}	robot R_1 available

Table 2. **Transitions assigned to Petri net model of the assembly cells**

t_1	conveyor C_1 acquires its left robot R_1
t_4	conveyor C_2 acquires its left robot R_2
t_7	conveyor C_3 acquires its left robot R_3
t_2	conveyor C_1 acquires its right robot R_2
t_5	conveyor C_2 acquires its right robot R_3
t_8	conveyor C_3 acquires its right robot R_1
t_3	conveyor C_1 releases R_1 and R_2
t_6	conveyor C_2 releases R_2 and R_3
t_9	conveyor C_3 releases R_3 and R_1

To prevent occurrence of deadlock states, the process (which is modeled by the Petri net in Figure 5) must be supervised by the deadlock prevention supervisor, which consists of control places connected to the Petri net. Calculation of the control places could be done using the algorithm for generating deadlock-free Petri net (Figure 3). Following the analyses of the Petri net, it has been determined that there are 7 minimal siphons in the net. These minimal siphons are as follows:

$S_1 = \{p_1, p_2, p_3\}$,
$S_2 = \{p_3, p_5, p_6, p_{10}\}$,
$S_3 = \{p_6, p_8, p_9, p_{11}\}$,
$S_4 = \{p_3, p_6, p_9, p_{10}, p_{11}, p_{12}\}$,
$S_5 = \{p_2, p_3, p_9, p_{12}\}$,
$S_6 = \{p_4, p_5, p_6\}$,
$S_7 = \{p_7, p_8, p_9\}$.

After calculating control places for siphons S_1, S_2, S_3, S_5, S_6, S_7 it has been determined that all control places satisfy case d) given by equation (13). Those siphons must then satisfy (14) (they must be initially marked) and there is no need to add any control places to the Petri net, because they cannot be emptied. After calculating control places for siphon S_4 it has been determined that control place for this particular siphon satisfies (10). For this critical minimal siphon a control place will prevent emptying of the siphon, and must be added to the Petri net in Figure 5.

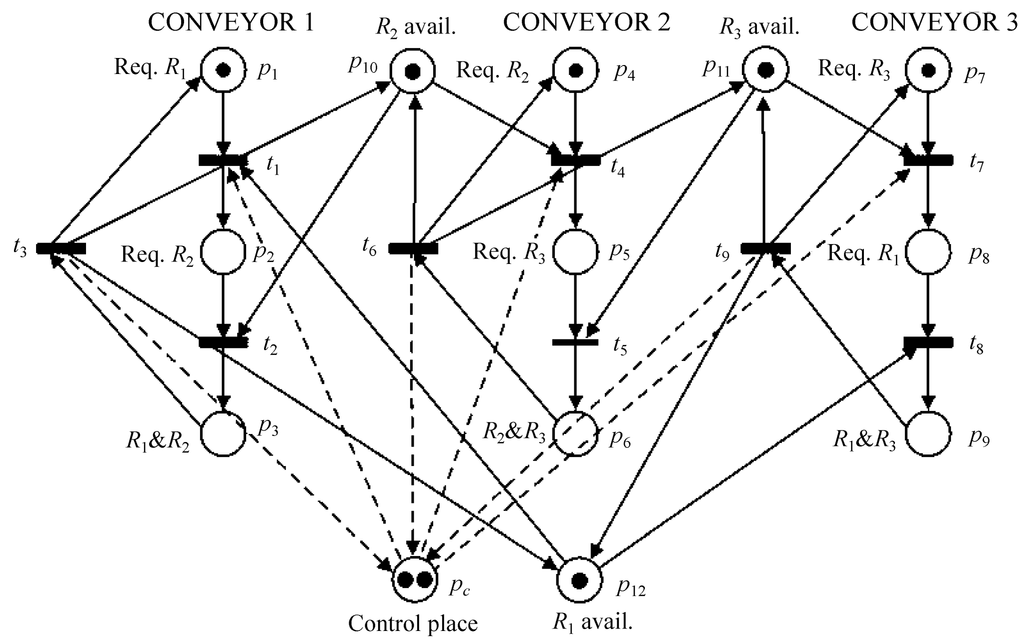

Fig. 6 Deadlock-free Petri net with control place

The condition that needs to be fulfilled is that the sum of tokens in this siphon is greater or equal to 1 for all reachable markings (4). The constraint is:

$$\mathbf{m}_3 + \mathbf{m}_6 + \mathbf{m}_9 + \mathbf{m}_{10} + \mathbf{m}_{11} + \mathbf{m}_{12} \geq 1. \quad (15)$$

Synthesis of control place p_c[1)] that fulfills the given condition could be done by the method which was explained in the section 3. Figure 6 shows a deadlock-free Petri net with control place p_c.

After adding control place p_c, algorithm for generating the deadlock-free Petri net detects that there are no more deadlock states, and the algorithm ends.

5 CASE STUDY II: DEADLOCK PREVENTION SUPERVISOR IN A MARINE TRAFFIC SYSTEM

In this section, the model of a canal traffic system (Figure 7) is made using a Petri net. The Petri net (Figure 8) describes the vessel traffic in the system. After analyzing the Petri net and applying the algorithm for deadlock prevention described in section 3, it is possible to detect deadlock states and design a deadlock prevention supervisor.

The considered canal traffic system consists of the three canals (P_1, P_2 and P_3) of 1000 m length each, and four basins (S_1, S_2, S_3 and S_4) of 100 m length each [9]. The vessels on the left end of the canal system A_i wait for the passage to the right in the *direction A*, and vessels B_i wait for the passage to the left in the *direction B*. A vessel can move through the canal system by its own propulsion or by hauling vehicles at the sides of the canal, which can tow the vessel. The vessel's speed is always greater in a canal than in a basin.

The canals allow the vessels to move in one direction only, and basins allow the passing of vessels in opposite directions. The capacity of a canal and of a basin is 1 vessel. Therefore, the vessels cannot enter a canal or a basin that is occupied (in which there is another vessel), but they have to wait until the canal or basin is empty. The vessels can wait in canals or in basins only if there is a danger of deadlock.

All basins and canals represent resources of canal and basin system. If a particular resource is occupied, and there are vessels waiting to use them, then these vessels must wait for the availability of the occupied resource at the exit of the resource in which they are. When the resource becomes available, it is occupied by awaiting vessels at random. The maximum number of vessels which can be in the canal system is 7 (Figure 7). However, due to the reasons of safety, this number is limited to 6 vessels.

The traffic in the canal system can be fully automated using the computer system which controls traffic lights at the entrance into each resource in directions A and B. Sensors for detection of ves-

1) In the section 3 the sign for control place is p_c instead of c, because the coveyor is alredy signed by C.

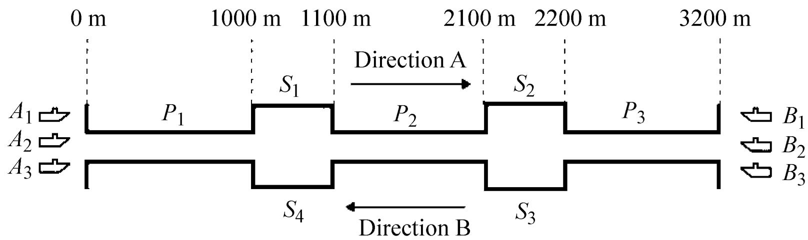

Fig. 7 System of canals and basins

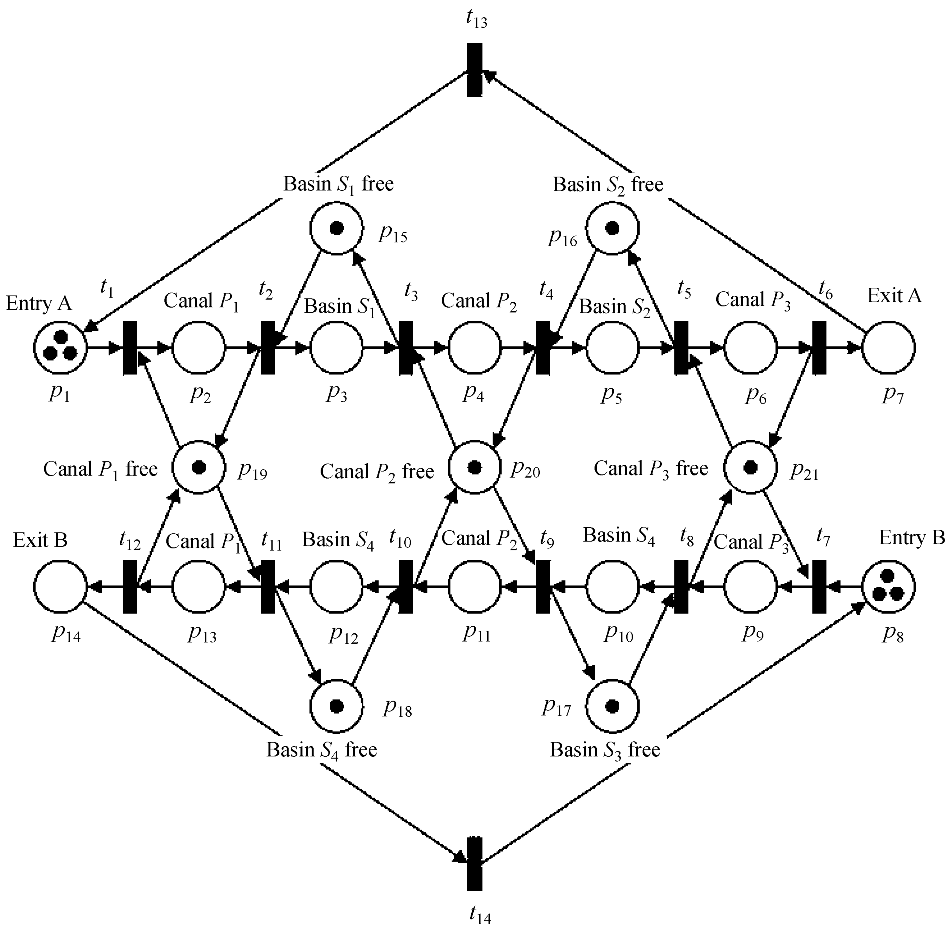

Fig. 8 Petri net of the system of canals and basins

sels' passing from one resource to the other are connected to the computer control system. The traffic signalization control system should not let vessels enter a resource as soon as it becomes available, because it can cause a forbidden deadlock state. In order to avoid such states, the traffic control system is controlled by a supervisor, which has the function of deadlock prevention. The supervisor is required to be the maximal permissible i.e. not hindering the passage of the vessels, but only in the event of immediate danger of a deadlock.

The Petri net in Figure 8 shows the model of traffic system in Figure 7, and it comprises the places p_1–p_7 which represent discrete states in the process of the vessels moving in *direction A*, and places p_8–p_{14} for moving in *direction B*. The marking of places p_1–p_{14} denotes the number of vessels in the resource represented by that place. Firing of transitions t_1–t_6 causes the movement of to kens in places p_1–p_7, which corresponds to the movement of vessels in *direction A*, and firing of transitions t_7–t_7 causes the movement of tokens in places p_8–p_{14} (movement of vessels in *direction B*). Places p_{15}–p_{21} denote the availability of particular resources in the canal system. Initial markings of the Petri net in Figure 8 are

$$m_0(p) = [3,0,0,0,0,0,0,3,0,0,0,0,0,0,1,1,1,1,1,1,1]^T.$$

Places p_1 and p_8 have 3 tokens which denote the vessels waiting to get into the canal system (the maximum number of vessels that can be in the system is 6). In this initial marking all the resources are available and so the places p_{15}–p_{21} are marked.

Procedure for calculating a deadlock prevention supervisor begins with detection of deadlock states of the Petri net (Figure 8), which is done in the way of calculating the incidence matrix **A** and analyzing reachability graph [8].

In the Petri net in Figure 8 there are two deadlock states:

$$\mathbf{dm}_1 = [0,0,1,1,1,0,0,1,1,1,0,0,0,0,0,0,0,1,1,0,0]^T,$$

and

$$\mathbf{dm}_2 = [1,1,1,0,0,0,0,0,0,1,1,1,0,0,0,1,0,0,0,0,1]^T.$$

The Petri net is P-T net without source places, and we can apply the algorithm for finding critical minimal siphons. There is only one critical minimal siphon: $S_1 = \{p_4, p_{13}, p_{15}, p_{18}, p_{19}, p_{20}\}$, for which we can add the control place $c_1(p_{22})$. The control place puts a constraint to the set of reachable markings of the net in Figure 8. After adding the control place c_1 the deadlock state is checked again. The new deadlock state is:

$$\mathbf{dm}_3 = [0,0,1,1,1,0,0,1,1,1,0,0,0,0,0,0,0,1,1,0,1,1]^T.$$

It is obvious that the control place c_1 creates new siphons in the net, so it is necessary to find new critical minimal siphons and new control places once again. After the second and third iteration of the deadlock prevention algorithm, new siphons are found:

$$S_2 = \{p_6, p_{11}, p_{16}, p_{17}, p_{20}, p_{21}\}$$

and

$$S_3 = \{p_4, p_5, p_{11}, p_{12}, p_{22}, p_{23}\}.$$

So, new control places $c_2(p_{23})$ and $c_3(p_{24})$ need to be added.

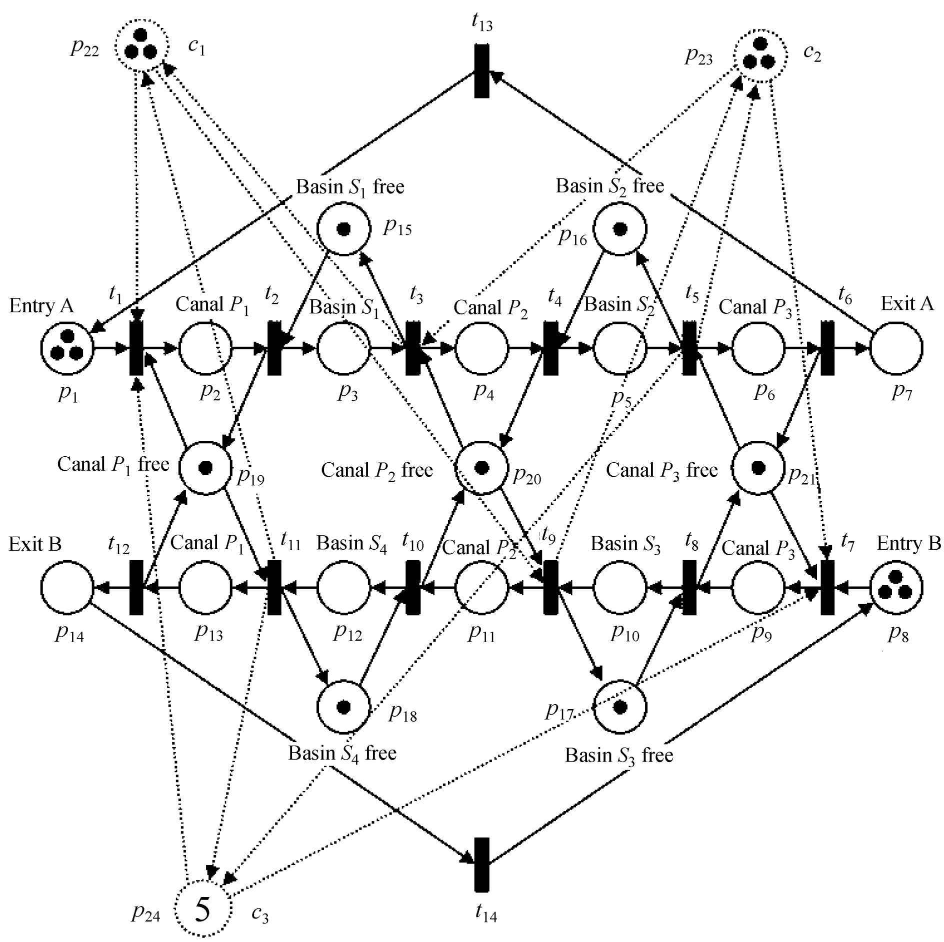

Fig. 9 Deadlock-free Petri net with deadlock prevention supervisor

The calculated supervisor comprises three control places $c_1(p_{22})$, $c_2(p_{23})$, $c_3(p_{24})$. The supervisor together with the Petri net in Figure 8 forms a new composite Petri net (Figure 9). The deadlock state for the net in Figure 9 has been checked again, and it can be concluded that there are no new deadlock states and no redundant control places, and the calculation for deadlock prevention supervisor is completed. The Petri net in Figure 9 is deadlock-free. By using the equation (9) we can get initial marking of control places $m_0(c_1)=3$, $m_0(c_2)=3$ and $m_0(c_3)=5$.

The developed deadlock prevention supervisor for the analyzed marine traffic system is verified by simulation in Matlab/Simulink environment. Three vessels in direction A and three vessels in direction B are trying to pass through the canal and basins system. The speed of the vessels in canals in direction A is 2 m/s, and in direction B 1 m/s, and the speed of the vessels in basins for both directions is 0.5 m/s. The simulation without the deadlock prevention supervisor shows that only one vessel in direction A can pass through the canals and traffic systems, and the others vessels are deadlocked in

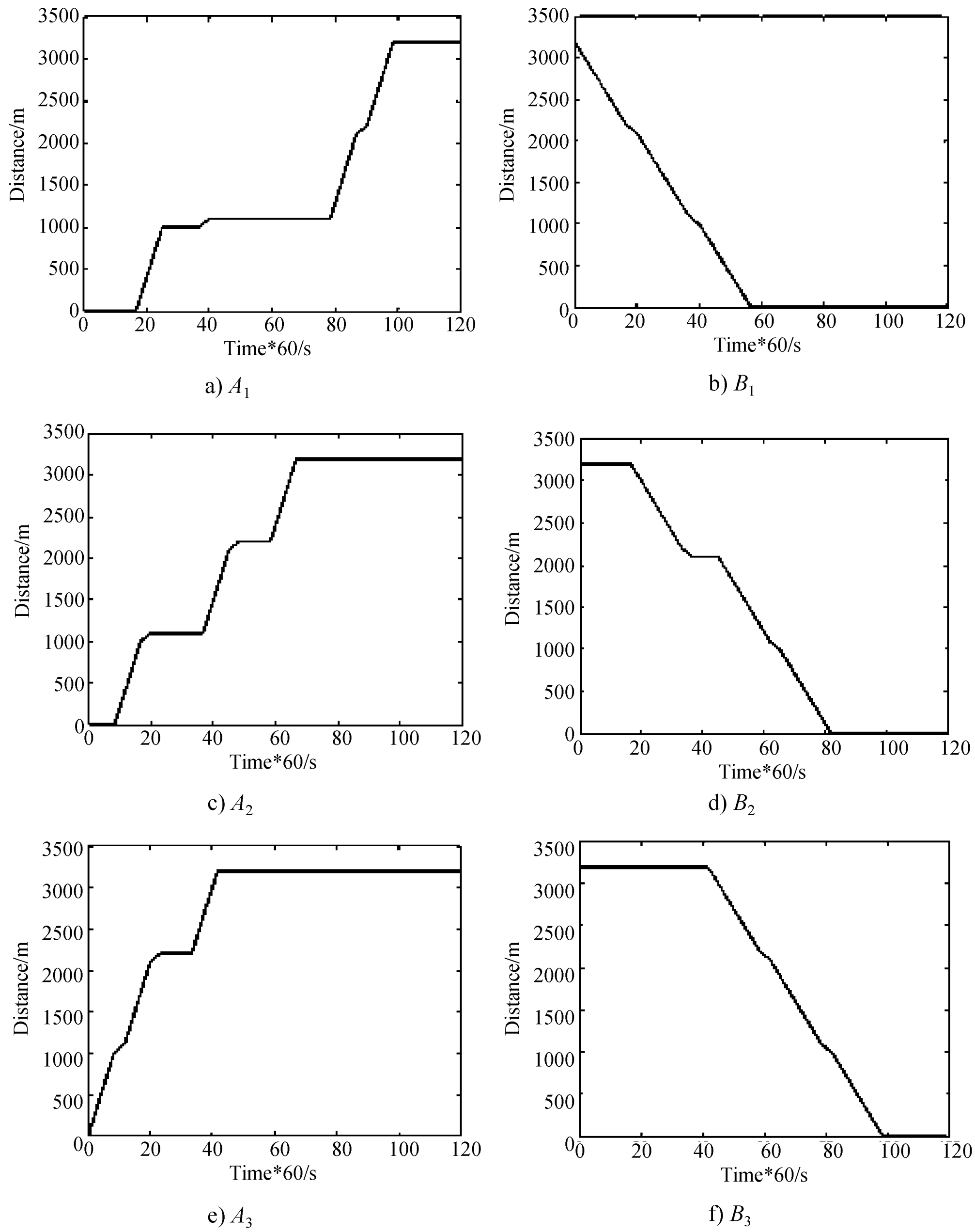

Fig. 10 The vessels movements through the canal system

canals P_1, P_2, and basins S_1, S_2, S_4. Two vessels in direction A are deadlocked in P_1 and S_4, and three vessels in direction B are deadlocked in S_1, P_2 and S_2. Each vessel is waiting for another, so the circular wait occurs, and the vessels must wait for ever.

But, if the deadlock prevention supervisor is applied, which controls the vessels traffic by traffic lights; it is possible to avoid the deadlocks. Figures 10 a)–f) show the vessels movement when the deadlock prevention supervisor is added to the traffic system. In that case, only the vessel B_1 is passing through the canals and traffic systems without waiting. The other vessels must wait a couple of minutes in the canals and basins, but in the end, with some delay, all of them pass through the canals and traffic systems.

6 CONCLUSION

A new method is proposed for calculating the maximal permissible deadlock prevention supervisor in Discrete Event Systems (DES), but it is also applicable to the continuous-variable dynamic systems that can be viewed as DES at higher level of abstraction. The Petri nets are used for creating the models of the systems, which enable analysis of possible deadlock states as well as the deadlock prevention control places calculation. These control places are added to the Petri net ensuring that the critical minimal siphons in the net, which would cause the deadlock states, are never emptied. The proposed algorithm generates as few control places as possible because it does not search for all uncontrolled minimal siphons in the Petri net, and adds control places until all minimal siphons in the net are controlled, because uncontrolled minimal siphons are not necessary condition for deadlock. The algorithm searches for deadlock states in the Petri net every time after adding another control place, and terminate in case of deadlock-free Petri net. The method is suitable for smaller Petri nets, as for the larger Petri nets the procedure of finding critical minimal siphons can be a time consuming process. In that case the large Petri net must be reduced to the smaller one. Finally, we illustrate our approach for calculating the deadlock prevention supervisor in two systems: in a manufacturing system and in a marine traffic system. A deadlock in the manufacturing system can occur during the production as the consequence of the improper sharing of two or more resources between two or more operations. In marine traffic systems a deadlock of the vessels is possible in case of inadequate use of canals and basins. Deadlock prevention supervisors must disallow all dangerous events in the both systems that can cause deadlock. The deadlock prevention supervisor of a marine traffic system is verified using computer simulation. Based on the final deadlock-free Petri net, the synthesis of a real supervisor could be easily done.

7 REFERENCES

[1] K. Barkaoui, J. Abdallah, **Deadlock Avoidance in FMS Based on Structural Theory of Petri Nets.** IEEE. Symposium on Emerging Technologies and Factory Automation, pp. 499–510, 1995.

[2] J. Ezpelta, J. Colom, J. Martinez, **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.

[3] M. Iordache, J. Moody, P. Antsaklis, **Automated Synthesis of Deadlock Prevention Supervisor Using Petri Nets.** Technical report of the ISIS Group at the University of Noter Dame, ISIS-2000-003, May 2000.

[4] L. Lautenbach, H. Ridder, **The Linear Algebra of Deadlock Avoidance – A Petri Net Approach.** Resarch Report of Institute for Computer Science, University of Koblenz, Germany, 1996.

[5] F. Lewis, H. Huang, D. Tacconi, A. Gurel, O. Pastravanu, **Analysis of Deadlocks and Circular Waits Using a Matrix Model for Discrete Event Systems.** Automatica, vol. 34, no. 9, pp. 0–19, 1998.

[6] T. Murata, **Petri Nets: Properties, Analysis and Applications.** Proc. of the IEEE, vol. 77, no. 3, pp. 541–580, 1989.

[7] D. Kezić, N. Perić, I. Petrović, **Deadlock Prevention in Manufacturing Systems Based on Petri Nets.** Proc. of 11th International Power Electronics and Motion Control Conference, vol 4, pp. 320–327, Riga, Latvia, 2–4, Sept. 2004.

[8] D. Kezić, **Petri nets Based Deadlock Prevention in Discrete Event Systems.** Doctoral thesis, Faculty of Electrical Engineering and Computing, University of Zagreb, 2004.

[9] D. Kezić, N. Perić, I. Petrović, **A Petri Net Approach of Deadlock Prevention in Marine Traffic System.** Proc. of the IEEE International Symposium on Industrial Electronics 2005, vol. 1, pp. 217–222, Dubrovnik, Croatia, 20–23 June 2005.

Algoritam za sprječavanje zastoja temeljen na uzastopnoj kontroli sifona Petrijeve mreže. Članak opisuje formalnu metodu proračuna nadzornika za sprječavanje zastoja korištenjem Petrijevih mreža. Predloženi algoritam koristi stablo dostupnih stanja za detekciju stanja zastoja i metodu uzastopne kontrole sifona za sintezu nadzornika za sprječavanje zastoja. Nadzornik je najviše dopuštajući i sadrži najmanji broj kontrolnih mjesta. Algoritam je namijenjen za reverzibilne ili djelomično reverzibilne P-T Petrijeve mreže, ali se može koristiti i za obične Petrijeve mreže. Proračun nadzornika pokazan je na dva primjera. Prvi primjer prikazuje sintezu nadzornika za sprječavanje zastoja u fleksibilnom proizvodnom sustavu s tri robota i tri proizvodne trake, gdje se zastoj može dogoditi zbog međusobnog natjecanja transportnih traka za angažiranjem robota te zbog nepredvidljivosti trajanja tih angažmana. Drugi primjer prikazuje sintezu nadzornika u pomorskom prometnom sustavu, gdje se opasne situacije zastoja plovila mogu dogoditi poradi neodgovarajućeg pomicanja plovila kroz sustav. Da bi se to izbjeglo, promet plovila se nadzire i upravlja pomoću svjetlosne signalizacije korištenjem nadzornika za sprječavanje zastoja, koji je odgovoran za zaustavljanje plovila samo u slučaju opasnog stanja te dok to stanje ne nestane.

Ključne riječi: nadzornik za sprječavanje zastoja, Petrijeva mreža, kontrola sifona

AUTHORS' ADDRESSES

Doc. dr. Danko Kezić
Faculty of Maritime Studies, Dept. of El. Eng.
University of Split, Zrinsko-Frankopanska 38, 21000 Split, CROATIA
E-mail: danko.kezic@pfst.hr WWW: http://www.pfst.hr

Prof. dr. Nedjeljko Perić, Prof. dr. Ivan Petrović
Faculty of Electrical Engineering and Computing
University of Zagreb, Unska 3, 10000 Zagreb, CROATIA
E-mail: nedjeljko.peric@fer.hr WWW: http://www.fer.hr/apr
ivan.petrovic@fer.hr WWW: http://www.fer.hr/apr

Received: 2006-03-30