Safety Discrete Event Models for Production Lines with Shared Resources

C. Ciufudean*, A.B. Larionescu*, A. Graur*, C. Filote*, C. Petrescu**
* "Stefan cel Mare" University of Suceava, 1 Universitatii Str., Suceava, Romania
** Electrical Engineering Co., 313 Splaiul Unirii Bvd., Bucharest, Romania

Abstract—**In this paper we study the problem of deadlock avoidance of production lines with shared resources. We use the Petri net model for production lines and a restrictive policy which prevents some enabled transition from firing for avoiding deadlock in the system. This is the safety control for non-stochastic discrete event systems. We generalise this notion of safety to the setting of stochastic discrete event systems, modelled as Markov chains. We propose a restriction policy for this generalised case.**

I. INTRODUCTION

Given a model of a production line and a specification of the desired behaviour for the controlled system, the main objective to achieve a certain throughput is to synthesise the appropriate controller to realise the specified behaviour. In general production lines, raw products of various types enter the system at discrete point of times and are processed concurrently. Usually, these systems shares a limited number of resources such as machines, robots and buffers, and each product has a particular operation routing that determines the order in which resources must be assigned to the product. The concurrent flow of multiple products in a system, which all competes for a finite set of resources, can lead to a deadlock situation. In such resource-shared systems, deadlocks constitute a major issue to be addressed at the design and operation phases. Many efforts have been focused on the problem of deadlock in a Flexible Manufacturing System (FMS) [1], [2], [3]. Some of them adopted Petri net (PN) models as a formalism to describe FMS's and to develop deadlock avoidance policies. Banaszak and Krough [4] considered a class of Petri net model and proposed the deadlock avoidance algorithm (DAA) for FMS with concurrently competing process flows. Minoura and Ding [2] developed a method based on the untimed PN formalism to synthesise deadlock avoidance controllers that keep an FMS live and may achieve a high resource utilisation under a given dispatching policy. Ezpeleta et al. [3] solved the deadlock avoidance problem using the concept of siphons. Our paper studies the important issues of deadlock avoidance in production lines using PN and Markov chains based techniques. First we present [4] the conditions in which the system is in a deadlock situation. Then we present a restriction policy to avoid the deadlocks. In some cases, despite the fact that this policy is minimally restrictive, a deadlock may occur in an extreme case, when the resources involved in a concurrent operation are simultaneously out of order. This case is studied with a Markov chain because the occurrence of breakdowns in system represents a stochastic process. An example will illustrate our approach.

II. PN MODELS OF CONCURRENT OPERATIONS

Let us consider a production line with m types of resources, denoted by $r_1, r_2, \ldots, r_m$ and n different types of products, denoted by $q_1, q_2, \ldots, q_n$.

The manufacturing process of each product is supposed to be defined as a sequence of resource utilisation. We denote by $u(q_i)$ the sequence corresponding to product type q_i. First, we adopt a PN model for the production line which is similar to [4]. There are four workstations W_1, W_2, W_3 and W_4 served by a transport system S and two types of products q_1, q_2. Each workstation has an input buffer to hold products to be processed and an output buffer to hold products for which the process step has been completed. Each buffer has a capacity of five. Suppose that there is a single transport resource in S and a single machine M_i in workstation W_i, $i = 1,\ldots, 4$; then the set of resources is $R = \{I_1, \ldots, I_4, O_1, \ldots, O_4\}$.

Labels $I_1,\ldots, I_4$ denote the input buffers, and $O_1, \ldots, O_4$ denote the output buffers for W_i. Suppose q_1, q_2 are processed, respectively, by sequencing machines M_3, M_1, M_2, M_1 and M_2, M_3, M_4 in order. The process sequences are specified as follows: $u(q_1) = (I_3, O_3, I_1, O_1, I_2, O_2, I_1, O_1)$, and $u(q_2) = (I_2, O_2, I_3, O_3, I_4, O_4)$. Transition firing corresponds to one process step completing and the next process step starting. The tokens in place p^i_0 represent the number of products of type q_i to be initiated and tokens in places $p^i_{L_i+1}$ represent the number of completed products of q_i.

L_i is the length of the sequence for product q_i. Places $p^i_1, p^i_2, p^i_{L_i}$ correspond to process steps, and each step requires only one resource.

A resource place is assigned to each type of resource r, denoted by a_r, tokens in place a_r indicate available resources of type r, the initial marking is defined as $m_0(a_r)=C_r$, where C_r is the available capacity of type r. The process step p^i_j using resource r, denoted by $R(p^i_j)$=r, is represented by an arc from a_r to t^i_j and an arc from t^i_{j+1} to a_r, $i = 1, \ldots, n, j = 1, \ldots, L_i$.

Fig. 1 shows the model for our production line.

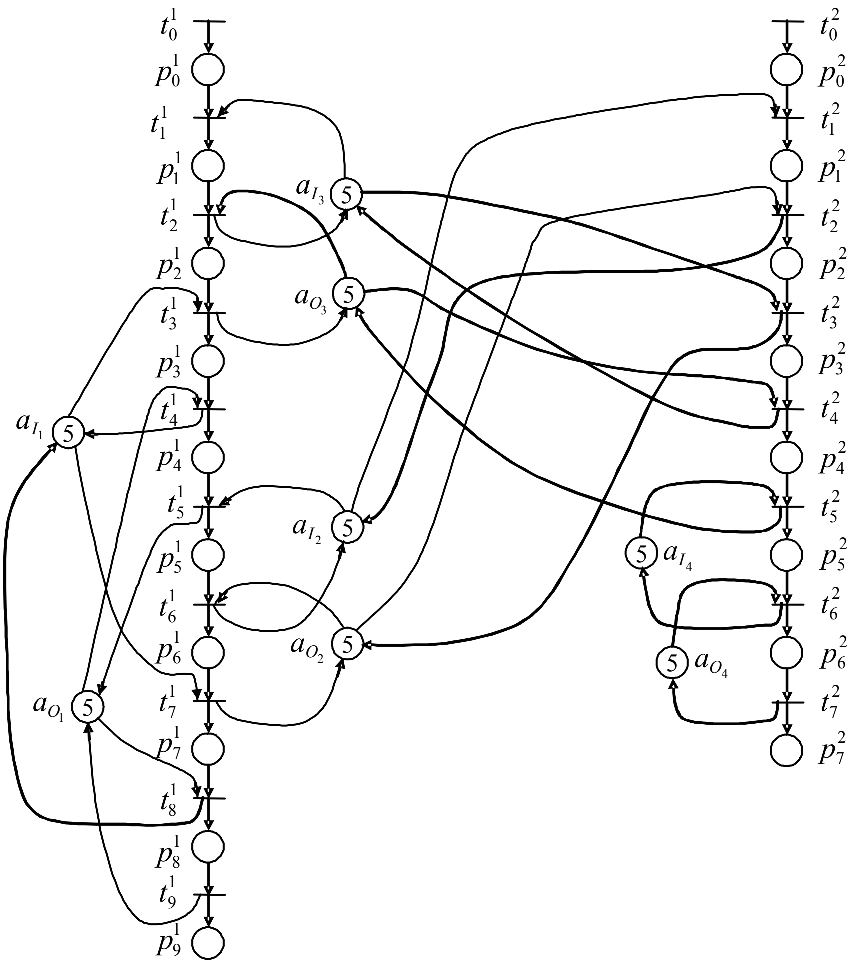

Figure 1. PN model of a production line

The complete PN model for a production line is described by:

$$P = P_p \cup P_r \tag{1}$$

where:

$$P_p = \{p^i_j,\ i = 1, 2, \ldots, n,\ j = 0, 1, \ldots, L_i+1\} \tag{2}$$

$$P_r = \{a_{r_i},\ i = 1, 2, \ldots, m\} \tag{3}$$

$$T = \{\ t^i_j,\ i = 1, 2, \ldots, n, j = 0, 1, \ldots\ L_i+1\} \tag{4}$$

$$I = \{(p^i_j, t^i_{j+1}),\ i = 1, 2, \ldots, n,\ j = 0, 1, \ldots, L_i\} \cup$$
$$\cup\ \{(\ a_{R(p^i_j)}, t^i_j\ ),\ i = 1, 2, \ldots, n,\ j = 1, \ldots, L_i\} \tag{5}$$

$$O = \{(\ t^i_j, p^i_j\ ),\ i = 1, 2, \ldots, n,\ j = 0, 1, \ldots, L_{i+1}\} \cup$$
$$\cup\ \{(t^i_{j+1}, a_{R(p^i_j)}),\ i = 1, 2, \ldots, n,\ j = 1, \ldots, L_i\} \tag{6}$$

$m_0(p) = C_r$, for $p_r \in \{\ a_{r_i},\ i = 1, 2, \ldots, n\}$, otherwise $m_0(p) = 0$.

For our production line $i = 1, 2; j = 1, 2, \ldots, 9; C_r = 20$.

When one or more sorts of products enter a production line, it is possible to reach a deadlock situation. For example, if the marking m of the PN model in Fig. 1 is defined by:

$$m(p) = \begin{cases} 5, \text{ if } p \in \left\{p_3^1, p_4^1, p_5^1, p_6^1, a_{I_3}, a_{I_4}, a_{O_3}, a_{O_4}\right\} \\ \\ 0, \text{ otherwise} \end{cases} \tag{7}$$

then transitions t^1_4, t^1_5, t^1_6 and t^1_7 can not fire and are in deadlock. We consider for deadlock the next definition [4]:

Given sets of resources R and products Q and a PN, G, a set of transitions D is said to be in deadlock for a marking $m \in RM(G, m_0)$ if:

1) all transitions in D are process enabled under the marking m, and

2) no transition in D is resource enabled for any marking $m' \in RM(G, m)$.

We note that from definition of deadlock for a PN G, if transition sets D_1 and D_2 are in deadlock for a marking $m \in RM(G, M_0)$, then $D_1 \cup D_2$ is in deadlock for the marking m.

In [5] is proved the following minimally restrictive policy: If a PN, G, contains any deadlock structure, then it will reach some marking m for its initial marking m_0 such that $D(m)\neq\varnothing$. So, to avoid deadlock, it is necessary to restrict the number of tokens in every deadlock structure, D, meaning that the number of tokens in ${}^0D$ must be not greater than $\sum_{t\in R(D)} C_r - 1$, where ${}^0D$ denotes the union set of ${}^0t$ for all $t\in D$, i.e., ${}^0D=\{{}^0t \,/\, t\in D\}$.

For a given transition $t\in T$, we let ${}^0t$ denote the process place in P_p, which are input places for t. A transition $t\in T$ is process enabled if $m({}^0t) \geq 1$. For the given example in Fig. 1, we need to construct the restriction policy by restricting the number of tokens in ${}^0D$ for each deadlock structure D of PN within $\sum_{r\in R(D)} C_r - 1$. The deadlock structures of PN from the left side in Fig. 1 are:

$$D_1 = \{t_4^1, t_5^1, t_6^1, t_7^1\}, \tag{8}$$

$$D_2 = \{t_5^1, t_6^1, t_7^1, t_8^1\}, \tag{9}$$

$$D_3 = D_1\cup D_2 = \{t_4^1, t_5^1, t_6^1, t_7^1, t_8^1\}. \tag{10}$$

Since $R(D_1)=R(D_2)=R(D_3)=\{I_1, O_1, I_2, O_2\}$ and $D_1\subseteq D_3$, $D_2\subseteq D_3$, the restriction policy for deadlock avoidance only needs to restrict the number of tokens in the places of ${}^0D_3$ within:

$$\sum_{r\in R(D_3)} C_r - 1 = C_{I_1} + C_{O_1} + C_{I_2} + C_{O_2} - 1 =$$
$$= 5 + 5 + 5 + 5 - 1 = 19. \tag{11}$$

For example, we consider a marking $m\in RM(G, m_0)$ defined by:

$$m(p) = \begin{cases} 5 \text{ if } p\in \{p_3, p_4, p_5, p_6, a_{I_3}, a_{O_3}\} \\ 0, \text{ otherwise} \end{cases} \tag{12}$$

where transitions t_4, t_5, t_6, t_7 are in deadlock. $D_1=\{t_4, t_5, t_6, t_7\}$ is a deadlock structure and $D_1 \subset D_3$. The marking m is not reachable under the control, because there are allowed at most 19 tokens in the places of ${}^0D_3 = \{p_3, p_4, p_5, p_6, p_7\}$. This number, i.e. 19, is the largest number of tokens which can enter in places of ${}^0D_3$ and no deadlock will occur.

The PN realisation with the restriction policy is:

$$X = X_3 = (P_X, \{t_3, t_8\}, I_X, O_X, m_{X_0}) \tag{13}$$

as shown in Fig. 2.

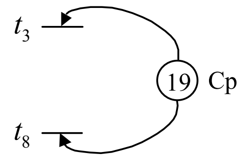

Figure 2. An example of PN restrictive deadlock avoidance policy

For the entire PN in Fig. 1, using the same principle we have the next deadlock structures, and unions of them:

$$D_0 = \{t_4^1, t_5^1, t_6^1, t_7^1\}, \tag{14}$$

$$D_1 = \{t_5^1, t_6^1, t_7^1, t_8^1\}, \tag{15}$$

$$D_2 = \{t_4^1, t_5^1, t_6^1, t_7^1, t_2^2\}, \tag{16}$$

$$D_3 = \{t_5^1, t_6^1, t_7^1, t_8^1, t_2^2\}, \tag{17}$$

$$D_4 = \{t_2^1, t_3^1, t_4^1, t_5^1, t_2^2, t_3^2\}, \tag{18}$$

$$D_5 = \{t_3^1, t_4^1, t_5^1, t_6^1, t_3^2, t_4^2\}. \tag{19}$$

The PN restrictive avoidance policy for the PN in Fig.1 is given in Fig. 3.

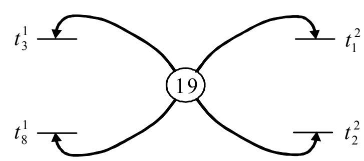

Figure 3. The PN restrictive deadlock policy for the net in Fig. 1

III. CONTROLLERS FOR STOCHASTIC SYSTEMS

The previous paragraph presented an efficient method to avoid the deadlocks when the number of any key kind resources is greater than one [4], [5]. But, what is to do when the number of key resources is zero? In this case, the deadlocks appear because no transition is resource enabled for any marking. This case may appear when key resources are machines with the same reliability and with the same availability. Because the appearance of breakdowns for machines is a stochastic process, we shall study it with Markov chains by associating a probability measure to each state transition. We shall study the safety of Markov chains [6] by an example. Consider a single machine which operates in either of its two states: “up” and “down”. Let the probability that the machine maintains its current state at the next step is given by p (respectively, q) if the current state is up (respectively, down). Then the state set of the machine is given by A={up, down}, and the state transition matrix is given by:

$$P_A = \begin{pmatrix} p & 1\text{-}p \\ 1\text{-}q & q \end{pmatrix} \tag{20}$$

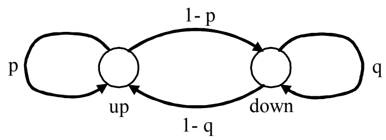

Figure 4. Markov chain model of a machine

Fig. 4 illustrates graphically the above relation, where p represents the machine function of intensity of usage and q the function of intensity of maintenance.

The entries of the state transition matrix can be controlled at any given state. Two types of control are underlined, namely, the intensity of usage, and the intensity of maintenance. In the up state, p is an increasing function of the intensity of maintenance, and a decreasing function of the intensity of usage. In the down state, q is a decreasing function of the intensity of maintenance, and it does not depend on the intensity of usage, since the machine is not used in its down state.

We consider the definition of safety of a Markov chain given in [6]:

A given Markov chain with state transition matrix $P\in[0,1]^{n \times n}$ is said to be safe with respect to m if the state probability distribution vector remains bounded above by m at all steps, i. e., for all $k\geq 0$, $\Pi_0 \cdot P^k \leq m$.

We use $\Pi_m=\{\pi\in\Pi \ / \ \pi\leq m\}$ to denote the set of all safe state probability distribution vectors. Here, $m\in[0,1]^n$ denotes a unit interval valued row vector that imposes a safety specification. For the considered example (Fig. 4) suppose it is desired that at any step the machine is never down with probability more than 25%. Then the safety specification for the machine is given by $m=[1, 1/4]$, where $m_1=1$ implies that the probability of being in the up state can be anything, and $m_2=1/4$ implies that the probability of being in the down state must not exceed $1/4$ =25%. Following the algorithm given in [6], we obtain a necessary and sufficient condition on P_A so that the state probability distribution vectors of the controlled Markov chains remains safe at all steps, i.e., when $\Pi_0\in\Pi_m$, we also have:

$$\Pi_0 \cdot P_A \in \Pi_m. \tag{21}$$

We obtain constrains p and q which satisfy the above controlled Markov chain. For the given example, assuming that $p\geq 1\text{-}q$, for a state of safety enforcing controller with state matrix:

$$P_A = \begin{pmatrix} p & 1\text{-}p \\ 1\text{-}q & q \end{pmatrix} \tag{22}$$

we have:

$$(p \geq 1\text{-}q) \cap (3p\text{-}q \geq 2). \tag{23}$$

The above relations are true for the set of safe state probability distribution vector:

$$\Pi_m = \{ \pi \leq [1,\ 1/4]\} \tag{24}$$

In [7], [8], we built an algorithm to determine the availability of a FMS modeled with a Markov chain with recovery. That model shows that the availability of a system can decrease to zero if the repair factor of resources is less than the breakdown factor of resources. So, if all the key resources have the same reliability, or, else said, they have the same state safety-enforcing controller, then a deadlock situation occurs in the system.

In this situation the PN model has no transition resource enabled, as we discussed earlier.

In this paragraph we show that there is always a safe state controller for a resource system. In order to avoid the occurrence of deadlock situations in a multi-resource system, we propose the following policy: we admit the PN model discussed in the second paragraph for which we propose the restrictive relation:

$$1 \leq \text{Number of tokens in } {}^0D \text{ for any deadlock structure} \leq \sum_{r\in R(D)} C_r - 1 \tag{25}$$

where the involved terms have the signification given in the second paragraph.

The above relation involves that the key resources must not have the same availability (respectively, they must not have the same state safety-enforcing controller).

This goal can be achieved either when we use different types of key resources, or, when it is necessary to use the same type of key resources, these must not have the same service life. That means, they must be utilised starting from different moments of time, and at different intensity of performance requirements.

IV. ILLUSTRATIVE EXAMPLE: A MARKOV MODEL FOR EVALUATING THE AVAILABILITY OF FMS

For the flexible manufacturing system depicted in Fig.5, we assume that the machines are failure-prone, while the load/unload station and the conveyor are extremely reliable.

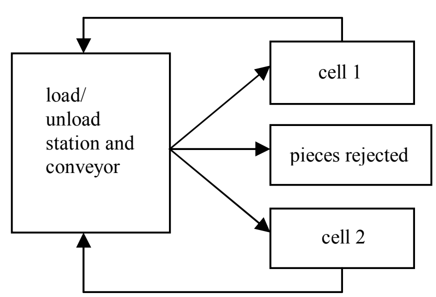

Figure 5. Logical model for a flexible manufacturing system

Assuming the failure times and the repair times to be exponentially distributed, we can formulate the state process as a continuous time Markov chain (CTMC). The state process is given by $\{X(u), u \geq 0\}$ with state space $S=\{(ij), i\in\{0,1,2\}, j\in\{0,1\}\}$, where i denotes the number of machine working, and j denotes the status of the material handling system (load station and conveyor): up "1", and down "0". We consider the state time independent (or time dependent) failure case and the operation dependent failure case separately.

A. Time dependent failures

In this case, the component fails irrespective of whether the system is operational or not. All failure states are recoverable. Let r_a and r_m denote the repair rates of the material handling system (MHS) and a machine, respectively. The state process is shown in Fig. 6a.

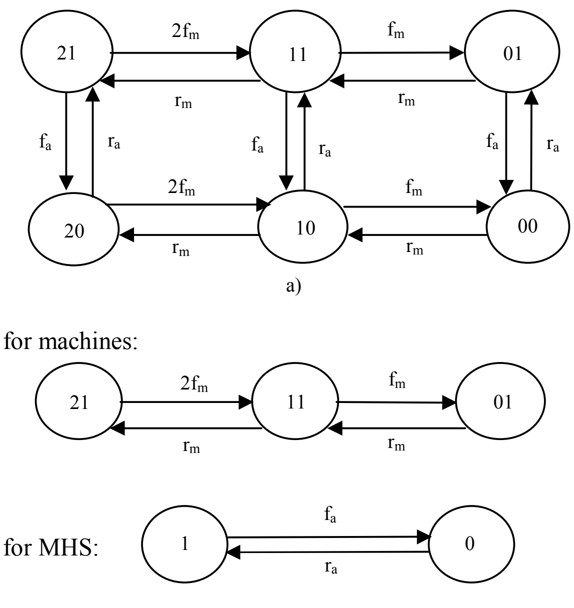

Figure 6. State process of a FMS with time-dependent failures: (a) State process for a state-independent failure model, (b) Decomposed failure/repair process

Because the failure/repair behaviour of the system components is independent, the state process can be decomposed into two CTMCs as shown in Fig. 6b. Analytically, the state process is expressed by relations: $S_0 = \{(21), (11)\}$ and $S_F = \{(20), (10), (00)\}$. For each state in S_F no production is possible since the MHS or both the machines are down. In Fig. 6b the failure/repair behaviour of each resource type (machines or MHS) is described by a unique Markov chain. Thus, the transient state probabilities, $p_{ij}(t)$, can be obtained from relation:

$$p_{ij}(t) = p_i(t)\cdot p_j(t) \tag{26}$$

where $p_i(t)$ is the probability that i machines are working at time t for $i = 0,1,2$.

The probability $p_i(t)$ is obtained by solving (separately) the failure/repair model of the machines. We note with $p_j(t)$ the probability that j MHS (load/unload station and conveyor) are working at instant t, for $j = 0,1$. Let f_a and f_m denote the failure rates of the MHS and of a machine respectively.

B. Operation dependent failures

Assume that when the system is functional, the resources are all fully utilized. Since failures occur only when the system is operational, the state space is: $S = \{(21), (11), (20), (10), (01)\}$, with $S_0 = \{(21), (11)\}$, $S_F = \{(20), (10), (01)\}$. The Markov chain model is shown in Fig. 7. Transitions representing failure will be allowed only when the resource is busy. Transitions rates can however be computed as the product of the failure rates and percentage utilization of the resource. If T_k^{ij} represents the average utilization of the k^{th} resource in the state (ij), the transition rates are given in Fig. 7.

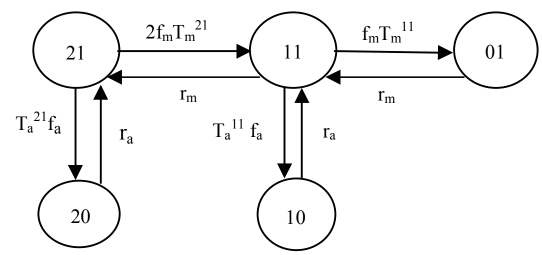

Figure 7. State process of a FMS with state-dependent failures

C. Numerical example

For the FMS presented in this paper, in the Table 1 are given the failure/repair data of the system components. We note with T_k^{ij} the average utilization of the system of the k^{th} resource in state (ij); $T_k^{ij} = 1$ since the utilization in each operational state is 100% for all $i=\{0,1,2\}$, $j=\{0,1\}$, $k=4$. The other notations used in table 1 are: f is the exponential failure rate of resources, r is the exponential repair rate of resources, N_p is the required minimum number of operational machines in cell p, $p = \{1,2\}$, and n_p is the total number of machines in cell p.

TABLE 1
DATA FOR THE NUMERICAL STUDY

	R	F	N_p	n_p	T_k^{ij}
Machines	1	0,05	1	2	1
MHS	0,2	0,001	1	1	1

From Fig. 6 and Fig. 7 we calculate the corresponding infinitesimal generators and after that, the probability vector of CTMC. With relation (1) we calculate the availability of FMS given in this article. The computational results are summarized in Table 2 for the state process given in Fig. 2 (FMS with time-dependent failures), and respectively in Table 3 for the state process given in Fig. 7 (FMS with state-dependent failures). We consider the system operation over an interval of 24 hours (three consecutive shifts).

TABLE 2
COMPUTATIONAL RESULTS FOR THE FMS IN FIG. 6

Time hour	Machines	MHS	System Availability
0	1.0000	1.0000	1.0000
1	0.9800	0.9548	0.9217
4	0.9470	0.8645	0.7789
8	0.9335	0.8061	0.7025
12	0.9330	0.7810	0.6758
16	0.9331	0.7701	0.6655
20	0.9330	0.7654	0.6623
24	0.9328	0.7648	0.6617

TABLE 3
COMPUTATIONAL RESULTS FOR THE FMS IN FIG. 7

Time hour	Machines	MHS	System Availability
0	1.0000	1.0000	1.0000
1	0.9580	0.9228	0.9001
4	0.9350	0.8228	0.7362
8	0.9315	0.8039	0.7008
12	0.9310	0.7798	0.6739
16	0.9320	0.7688	0.6632
20	0.9318	0.7639	0.6598
24	0.9320	0.7636	0.6583

The results of the availability analysis of the flexible manufacturing system are illustrated in Fig. 8, which depicts the availability of the system as a function of the time. The numbers x = 2, 3 indicate the system in Fig. 6, respectively Fig. 7. One can see from Fig. 8 that the layout with FMS with time-dependent failures is superior to that with FMS with state-dependent failures.

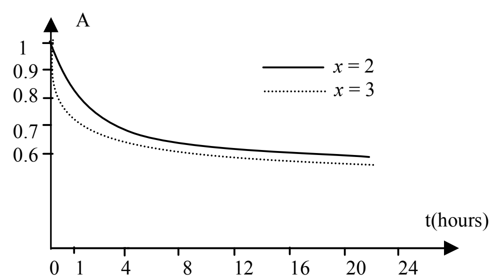

Figure 8. Availability analysis of the flexible manufacturing system given in Fig. 5

V. CONCLUSIONS

Maintenance philosophy necessitates the development of new operation maintenance decision-making tools.

Increasing performance requirements will result in increasing stresses applied to structural components that lead to greater damage accumulation in the components and shorter service life. Therefore, performance reliability obviates the need for optimal control subject to reliability constraints. For healthy components, higher performance may be achieved while maintaining the requisite reliability. The same is not true for ageing components where a decision to perform maintenance may be necessary before higher performance is achievable [7].

In this paper we have studied the problem of deadlock avoidance in PN models representing production lines with shared resources. We have presented a restriction policy for avoiding deadlocks. An advantage of the restriction policy procedure is that it is based both on PN model and Markov chains model. An illustrative example using analytical technique for the availability evaluation of the flexible manufacturing systems was presented. The novelty of the approach is that the construction of large Markov chains is not required. Using a structural decomposition, the manufacturing system is divided into cells. For each cell a Markov model was derived and the probability was determined of at least Ni working machines in cell i, for $i = 1,2,..,n$ and j working material handling system at time t, where N_i and j satisfy the system production capacity requirements. The model presented in this paper can be extended to include other components, e.g., tools, control systems. The results reported here can form the basis of several enhancements, such as conducting performance studies of complex systems, with multiple part types. So, other policies may extend the proposed one.

REFERENCES

[1] N.Viswandham, Y.Narahari, T.L.Johnson, "Deadlock prevention and deadlock avoidance in FMS using Petri net models", *IEEE Trans. Robotics Automat.*, vol.6, no. 6, pp.438-456, 1990.
[2] A.Minoura, C.Ding, "A deadlock prevention method for a sequence controller for manufacturing", *Int. J. Robotics Automat*., vol. 6, no. 3, pp.234-240, 1991.
[3] J.Ezpeleta, J.M.Colom, J.Martinez, "A Petri net based deadlock prevention policy for FMS", IEEE Trans. Robotics Automat., vol.11, no. 2, pp.232-240, 1995.
[4] Z.A.Banaszak, B.H.Krogh, "Deadlock avoidance in FMS with concurrently competing process flows", *IEEE Trans. Robotics Automat*., vol. 6, no. 6, pp.452-460, 1990.
[5] K.Y.Xing, B.S.Hu, H.X.Chen, "Deadlock avoidance policy for PN modeling with shared resources", *IEEE Trans. Robotics Automat*., vol. 41, no. 2, pp.302-308, 1996.
[6] S.P.Sethi, Q.Zhon, "Hierarchical production controls in a stochastic two machine flow-shop with a finite interval buffer", *IEEE Trans. Robotics Automat.*, vol. 13, no. 1, pp.112-116 1997.
[7] C.Ciufudean, "Discrete event systems for modelling the railway traffic", Matrix Rom Publishing House, Bucharest, 2002.
[8] C.Ciufudean, A.B.Larionescu, "Estimation of the performances of the discrete event systems", *Advances in Electrical and Computer Engineering*, vol.3(10), no.2(20), pp.30-35, University of Suceava, Romania, 2003.
[9] C.Ciufudean, D.Popescu, "Modelling digital signal perturbations with stochastic Petri nets", *Advances in Electrical and Computer Engineering*, vol.4(11), no.1(21), pp.71-75, University of Suceava, Romania, 2004.