15th Triennial World Congress, Barcelona, Spain

ROBUSTNESS OF DEALOCK AVOIDANCE ALGORITHMS FOR SEQUENTIAL PROCESSES

Fu-Shiung Hsieh

Overseas Chinese Institute of Technology
Taiwan, R.O.C.

Abstract: Although deadlock avoidance issue has attracted much attention and has been extensively studied, most of the existing results assume reliable machines, which makes it difficult to apply existing deadlock avoidance algorithms to a real manufacturing system with unreliable machines. This paper presents the results to apply existing deadlock avoidance algorithms to systems with unreliable machines by analysing the robustness of the deadlock avoidance algorithms. Sequential production processes are considered in this paper and Petri Net is adopted as the tool for modelling and analysis of the sequential processes. The tolerable machine failure under which liveness property can be preserved is characterized. *Copyright © 2002 IFAC*

Keywords: deadlock, flexible manufacturing systems, algorithms, robustness.

1. INTRODUCTION

Deadlock is a highly undesirable situation in which a set of parts or jobs are requesting or waiting for resources held by other parts or jobs in the same set, with the set of parts or jobs in circular waiting. Deadlock issue has attracted much attention in the last decade in the context of manufacturing systems. However, most of the existing results (Reveliotis, 1999; Lawley, 1998a; Lawley, 1998b; Reveliotis, 1997; Cho, 1995; Ezpeleta, 1995; Hsieh, 1994; Wysk, 1991; Banaszak, 1990;Viswanadham, *et al*., 1990;) assume reliable machines, which is an inappropriate assumption for real manufacturing systems. These works focus on deadlocks issues in the context of flexible manufacturing systems. Banaszak and Krogh (Banaszak, 1990) proposed a production Petri net (PPN) to model concurrent job flow and dynamic resource allocation in FMS. In their work, a simple and low computational complexity deadlock avoidance scheme is proposed based on restriction policy based on current marking for allocating resources. Although such restriction policy is suitable for real-time control logic implementation, it may impose unnecessary constraints on resource allocation and degrades system performance. Hsieh (Hsieh, 1994) overcame the drawbacks of the above approach by formulating a deadlock avoidance controller (DAC) synthesis problem for a class of Petri net called CPPN. Ezpeleta et al. (Ezpeleta, 1995) formulated a policy that prevents deadlock by establishing the equivalence between deadlocks in a manufacturing system and unmarked siphons in a class of Petri net called S^3PR. Wysk et al. (Wysk, 1991) and Cho (Cho, 1995) proposed graph-theoretic models for deadlock detection and avoidance for manufacturing systems. Lawley (Lawley, 1998a; Lawley, 1998b) addressed the scalability, configurability and routing flexibility design issues of deadlock avoidance algorithms for manufacturing systems and proposed deadlock avoidance algorithms for manufacturing systems with flexible routing capability. However, these deadlock avoidance algorithms do not address the issues of handling uncertainties such as unreliable machines in manufacturing systems.

In manufacturing systems, uncertainties such as unreliable machines may occur stochastically. This pose problems to apply existing deadlock avoidance algorithms to manufacturing systems with machines that may fail stochastically. Unreliable machines pose challenges in design and control of production processes as machine failure may bring the system to a dead state and has negative impacts on scheduled production activities. Feasibility of tailoring existing deadlock avoidance algorithms for manufacturing systems with reliable resources to systems with unreliable resources requires further study.

An interesting question is whether existing deadlock avoidance algorithms posses desirable robust properties for handling such uncertainties. In control theory, a controller designed for a plant often exhibits some degree of robustness with respect to uncertainties. That is, as long as the unmodeled dynamics is within some safety margin, the real system stays stable using the nominal controller. A similar line of thinking motivates the research of this paper. An interesting question is whether there exists any safety margin within which a nominal deadlock avoidance algorithm still works. Is it possible to characterize the safety margin quantitatively? There is apparently a lack of research regarding evaluation of the impacts of unreliable machines on the manufacturing systems. The goal of this paper is to quantitatively characterize the tolerable margin that machines may fail without the risk to enter a deadlock. It will provide much insight on development of deadlock avoidance algorithms for manufacturing systems with unreliable machines.

To focus on the characterization of safety margin for deadlock avoidance algorithms, we consider the class of sequential processes that have been extensively studied in existing literatures. To facilitate the analysis of sequential processes, a proper mathematical model is required. Although there are many literatures on modeling and design of production shop floor controllers (Jeng, 1997; Jeng, 1993; Zhou, 1992; Zhou 1991; Zhou, 1989; Narahari, 1985), Petri Net is chosen in this paper owing to its modelling power as well as analysis capability. This paper focuses on synthesis of Petri Net controllers that can operate in the presence of unreliable machines. In addition to a nominal deadlock avoidance, robustness property of the nominal deadlock avoidance is studied. Given a state or a marking in Petri Net's terminology, sufficient liveness conditions for a perturbation of the state or marking are established. Each liveness condition represents a subset of state space within which the perturbed system can be kept live. Safety margins that guarantees liveness property of the nominal system are established.

The remainder of this paper is organized as follows. Section 2 formulates the deadlock avoidance problem for sequential production processes based on Petri Nets. Section 3 presents the concept of token flow path in Petri Nets which will be used to characterize necessary and sufficient liveness conditions for sequential processes. Section 4 analyzes the robustness of deadlock avoidance algorithm. Tolerable machine failure under which liveness property can be preserved are established. Section 5 demonstrates the result of this paper by an example. Section 6 concludes this paper.

2. PETRI NET MODEL

This Section first introduces the sequential process that will be considered throughout the remaining of this paper. Then Petri Net is used to model this type of production processes. The class of sequential processes under consideration is defined as follows.

Definition 2.1: A sequential process is a process involving a sequence of operations. A sequential process is said to be a re-entrant process if there exists at least one two operations requiring the same resource.

A Petri Net (PN) G is a five-tuple $G = (P, T, I, O, m_0)$, where P is a finite set of places with cardinality $|P|$, T is a finite set of transitions, $I \subset P \times T$ is a set of transition input arcs, $O \subset T \times P$ is a set of transition output arcs, and $m_0 : P \rightarrow Z^{|P|}$ is the initial marking of the PN with Z as the set of nonnegative integers. The marking of G is a vector $m \in Z^{|P|}$ that indicates the number of tokens in each place and is a state of the system. The readers may refer to (Murata, 1989) for further definitions such as enabled transitions, transition firing rules and the set of reachable markings of the PN G from an initial marking m_0, denoted as $R_\infty(G(m_0))$.

Consider a set $\boldsymbol{J}$ of sequential processes. Typically, a type-j sequential process s_j can be modelled as a sequence of transitions as $t_j(1)t_j(2)t_j(3)....t_j(n_j)$, where $j \in \boldsymbol{J}$. The Petri Net model, called job subnet, associated with a type-j production process is represented as GJ_j. Let $\boldsymbol{R}(G)$, abbreviated as $\boldsymbol{R}$, be the set of resource types in the manufacturing system that corresponds to G. Assume that a unit of resource can only be involved in one operation at a time. Suppose that a type-r resource, $r \in \boldsymbol{R}$, may be involved in a number of activities, where each activity consists of a sequence of operations using type-r resources sequentially and can be modelled as a resource activity circuit. All the resource activity circuits of a type-r resource, $r \in \boldsymbol{R}$, can be represented as a Petri Net model called resource subnet of type-r resource as GR_r.

To execute an operation, resource(s) and part(s) involved in an operation have to be synchronized. Resource subnets GR_r and job subnets GJ_j are merged to form a PN that models the interactions among operations, resources and jobs in the manufacturing system. In a manufacturing system, each resources such as machine or robot have exogenous control points. To model control points in Petri Net, we define control places and controlled transitions as follows.

Definition 2.2: A control place p_c, represents a control point for enabling or disabling a transition such as a job loading transition or a resource allocation transition. A control place is denoted by a small square box There is a directed arc from p_c to the corresponding controlled transition. A controlled transition may be fired as many times as the number of tokens in the control place. Let P be the set of places, P_c be the set of control places, T_u be the set of of uncontrolled transitions and T_c be the set of controlled transitions resource allocation and job loading operations, where $P \cap P_c = \Phi$.

Definition 2.3: A CPPN is defined as an eight tuple $G_c = (P, P_c, T_u, T_c, I, O, m_0, u)$, abbreviated as $G_c(m_0, u)$, where u is a control policy defined based on control action of a given CPPN as follows.

Definition 2.4: A control action a is a vector in $Z^{|P_c|}$ that determines how many times that each transition in T_c may be fired concurrently. We will use $a(p_c)$ to denote the number of tokens in control place p_c under control action a . A control policy u is a mapping $u : \boldsymbol{M}_0(G_c) \to (Z^{|P_c|})^{\infty}$ that generates a sequence{ a_n }of control actions for the CPPN G_c based on its initial marking, where $\boldsymbol{M}_0(G_c)$ is the set of admissible initial markings of G_c and is defined as follows.

Definition 2.5: The set of admissible initial markings of CPPN G_c is defined as $\boldsymbol{M}_0(G_c)$ $\equiv \{m \mid m(p) = 0 \ \forall\ p \in P - P_o$ and $m(p) \geq 0$ $\forall\ p \in P_o\}$ and is abbreviated as $\boldsymbol{M}_0$, where P_o the set of all idle state places for all types of resources of G_c .

Note that $\boldsymbol{M}_0(G_c)$ denotes the system states of CPPN G_c under which all resources are in idle state. The set of feasible initial markings for G_c is denoted as $\boldsymbol{M}_0(G_c)$ ={ $m \mid m \in \boldsymbol{M}_0(G_c)$ and there exists a control policy u under which $G_c(m,u)$ is live.} and is abbreviated as $\boldsymbol{M}_0$. Obviously, $\boldsymbol{M}_0 \subseteq \boldsymbol{M}_0$. The set of reachable markings from $m_0 \in \boldsymbol{M}_0(G_c)$ under a control policy u is denoted as $R_\infty(G_c(m_0,u))$ and is abbreviated as $R_\infty(m_0,u)$. For an initial marking $m_0 \in \boldsymbol{M}_0(G_c)$, the set of all admissible reachable markings of G_c is denoted as $\boldsymbol{M}(m_0) \equiv \{m \mid m \in \bigcup_{u \in \boldsymbol{U}} R_\infty(m_0,u)$, where $\boldsymbol{U}$ is the set of all control policy of G_c $\}$ and is abbreviated as $\boldsymbol{M}$.

Definition 2.6: $G_c(m_0,u)$ is live if all the transitions in G_c can be fired infinitely from m_0 under the given u . A control action a is called a valid control action if there exists some control policy that keeps the CPPN live after execution of a .

3. LIVENESS CONDITIONS BASED ON TOKEN FLOW PATHS

A feasible condition to maintain the liveness of a CPPN under some control policy is to have sufficient resources available. A feasible initial marking for a given CPPN G_c can be calculated based on the concept of minimal resource requirement (MRR). The MRR for the existence of a control policy u that keeps G_c live is defined as follows.

Definition 3.1: $\boldsymbol{M}_0^*(G_c)$ = { $m \mid m \in \boldsymbol{M}_0(G_c)$ and for any $m' \in \boldsymbol{M}_0(G_c)$ with $m' < m$, there does not exist any control policy u' under which $G_c(m',u')$ is live} denote the set of minimal feasible initial markings of G_c . $\boldsymbol{M}_0^*(G_c)$ is abbreviated as $\boldsymbol{M}_0^*$ when it is clear from the context.

There are two equivalent ways to represent $\boldsymbol{M}_0^*(G_c)$. One is represented by a vector $N^* \in Z^{|\boldsymbol{R}|}$ and the other is represented by a marking $m^* \in \boldsymbol{M}_0^*(G_c)$, with $m^*(p_r(0)) = N^*(r)$ and $p_r(0)$ denotes the idle state place corresponding to type- r resources.

As a CPPN is constructed by merging a number of mutually interacting resource subnets and job subnets, we will first decompose G_c into a number of decomposed subnets, G_j , one for each $j \in \boldsymbol{J}$. A minimal resource requirement for firing all transitions in G_c can then be calculated based on MRR of G_j $\forall\ j \in \boldsymbol{J}$.

Definition 3.2: N_j^*, a vector in $Z^{|\boldsymbol{R}|}$, denotes the set of resources required for firing the sequence, $s_j = t_j(1)t_j(2)t_j(3)....t_j(n_j)$ of transitions. Let R_t , a vector in $Z^{|\boldsymbol{R}|}$, denotes the resource requirement for firing transition t .

$N_j^* = R_{t_j(1)} \oplus R_{t_j(2)} \oplus R_{t_j(3)} \oplus \ldots \oplus R_{t_j(n_j)}$, where $\oplus$ takes the larger of the two vectors element by element. A MRR N^* for G_c can be obtained by calculating $N^* = N_1^* \oplus N_2^* \oplus N_3^* \oplus \ldots \oplus N_{|\boldsymbol{J}|}^*$.

Theorem 3.1: There exists a control policy u such that $G_c(m,u)$ is live if and only if there exists a $m^* \in \boldsymbol{M}_0^*$ and a sequence of control actions that bring m to a marking $m' \in \boldsymbol{M}_0$ with $m' \geq m^*$, where $m \in \boldsymbol{M}(m_0)$.

Theorem 3.1 implies that as long as the set of resources that can be released from the current system state dominates the MRR, the liveness of the system can be maintained. By exploiting the structure of the sequential production processes, release of resources can be evaluated based on the acyclic marked graph MG_j associated with type-

j production process. A procedure to obtain MG_j based on decomposition of a given CPPN G_c has been proposed in (Hsieh,1994). Please refer to (Hsieh,1994) for details. Let P_{jr} denote the set of places to which a resource in use by type- j production process may be released and return to idle state.

Definition 3.3:The total number of tokens in a token flow path π in an acyclic type- j marked graph MG_j under submarking m^j is denoted as $\pi(m_j) = \sum_{p\in\pi} m_j(p)$.

Definition 3.4: Let $\Gamma_{jr}(p_{ro})$ denote the set of token flow paths for type- r resources ending with a place $p_{ro} \in P_{jr}$.

As each acyclic Marked Graph MG_j, $j \in \boldsymbol{J}$, is a deterministic Petri Net, the number of type- r resources that may stay at or be released to the idle state place of type- r resources under control action a and marking m is $\sum_{j\in \boldsymbol{J}} \sum_{p_{ro}\in P_{jr}} \min_{\pi\in\Gamma_{jr}(p_{ro})} \pi(m_j)$.

Combining the above result with Theorem 3.1, the following Corollary holds.

Corollary 3.1: There exists a control policy u such that $G_c(m,u)$ is live if

$$\sum_{j\in \boldsymbol{J}} \sum_{p_{ro}\in P_{jr}} \min_{\pi\in\Gamma_{jr}(p_{ro})} \pi(m_j) \ge m^*(p_r(0)),$$

where $m \in \boldsymbol{M}(m_0)$.

4. ROBUSTNESS ANALYSIS OF DEADLOCK AVOIDANCE ALGORITHM

In this Section, we characterize the tolerable machine failure based on the condition of Corollary 3.1. To convey the idea, consider the inequality stated in Corollary 3.1:

$$\sum_{j\in \boldsymbol{J}} \sum_{p_{ro}\in P_{jr}} \min_{\pi\in\Gamma_{jr}(p_{ro})} \pi(m_j) \ge m^*(p_r(0)) \qquad (1)$$

Inequality (1) implies that $\min_{\pi\in\Gamma_{jr}(p_{ro})} \pi(m_j)$ remains intact as long as the change $\delta m_j(p)$ of the decrease in the number of tokens of a place p does not reduce the sum of tokens in any of the token flow path $\pi \in \Gamma_{jr}(p_{ro})$ such that $\pi(m'_j) \le \min_{\pi\in\Gamma_{jr}(p_{ro})} \pi(m_j)$, where m'_j denote the perturbed marking after $\delta m_j(p)$ units of tokens have been removed from m_j. Based on this observation, the following definition is required to convey the above concept. The set of all token flow paths is devided into two categories: critical paths and non-critical paths. Removing one token from a place in a non-critical path has no effect on the liveness property of a CPPN. Removing one token from a place in a critical path will reduce the number of tokens that will be released and may destroy the liveness property of the CPPN.

Definition 4.1: Under control action a and marking m, a token flow path in $\Gamma_{jr}(p_{ro})$ with the total number of tokens along the path equal to $\min_{\pi\in\Gamma_{jr}(p_{ro})} \pi(m_j)$ is called a critical path. The set of critical paths in $\Gamma_{jr}(p_{ro})$ is denoted as $\Gamma^c_{jr}(p_{ro}) = \{\ \pi \mid \pi(m_j) = \min_{\pi\in\Gamma_{jr}(p_{ro})} \pi(m_j)$ and $\pi \in \Gamma_{jr}(p_{ro})\ \}$.

Definition 4.2: $\Gamma^n_{jr}(p) = (\ \Gamma_{jr} - \Gamma^c_{jr}(p)\ ) \cap \Gamma_{jr}(p)$ represents the set of non-critical paths for Type- r resources in MG_j under control action a and marking m, where $p \in P_{jr}$.

Based on the above definition, the main result is stated as follows.

Theorem 4.1: Given a CPPN $G_c(m,u)$ under control action a and marking m, where $m \in R_\infty(G_c(m_0,u))$ and $m_0 \in \boldsymbol{M}^*_0$, for any change $\delta m \in \boldsymbol{\delta M}_p(m)$ with $\delta m_j(p) = \delta_p$ for some $j \in \boldsymbol{J}$, the number of type- r resources that will be released to p_{ro} is decreased by $\delta\gamma(p_{ro}, m_j) = \delta_p * \alpha(j, p_{ro}, p, m_j)$ - $\beta(j, p_{ro}, p, m_j)$, where

$\alpha(j, p_{ro}, p, m_j) = 0$ and $\beta(j, p_{ro}, p, m_j) = 0$ if $\Gamma^c_{jr}(p_{ro}) \cap \Gamma_{jr}(p) = \Phi$, and $\Gamma^n_{jr}(p_{ro}) \cap \Gamma_{jr}(p) = \Phi$,

$\alpha(j, p_{ro}, p, m_j) = 1$ and $\beta(j, p_{ro}, p, m_j) = 0$ if $\Gamma^c_{jr}(p_{ro}) \cap \Gamma_{jr}(p) \ne \Phi$, and $\Gamma^n_{jr}(p_{ro}) \cap \Gamma_{jr}(p) = \Phi$,

$\alpha(j, p_{ro}, p, m_j) = 0$ and $\beta(j, p_{ro}, p, m_j) = 0$ if $\Gamma^c_{jr}(p_{ro}) \cap \Gamma_{jr}(p) = \Phi$, and $\Gamma^n_{jr}(p_{ro}) \cap \Gamma_{jr}(p) \ne \Phi$, and

$$\min_{\pi\in\Gamma^n_{jr}(p_{ro})\cap\Gamma_{jr}(p)} \pi(m_j) - \min_{\pi\in\Gamma^c_{jr}(p_{ro})} \pi(m_j) - \delta_p \ge 0$$

$\alpha(j, p_{ro}, p, m_j) = 1$ and

$\beta(j, p_{ro}, p, m_j) = \min_{\pi\in\Gamma^n_{jr}(p_{ro})\cap\Gamma_{jr}(p)} \pi(m_j) - \min_{\pi\in\Gamma^c_{jr}(p_{ro})} \pi(m_j)$ if $\Gamma^c_{jr}(p_{ro}) \cap \Gamma_{jr}(p) = \Phi$, and $\Gamma^n_{jr}(p_{ro}) \cap \Gamma_{jr}(p) \ne \Phi$, and

$$\min_{\pi\in\Gamma^n_{jr}(p_{ro})\cap\Gamma_{jr}(p)} \pi(m_j) - \min_{\pi\in\Gamma^c_{jr}(p_{ro})} \pi(m_j) - \delta_p < 0,$$

$\alpha(j, p_{ro}, p, m_j) = 1$ and $\beta(j, p_{ro}, p, m_j) = 0$ if $\Gamma^c_{jr}(p_{ro}) \cap \Gamma_{jr}(p) \ne \Phi$, and $\Gamma^n_{jr}(p_{ro}) \cap \Gamma_{jr}(p) \ne \Phi$.

Based on Theorem 4.1, we characterize the set of tolerable single place token loss that leaves the system live as follows.

Definition 4.3:
Let
$\boldsymbol{\delta}\overline{\boldsymbol{M}}_p(m) \equiv \{\delta m \mid \delta m \in \boldsymbol{\delta M}_p(m) \text{ and } \delta m \text{ satisfies } (2)\}$.

$$\sum_{j\in \boldsymbol{J}} \sum_{p_{ro}\in P_{jr}} \min_{\pi\in\Gamma_{jr}(p_{ro})} \pi(m_j) - \delta_p * \alpha(j, p_{ro}, p, m_j) + \beta(j, p_{ro}, p, m) \geq m^*(p_r(0)),\ \forall r \in \boldsymbol{R} \qquad (2)$$

It follows directly from Theorem 4.1 and the above definition that the following theorem offers a sufficient liveness condition for tolerable token loss of a single place.

Theorem 4.2: Given a CPPN G_c that can be kept live under DAC under control actio a and marking m, where $m \in R_\infty(G_c(m_0))$ and $m_0 \in \boldsymbol{M}_0^*$, for any change $\delta m \in \boldsymbol{\delta}\overline{\boldsymbol{M}}_p(m)$, there exists a control policy u' under which $G_c(m-\delta m, u')$ is live.

5. AN ILLUSTRATIVE EXAMPLE

To access the robustness property, the following example that has been considered in (Banaszak, 1990) and (Hsieh, 1994) is reproduced here. Consider the system with marking m_1 and control action a as shown in Figure 1. As there is only one type of production process, $\boldsymbol{J} = \{1\}$. For this example, in (Hsieh, 1994), the author has demonstrated that this controlled Petri Net can be kept live under some deadlock avoidance control (DAC) algorithm. The following scenario is created to check the robustness property of the deadlock avoidance algorithm.

Note that $\Gamma_{13}(p_{12}(2)) = \{\pi_1, \pi_2, \pi_3, \pi_4, \pi_5\}$, where
$\pi_1 = p_0 p_1 p_2 p_3 p_{12}(2)$ with $\pi_1(m_1) = 6$,
$\pi_2 = p_{10}(1) p_1 p_2 p_3 p_{12}(2)$ with $\pi_2(m_1) = 9$,
$\pi_3 = p_{11}(1) p_2 p_3 p_{12}(2)$ with $\pi_3(m_1) = 8$,
$\pi_4 = p_{12}(1) p_3 p_{12}(2)$ with $\pi_4(m_1) = 8$ and
$\pi_5 = p_{13}(1) p_{12}(2)$ with $\pi_5(m_1) = 1$.
As $\pi_5(m_1) = \min_{\pi\in\Gamma_{13}(p_{12}(2))} \pi(m_1)$, π_5 is a critical token flow path.
As $\pi_i(m_1) > \min_{\pi\in\Gamma_{13}(p_{12}(2))} \pi(m_1)$ for $i \in \{1,2,3,4\}$,
π_1, π_2, π_3 and π_4 are not critical token flow paths.

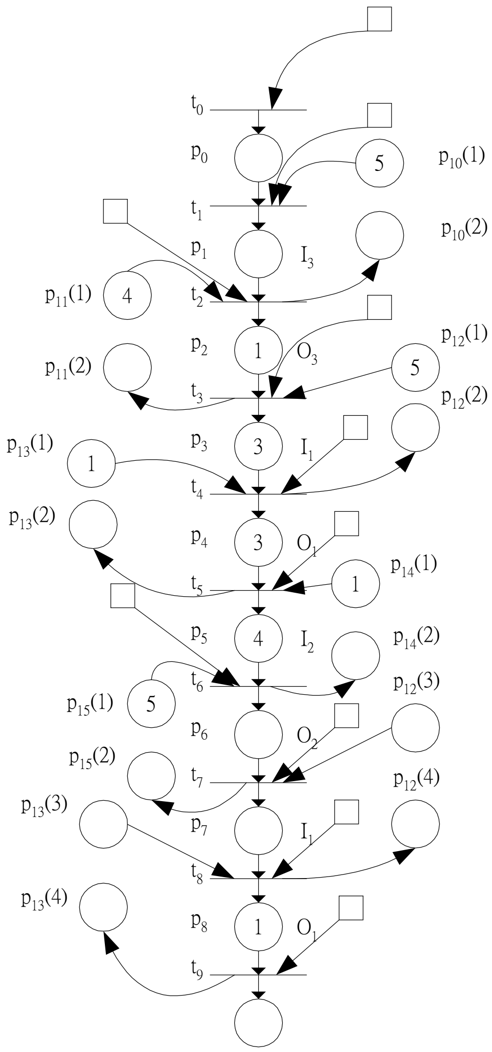

Fig. 1. Token allocation under a marking and control action.

A machine in failure mode can be modelled as removal of tokens from a place in the Petri Net model. Therefore removing one or two units of machines from place p_3 still preserves the liveness property of the resulting Petri Net.

CONCLUSIONS

Uncertainties such as machine failure poses challenges in design and control of manufacturing systems as resource failures may reduce the number of available resources which may in turn result in deadlocks or lead to dead states. As existing papers assume reliable machines in development of deadlock avoidance algorithms, there is a gap between the development of theory and application to real manufacturing systems. To bridge this gap, feasibility of tailoring existing deadlock avoidance algorithms for manufacturing systems with reliable resources to systems with unreliable resources requires is studied in this paper. The main results show that the existing DAC algorithm possesses desirable robustness properties that is capable of dealing with unreliable machines. The concept of critical path is proposed to quantitatively characterize the safety margins with which the system remain can still be kept live using the remaining properly functioned resources. The results of this paper shows that it is feasible to tailor existing deadlock avoidance algorithms to systems with unreliable machines.

REFERENCES

Banaszak Z. and B. Krogh, (1990) Deadlock avoidance in flexible manufacturing systems with concurrently competing process flows, *IEEE Trans. Robot. Automat.*, vol. 6, pp. 724-734.

Cho H., T. K. Kumaran and R. A. Wysk, (1995) Graph-theoretic deadlock detection and resolution for flexible manufacturing systems, *IEEE Trans. Robot. Automat.*, vol.11, pp. 413-421.

Ezpeleta J., J.M. Colom, and Martinez, (1995) A Petri Net based deadlock prevention policy for flexible manufacturing systems, *IEEE Trans. Robot. Automat.*, vol.11, pp. 173-184.

Hsieh F.S. and S.C. Chang, (1994) Dispatching driven deadlock avoidance controller synthesis for flexible manufacturing systems, *IEEE Trans. Robot. Automat.*, vol.10, pp. 196-209.

Jeng M. D. and F. DiCesare, (1993) A review of synthesis techniques for Petri nets with applications to automated manufacturing systems, *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 1, pp. 301-312.

Jeng M. D., (1997) Petri Nets for Modeling Automated Manufacturing Systems with Error Recovery, *IEEE Trans. Robot. Automat.*, vol.13, no. 5, pp. 752-760.

Lawley M. A., (1998a) Deadlock avoidance for production systems with flexible routing, *IEEE Trans. Robot. Automat.*, vol.15, pp. 497-509.

Lawley M. A., Spyros A. Reveliotis and Placid M. Ferreira, (1998b) A correct and scalable deadlock avoidance policy for flexible manufacturing systems, *IEEE Trans. Robot. Automat.*, vol. 14, pp. 796-809.

Murata T., (1989) Petri Nets: Properties, Analysis and Applications, *Proceedings of the IEEE*, vol. 77, No. 4.

Narahari Y. and N. Viswanadham, (1985) A Petri net approach to the modeling and analysis of flexible manufacturing systems, *Annals Oper. Res.*, vol. 3, pp. 449-472.

Reveliotis S.A., M.A. Lawley and P.M. Ferreira, (1997) Polynomial-Complexity Deadlock Avoidance Policies for Sequential Resource Allocation Systems," *IEEE Trans. Automat. Contr.*, vol.42, pp. 1344-1357.

Reveliotis S.A., (1999) Accommodating FMS operational contingencies through routing flexibility, *IEEE Trans. Robot. Automat.*, vol.15, pp. 3-19.

Viswanadham N., Y. Narahari, and T.L. Johnson, (1990) Deadlock prevention and deadlock avoidance in flexible manufacturing systems using Petri net models, *IEEE Trans. Robot. Automat.* Vol. 6, pp. 713-723.

Wysk R. A., N.S. Yang, and Joshi, (1991) Detection of deadlocks in flexible manufacturing cells, *IEEE Trans. Robot. Automat.* vol. 7, pp. 853-859.

Zhou M. C., F. DiCesare and A.A. Desrochers, (1992) A hybrid methodology for synthesis of Petri net models for manufacturing systems, *IEEE Trans. Robot., Automat.*, vol. 8, no. 3, pp. 350-361.

Zhou M. C. and F. DiCesare, (1991) Parallel and sequential mutual exclusions for Petri net modeling for manufacturing systems with shared resources, *IEEE Trans. Robot., Automat.*, vol. 7, no. 4, pp. 515-527.

Zhou M.C. and F. DiCesare, (1989) Adaptive design of Petri net controllers for error recovery in automated manufacturing systems, *IEEE Trans. Syst., Man, Cybern.*, vol. 19, no. 5, pp. 963-973.