Robustness of a Class of Non-ordinary Petri Nets

Fu-Shiung Hsieh#1

Department of Computer Science and Information Engineering, Chaoyang University of Technology
Taiwan, R.O.C.

1fshsieh@cyut.edu.tw

***Abstract*—Robustness analysis provides an alternative way to analyze a perturbed system to quickly respond to resource failures. Non-ordinary Petri nets have weighted arcs and have the advantage to compactly model operations requiring multiple parts or resources. In this paper, we consider a class of non-ordinary flexible assembly/disassembly Petri net (NFADPN) models and study its robustness properties. Due to the routing flexibility in NFADPN, there may exist different ways to accomplish the tasks. To take advantage of the alternative routes without enforcing liveness for the whole system, we generalize the concept of persistent production proposed to NFADPN. We propose a condition for persistent production. We extend robustness analysis to NFADPN by exploiting its structure. We identify several patterns of resource failures and characterize the conditions to maintain operation in the presence of resource failures.**

***Keywords*—Petri net, robustness, failure**

1. INTRODUCTION

Assembly systems have attracted a lot of attention recently. For example, Fanti *et al*. [9] dealt with the deadlock problem in automated assembly systems based on discrete event system theory. Many supervisory control problems in discrete event systems can be modeled and analyzed by Petri nets (CPN) [1]-[3]. Roszkowska *et al*. [10] defined a Petri net model for assembly/disassembly processes and proposed a control policy to determine the reserved buffer spaces to guarantee enough spaces for assembly to continue. Wu, Zhou and Li [5] proposed a resource-oriented Petri nets to tackle deadlocks in a class of flexible assembly systems. In addition to the development of supervisory control algorithm, design of robust supervisory controllers for manufacturing systems with unreliable resources has received significant attention recently [20], [6]-[8], [16]-[19]. Unreliable resources, which may result from a variety of causes, including unreliable components, defective parts, faulty sensors, and so forth, are a common problem in real systems. Resource failures pose challenges in control of discrete event systems. In [16]-[18], Lawley, Chew, Wang and Sulistyono developed supervisory controllers to ensure robust deadlock-free operation for systems with unreliable resources. Another approach for analyzing the robustness of manufacturing systems is based on Petri nets. In case of resource failures, re-analysis of the perturbed system is required. However, reanalysis based on existing reachability tree method is inefficient due to the state space explosion problem. Robustness analysis provides an alternative way to determine whether the operation of a perturbed system or some parts of it can still be maintained in case of resource failures. In [20] and [6]-[8], Hsieh analyzed the robustness property of several subclasses of ordinary Petri nets, including controlled production Petri net (CPPN)[20], controlled assembly Petri net (CAPN)[6], controlled assembly/disassembly processes Petri nets (CADPN) [7] and controlled assembly Petri net with alternative routes (CAPN-AR) [8]. However, the aforementioned models and analysis are based on ordinary Petri nets. Non-ordinary Petri nets [11]-[12], which have weighted arcs, have the advantage to compactly model operations requiring multiple parts or resources. However, robustness analysis for non-ordinary Petri net models has not been studied. Our interest in this paper is to propose an effective methodology to model and analyze the robustness of flexible assembly/disassembly systems based on non-ordinary Petri nets.

In this paper, we study the robustness of a class of non-ordinary Petri nets called NFADPN. A bottom-up approach [14]-[15] is adopted in this paper to construct the NFADPN models. The NFADPN model is constructed by merging the job subnets with resource subnets [6], [8] to capture the interactions between job subnets and resource subnets. Due to the routing flexibility in NFADPN, there may exist different ways to accomplish the tasks. The routing flexibility in NFADPN significantly enhances the robustness of the system by allowing a product to follow the routes in any completely connected subprocesses as long as the resource requirements can be met. However, alternative routes in NFADPN also add complexity to the analysis of the system. As long as there exists a set of completely connected subprocesses for certain type of products, the production of that type of products can still be maintained without requiring the whole NFADPN to be live.We characterize the robustness property for NFADPN by exploiting the net structure.

The remainder of this paper is organized as follows. In Section 2, we introduce a bottom-up approach to construct the NFADPN model for flexible assembly/disassembly systems in non-ordinary Petri nets. We propose uncertainties model based on the NFADPN model and study the condition for persistent production in Section 3. In Section 4, we analyze the robustness of NFADPN. We conclude this paper in Section 5.

2. NON-ORDINARY PETRI NET MODEL

The production processes of a flexible assembly/disassembly system can be viewed as the interactions between the (non-reusable) parts (or components) and (reusable) resources. The precedence constraints and the dependency between different parts (or components) can be can be described by a workflow model. A resource involved in the operations of a workflow can be described by a set of activities. Petri net is an effective tool for modeling the workflows of parts and activities of resources. To model a flexible assembly process using Petri net, a brief introduction to Petri net is given first. A non-ordinary Petri Net (PN) G is a four-tuple $G=(P,T,W,m_0)$, abbreviated as $G(m_0)$, where P is a finite set of places with cardinality $|P|$, T is a finite set of transitions, $W:(P\times T)\cup(T\times P)\rightarrow\{0,1,2,3,...\}$ is a weight function to specify set of transition input/output arcs, $m_0: P\rightarrow Z^{|P|}$ is the initial marking of the PN with Z as the set of nonnegative integers. A 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. We use ${}^{\bullet}t$ and $t^{\bullet}$ to denote the set of input places and the set of output places of transition t, respectively. All Petri nets are assumed to be ordinary in this paper. A transition t is enabled and can be fired under a marking m if and only if $m(p)>0 \quad \forall p\in{}^{\bullet}t$. Firing a transition once removes one token from each of its input places and adds one token to each of its output places. A marking m' is reachable from m iff there exists a firing sequence of transitions bringing the net from m to m'. The reachability set $R(m_0)$ denotes all the markings that are reachable from m_0. A Petri net $G=(P,T,W,m_0)$ is said to be live if, no matter what marking has been reached from m_0, it is possible to ultimately fire any transition of G by progressing through some further firing sequence. Please refer to [2] for a tutorial on Petri nets.

To capture the workflows of parts in flexible assembly/disassembly systems, we introduce the workflow net model. We use ${}^{\bullet}p$ and $p^{\bullet}$ to denote the set of input transitions and the set of output transitions of p, respectively.

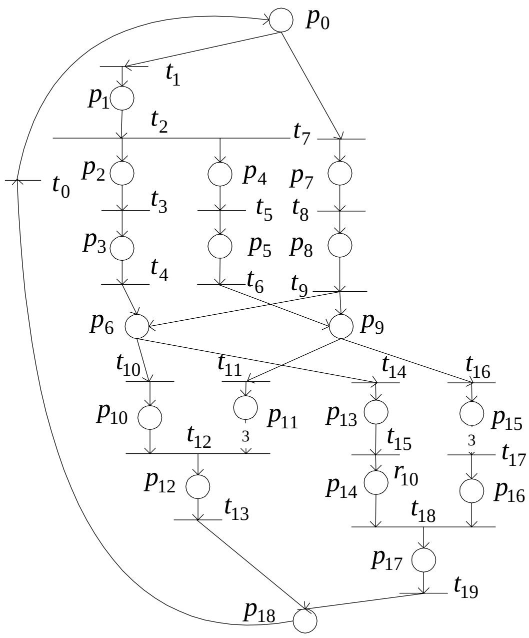

Figure 1

Definition 2.1: A PN G is a workflow net if and only if (i) G has a source place ε and a sink place θ with ${}^{\bullet}\varepsilon=\Phi$ and $\theta^{\bullet}=\Phi$ and (ii) if we add a new transition t to G to connect θ with ε, the resulting PN G' is strongly connected. The augmented workflow net G' associated with G is the strongly connected PN obtained by add a new transition t to G to connect θ with ε.

Fig. 1 shows an augmented workflow net.

The workflows of parts are modeled by job subnets defined as follows.

Definition 2.2: A type- j job subnet $GJ_j=(P_j^J, T_j^J, W_j^J, m_{j0})$ is an augmented workflow net, where $\varepsilon_j\in P_j^J$ denotes the source place and $\theta_j \in P_j^J$ denotes the sink place and $m_{j0}(p) = 0 \; \forall p \in P_j^J$. That is, no job is in process under m_{j0} .

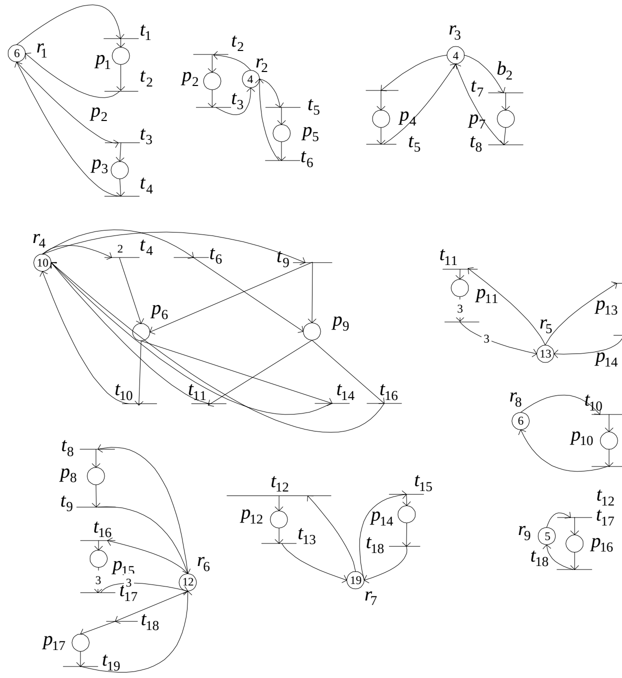

Figure 2 G_r , $r \in \boldsymbol{R} = \{r_1, r_2, ..., r_9\}$

GJ_j only model the behaviors of workflows without taking into account the interactions with resources. To take into account the interactions with resources, Petri net models of resources are proposed. Let $\boldsymbol{R}$ denote the set of different types of resources. A type-r resource may take part in a set $\Omega_r = \{1,2,3,\ldots,K_r\}$ of activities, where $r \in \boldsymbol{R}$. To model an activity $k \in \Omega_r$ in Petri net, we use a transition t_a^k to represent allocation of resources and use t_d^k to represent de-allocation of resources after completing the operations in it. The $k-th$ activity of type-r resources is described by a directed path $p_1 t_1 p_2 t_2 p_3 t_3 ..... p_n t_n p_{n+1}$ starting with place p_{ri}^k and ending with place p_{ro}^k , where $p_1 = p_{ri}^k$, the idle state before allocating a type-r resource, $p_{n+1} = p_{ro}^k$, the idle state after de-allocating a type-r resource, $t_1 = t_a^k$ and $t_n = t_d^k$.

Let $w_i = W(p_i, t_i)$ and $v_i = W(t_i, p_{i+1})$ for $i = 1,2,3,..., n$. As resources conserve, we assume $\frac{v_n}{1} \frac{v_{n-1}}{w_n} \frac{v_{n-2}}{w_{n-1}} ... \frac{v_{n-k}}{w_{n-k+1}} ... \frac{1}{w_1} = 1$. The Petri net model for the $k-th$ activity is described by a Petri net G_r^k defined as follows.

Definition 2.3: The $k-th$ activity of type-r resources is represented by the Petri net $G_r^k = (P_r^k, T_r^k, W_r^k, m_{r0}^k)$. Remark that $T_r^k \cap T_r^{k'} = \Phi$ for $k \neq k'$.

For each $r \in \boldsymbol{R}$, we merge places p_{ri}^k and p_{ro}^k for all $k \in \Omega_r$ into one place p_{r0} . The type-r resource subnet G_r is defined by G_r = $\|_{k \in \Omega_r} G_r^k = G_r^1 \| G_r^2 \| \ldots G_r^{K_r}$ as follows.

Definition 2.4: The Petri net G_r = $\|_{k \in \Omega_r} G_r^k$ = (P_r, T_r, W_r, m_{r0}) denotes the type-r resource subnet, where $m_{r0}(p_{r0}) > 0$ and $m_{r0}(p) = 0 \; \forall p \in P_r \setminus \{p_{r0}\}$. $P_o = \{p_{r0} | r \in \boldsymbol{R}\}$ denotes the set of all resource idle places.

Fig. 2 shows nine resource subnets, where $P_o = \{r_1, r_2, ..., r_9\}$.

To capture the interactions among resources and process workflows, we apply the operation " $\|$ " (defined in [6]-[8]) to merge two PNs through common places, transitions, or arcs. The Petri net model $G(m_0) = \|_{r \in \boldsymbol{R}} GR_r \; \|_{j \in \boldsymbol{J}} GJ_j$ is constructed by merging the resource subnets GR_r , $r \in \boldsymbol{R}$, with job subnets GJ_j , $j \in \boldsymbol{J}$, where $\boldsymbol{J}$ denotes the set of different types of processes in the system, $\boldsymbol{R}$ denotes the set of resource types in the system, u is a control policy and the initial marking m_0 of G belongs to $\boldsymbol{M}_0$ is defined below.

Definition 2.5: $\boldsymbol{M}_0 \equiv \{m \mid m(p) = 0 \; \forall \; p \in P \setminus P_o$ and $m(p) > 0 \; \forall \; p \in P_o\}$ denotes the set of initial marking of G , where P_o is the set of all idle places of all types of resources.

Based on G , a non-ordinary controlled Petri net G_c is defined as follows.

Definition 2.6: A non-ordinary flexible assembly/disassembly Petri net (NFADPN) G_c is defined based on G as a five-tuple G_c = (P, T, W, m_0, u) , abbreviated as $G_c(m_0, u)$, where P is a finite set of places with cardinality $|P|$, T is a finite set of transitions, $W : (P \times T) \cup (T \times P) \rightarrow \{0,1,2,3,...\}$ is a weight function to specify set of transition input/output arcs, $m_0 : P \rightarrow Z^{|P|}$ is the initial marking of the PN with Z as the set of nonnegative integers, $T = T_c \cup T_u$, T_c is the set of controlled transitions, T_u is the set of uncontrolled transitions and u is a control policy defined based on control action of a given NFADPN as follows.

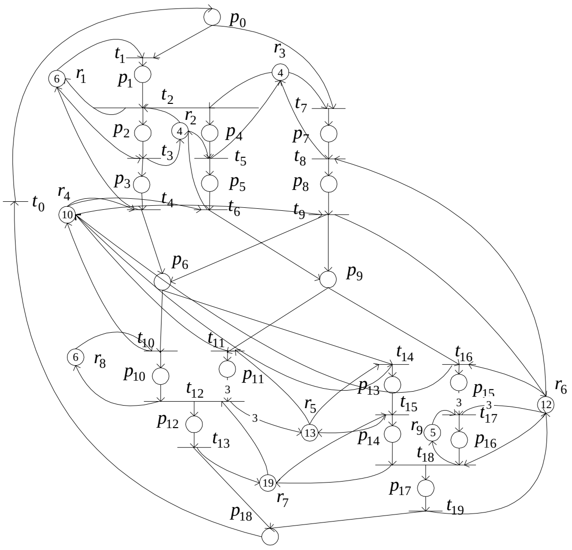

Figure 3(a) A NFADPN G_c

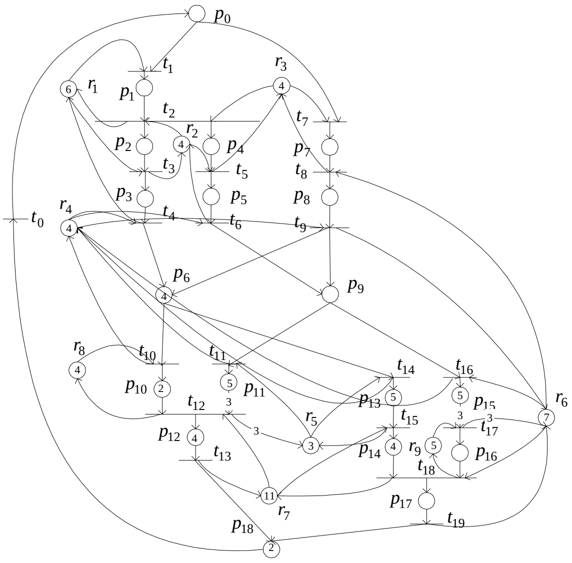

Figure 3(b) A NFADPN G_c

Definition 2.7: A control action a is a vector in $Z^{|T_c|}$ that determines how many times that each transition in T_c may be fired concurrently, where $a(t)$ denotes the number of times transition t can be fired under control action a. A control policy u is a mapping $u: R_c(m_0) \to Z^{|T_c|}$ that generates a sequence $\{a_n\}$ of control actions for the NCPN G_c based on its marking $m \in R_c(m_0)$, where $R_c(m_0)$ is the set of all reachable markings of G_c from an initial marking m_0.

Definition 2.8: $G_c(m_0, u)$ is live if no matter what marking hs been reached from m_0, it is possible to ultimately fire any transition of G_c by progressing through some further firing sequence under u. A control action a is called an admissible control action if there exists a control policy that keeps the NFADPN live after executing a.

Fig. 3(a) illustrates a NFADPN G_c, where the number in each place denotes the number of tokens. G_c may evolve to the marking as shown in Fig. 3(b).

3. CHARACTERIZATION OF UNCERTAINTIES

Due to the existence of alternative routes, it is not required for each transition in G_c to be live to maintain production. To characterize the condition that maintains the operation of a system without enforcing the liveness of the entire net, the concept of persistent production is introduced. A type-j job can still be completed as long as there exists a transition $t_j^f \in {}^{\bullet}\theta_j$ that can be kept live. To capture this property, we define a process as persistent as follows.

Definition 3.1: Let $j \in \boldsymbol{J}$. A NFADPN G_c is j-persistent under a marking if and only if there exists a control policy under which transition $t_j^f \in {}^{\bullet}\theta_j$ is live. A NFADPN G_c is persistent under a marking if and only if G_c is j-persistent for all $j \in \boldsymbol{J}$.

To analyze the effects of uncertainties on a NFADPN G_c, we extend G_c as follows.

Definition 3.2: A NFADPN with uncertainties (NFADPN U) $G_c^{\Delta}(m_0, u) = (P, T, W, \Delta, m_0, u)$ is an six-tuple defined based on NFADPN G_c = (P, T, W, m_0, u) by introducing a mapping $\Delta: R_c(m_0) \to Z^{|P|}$ that specifies the maximal perturbation $\Delta(m)$ for each marking $m \in R_c(m_0)$, with $0 \le \Delta(m) \le m$ and $\Delta(m) \in Z^{|P|}$, $m_0: P \to Z^{|P|}$ is the initial marking of the PN with Z as the set of nonnegative integers, and u is a mapping $u: R_c^{\Delta}(m_0) \to Z^{|T_c|}$ that generates a sequence $\{a_n\}$ of control actions for CPNU G_c^{Δ} from m_0, where $R_c^{\Delta}(m_0)$ $=\{$ $m'|$ $m' \in R_c(m-\delta)$, where $m \in R_c(m_0)$, $0 \le \delta \le \Delta(m)$ $\}$is the set of all reachable markings of G_c^{Δ} from m_0.

Liveness of a NCPNU G_c^{Δ} is defined as follows.

Definition 3.3: G_c^{Δ} is live under marking m_0 if there exists a control policy u' under which $G_c(m', u')$ is live for each $m' \in R_c^{\Delta}(m_0)$.

Definition 3.4: For two uncertainty functions Δ_1 and Δ_2, we say Δ_1 dominates Δ_2, denoted by $\Delta_1 \geq \Delta_2$, if $\Delta_1(m) \geq \Delta_2(m)$ for each $m \in R_c(m_0)$. We say Δ_1 strictly dominates Δ_2, denoted by $\Delta_1 > \Delta_2$, if $\Delta_1 \geq \Delta_2$ and $\Delta_1(m) > \Delta_2(m)$ for some $m \in R_c(m_0)$.

Definition 3.5: An uncertainty function Δ_1 is a maximal tolerable uncertainty function if $G_c^{\Delta_1}$ is live under marking m_0 and $G_c^{\Delta_2}$ is not live under marking m_0 for any uncertainty function Δ_2 with $\Delta_2 > \Delta_1$.

If G_c^{Δ} is live under marking m_0, $G_c^{\Delta'}$ must be live under marking m_0 for any $\Delta' \leq \Delta$.

Given a NFADPN G_c with marking m, an important issue is to find the maximal tolerable uncertainties $\Delta(m)$. However, the computational complexity to find the maximal tolerable uncertainties of a general NFADPN G_c grows exponentially with the size of the nets.

Definition 3.6: G_c^{Δ} is j-persistent under marking m_0 if there exists a control policy u under which $G_c(m', u)$ is j-persistent for each $m' \in R_c^{\Delta}(m_0)$. G_c^{Δ} is persistent under marking m_0 if and only if G_c^{Δ} is j-persistent for all $j \in \boldsymbol{J}$.

Definition 3.7: G_c^{Δ} is persistent under marking m_0 if there exists a control policy u under which $G_c(m', u)$ is persistent for each $m' \in R_c^{\Delta}(m_0)$.

Definition 3.8: $\boldsymbol{M}_j^*$ denotes the set of initial markings of G_c with minimal resources for the existence of a control policy to keep t_j^f live for all $j \in \boldsymbol{J}$, where $\boldsymbol{M}_j^* \subset \boldsymbol{M}_0$. The set of resources under $m_j^* \in \boldsymbol{M}_j^*$ is called a minimal resource requirement (MRR) of type- j jobs.

In this paper, we assume $m_0 \geq m^* = m_1^* \oplus m_2^* \oplus m_3^* \oplus ... \oplus m_{|\boldsymbol{J}|}^*$.

Property 3.1: Given a NFADPN G_c with marking $m \in R(m_0)$, there exists a control policy u such that G_c is persistent under m if and only if there exists a sequence of control actions that bring m to a marking $m' \in \boldsymbol{M}_0$ with $m'(p_{r0}) \geq m_j^*(p_{r0})\ \forall r \in \boldsymbol{R}, \forall j \in \boldsymbol{J}$, where $m_j^* \in \boldsymbol{M}_j^*$.

Application of Property 3.1 requires computation of m_j^* and testing reachability of a marking m' that covers $m_j^*\ \forall j \in \boldsymbol{J}$. A heuristics to obtain a lower bound of m_j^* can be found by firing an arbitrary permutation of the transitions in T_j^J. Direct application of reachability tree method to this problem is computationally feasible only for small Petri nets [1]-[2]. Instead, we propose an efficient but sufficient only test procedure by exploiting the net structure.

4. ROBUSTNESS ANALYSIS

Let u denote control policy under the nominal supervisory control algorithm. Suppose NFADPN G_c evolves under u from marking m_0, where $m_0 \geq m_{C_j}^*\ \forall j \in \boldsymbol{J}$. Suppose $m \in R_c(m_0)$ is a nominal marking reached under u. Under control policy u, G_c is persistent under each reachable marking m by maintaining the liveness of G_{C_j} for each $j \in \boldsymbol{J}$, where $C_j \in \boldsymbol{C}_j$. Robustness is concerned with the influence of resource failures on the operation of G_c. We model resource unavailability as a perturbation, δ, $0 \leq \delta \leq \Delta(m)$ in nominal marking m. Depending on the structure of δ, its impact on the execution of a task vary. As G_c is persistent under each reachable marking m by maintaining the liveness of each G_{C_j} for each $j \in \boldsymbol{J}$ and each G_{C_j} can be decomposed into a set S_j of subprocesses by splitting each place $p \in \hat{P}_j$ into $\left| {}^\bullet p \cup p^\bullet \right|$ places, we analyze the effects of δ by exploiting the structure of $MG_b\ \forall b \in S_j$ for each $j \in \boldsymbol{J}$. We first consider perturbation δ in a single place p. In this case, $\delta \in \Delta(m)$, $\delta(p) > 0$ and $\delta(p') = 0$ otherwise. The perturbation δ can be broken down into components δ_b, $b \in S_j$, $j \in \boldsymbol{J}$. Let $b \in S_j$ be the subprocess to which place p belongs. In this case, $\delta_{b'} = \underline{0}\ \forall b' \neq b$, $\delta_b(p) > 0$ and $\delta_b(p') = 0$ otherwise.

For a perturbation δ at place p under marking m with $\delta(p) > 0$, the reduction on the number of tokens that will return to place $p' \in R_{br}^o$ is $\gamma_{p'}(m_b) - \gamma_{p'}(m_b - \delta)$. The influence of δ on the number of type- r resources that can be released can be characterized as follows.

Definition 4.1: The influence of perturbation δ on place $p' \in R_{br}^o$ can be computed as follows.

$\varpi_b(m,\delta,r) = \sum_{p' \in R^o_{br}} \gamma_{p'}(m_b) - \gamma_{p'}(m_b - \delta)$.

The function $\varpi_b(\delta,r)$ represents the reduction in the number of resources that can be released in MG_b . Therefore, a sufficient condition to guarantee persistent production is as follows.

Theorem 4.1: Let u denote the nominal supervisory control algorithm with $C_j \in \boldsymbol{C}_j$ selected for each $j \in \boldsymbol{J}$. Suppose NFADPN G_c evolves under u from marking m_0 , where $m_0 \geq m^*_{C_j}\ \forall j \in \boldsymbol{J}$. Suppose $m \in R_c(m_0)$ is a nominal marking reached under u . Let $\bar{\delta} = \Delta_p(m)$, where $p \in P$, be defined as follows $0 \leq \bar{\delta} = \Delta_p(m) \leq m,\quad 0 < \bar{\delta}(p) \leq m_b(p)$, $\bar{\delta}(p') = 0\ \forall p' \in P \setminus \{p\}$ and $\varpi_b(m,\bar{\delta},r) = 0 \forall r \in \boldsymbol{R}$ for each $m \in R_c(m_0)$. Then $G_c^{\Delta_p}$ is persistent under marking m .

5. Conclusion

Unreliable resources may cripple the operation of manufacturing systems unexpectedly and degrade the system performance. One of the approaches to analyze the effects of resource failures is to study the robustness of the systems. Robustness refers to the resource failures tolerable under a certain system state. In this paper, we study the robustness of a class of non-ordinary Petri nets called NFADPN. Due to the routing flexibility in NFADPN, there may exist different ways to accomplish the tasks. The routing flexibility in NFADPN significantly enhances the robustness of the system by allowing a product to follow the routes in any completely connected subprocesses as long as the resource requirements can be met. However, alternative routes in NFADPN also add complexity to the analysis of the system. To take advantage of the alternative routes without enforcing liveness for the whole system, we generalize the concept of persistent production to NFADPN. As the NFADPN model grows rapidly with the scale of the problem, it is computationally infeasible to analyze complex problems based on reachability tree. We propose an analysis method by which the problem can be scaled through structural decomposition of NFADPN to enforce persistent production for a NFADPN in nominal state. We propose sufficient persistent production conditions to verify whether a certain type of resource failures is tolerable. A nominally persistent NFADPN is still persistent under a perturbed state as long as the sufficient condition holds.

Acknowledgment

This paper is supported in part by National Science Council of Taiwan under Grant NSC97-2410-H-324-017-MY3.

References

[1] M.C. Zhou, M. Jeng and M. P. Fanti (editors), Deadlock Resolution in Computer-Integrated Systems, IEEE Transactions on System, Man and Cybernetics, Part A: Systems and Humans, Special Issue, vol. 34, no.1, 2004.

[2] Tadao Murata, ”Petri Nets: Properties, Analysis and Applications,” *Proceedings of the IEEE*, vol. 77, No. 4, April 1989.

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

[4] F. S. Hsieh, “Robustness Analysis of a Class of Non-ordinary Controlled Petri Nets,” Proceedings of the 2006 IEEE International Conference on System, Man and Cybernetics, pp. 3068-3073, Taipei. , Oct. 8-11, 2006.

[5] Naiqi Wu; MengChu Zhou; ZhiWu Li, “Resource-Oriented Petri Net for Deadlock Avoidance in Flexible Assembly Systems,” IEEE Transaction on System, Man and Cybernetics, Part A, Vol. 38, No. 1, pp. 56-69, 2008.

[6] F. S. Hsieh,”Fault Tolerant Deadlock Avoidance Algorithm for Assembly Processes”, IEEE Transactions on System, Man and Cybernetics, Part A: Systems and Humans, Vol. 34, No. 1, pp. 65-79, 2004.

[7] F. S. Hsieh, “Robustness analysis of Petri nets for assembly/disassembly processes with unreliable resources,” Automatica, vol. 42, pp.1159-1166, 2006.

[8] F. S. Hsieh, “Analysis of Flexible Assembly Processes based on Structural Decomposition of Petri Nets,” IEEE Transaction on System, Man and Cybernetics, Part A, Vol. 37, No. 5, pp. 792-803, 2007.

[9] M.P. Fanti, B. Maione, and B. Turchiano, “Event Control for Deadlock Avoidance in Assembly Systems,” *Proceedings of 1997 IEEE Int. Conf. on Syst., Man, Cybern.*, vol. 4, pp. 3756-3761, Oct. 12-15, 1997.

[10] E. Roszkowska, ” Supervisory control for deadlock avoidance in compound processes,” IEEE Transactions on System, Man and Cybernetics, Part A: Systems and Humans, Vol. 34, No. 1, pp. 52-64, 2004.

[11] Young Cheol Cho and Wook Hyun Kwon,” On forbidden state problems in a general

class of non-ordinary controlled Petri nets," *Proceedings of 1998 IEEE Int. Conf. on Syst., Man, Cybern.*, vol. 1, pp. 90-95, Oct. 11-14, 1998.

[12] Jonghun Park and Reveliotis, S.A.," Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings," IEEE Transactions on Automatic Control vol. 46, no. 10, pp. 1572-1583, 2001.

[13] Mu Der Jeng; DiCesare, F.; A review of synthesis techniques for Petri nets with applications to automated manufacturing systems, IEEE Transactions on System, Man and Cybernetics, Part A: Systems and Humans, Vol. 23, No. 1, pp. 301-312, 1993.

[14] Koh, I.; DiCesare, F.; Modular transformation methods for generalized Petri nets and their application to automated manufacturing systems, IEEE Transactions on System, Man and Cybernetics, Part A: Systems and Humans, Vol. 21, No. 6, pp. 1512-1522, 1991.

[15] Ferrarini, L.; Trioni, M.; Modeling shared resources with generalized synchronization within a Petri net bottom-up approach, IEEE Transactions on System, Man and Cybernetics, Part A: Systems and Humans, Vol. 26, No. 4, pp. 653-659, 1996.

[16] Song Foh Chew; Lawley, M.A.; Robust supervisory control for Production Systems with multiple resource failures, IEEE Trans. Automation Science and Engineering, vol. 3, no. 3, pp. 309-323, July 2006.

[17] Shengyong Wang; Song Foh Chew; Lawley, M.A.; Using Shared-Resource Capacity for Robust Control of Failure-Prone Manufacturing Systems, IEEE Transactions on System, Man and Cybernetics, Part A: Systems and Humans, Vol. 38, No.3, pp. 605-627, 2008.

[18] M. Lawley, and W. Sulistyono, "Robust supervisory control policies for manufacturing systems with unreliable resources," IEEE Trans. Robot. Automat., vol. 18, no. 3, pp. 346-359, June 2002.

[19] M. A. Lawley,"Deadlock avoidance for production systems with flexible routing," *IEEE Trans. Robot. Automat.*, vol.15, pp. 497-509, Oct. 1998.

[20] F. S. Hsieh, "Robustness of deadlock avoidance algorithms for sequential processes", Automatica (39) p.1695-1706, 2003.

[21] L.S.H. de Mello and A.C. Sanderson,"A correct and complete algorithm for the generation of mechanical assembly sequences," *IEEE Trans. Robot. Automat.*, vol.7, pp. 228-240, Feb. 1991.

[22] W. M. P. van der Aalst, "The application of Petri nets to workflow management," *J. Circuits, Syst. Comput.*, vol. 8, no. 1, pp. 21–66, 1998.