engineering proceedings

MDPI

Proceeding Paper

Design of Maximally Permissive Controllers for Solving Deadlock Problems in Flexible Manufacturing Systems †

Yen-Liang Pan [1,*], Wen-Yi Chuang [2], Kuang-Hsiung Tan [2] and Ching-Yun Tseng [3]

1 Department of Aeronautical Engineering, Chaoyang University of Technology, Taichung 413310, Taiwan
2 Department of Electrical and Electronic Engineering, Chung Cheng Institute of Technology, National Defense University, Taoyuan 335009, Taiwan; king731107@gmail.com (W.-Y.C.); s913115@gmail.com (K.-H.T.)
3 Mathematics and Physics Division, General Education Center, R.O.C. Air Force Academy, Kaohsiung City 820009, Taiwan; xxzz5801@gmail.com
* Correspondence: peterpan960326@gmail.com
† Presented at the 2024 IEEE 7th International Conference on Knowledge Innovation and Invention, Nagoya, Japan, 16–18 August 2024.

Abstract: Industry 5.0 aims to integrate humans and machines to achieve greater productivity, personalization, and sustainable development in the production process. Built on the foundation of Industry 4.0 which emphasizes automation, digitalization, and intelligent production processes, Industry 5.0 highlights the importance of human resources in modern manufacturing. Robotic arms have replaced traditional manpower, particularly in flexible manufacturing systems. However, integrating advanced machinery into workflows has increased competition in terms of securing resources, resulting in frequent deadlocks. Various deadlock prevention policies have been proposed to address this issue. Despite these efforts, resolving system deadlocks while achieving the optimal number of reachable states remains challenging. Based on existing research, we have developed a novel deadlock recovery method applicable to various flexible manufacturing systems. We designed an adaptable system and a controller that can restore the system to its fully operational state.

Keywords: Petri Net; flexible manufacturing systems; controller synthesis

Academic Editors: Teen-Hang Meen, Chun-Yen Chang and Cheng-Fu Yang

Published: 24 February 2025

Citation: Pan, Y.-L.; Chuang, W.-Y.; Tan, K.-H.; Tseng, C.-Y. Design of Maximally Permissive Controllers for Solving Deadlock Problems in Flexible Manufacturing Systems. *Eng. Proc.* **2025**, *89*, 10. https://doi.org/10.3390/engproc2025089010

1. Introduction

System deadlocks are caused repeatedly in flexible manufacturing systems (FMSs) [1,2], resource allocation systems (RASs) [3,4], and automated manufacturing systems (AMSs) [5–7], so solving problems related to these deadlocks is a significant and urgent task. The Petri Net (PN) theory is the one most widely utilized for dealing with deadlock problems. In this study, we adopted reachability graph analysis as our main technique, which with structural analysis are the two main analyzing techniques in PN theory. Our deadlock recovery strategy has demonstrated permissiveness and an optimal performance. Unlike the traditional deadlock prevention strategy, our deadlock recovery strategy improves the system liveness of deadlock states rather than forbidding them. Lu et al. [8] developed a resource flow graph (RFG)-based deadlock recovery method to mitigate resource competition. This method features a low computation cost and an intuitive application, with no need to calculate the whole reachability graph. Tseng et al. [9] improved the RFG-based method to reduce the number of controllers and maintain the same performance.

We have developed an optimal deadlock recovery method to solve system deadlock problems. The existing RFG-based methods [8,9] built transition-based controllers that release the resources from all operations of the RFG and return them to idle and resource places. However, this method requires a lot of reallocating of whole resources. We rearranged the partial resources in the RFG to maintain permissiveness and reachability. An example of a system of simple sequential processes with resources(S^3PR) net, modeled on the PN theory, was introduced for demonstration.

The rest of this article is laid out as follows: Section 2 provides an overview of the basic PN theory. Section 3 clarifies the central concept and the definition of an HRC. Section 4 introduces a classic PN model of S^3PR net to demonstrate the proposed method and compare it with existing works. The final section gives a summary and our conclusions.

2. Preliminaries

According to basic PN theory [10], the PN model of S^3PR net is a four-tuple $N = (P, T, F, W)$. A PN is illustrated as a directed bipartite graph, which is also called a Place/Transition net. P and T are two nonempty disjoint finite sets of places and transitions, respectively, where $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$. F denotes the set of arcs connecting places and transitions, such that $F \subseteq (P \times T) \cup (T \times P)$. $W : F \to \mathbb{N}^+$ is the weighted value of each arc in F. Each place of N is marked with one or more tokens at the current time to represent its property. A marking (or state) is a one-dimensional array mapping $M : P \to \mathbb{Z}$ and M_0 is the initial marking of N. $M(p)$ denotes the marking of any place $p \in P$ within M.

Identical to graph theory, $^\bullet x$ and $x^\bullet$ indicate the pre- and post-set of the node x of a PN model, no matter whether x is a place or transition. Given a transition $t \in T$ for firing at marking M, $\forall p \in {^\bullet t} : M(p) \geq W(p, t)$, which is also denoted as $M[t\rangle$. $\neg M[t\rangle$ means t is disabled for the firing at marking M. An S^3PR net contains idle, operation, and resource places, which are denoted as P^0, P_A, and P_R. Each sequential process of an S^3PR consists of one idle place and certain operation places, and different processes are connected by various shared resources.

Definition 1. *Let $N = \left(P^0 \cup P_A \cup P_R,\ T,\ F,\ W,\ M_0\right)$ be an S^3PR net which holds the following:*

(1) P_A is the set of operation places, $\forall p_a \in P_A$, $M_0(p_a) = 0$.

(2) P_R is the set of resource places, $\forall p_r \in P_R$, $^\bullet p_r \cap p_r^\bullet = \emptyset$; $\forall p_a \in P_A$, $t_i \in {^\bullet p_a}$, $t_j \in p_a^\bullet$, $\exists p_r \in P_R$, $^\bullet t_i \cap P_R = t_j^\bullet \cap P_R = \{p_r\}$; $\forall p_r \in P_R$, $^{\bullet\bullet} p_r \cap P_A = p_r^{\bullet\bullet} \cap P_A \neq \emptyset$.

(3) P^0 is the set of idle places and each sequential process contains one idle place p^0.

Using reachability graph analysis, all markings that were reachable from the initial one were found for further study. A reachability graph consists of nodes, which are labeled as markings, and arrows, labeled with transitions. $R(N, M_0)$ denotes the reachability graph of the PN model N with the initial marking M_0. $\mathfrak{M}$ indicates the set of all markings, including M_0. A reachable marking M can be reached by firing a sequence of transitions $\sigma = t_1 t_2 t_3 \ldots t_{n-1} t_n$, where $M_0[\sigma\rangle M$. A deadlock marking M_D does not reach any other ones due to all transitions being disabled. To develop an optimal deadlock recovery solution, we used a control transition instead of a control place to deal with the deadlock problem. A control transition is a generated transition that is enabled at certain deadlock markings.

3. Hold and Request Circuit-Based Deadlock Recovery Policy

In this study, we built an optimal deadlock recovery method to provide maximum permissiveness and system liveness. A structure named a hold and request circuit (HRC) was used to enhance efficiency.

3.1. HRC

In S^3PR systems, multiple production processes usually request shared resources to follow the steps of production. When any resource is held by one of the relevant processes while it waits for other resources, the others cannot obtain this resource again. System deadlocks caused by a lack of shared resources are undesirable and require full recovery. Via the specific subnet HRC, the flow between the resources and operation is checked, and then a set of well-designed control transitions is added to the system to reallocate those held resources.

An HRC contains operation and resource places, with connecting arrows between them used to present the flow relationships of the competing resources. As aforementioned, a deadlock occurs when resources are held by certain processes and these processes are waiting for other resources to carry out their next operation. When partial operation places and resource places are at a standstill and keep waiting, circular holding and waiting occur, which is one of the conditions of a deadlock. It is necessary to check all HRCs and build corresponding control transitions.

Definition 2. *Given an S^3PR net $N = \left(P^0 \cup P_A \cup P_R,\ T,\ F,\ W,\ M_0\right)$, a valid HRC $\mathbb{G}_{HRC}(\mathcal{V},\mathcal{U})$ holds:*

1. *(1) $\mathcal{V}$ is the set of nodes in the HRC, including operation places $\mathcal{V}_A \subseteq P_A$ and resource places $\mathcal{V}_R \subseteq P_R$, such that $\mathcal{V} = \mathcal{V}_A \cup \mathcal{V}_R$.*
2. *(2) $\mathcal{U}$ is the set of edges in the HRC, implying a flow of circular holding and waiting. There are two types of resource flow: requesting resources and releasing them from operation, where $\mathcal{U} = \{v_a \to v_r \mid \exists t \in T,\ (v_r,\ t) \cup (t,\ v_a) \subseteq F\} \cup \{v_a \to v_r \mid \exists t \in T,\ (v_r,\ t) \cup (v_a,\ t) \subseteq F\}$.*
3. *(3) Given an HRC with $\mathcal{V}_A = \bigcup_{i \in \mathbb{N}^+}^{n} v_{a_i}$ and $\mathcal{V}_R = \bigcup_{i \in \mathbb{N}^+}^{n} v_{r_i}$, $\forall v_a \in \mathcal{V}_A,\ v_r \in \mathcal{V}_R,\ {}^\bullet v_a \cup v_a^\bullet \subseteq \mathcal{V}_R,\ {}^\bullet v_r \cup v_r^\bullet \subseteq \mathcal{V}_A$.*
4. *(4) It is noteworthy that there is no repetition of any node in each HRC.*

3.2. Control Transition

Each valid HRC can form one corresponding control transition by reallocating resources. According to the concept of circular holding and waiting, the products and resources from idle and resource places are held in operation places. Therefore, control transitions are designed to solve these standstill conditions, release the products and the resources from operation places, and send them to idle places and resource places.

Definition 3. *Given an HRC $\mathbb{G}_{HRC}(\mathcal{V},\mathcal{U})$, its corresponding control transition t_c has input arcs from all $v_a \in \mathcal{V}_A$, output arcs into all of $v_r \in \mathcal{V}_R$, and an idle place $p^0 \in P^0$ belonging to the same process as v_a.*

3.3. Efficiency Enhancement

In the HRC-based deadlock recovery method, all valid HRCs of the system are found, and related control transitions are generated. All the operation places and resource places of the HRCs are considered within obtaining solutions, but they also bring the requirement of higher consumption for the rearrangement of resources. We assume that not all nodes of the HRCs are necessary for developing solutions. An S^3PR net contains two or more sequential processes with shared resources, with each process being a circuit of one idle place and one operation place. Each pair of operation places from the same HRC belongs to the same process, i.e., they share the same idle place p^0. In a deadlock marking, all operation places in the HRC cannot progress any further. Given two operation places v_a^1 and v_a^2 belong to the same HRC and share the same idle place p^0, the resources in v_a^1 and v_a^2

are not sent to the next operation place in this production circuit. As v_a^1 is closer to the end of this process than v_a^2, it releases its resources to partially recover the processing sequence. That is to say, in any sequential process, only one operation place needs to be considered when generating a control transition.

4. Experimental Results

A classic example of S^3PR net was used for the demonstration of our proposed HRC-based method.

Example 1. *Considering the PN model in Figure 1, there are 19 places and 14 transitions in total, where* $P^0 = \{p_1, p_8\}$, $P_A = \{p_2, p_3, p_4, p_5, p_6, p_7, p_9, p_{10}, p_{11}, p_{12}, p_{13}\}$, *and* $P_R = \{p_{14}, p_{15}, p_{16}, p_{17}, p_{18}, p_{19}\}$. *The initial marking* $M_0 = 6p_1 + 6p_8 + p_{14} + p_{15} + p_{16} + p_{17} + p_{18} + p_{19}$.

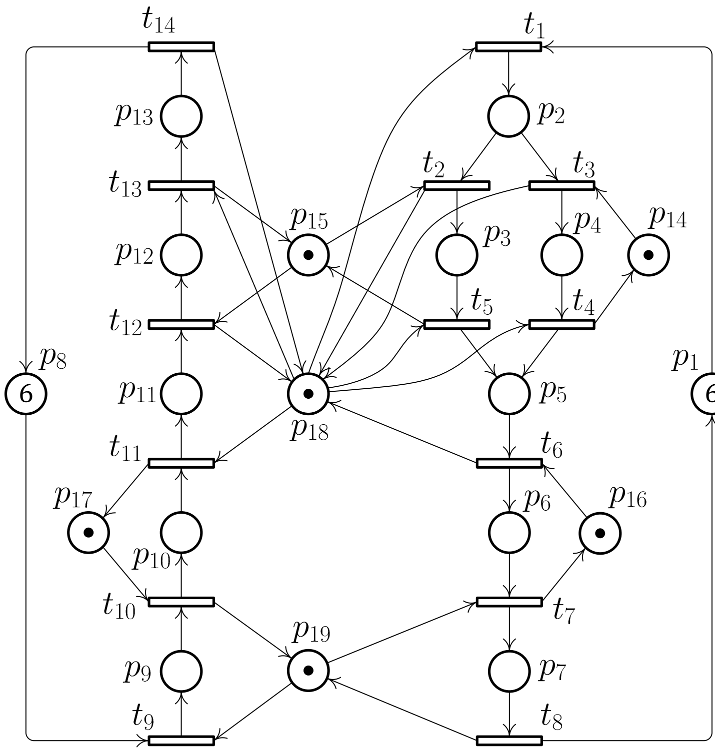

Figure 1. Example of S^3PR net.

According to HRC-based deadlock recovery policies [8,9], there are four HRCs in Example 1. Their contents are shown in Table 1.

Table 1. Contents of all HRCs.

HRC	$\mathcal{V}_A$	$\mathcal{V}_R$	$\mathcal{U}$
$\mathbb{G}_{HRC_1}$	p_2, p_4	p_{14}, p_{18}	$p_2 \to p_{14}, p_{14} \to p_4, p_4 \to p_{18}, p_{18} \to p_2$
$\mathbb{G}_{HRC_2}$	p_3, p_{11}	p_{15}, p_{18}	$p_{11} \to p_{15}, p_{15} \to p_3, p_3 \to p_{18}, p_{18} \to p_{11}$
$\mathbb{G}_{HRC_3}$	p_{11}, p_{12}	p_{15}, p_{18}	$p_{11} \to p_{15}, p_{15} \to p_{12}, p_{12} \to p_{18}, p_{18} \to p_{11}$
$\mathbb{G}_{HRC_4}$	p_5, p_6, p_9, p_{10}	$p_{16}, p_{17}, p_{18}, p_{19}$	$p_{18} \to p_5, p_5 \to p_{16}, p_{16} \to p_6, p_6 \to p_{19}$, $p_{19} \to p_9, p_9 \to p_{17}, p_{17} \to p_{10}, p_{10} \to p_{18}$

There are two sequential processes in the net in Figure 1, including $p_1t_1p_2t_2p_3t_5(t_3p_4t_4)$ $p_5t_6p_6t_7p_7t_8$ and $p_8t_9p_9t_{10}p_{10}t_{11}p_{11}t_{12}p_{12}t_{13}p_{13}t_{14}$. Only $\mathbb{G}_{HRC_4}$ has more than one operation place belonging to each sequential process; the other HRCs have one operation place each. According to Section III-C, $\mathbb{G}_{HRC_4}$ can be replaced by $\mathbb{G}^{\star}_{HRC_4} = (\mathcal{V}^{\star}_4, \mathcal{U}^{\star}_4)$ when $\mathcal{V}^{\star}_4 = \{p_6, p_{10}, p_{16}, p_{17}\}$. The newly generated control transitions are shown in Table 2.

Table 2. Generated control transitions.

t_{c_i}	$^{\bullet}t_{c_i}$	$t^{\bullet}_{c_i} \cap P_R$	$t^{\bullet}_{c_i} \cap P^0$
t_{c_1}	p_2, p_4	p_{14}, p_{18}	$2p_1$
t_{c_2}	p_3, p_{11}	p_{15}, p_{18}	p_1, p_8
t_{c_3}	p_{11}, p_{12}	p_{15}, p_{18}	$2p_8$
t_{c_4}	p_6, p_{10}	p_{16}, p_{17}	p_1, p_8

After four control transitions $t_{c_1}, t_{c_2}, t_{c_3}, t_{c_4}$ are added to the original net in Figure 1, the controlled net is checked with no any reachable deadlock marking by simulation software. The comparative results of the HRC-based methods are shown in Table 3, where the proposed method provides a better performance.

Table 3. Comparison of all RFG-based methods.

Methods	$\mid\mathfrak{M}\mid$	$\mid T_c \mid$	$\mid {}^{\bullet}T_c \cup T^{\bullet}_c \mid$	Liveness
[8]	282	6	42	Deadlock-free
[9]	282	4	30	Deadlock-free
Proposed	282	4	24	Deadlock-free

5. Conclusions

We developed an HRC-based deadlock recovery policy that led to an improvement in performance and efficiency. The experimental and comparative results showed that the proposed method is better than other HRC-based ones. In future research, it will be necessary to consider a more complicated system and enhance that system's performance.

Author Contributions: Conceptualization, Y.-L.P. and C.-Y.T.; methodology, W.-Y.C.; software, W.-Y.C. and K.-H.T.; validation, W.-Y.C. and C.-Y.T.; formal analysis, Y.-L.P. and W.-Y.C.; investigation, Y.-L.P.; resources, K.-H.T.; data curation, W.-Y.C.; writing—original draft preparation, W.-Y.C. and C.-Y.T.; writing—review and editing, Y.-L.P. and K.-H.T.; visualization, C.-Y.T.; supervision, Y.-L.P.; project administration, Y.-L.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Košťál, P.; Sobrino, D.R.D. Flexible Manufacturing System for Drawingless Manufacturing. *Key Eng. Mater.* **2013**, *581*, 527–532. [CrossRef]
2. Liu, D.; Li, Z.; Zhou, M. Hybrid Liveness-Enforcing Policy for Generalized Petri Net Models of Flexible Manufacturing Systems. *IEEE Trans. Syst. Man Cybern. Syst.* **2013**, *43*, 85–97. [CrossRef]
3. Lee, S.; Tilbury, D.M. Deadlock-Free Resource Allocation Control for a Reconfigurable Manufacturing System With Serial and Parallel Configuration. *IEEE Trans. Syst. Man Cybern. C* **2007**, *37*, 1373–1381. [CrossRef]

4. Reveliotis, S.; Fei, Z. Robust Deadlock Avoidance for Sequential Resource Allocation Systems With Resource Outages. *IEEE Trans. Automat. Sci. Eng.* **2017**, *14*, 1695–1711. [CrossRef]
5. Chen, C.; Hu, H. Extended Place-Invariant Control in Automated Manufacturing Systems Using Petri Nets. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 1807–1822. [CrossRef]
6. Alzalab, E.A.; El-Sherbeeny, A.M.; El-Meligy, M.A.; Rauf, H.T. Trust-Based Petri Net Model for Fault Detection and Treatment in Automated Manufacturing Systems. *IEEE Access* **2021**, *9*, 157997–158009. [CrossRef]
7. Feng, Y.; Xing, K.; Zhou, M.; Wang, X.; Liu, H. Robust Deadlock Prevention for Automated Manufacturing Systems With Unreliable Resources by Using General Petri Nets. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 3515–3527. [CrossRef]
8. Lu, Y.; Chen, Y.; Li, Z.; Wu, N. An Efficient Method of Deadlock Detection and Recovery for Flexible Manufacturing Systems by Resource Flow Graphs. *IEEE Trans. Automat. Sci. Eng.* **2022**, *19*, 1707–1718. [CrossRef]
9. Tseng, C.-Y.; Chen, J.-C.; Pan, Y.-L. An Improved Deadlock Recovery Policy of Flexible Manufacturing Systems Based on Resource Flow Graphs. *IEEE Access* **2024**, *12*, 65202–65212. [CrossRef]
10. Murata, T. Petri Nets: Properties, Analysis and Applications. *Proc. IEEE* **1989**, *77*, 541–580. [CrossRef]