Information Sciences 346–347 (2016) 17–28

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Robust deadlock control for automated manufacturing systems with an unreliable resource

Yunchao Wu, Keyi Xing*, Jianchao Luo, Yanxiang Feng

State Key Laboratory for Manufacturing Systems Engineering, and Systems Engineering Institute, Xi'an Jiaotong University, Xi'an 710049, China

ARTICLE INFO

Article history:
Received 7 September 2014
Revised 22 November 2015
Accepted 24 January 2016
Available online 11 February 2016

Keywords:
Automated manufacturing system (AMS)
Robust control
Deadlock
Petri net
Unreliable resource

ABSTRACT

So far, most of deadlock control policies are proposed based on the assumption that automated manufacturing systems (AMSs) have no unreliable resources. While in real manufacturing systems, resource failure is inevitable and will reduce the number of available resources. This paper focuses on the robust deadlock control problem for AMSs with an unreliable resource. Petri nets are used to model the unreliable systems, and a subclass of the ordinary and conservative Petri nets, known as system of simple sequential process with resources (S^3PR), is studied. A resource failure and recovery net is added to describe the resource failure and recovery. To prevent each siphon from being emptied, the concept of constraint set for a strict minimal siphon is introduced. By limiting the number of tokens in each constraint set, a robust deadlock controller is devised. It is proved that our controller can guarantee the liveness of the controlled system no matter one resource fails or not. Finally, some examples are provided to illustrate the validity of the proposed robust deadlock controller.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

Deadlock-free resource allocation has been an active research area of automated manufacturing systems (AMSs) for the several past decades. Researchers have made tremendous progress in computational, structural and control aspects of the deadlock-free resource allocation problem [1–5,7–9,13–15,17–20,22,23,25–27,29]. Three kinds of strategies, deadlock prevention, deadlock avoidance, and deadlock detection and recovery, are usually used to handle deadlock problems [17,29]. Deadlock prevention policies [2–5,8,14,18–20,22] usually use offline computational mechanisms to design a controller to ensure that deadlocks never occur. While in deadlock avoidance policies [26,27], a state is monitored online and resource is granted to process only if the resulting state is safe. In deadlock detection and recovery policies, however, resources are granted to a process without any check, the statuses of resource allocation and request are examined periodically to determine whether the system is in deadlocks, and if a deadlock is found, the system recovers from it by aborting one or more deadlocked processes.

Until now, most of deadlock control policies prevent deadlocks on condition that resources in considered systems are assumed not to fail, whereas this assumption is unrealistic for most real manufacturing systems. In case of resource failure, the existing deadlock control policies are always no longer in force and deadlocks may occur, then the system has to stop

* Corresponding author. Tel.: +86 13571897199.
E-mail addresses: wuyunchao@stu.xjtu.edu.cn (Y. Wu), kyxing@mail.xjtu.edu.cn (K. Xing), luo770826336@stu.xjtu.edu.cn (J. Luo), fengyxss@stu.xjtu.edu.cn (Y. Feng).

http://dx.doi.org/10.1016/j.ins.2016.01.049
0020-0255/© 2016 Elsevier Inc. All rights reserved.

and wait for the failed resource to be repaired, which will cause non-productive time, inefficiency and loss. Thus, it is necessary to develop an effective and robust deadlock controller to ensure that deadlocks cannot occur even if some resources break down.

Only a few researchers have studied on the robustness of deadlock control [6,10–12,16,21,24,28]. Hsieh analyzed the robustness property of several subclasses of Petri nets (PNs) with unreliable resources, including controlled production PNs [10], controlled assembly PNs [11], and non-ordinary PNs for flexible assembly/ disassembly processes [12]. Lawley and Sulistyono [16] proposed a robust supervisory control policy for the system with single unreliable resource. Neighborhood constraints and Banker's Algorithm are used to allocate system buffer spaces, so that when the unreliable resource fails, the system can continue to produce all part types not requiring it. Chew and Lawley [6] used the neighborhood constraints and Banker's Algorithm in [16] to a more general class of system with multiple unreliable resources, the new algorithm can make sure that if any subset of unreliable resources fails, the residual system can continue to produce all part types not requiring failed resources. Wang et al. [24] developed a robust supervisory control policy for single-unit resource allocation system with unreliable resources. For an AMS with one specified unreliable resource, Yue and Xing [28] modified Banker's Algorithm in [16] to achieve more reachable states. Liu et al. [21] added recovery subnets and controllers to the system of simple sequential process with resources (S^3PR) to deal with resource failures. The controlled net is live when no resource fails, and deadlocks are prevented even if some resources fail to work.

This paper focuses on the robust deadlock control problem in AMSs with an unreliable resource and only considers the case where the unreliable resource (machine or robot) can fail, but its buffer still can be assigned work-pieces. For S^3PRs with an unreliable resource, a robust deadlock controller is proposed. A resource failure and recovery net is added to describe the resource failure and recovery by PNs. The concept of constraint set of an strict minimal siphon (SMS) is defined. It is an extension of a complementary set of SMS, and contains all operation places using the unreliable resource in the SMS. If the number of tokens in a constraint set of an SMS is restricted, the SMS cannot be empty at any reachable markings even if one resource in it fails to work. Then based on constraint sets, related control places and arcs are added. Our controller has the following characteristics, which is defined as "1-robustness" in this paper:

(1) The controlled system is live in the absence of resource failure;
(2) The failure does not lead deadlock in the controlled system and during a resource failure, the controlled system can still process all types of parts infinitely;
(3) After the failed resource is repaired, it can be returned to the system to continue working without causing a deadlock.

Compared with [21], our controller is different in that: (1) Our controller contains no inhibitor arcs and the controlled PN is an ordinary PN; (2) There are some states in [21] that the controlled system has to wait for the failed resources to be repaired, which are called as "waiting-for-repair states"; while all waiting-for-repair states are prohibited in our controlled system.

The rest of this paper is organized as follows. Section 2 reviews the basics of PNs, S^3PR and its deadlock controller. Section 3 develops the PN model of AMSs with an unreliable resource and introduces the concept of 1-robust controller. A method for designing a 1-robust deadlock controller is proposed in Section 4. Section 5 gives some examples. Section 6 is a conclusion of this paper.

2. Petri net and S^3PR class

This section is a brief presentation of Petri nets and S^3PR class, and a design method of deadlock controller for S^3PR is also introduced. Readers may refer to [8,17,26,29] for more details.

2.1. Basics of Petri nets

A Petri net (PN) is a three-tuple $N = (P, T, F)$, where P is a finite set of places and T is a finite set of transitions. $F \subseteq (P \times T) \cup (T \times P)$ is the set of directed arcs.

Let $N = (P, T, F)$ be a PN. Given a node $x \in P \cup T$, the preset of x is defined as ${}^\bullet x = \{y \in P \cup T | (y, x) \in F\}$, and the postset of x is defined as $x^\bullet = \{y \in P \cup T | \ (x, y) \in F\}$. These notations can be extended to a set. For example, if $X \subseteq P \times T$, then ${}^\bullet X = \cup_{x \in X} {}^\bullet x$ and $X^\bullet = \cup_{x \in X} x^\bullet$. N is *pure* if $(x, y) \in F$ implies $(y, x) \notin F$. A *state machine* is a PN in which each transition has exactly one input and one output place.

Let S be a nonempty subset of places. S is a *siphon* (*trap*) if ${}^\bullet S \subseteq S^\bullet$ ($S^\bullet \subseteq {}^\bullet S$) holds. A siphon is said to be *minimal* if there is no siphon contained in it as a proper set. A minimal siphon is said to be *strict* if it does not contain a trap. A strict minimal siphon is denoted as *SMS* for short.

A marking or state of N is a mapping $M : P \to Z^+$, where Z^+ is the set of natural numbers. Given a place $p \in P$ and a marking M, $M(p)$ denotes the number of tokens in p at M. Let $B \subseteq P$ be a set of places, the sum of tokens in all places of B at M is denoted by $M(B)$, i.e., $M(B) = \sum_{p \in B} M(p)$. A PN N with an initial marking M_0 is called a *marked PN*, denoted as (N, M_0).

Given a marking M, a transition t is *enabled* at M if $\forall p \in {}^\bullet t, M(p) \geq 1$. This fact can be denoted by $M[t >$. We use $M[t > M'$ to denote that firing transition t and reaching a new marking M'. A marking M is said *reachable* from M_0 if there exists a sequence of transitions $\sigma = t_0 t_1 \ldots t_n$ and markings $M_1, M_2, \ldots, M_n (n \in Z^+)$ such that $M_0[t_0 > M_1[> t_1 > M_2 \ldots M_n[t_n > M$

Download English Version:

https://daneshyari.com/en/article/392410

Download Persian Version:

https://daneshyari.com/article/392410

Daneshyari.com