'91 KACC 1991. 10. 22~24

CHECKING LIVENESS IN PETRI NETS USING SYNCHRONIC VARIABLES

Inseon Koh and Frank DiCesare

Department of Electrical, Computer, and Systems Engineering and
Center for Manufacturing Productivity and Technology Transfer
Rensselaer Polytechnic Institute, Troy, New York 12181-3590

Abstract

In this paper we present how the deviation bound, which is a synchronic variable, can be used for checking liveness in Petri nets. Also, the deviation bound will be applied to detect or avoid deadlock situations and to characterize concurrency against sequential behaviors in automated manufacturing systems. In the current stage, we restrict the applicable domain of these methods to the Petri net structure that can be synthesized by combining common transitions or common places or common paths of Live-and-Bounded circuits.

1. Introduction

Petri net theory has been developed considerably since its origin by C. A. Petri in his 1962 Ph. D. dissertation on the study of the communication protocols between the components of a computer system. Since then Petri nets have become well known as a suitable tool for the modeling and analysis of Discrete Event Dynamic Systems.

Modeling a system using Petri nets has many advantages compared to other modeling schemes such as Markov chains, Queuing models, and general discrete event simulation languages. The graphic representation of Petri nets makes the models relatively simple and legible, and the well developed analysis methods such as invariant analysis and reachability tree can detect certain anomalies of system behaviors. Furthermore, the constructed models using Petri nets can be directly applied for realizing control system software. Because of the various advantages mentioned above, Petri nets have been widely used to model various domains such as circuit analysis, communication, computer systems, manufacturing systems, and knowledge representation.

Although Petri nets are supported by well developed mathematical theories, the analysis methods for Petri nets have some drawbacks. If the modeled system is large, we will be confronted by an inherent complexity problem. It is well known that a major weakness of Petri nets is the difficulty of analyzing large Petri nets. In other words, the set of all reachable markings of large Petri nets cannot be analyzed in practice. Reduction and synthesis methods have been suggested to cope with this problem. The details of the previous works related to reduction and synthesis methods of Petri nets are described in [11].

The concurrent flow of multiple sub-processes is one of the vital characteristics of automated manufacturing systems. Deadlock is a side issue if the sub-processes in automated manufacturing systems do not require shared information or shared resources during their operations. Usually many sub-processes compete for a finite number of shared resources. This often leads to a deadlock situation.

In an improperly designed automated manufacturing systems, the only remedy for deadlock may be manual clearing of buffers or machines, and restart of the system from an initial condition which is known to produce deadlock-free operation under nominal production conditions. Both the lost production and the labor cost in resetting the system in this way can be avoided by proper design [21]. The cooperation between sub-processes must be controlled to ensure correct operation of the overall system.

These problems can be solved by designing deadlock-free systems. Although each unit of the system functions normally, the system as a whole may have some critical synchronization errors (e.g., deadlocks). The trivial solution for avoiding such errors is to prohibit shared resources. However, from the point of view of optimizing resource utilization, this approach is not satisfactory. The approach in this paper is to develop systematic synthesis methods by calculating synchronic variables, which can measure the mutual dependency between two sub-processes, especially when the shared resources exist between sub-processes. If the interactions (dependency) among sub-processes are properly controlled, then deadlock situations can be avoided.

The synchronic concept was introduced by C. A. Petri [17] using synchronic distance for measuring dependency between the firing of transition subsets. The notion of synchronic distance was originated from the S-completion of condition/event systems [6]. According to Dr. Petri, all properties of a system should be able to be described in terms of synchronic distances. However, it is known that the case is true for some classes of Petri nets, but not for all classes of Petri nets.

Since then, many different definitions and notations have been used to investigate synchronic concepts. Four synchronic variables: Deviation Bound(DB), Fairness Bound(FB), Synchronic Lead(SL), and Synchronic Distance (SD) have been formulated.

Synchronic variables are rarely used for designing or analyzing real systems although some directions are shown in [6]. Some implementation of synchronic concepts has been accomplished, but more work is required. Kluge and Lautenbach [9] use the weighted synchronic distance to design a dynamic priority scheme for resolving multiple conflicts (memory access allocation) among competing channel processes in high-performance computer systems. This approach is different from other methods usually used in computer systems. It is neither probabilistic nor deterministic nor priority-oriented, but well defined and deadlock-free.

Synchronic variables can also be applied to detect or avoid deadlock situation and to characterize concurrency against sequential behavior in certain circumstances. In an automated manufacturing system, resources and information are shared among several processes. This sharing should be controlled or synchronized to insure the correct operation of the overall system.

Kanban, another name for the Just-in-Time policy is a way of introducing a synchronic distance between a pair of operations [18]. This policy was proposed in Japan to minimize in-process inventories. The synchronic values of two operations in a Just-in-Time policy should be calculated and controlled. Intuitively, we know that the control of synchronization is very important in the Just-in-Time policy.

Deadlocks in manufacturing systems have been studied in Petri nets without using synchronic variables. Banaszak and Krogh [1] developed a Petri net model of concurrent job flow and dynamic resource allocation in an FMS and presented deadlock avoidance algorithm by introducing the notion of a restriction policy, which is a feedback policy for excluding some enabled transitions from the current resource allocation alternatives. They provide three FMS examples for the illustration of their algorithm. They show that the conventional deadlock avoidance algorithm mainly used in computer operating systems are not efficient for manufacturing systems.

This paper describes the interactions between subprocesses, which frequently appear in manufacturing environments, using deviation bound. The interactions studied in the context of Petri nets are common places, common transitions, and common place and path(Transition-Transition-Path). Usually, the first represents "OR" operations, the second "AND" operations, and the last represents shared resource operations.

2. Synchronic Variables and LB-Circuit

In this section, we explain the basic concept of synchronic variables and introduce a Live-and-Bounded Circuit (LB-circuit) that can describe a primitive activity unit in manufacturing systems. We assume that the reader is familiar with basic Petri net theory. Because of many different notations used in literature, the definition of a Petri net is reviewed. For details, please refer to [14, 16].

A Petri net N is defined by a 5-tuple, N=(P, T, I, O, m_0), where:

(1) $P=\{p_1, p_2, ..., p_m\}$, a finite set of places, $m \geq 0$.
(2) $T=\{t_1, t_2, ..., t_n\}$, a finite set of transitions, $n \geq 0$, such that $P \cap T = \emptyset$.
(3) I: T --> $\mathbb{N}^m$ is an input function, $\mathbb{N}$ is a non-negative integer.
(4) O: T --> $\mathbb{N}^m$ is an output function.
(5) $m_0 \in \mathbb{N}^m$ is an initial marking.

Four synchronic variables mentioned before are originated from the synchronic distance. Therefore, we will explain the notion of the synchronic distance first, and give the formal definition of the deviation bound that is used for measuring liveness of systems in this paper. The concept of the synchronic distance is intuitively explained as follows.

The synchronic distance between $\{t_1, t_2\}$ and $\{t_3, t_4\}$ shown in Figure 1(a) is calculated by adding the place s shown in (b) such that ${}^{\bullet}s=\{t_1, t_2\}$, $s^{\bullet}=\{t_3, t_4\}$. Then the capacity of tokens of the place s at least should be 2, when $m_0(s)=0$. Because, whenever t_1 and t_2 fire, each transition

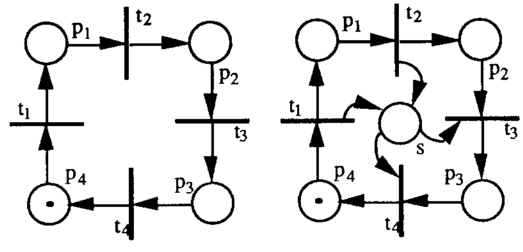

Fig.1 Synchronic distance (a) Original net (b) Modified net

puts a token into the place s, and by subsequent firing of t_3 and t_4, the two tokens in s are removed again, therefore we find the variance of the number of tokens of the place s will never exceed 2. This maximal variance of the number of tokens on s is called the synchronic distance between $\{t_1, t_2\}$ and $\{t_3, t_4\}$.

If we change the initial marking of Figure 1(a) such that $m_0(P)=[0, 0, 1, 0]^T$ and hold the initial marking of the place s, then the behavior of the original net is changed. The original net is live and bounded, but the modified net is no longer live. To be consistent with the previous case, we put one token in the place s. This does not affect the maximal variance of the number of tokens in s. The synchronic distance between $\{t_1, t_2\}$ and $\{t_3, t_4\}$ remains 2. The synchronic distance is considered the maximal variance of the number of tokens in the place s while maintaining the system behavior of the original net.

The synchronic distance between t_1 and t_2 for the choice node shown in Figure 2 can be calculated in the same manner as above.

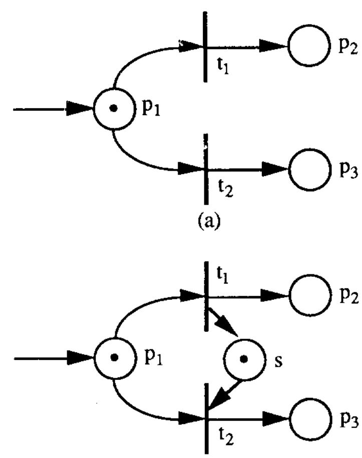

Fig. 2 Synchronic distance for a choice node (a) Original net (b) modified net

To preserve the system behaviors of the original net shown in Figure 2 (a), we should initially place a token in s shown in (b). The synchronic distance in this case is 1. If we assume that the above system is cyclic, then the synchronic distance is 2. Synchronic distance can be interpreted as the minimum token capacity of the place s while the system behaviors are not changed.

A synchronic distance of 1 means that two events can only occur alternately. A synchronic distance of 0 means "coincidence", meaning that the related input events and output events are not distinguishable from each other in time and space, and this is only possible if two sets of events are equal [6].

Deviation bound (DB) can be explained in the same way with the synchronic distance. The difference is that the second variable should not be fired in the case of the deviation bound. For example, the deviation bound between $\{t_1, t_2\}$ and $\{t_3, t_4\}$ shown in Figure 1 is the maximum number of tokens in the place s without firing the set of transitions $\{t_3, t_4\}$. In the following, the formal definition of Deviation bound(DB) is described from the literature [18].

Deviation Bound (DB)

$$DB(T_i, T_j)=\sup\{\bar{\sigma}(T_i) \mid \sigma\in L(N,M),\ M\in R(N, m_0),\ \bar{\sigma}(T_j)=0\}$$

where σ:A firing sequence; $\bar{\sigma}$: A characteristic vector (Firing count vector); $\bar{\sigma}(t_i)$: Number of occurrences of t_i in σ;

T_i : Subset of T; $\bar{\sigma}(T_i)= \sum_{t_j\in T_i} \bar{\sigma}(t_j)$;

$L(N, m_0)$: The set of all firing sequences from m_0.

We first introduce the concept of a Live-and-Bounded Circuit (LB-Circuit) in Petri nets and later formally define the LB-circuit. Let us consider the Petri net with two transitions t_1, t_2 and one place p_1 as shown in Figure 3. The transition t_1 produces five tokens and the transition t_2 consumes two tokens when these two transitions are fired once. We define an arc ratio as the integer part of this production/consumption ratio.

If the arc ratio is less than one and t_1 fires once, then the transition t_2 cannot be enabled. If the arc ratio is greater than two, then the transition t_2 can be fired more than two times (two-more enabled). If the arc ratio is exactly one, then the transition t_2 can be enabled just once. The remaining tokens in the place p_1 (the remainder of the arc ratio) cannot contribute to firing the transition t_2 when the transition t_1 is fired once. However, these remaining tokens are closely related to the boundedness of the net if the net is a circuit. We also know that the remaining tokens are not related to the liveness of the net if the net is a circuit.

The arc ratio r_{12} between two transitions t_1, t_2 and the place p_1 is calculated by

$$r_{12}=INT(O(p_1,t_1)/I(p_1,t_2)) \quad \text{if } I(p_1,t_2)\neq 0,$$
$$r_{12}=0 \quad \text{if } I(p_1,t_2)=0.$$

The remainder s_{12} between two transitions t_1, t_2 and the place p_1 is calculated by

$$s_{12}=MOD(O(p_1,t_1)/I(p_1,t_2)) \quad \text{if } O(p_1,t_1)/I(p_1,t_2)\geq 1,$$
$$s_{12}=0 \quad \text{if } O(p_1,t_1)/I(p_1,t_2)<1.$$

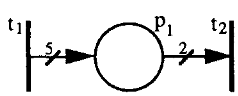

Fig. 3 An arc ratio r_{12}=2, and a remainder s_{12}=1.

If we serially add the place p_2 and the transition t_3 as shown in Figure 4, then the arc ratio r_{13} is calculated by

$$r_{13}=INT(r_{12}\times O(p_2,t_2)/I(p_2,t_3)) \quad \text{if } I(p_2,t_3)\neq 0,$$
$$r_{13}=0 \quad \text{if } I(p_2,t_3)=0.$$

Also, the remainder s_{13} is calculated by

$$s_{13}=s_{12}+MOD(r_{12}\times O(p_2,t_2)/I(p_2,t_3)) \quad \text{if } r_{12}\times O(p_2,t_2)/I(p_2,t_3)\geq 1,$$
$$s_{13}=s_{12}+r_{12}\times O(p_2,t_2) \quad \text{if } r_{12}\times O(p_2,t_2)/I(p_2,t_3)<1.$$

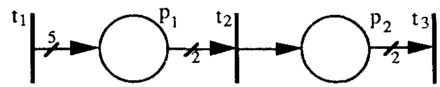

Fig. 4 An arc ratio r_{13}=1, and a remainder s_{13}=1.

In the same way mentioned above, we can recursively calculate an arc ratio r_{1n} and a remainder s_{1n} as follows.

Arc Ratio and Remainder

The arc ratio r_{1n} and the remainder s_{1n} of an alternating sequence of transitions and places, $t_1p_1t_2p_2 \ldots t_ip_i \ldots p_{n-1}t_n$ (the first node and the last node should be transitions) are defined as follows.

$$r_{1n}=INT(r_{1n-1}\times O(p_{n-1},t_{n-1})/I(p_{n-1},t_n)) \text{ if } I(p_{n-1},t_n)) \neq 0,$$
$$r_{1n}=0 \text{ if } I(p_{n-1},t_n)) = 0.$$
$$s_{1n}= \sum_{i=1}^{n-1} s_{1i}+MOD(r_{1n-1}\times O(p_{n-1},t_{n-1})/I(p_{n-1},t_n))$$
$$\text{if } r_{1n-1}\times O(p_{n-1},t_{n-1})/I(p_{n-1},t_n)\geq 1,$$
$$s_{1n}= \sum_{i=1}^{n-1} s_{1i} +r_{1n-1}\times O(p_{n-1},t_{n-1})$$
$$\text{if } r_{1n-1}\times O(p_{n-1},t_{n-1})/I(p_{n-1},t_n)<1.$$

Using the arc ratio and remainder, we define an LB-circuit as follows.

LB-Circuit

An LB-circuit is a directed path $t_1p_1t_2p_2 \ldots t_ip_i \ldots t_{n+1}$, such that

(1) $t_1=t_{n+1}$.
(2) $r_{1n+1}=1$, $s_{1n+1}=0$.
(3) the circuit is live.

An LB-circuit represents a primitive activity unit of the overall system. The complex behaviors of entire systems can be described by combining LB-circuits fused along common transitions or places or paths. In the next section, we present how the interaction between sub-systems are described by deviation bounds. Our methods concentrate on preserving liveness and boundedness of systems, which are the minimum necessary conditions for automated manufacturing systems.

3. Measuring Liveness Using Deviation Bound

In this section we present a theorem which can measure the liveness level of the combined net if the liveness-level of each of the sub-nets to be fused by common places or common Place-Place-Path is known. Because of the lack of space, the theorem is presented without proof. We also show that the theorem can be used for synthesizing sub-Petri nets. Then, we describe the interactions between sub-Petri nets fused by common transitions using deviation bounds. Finally, shared resource problems are studied in terms of Petri nets.

Case I: Common Places and Common Place-Place-Path

Theorem 1.
The liveness level of the combined net $N(N_1+N_2)$ is $Min\{Level(N_1), Level(N_2)\}$, if $Min\{DB(T_1, T_2), DB(T_2, T_1)\}=\infty$.

The synthesis cases that can satisfy the condition of Theorem 1 are illustrated in Figure 5.

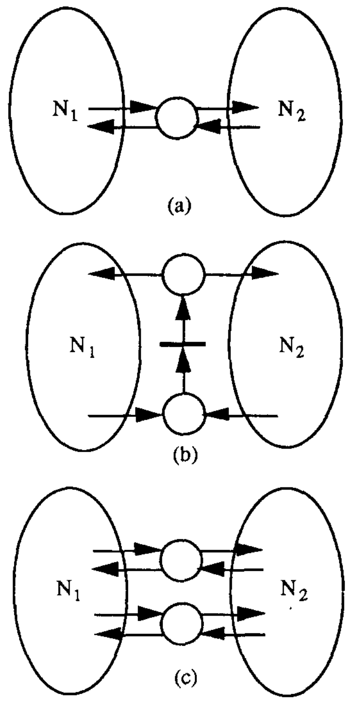

Fig. 5 Combined nets fused by common Place-Place-Paths and common places

Figure 5 (a) and (b) show that the combined nets are fused by common Place-Place-Path. In Figure 5 (c), the combined net is fused by common places. This can be used for the shared resource problems in manufacturing systems. For example, if two sub-systems are live and bounded and the synchronic distance between sub-processes is infinite, then the whole system can be live and bounded using the above synthesizing methods.

But, as seen in Figure 5, the combine nets are fused by common places or a common Place-Place-Path to satisfy the condition $Min\{DB(T_1, T_2), DB(T_2, T_1)\}=\infty$. Unfortunately, the deviation bound is not a precise measure for checking all methods of combining nets. We consider the case of common transitions in the followings.

Case II: Common Transitions

As a simple example, if the combined net is fused by common transitions, the above condition can no longer be satisfied as seen in Figure 6.

The next topic is to handle this case in which the combined net is fused by common transitions. Let us consider the more detailed example shown in Figure 7. Two circuits (LB-circuits) shown in (a) and (b) are live and bounded. If these kinds of circuits are fused by the common

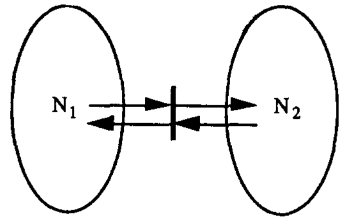

Fig. 6 A combined net fused by a common transition

transition t_c, then the liveness of the combined net shown in (c) may be changed. The question arises as to how this change(liveness) can be described by deviation bounds.

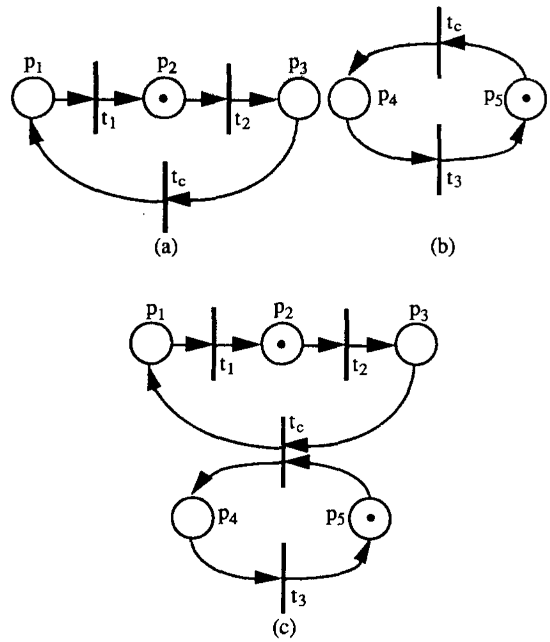

Fig. 7 A combined net fused along a common transition

The deviation bounds of the nets in Figure 7 are calculated as follows. The subscript "pre" denotes before-fusion and the subscript "post" denotes after-fusion. Here T_i is the set of transitions for the sub-Petri net i.

$DB(T_1, T_2)_{pre}=\infty$ $\quad$ $DB(T_1, T_2)_{post}=1$
$DB(T_2, T_1)_{pre}=\infty$ $\quad$ $DB(T_2, T_1)_{post}=0$

This is intuitively true, because the DB of the combined net in this case cannot exceed some positive number k. Actually $DB(T_1, T_2)_{post}$ and $DB(T_2, T_1)_{post}$ depend on the initial marking of each LB-circuit. From minor changes of the initial marking such as $m_0(p_1)=0$, $m_0(p_2)=0$, $m_0(p_3)=1$, $m_0(p_4)=1$, $m_0(p_5)=0$, the deviation bounds change as follows.

$DB(T_1, T_2)_{post}=0$ $\quad$ $DB(T_2, T_1)_{post}=1$

As mentioned before, in this case, even if the deviation bound of two LB-circuits is not infinite, the combined net is still live.

Let us assume that the t_c shown in Figure 7 (c) can be enabled when it is to be fired, and calculate the deviation bound between $\{T_1\text{-}t_c\}$ and $\{T_2\text{-}t_c\}$.

$DB(\{T_1\text{-}t_c\}, \{T_2\text{-}t_c\})=DB(\{t_1, t_2\}, \{t_3\})=\infty$
$DB(\{T_2\text{-}t_c\}, \{T_1\text{-}t_c\})=DB(\{t_3\}, \{t_1, t_2\})=\infty$

These results say that the liveness of the combined net fused by a common transition is preserved if the deviation bound between two sets of transitions excluding the common transition is infinite. The enabling condition previously stated can be easily satisfied if the nets to be fused are restricted to LB-circuits. This can be used for synthesizing Petri nets, in which the combined net is fused by a common transition if we can identify the enabling condition. As a matter of fact, this is not in conflict with Theorem 1.

The results can be extended to multiple common transitions as shown in Figure 8 .

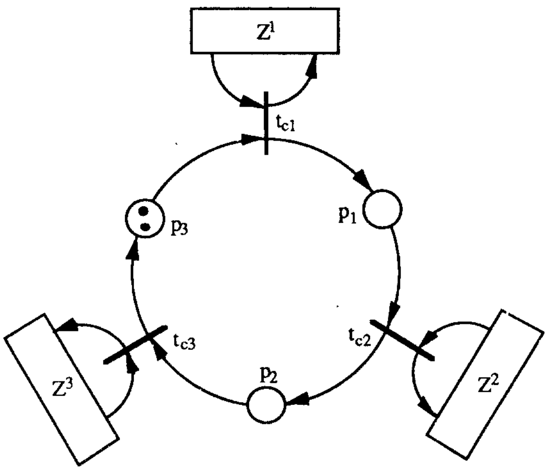

Fig. 8 A combined net fused by common transitions

The system consists of three sub-systems described by Z_1, Z_2, and Z_3. Three sub-systems must be coordinated with each other, and the coordination is described by an LB-circuit fused by three common transitions of sub-systems. We assume that the sub-systems are live and bounded. Theorem 1 can not be applied to this case, because this is not a common Place-Place-Path. Let us calculate the deviation bound. We know that the three common transitions can be enabled when those are to be fired.

$DB(\{T_1\text{-}t_{c1}\}, \{t_{c2}, t_{c3}\})=\infty \quad DB(\{t_{c2}, t_{c3}\}, \{T_1\text{-}t_{c1}\})=\infty$

From this, we know that the combined net of Z_1 and LB-circuit is live. In the same way as above,

$DB(\{T_2\text{-}t_{c2}\}, \{t_{c1}, t_{c3}\})=\infty \quad DB(\{t_{c1}, t_{c3}\}, \{T_1\text{-}t_{c2}\})=\infty$
$DB(\{T_3\text{-}t_{c3}\}, \{t_{c1}, t_{c2}\})=\infty \quad DB(\{t_{c1}, t_{c2}\}, \{T_3\text{-}t_{c3}\})=\infty$

we can conclude that the whole net is live and bounded.

If we keep the structure of the net, it can be generalized to n common transitions. The physical meaning of this system, especially for manufacturing systems, is that the LB-circuit distributes the shared resources to sub-systems while keeping some kind of priority scheme.

Case III: Shared Resources (Common Place and Transition-Transition-Path)

Deadlock has been widely studied in the context of operating systems. In automated manufacturing environments, the deadlock situation should be avoided to process sub-activities smoothly. Generally, deadlocks occur in handling shared resources in manufacturing systems. The general shared resource problems in manufacturing systems can be interpreted in Petri nets as shown in Figure 9.

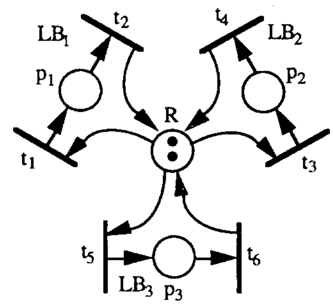

Fig. 9 The general shared resource problem in Petri nets

The system consists of three LB-circuits that are fused by the common place R. Three LB-circuits are $LB_1(t_1p_1t_2R)$, $LB_2(t_3p_2t_4R)$, and $LB_3(t_5p_3t_6R)$. The place R represents shared resources. For example, it may represent the number of robots available. We consider only the case when place R has an initial marking (tokens) less than the number of combined LB-circuits. If the place R has three tokens, then R has enough tokens to process three sub-processes independently. Deadlocks cannot occur in that situation.

The path $t_1p_1t_2$ shown in Figure 9 represents a sub-process requiring the shared resource in R. In the same way, the paths $t_3p_2t_4$ and $t_5p_3t_6$ represent other sub-processes. The system itself is live and bounded, ie., there is no deadlock and the system is stable based on the theory of the common Place-Place-Path [11].

However, the possibility of deadlock could increase if two sub-processes, for example, the sub-processes $t_1p_1t_2$ and $t_3p_2t_4$, interact together in another sub-system. One purpose of this paper is to describe the interaction using the deviation bound. Thus we can analyze the system's liveness.

For simplicity, just consider two sub-processes interacting with each other in another sub-system S_1 as shown in Figure 10. Two sub-processes, the paths $t_1p_2t_2$ and $t_3p_4t_4$, have interaction in the sub-system S_1 that is described by the LB-circuit $(t_1p_2t_2p_3t_3p_4t_4p_1t_1)$. The place R is a shared resource place. We assume that the initial condition of the net is $[2, 0, 0, 0, 1]^T$ as shown in Figure 10 (a). From the structure of the net, we know the activity of the sub-process $t_1p_2t_2$ precedes the activity of the sub-process $t_3p_4t_4$.

We observe that if the sub-process $t_1p_2t_2$ can put the same number of tokens as the place p_1 into place p_3 without activating the sub-process $t_3p_4t_4$, and if the sub-process $t_3p_4t_4$ can produce the same number of tokens as p_3 to the place p_1 without activating the sub-process $t_1p_2t_2$, then the system is live and bounded.

In the case of (b), the sub-process $t_1p_2t_2$ can produce only one token in the place p_2 without activating the sub-process $t_3p_4t_4$, because of the place p_5. The system is bounded, not live. If the place p_5 has two tokens initially, then it can produce two tokens into the place p_3. Therefore, the system is live. If the places p_1 and p_5 have one token each, then the net is still live. These conditions can be described by the deviation bound as follows.

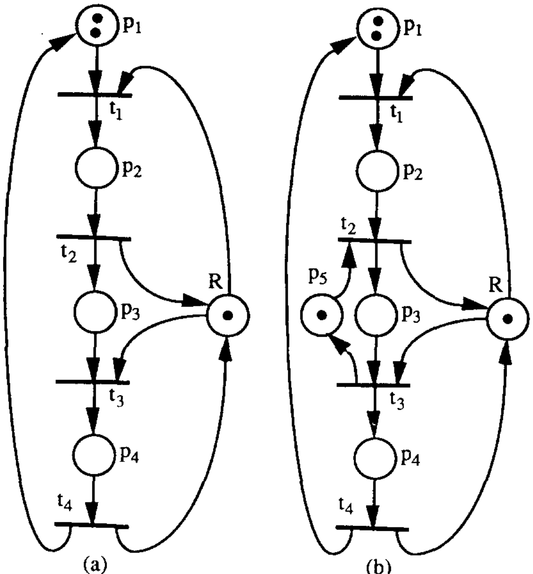

Fig. 10 Two sub-processes interacting with each other

The system shown in Figure 10 is live and bounded if
$m_0(p_1) \le DB(\{t_1, t_2\}, \{t_3, t_4\})$.

Two set of transitions, $\{t_1, t_2\}$ and $\{t_3, t_4\}$, are the sub-processes interacting with each other. This result can be directly applied to generalized Petri nets.

4. Conclusions

This paper shows the usefulness of synchronic variables for analyzing and designing discrete event systems. The interactions between sub-systems, which frequently appear in manufacturing environments, are analyzed using deviation bounds. In this paper, these interactions are described by common places, transitions, paths, and place plus path. We did not formalize all the results. The formal presentation will appear in our next paper, and their application to colored Petri nets is a topic for future research.

Acknowledgements

The authors acknowledge the research support by the industrially sponsored Computer Integrated Manufacturing program of the Center for Manufacturing Productivity and Technology Transfer at Rensselaer.

References

[1] Z. A. Banaszak and B. H. Krogh, "Deadlock Avoidance in Flexible Manufacturing Systems with Concurrency Competing Process Flows," *IEEE Trans. Robotics Automat.*, vol. 6, no. 6, pp. 724-734, 1990.

[2] G. Berthelot, "Transformations and Decompositions of Nets," *LNCS: Advances in Petri Nets 1986*, vol. 254, part 1, pp. 359-376, 1986.

[3] A. N. Choudhary, W. H. Kohler, J. A. Stankovic and D. Towsley, "A Modified Priority Based Probe Algorithm for Distributed Deadlock Detection and Resolution," *IEEE Trans. Soft. Eng.*, vol. 15, no. 1, pp. 10-17, 1989.

[4] A. Datta, D. Harms and S. Ghosh, "Deadlock Avoidance in Real Time Resource Sharing Distributed Systems: An approach using Petri nets," *Proc. IEEE Real-Time Syst. Symp.*, pp. 49-61, 1984.

[5] A. A. Desrochers, *Modeling and Control of Automated Manufacturing Systems.* IEEE Computer Society Press, Washington, 1990.

[6] U. Goltz and Y. Chong-Yi, "Synchronic Structure," *LNCS: Advances in Petri Nets*, pp. 233-252, 1985.

[7] U. Goltz, "Synchronic Distance," *LNCS*, part 1, pp. 338-358, 1986.

[8] R. T. Jacob and I. P. Page, "Synthesis of Mutual Exclusion Solutions Based on Binary Semaphores," *IEEE Trans. Soft. Eng.*, vol. 15, no. 5, pp. 560-568, 1989.

[9] W. E. Kludge and K. Lautenbach, "The Orderly resolution of Memory Access Conflicts among Competing Channel Processes," *IEEE Trans. Comput.*, vol. C-31, no. 3, pp. 194-207, 1982.

[10] I. Koh and F. DiCesare, "Transformation Methods for Generalized Petri nets and Their Applications in Flexible Manufacturing Systems," *Proc. 2nd. Int. Conf. Comput. Integrated Manufact.*, pp 364-371, 1990.

[11] I. Koh and F. DiCesare, "Modular Transformation Methods for Generalized Petri nets and Their Applications in Automated Manufacturing Systems," to appear in *IEEE Trans. Syst., Man, and Cybernetics*, vol. 21, no. 6, 1991.

[12] I. Koh and F. DiCesare, "Synthesis Rules for Colored Petri nets and Their Applications in Automated Manufacturing Systems," *Proc. 6th IEEE Int. Symp. Intelligent Control*, pp. 152-157, 1991.

[13] T. Murata and J. Wu, "Some Considerations on Synchronic Distances in a Petri Net," *IEEE Int. Symp. Circuit Syst.*, pp. 483-486, 1985.

[14] T. Murata, "Petri Nets: Properties, Analysis and Application," *Proc. IEEE*, vol. 77, no.4, pp. 541-579, 1989.

[15] T. Murata, B. Shenker and S. M. Shatz, "Detection of Ada Static Deadlocks Using Petri Net Invariants," *IEEE Trans. Soft. Eng.*, vol. 15, no. 3, pp. 314-325, 1989.

[16] J. L. Peterson, *Petri Net Theory and the Modeling of Systems.* Prentice Inc., 1981.

[17] C. A. Petri, "Interpretations of Net Theory," St. Augustin: Gesellschft fur Mathematik und Datenverarbeitung Bonn, Interner Berich ISF-75-07, Second Edition, 1976.

[18] M. Silva, "Towards a Synchrony Theory for P/T Nets," in *Concurrency and Nets.* pp. 435-460, 1987.

[19] M. Singhal, "Deadlock Detection in Distributed Systems," *IEEE Comput.*, pp. 37-48, 1989.

[20] N. Viswanadham, T. L. Johnson, and Y. Narahari, "Performance Analysis of Automated Manufacturing Systems with Blocking and Deadlocks," *IEEE*, pp. 64-68, 1990.

[21] N. Viswanadham, Y. Narahari, and T. L. Johnson, "Deadlock Prevention and Deadlock Avoidance in Flexible Manufacturing Systems Using Petri Net Models, "*IEEE Trans. Robotics Automat.*, vol. 6, no. 6, pp. 713-723, 1990.

[22] B. E. Wojcik and Z. M. Wojcik, "Sufficient Condition for a Communication Deadlock and Distributed Deadlock Detection," *IEEE Trans. Soft. Eng.*, vol. 15, no. 12, pp. 1587-1595, 1989.

[23] M. C. Zhou and F. DiCesare, "Parallel and Sequential Mutual Exclusion for Petri Net Modeling of Manufacturing Systems with Shared Resources," *IEEE Trans.. Robotics Automat.*, vol. 7, no. 4, pp. 515-527, 1991.