REFERENCES

[1] D. M. Titterington, A. F. Smith, and U. E. Makov, *Statistical Analysis of Finite Mixture Distributions.* New York: Wiley, 1985.

[2] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Rev.*, vol. 26, pp. 195–239, Apr. 1984.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc. B,* vol. 39, pp. 1–38, 1977.

[4] N. Vlassis, G. Papakonstantinou, and P. Tsanakas, "Mixture density estimation based on maximum likelihood and test statistics," *Neural Process. Lett.,* vol. 9, pp. 63–76, Feb. 1999.

[5] D. F. Specht, "Probabilistic neural networks," *Neural Networks,* vol. 3, pp. 109–118, 1990.

[6] E. Parzen, "On the estimation of a probability density function and mode," *Ann. Math. Statist.,* vol. 33, pp. 1065–1076, 1962.

[7] H. G. C. Tråvén, "A neural network approach to statistical pattern classification by 'semiparametric' estimation of probability density functions," *IEEE Trans. Neural Networks,* vol. 2, pp. 366–377, May 1991.

[8] R. L. Streit and T. E. Luginbuhl, "Maximum likelihood training of probabilistic neural networks," *IEEE Trans. Neural Networks,* vol. 5, no. 5, pp. 764–783, 1994.

[9] S. Shimoji, "Self-organizing neural networks based on Gaussian mixture model for PDF estimation and pattern classification," Ph.D. dissertation, Univ. Southern California, Los Angeles, 1994.

[10] N. Vlassis, A. Dimopoulos, and G. Papakonstantinou, "The probabilistic growing cell structures algorithm," in *Proc. ICANN'97, Int. Conf. Artificial Neural Networks,* Lausanne, Switzerland, Oct. 1997, pp. 649–654.

[11] B. D. Ripley, *Pattern Recognition and Neural Networks.* Cambridge, U.K.: Cambridge Univ. Press, 1996.

[12] G. J. McLachlan, "On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture," *Appl. Statist.,* vol. 36, pp. 318–324, 1987.

[13] W. D. Furman and B. G. Lindsay, "Testing for the number of components in a mixture of normal distributions using moment estimators," *Comput. Statist. Data Anal.,* vol. 17, pp. 473–492, 1994.

[14] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions.* New York: Wiley, 1997.

[15] W. H. Press, S. A. Teukolsky, B. P. Flannery, and W. T. Vetterling, *Numerical Recipes in C,* 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1992.

[16] S. Ingrassia, "A comparison between the simulated annealing and the EM algorithms in normal mixture decompositions," *Stat. Comput.,* vol. 2, pp. 203–211, 1992.

Analysis of Modularly Composed Nets by Siphons

MuDer Jeng and Xiaolan Xie

***Abstract*—This paper uses siphons to analyze the class of Petri nets constructed by a modular approach in [5] for modeling manufacturing systems with shared resources. A resource point of view is taken. First the behavior of each resource is modeled using resource control nets, strongly connected state machines with one place being marked initially. Interactions among the resources are modeled through merging of common transition subnets. This paper provides conditions, expressed in terms of siphons, under which reversibility and liveness of the integrated model are obtained. Relations between siphons and circular-wait are formally established. Superiority of the siphon-based analysis over a previous analysis using circular wait is shown.**

***Index Terms*—Analysis, Petri nets, synthesis.**

I. INTRODUCTION

Modular approach is an efficient way to cope with the complexity in modeling a large-scale system. It consists in decomposing it into simple subsystems called modules, modeling each module and integrating the module models together to obtain the model of the whole system.

A major concern, when modeling a real-life system, is to check whether the Petri net model has desired qualitative properties such as liveness, boundedness, and reversibility. As long as manufacturing systems are concerned, the liveness ensures that blocking will never occur, the boundedness guarantees that the number of in-process parts is bounded, the reversibility enables the system to come back to its initial state from whatever state it reaches.

Due to the complexity of real-life systems, classical property checking methods such as coverability tree, invariant analysis and algebraic analysis (see [10]) hardly apply. There are two classes of methods for analyzing a large Petri net model. The first one is the reduction of Petri nets while preserving properties. Reduction rules have been proposed [2], [9]. The main disadvantage of this approach lies in the difficulty of finding reducible subnets.

The second class of methods includes synthesis methods which build the models systematically and progressively such that the desired properties are preserved all along the design process. Two synthesis approaches: top-down approach and bottom-up approach, have been proposed.

The top-down approach begins with an aggregate model of the system which is refined progressively to introduce more and more details. The basic refinement is the substitution of a place or a transition by a so-called well-formed block [12], [13], [15]. Conditions, under which the desired properties are preserved, are given. This approach is well suited to model systems composed of almost independent sub-systems. However, this approach loses its efficiency in case of strongly coupled sub-systems since it is impossible to find small aggregate models.

The bottom-up approach [1], [4], [5], [7], [8], [11], [14] starts from sub-system models and integrate them by merging some places and/or transitions. The disadvantage of the general bottom-up approach lies

Manuscript received June 26, 1998; revised April 1, 1999.

M. Jeng is with the Department of Electrical Engineering, National Taiwan Ocean University, Keelung, Taiwan, R.O.C. (e-mail: b0162@ind.ntou.edu.tw).

X. Xie is with INRIA/MACSI Team, ENIM-Ile du Saulcy, 57045 Metz, France (e-mail: xie@loria.fr).

Publisher Item Identifier S 1083-4427(99)05343-6.

mainly in the extreme difficulty of checking whether an integration preserves the desired properties. Modular modeling approaches can be considered as bottom-up approaches.

In this paper, we adopt the modular approach proposed in [5]. It is a resource-oriented approach and contains three steps. The first step consists in modeling the dynamic behavior of each resource using Resource Control Nets. An RCN is a strongly connected state machine with one place being marked initially. It is bounded, live and reversible. The second step consists in modeling the interactions between various resources by merging common transitions and common transition subnets. The third step checks whether the properties of the module models are preserved. Contributions of this paper concern step 3, the most difficult step. Siphons, structural properties of the net, are used to characterize the reversibility and the liveness of the integrated model. In particular, it is shown that the integrated model is reversible iff no siphon in it can eventually become empty. The latter condition of large resource control systems can be checked using various mathematical programming techniques proposed in [3]. Results of this paper are compared with those based on circular-wait proposed in [5].

This paper is organized as follows. Section II presents basic notions of Petri nets used in this paper. Section III introduces RCN's and RCN-merged nets. Section IV is devoted to the analysis of RCN-merged nets using siphons. Section V establishes relations between siphons and circular-wait. Section VI presents a manufacturing example.

II. Basic Notions of Nets

Consider an ordinary Petri net $G = (P, T, F, M_0)$ where P is the set of *places*, T is the set of *transitions*, $F \subseteq (P \times T) \cup (T \times P)$ is the set of *directed arcs*, and $M_0 : P \rightarrow \mathcal{N}$ is the *initial marking*, where $\mathcal{N}$ is the set of nonnegative integers. The set of input (resp. output) transitions of a place p is denoted by $\bullet p$ (resp. $p\bullet$). Similarly the set of input (resp. output) places of a transition t is denoted by $\bullet t$ (resp. $t\bullet$). For any subset of places S, $\bullet S$ (resp. $S\bullet$) denotes the set of transitions with at least one output (resp. input) place belonging to S.

A transition t is *enabled* and can be *fired* under a marking M iff $M(p) \geq 1, \forall p \in \bullet t$. The firing results in removing one token from each of its input places and adding one token to each of its output places. A marking M' is *reachable* from M iff there exists a firing sequence σ of transitions from M to M'. This gives $M' = M + C\bar{\sigma}$, where $C = [c_{ij}]$ is called the *incidence matrix* such that $c_{ij} = 1$ if $t_j \in \bullet p_i \backslash p_i \bullet$, $c_{ij} = -1$ if $t_j \in p_i \bullet \backslash \bullet\ p_i$, and $c_{ij} = 0$, otherwise; and $\bar{\sigma}$, called the *firing count vector*, is a vector whose i-th entry denotes the number of occurrences of t_i in σ. The *reachability set* $R(M)$ denotes all the markings that are reachable from M.

A transition t is said to be *live* if for any $M \in R(M_0)$, there exists a sequence of transitions firable from M that contains t. A Petri net is said to be live if all the transitions are live. A Petri net is said to be *deadlock-free* if at least one transition is enabled at every reachable marking. A place p is said to be *bounded* if there exists a constant K such that $M(p) \leq K$ for all $M \in R(M_0)$. A Petri net is said to be bounded if all the places are bounded. It is said to be *structurally bounded* if it is bounded whatever the initial marking is. A Petri net is said to be *reversible* if, for any $M \in R(M_0)$, M_0 is reachable from M.

State machines are Petri nets such that $\forall t \in T : |t\bullet| = |\bullet t| = 1$. *Marked graphs* are Petri nets such that $\forall p \in P : |p\bullet| = |\bullet p| = 1$. A *state machine component* $G' = (P', T', F', M'_0)$ of a Petri net G is a state machine and is a subnet of G consisting of places in P', their input and output transitions, and the related arcs. A Petri net is said *state machine decomposable* if it is covered by state machine components.

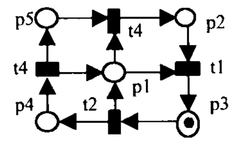

Fig. 1. Petri net.

A subset of places S is called a *siphon* if $\bullet S \subseteq S\bullet$, i.e., any input transition of S is also an output transition of S. It is called a *trap* if $\bullet S \supseteq S\bullet$. A siphon is *minimal* if it does not contain other siphons. The net of Fig. 1 has four siphons: $\{p_2, p_3, p_4, p_5\}$, $\{p_1, p_3, p_4\}$, $\{p_1, p_3, p_4, p_5\}$, $\{p_1, p_2, p_3, p_4, p_5\}$ and 4 traps: $\{p_2, p_3, p_4, p_5\}$, $\{p_1, p_2, p_3\}$, $\{p_1, p_2, p_3, p_4\}$, $\{p_1, p_2, p_3, p_4, p_5\}$. $\{p_2, p_3, p_4, p_5\}$ and $\{p_1, p_3, p_4\}$ are minimal siphons.

The following property (see [10]) shows the importance of siphons and traps in the detection of deadlocks.

Property 1: A siphon free of tokens at a marking remains token-free whatever the transition firings; A trap marked by a marking remains marked; For any marking such that no transition is enabled, the set of empty places forms a siphon.

In the following, a marking M such that no transition is enabled is called *dead marking*. A siphon S that eventually becomes empty is called *potential deadlock*. From Property 1 and from the definition of minimal siphons,

Property 2: A Petri net is deadlock-free if no minimal siphon eventually becomes empty.

Condition of Property 2 holds if every siphon contains a trap marked by M_0, i.e., the *Commoner condition* holds. Unfortunately, it does not hold for most Petri net models of many systems with shared resources. Deadlock-freeness of Petri net models of systems with shared resources has been extensively studied in [3]. Results needed in this paper can be summarized as follows.

Property 3: A siphon S can never become empty if either it contains a marked trap or $F(S) > 0$ with $F(S) = \min\{\sum_{p \in S} M(p) \mid M = M_0 + CY, M \geq 0, Y \geq 0\}$.

Note that a mathematical programming approach was proposed in [3] as well to avoid explicit enumeration of siphons. It allows one to check large Petri net models using powerful mathematical programming software packages.

III. RCN-Merged Nets

The modeling methodology proposed in [5] is resource oriented and is modular. The main steps include the modeling of the behavior of each resource using Petri nets, and the integration of resource net modules by taking into account interactions among resources. In [5] as well as in this paper, each resource is modeled using Resource Control Nets and the integration is realized through merging of common transition subnets

Definition 1: A Resource Control Net (RCN) is a strongly connected state machine (P, T, F, M_0) in which there exists one and only one place $p_r \in P$, called *resource place*, such that $M_0(p_r) \neq 0$. The remaining places are called *operation places*.

Definition 2: A transition subnet $G_\alpha = (P_\alpha, T_\alpha, F_\alpha, M_{\alpha 0})$ of a Petri net G is a subnet of G such that input transitions and output transitions of any place $p \in P_\alpha$ are transitions in T_α. In other words, the places of a transition subnet are local.

Definition 3: A Petri net $G = (P, T, F, M_0)$, obtained by merging n RCN's $\{G_s \mid G_s = (P_s, T_s, F_s, M_{s0}),\ s = 1, \ldots, n\}$ through common transitions and common transition subnets, is a net such

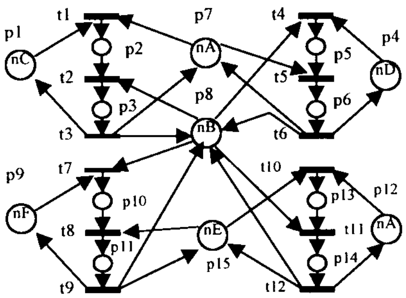

Fig. 2. An example (from [5]).

that: $P = P_1 \cup \cdots \cup P_n$, $T = T_1 \cup \cdots \cup T_n$, $F = F_1 \cup \cdots \cup F_n$, $M_0(p) = M_{s0}(p)$ if $p \in P_s$. Clearly, in the integrated model G, the common elements of any two RCN's form a transition subnet. In the following, G will be called an *RCN-merged net*.

Fig. 2 is an example taken from [5]. It allows us to illustrate the approach of this paper and to compare our results with those obtained in [5]. The net is obtained by merging 7 RCN's. Places p_1, p_4, p_7, p_8, p_9, p_{12} and p_{15} are resource places and the other places are operation places.

By construction, an RCN-merged net is State Machine Decomposable. Each RCN is a state machine component. The number of tokens in any state machine component remains constant whatever transition firings. Hence,

Property 4: An RCN-merged net is conservative and structurally bounded.

In the following, additional restrictions concerning the merging of RCN's are considered.

Restriction 1: At each common transition, there exists at most one input place that is an operation place.

Restriction 2: Common transition subnets should not include resource places.

Restriction 3: The Petri net G^* derived from the inetgrated model G by removing the resource places is an acyclic graph.

Restrictions 1 and 2 were considered in [5] as well. We repeat explanations given in [5] for completeness. A transition that has more than one operational input place corresponds to the synchronization of parallel processes or the assembly of several components. As a result, Restriction 1 is restrictive and its relaxation is an issue of future research.

Restriction 2 is natural for our approach that is resource-oriented. Since each type of resources is represented by an RCN where the resource place corresponds to the resource availability, it is then reasonable to associate with each type of resources a resource place, leading to Restriction 2. From this perspective, it seems restrictive to assume that, in each RCN, only the resource place is initially marked. Notice that a token in an operation place implies that a resource is in use. In most manufacturing systems, there exists a state such that all resources are available and there is no work-in-process. We choose such a state as the initial state. It is then natural that only the resource places are initially marked.

Restriction 3 is motivated by the fact that the net G^*, obtained by removing resource constraints, models flows of material or informations. These flows are usually acyclic and situations of Fig. 3 should not happen. Unfortunately, Restriction 3 rules out the possibility of rework and resource failures. Nevertheless, some common RCN models with rework and resource failures (see Fig. 4) can still be taken into account. For the model of Fig. 4, the reduction of places

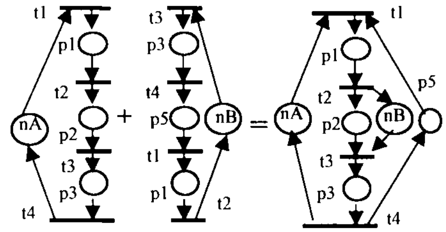

Fig. 3. Model that does not satisfy Restriction 3.

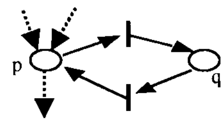

Fig. 4. Usual model for failures and rework.

p and q does not change the properties of the model and eliminate the loop containing p and q.

IV. PROPERTIES

It is interesting to see that reversibility of an RCN-merged net can be easily checked using siphons, properties usually used to characterize deadlocks.

Theorem 1: Suppose that Restrictions 1–3 hold. Then G is reversible iff no siphon of G can become empty.

Proof: $(\Rightarrow)$ Obvious since any siphon contains at least one resource place, any resource place is marked at M_0 and the net is reversible.

$(\Leftarrow)$ Consider a marking M_1 reachable from M_0. We show that M_0 can be reached starting from M_1. Consider as well the net G^*. From the structure of the net G and Restriction 3, G^* is an acyclic graph and its extremity nodes are transitions. Let H be the set of source transitions of G^*. Clearly, H is a subset of transitions of net G and the input places of any transition in H are resource places in G.

In order to show $M_0 \in R(M_1)$, starting from M_1, we apply the firing policy F which consists in not firing transitions in H. When applying to the net G^*, the firing policy F stops after a finite number of firings. Since G^* is obtained from G by removing resource places, any sequence of transitions firable in G is also firable in G^*. As a result, a marking at which none transition of G can fire under firing policy F is reached in a finite number of steps. Let M be this marking.

From the definition of M, no transition in $T - H$ is firable at M in the net G. If $M \neq M_0$, we show in the following that there exists an empty siphon. This contradicts the assumption of the theorem and concludes the proof.

In the remainder of the proof, we assume that $M \neq M_0$ and construct the empty siphon. The construction starts from the set R of RCN's with empty resource places at M, i.e. $R = \{s \mid M(p_{sr}) = 0, s = 1, \ldots, n\}$. We first prove by contradiction that $R \neq \phi$. Assume that $R = \phi$, i.e. every resource place is marked at M. Since $M \neq M_0$, at least one operation place p is marked. According to Restriction 1, p is the unique input place of all its output transitions that is an operation place. As a result, any transition in $p\bullet$ is firable at M. Since transitions in $p\bullet$ do not belong to H, this contradicts the assumption that no transition in $T - H$ is firable at M and implies that $R \neq \phi$.

For each element s in R, let S_s be the set of places p of P_s such that $M(p) = 0$ and that there exists a path γ in G_s connecting p to place p_{sr} with $M(\gamma) = 0$. For any input transition t of a place p in S_s, since t cannot be fired using policy F, the following three cases are possible: *Case 1*: $t \in p_{s'r}\bullet$ for some $s' \in R$ including the case $s' = s$, i.e. t is an output transition of an empty resource place; *Case 2*: $t \in q\bullet$ for some $q \in P_s$ such that $M(q) = 0$ which implies $q \in S_s$, i.e. t is an output transition of a place in S_s; and *Case 3*: $t \in H$, i.e. all input places of t are resource places, which leads to $t \in p_{sr}\bullet$ since t is a transition of G_s.

From the above reasoning, any input transition of a place belonging to $S = \bigcup_{s \in R} S_s$ is an output transition of a place in S. Hence S is a siphon and is empty at M. □

Consider the Petri net of Fig. 2. It has three minimal siphons that do not contain marked traps: $S_1 = \{p_8, p_3, p_6, p_{11}, p_{14}, p_{15}, p_7\}$, $S_2 = \{p_8, p_3, p_6, p_{11}, p_{14}, p_{15}, p_5\}$ and $S_3 = \{p_8, p_3, p_6, p_{11}, p_{14}, p_{10}, p_7\}$. From Property 3, the condition of Theorem 1 holds if $F(S_1) > 0, F(S_2) > 0$ and $F(S_3) > 0$. As in [3], these conditions can be expressed explicitly in terms of the initial marking. For this purpose, condition $F(S_i) = 0$ is considered. It corresponds to a set of equations/inequalities. Solving the equations and replacing the results into the other relations prove that $F(S_1) > 0, F(S_2) > 0$ and $F(S_3) > 0$ iff

$$(nE > nG) \vee (nA > nC) \vee (nB > nF + nD), \tag{1}$$

$$(nE > nG) \vee (nB > nF), \tag{2}$$

$$(nA > nC) \vee (nB > nD). \tag{3}$$

It can be shown that conditions (1–3) are necessary as well. These conditions are less restrictive than the following condition obtained in [5] using circular-wait: $((nA > nC) \vee (nB > nF + nD)) \wedge ((nE > nG) \vee (nB > nF + nD))$. The superiority of siphon-based approach will be confirmed in Section V.

Clearly, if the net is reversible, then any token in an operation place can be brought back to its related resource places and the net is deadlock-free. Furthermore, if tokens in a given place can no longer moved forward, using the same arguments, it can be shown that:

Corollary 1: Suppose that Restrictions 1–3 hold. Then there exists a marking $M \in R(M_0)$ and a place p such that $M(p) > 0$ and tokens in p can no longer move forward iff siphons of G can become empty.

Finally, let us notice that conditions of Theorem 1 and Corollary 1 can be checked using Property 3.

Under the reversibility, the RCN-merged net G is deadlock-free iff the initial marking is not a dead marking, i.e. at least one transition can fire. Furthermore, the liveness of the integrated model reduces to its potential liveness, a property easier to check.

In view of Theorem 1, it is obvious that:

Theorem 2: Suppose that Restrictions 1–3 hold, then G is live and reversible iff no siphon of G can become empty and every transition can fire at least once, i.e. every transition is *potentially firable*.

The following results can be used to check the potential firability of the transitions.

Restriction 4: At any common transition, there is at most one output place that is an operation place.

Theorem 3: Suppose that Restrictions 1–4 hold, then any transition of G is potentially firable.

Proof: Under the ongoing conditions, G^* is an acyclic Petri net such that all extremity nodes are transitions and that any transition has at most one input place and at most one output place.

For any transition t, there exists at least one elementary path $t_1 p_1 t_2 p_2 \ldots t_k p_k t$ in G^* connecting a source transition to t. Consider now the Petri net G. Clearly, $t_1 p_1 t_2 p_2 \ldots t_k p_k t$ is an elementary path in G as well and all input places of t_1 are resource places. As a result, t_1 is firable at M_0 and let M_1 be the marking obtained by firing t_1.

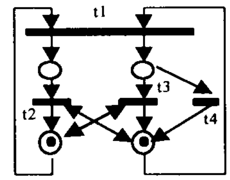

Fig. 5. RCN-merged net that is reversible but not live.

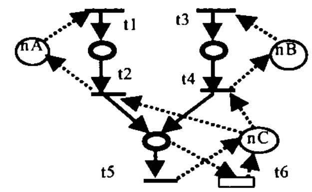

Fig. 6. An FCF-component.

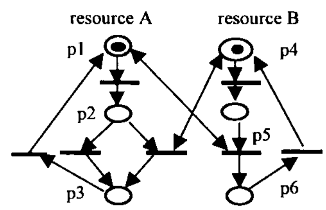

Fig. 7. Counter example.

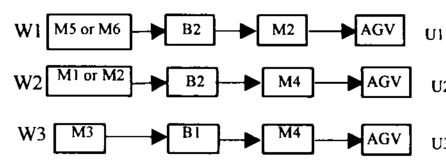

Fig. 8. BOM of U_1, U_2, and U_3.

Let us show that t_2 is firable at M_1. First, $M_1(p_1) = 1$. From Restriction 4, p_1 is the only operation place marked at M_1. This together with the conservation of RCN's implies that $M_1(p_{sr}) = M_0(p_{sr})$ for all s such that $p_1 \notin P_s$. From Restriction 1, any input place of t_2 except p_1 is a resource place p_{sr}. Clearly, $p_1 \notin P_s$, for all G_s such that $p_{sr} \in \bullet t_2$. This implies that t_2 is firable at M_1. Let M_2 be the marking obtained by firing t_2.

Similarly, it can be shown that t_3 is firable at $M_2, \ldots,$ and t is firable at M_k. □

Consider now the case where Restriction 4 does not hold. Checking potential firability becomes difficult as firing any transition not fulfilling Restriction 4 creates parallel processes. The potential firability is not always true as shown by the net of Fig. 5. It is obtained by merging two RCN's. Restrictions 1–3 hold but Restriction 4 does not hold. The net is reversible but transition t_3 cannot fire.

One way of verifying the potential firability of a given transition t is to check whether a sequence of transitions containing t can be found by either explicit enumeration or heuristic search. In the following, we use instead structural properties of the net. The basic idea is to remove the nondeterminism of the RCN-merged net by choosing for each operation place an output transition and to study the liveness of the resulting nets. This leads to the notion of *Forward-Conflict-Free components* (or FCF components).

Formally speaking, an FCF-component N_1 of a net N is a subnet generated by a subset T_1 of transitions having the following properties: i) each place in N_1 has one output transition and at least

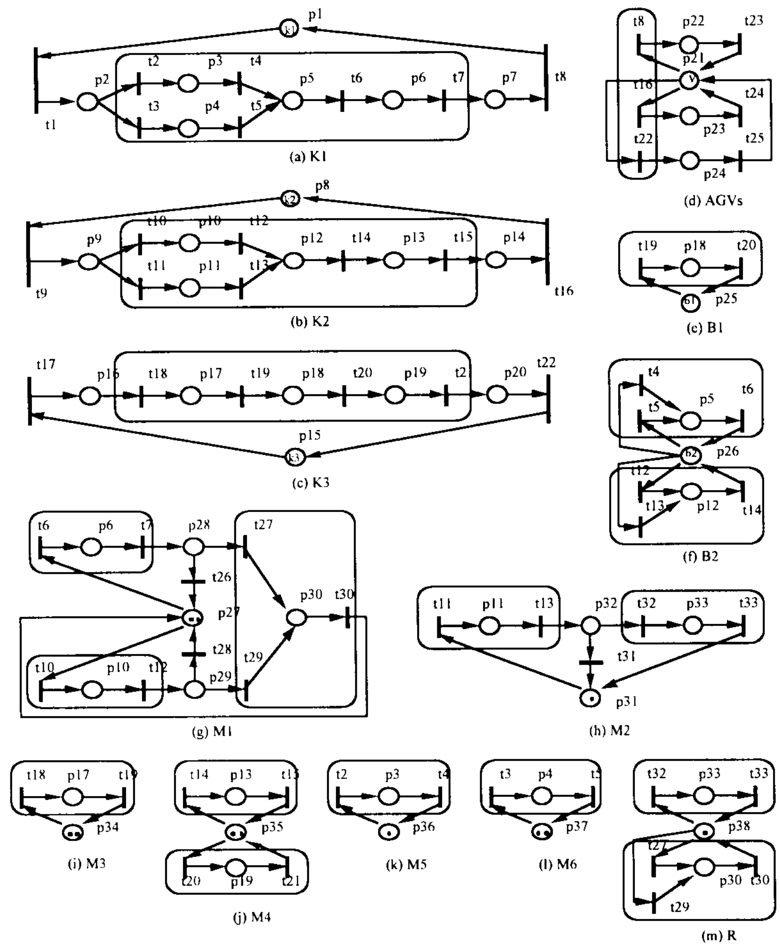

Fig. 9. RCN modules.

one input transition; ii) a subnet generated by T_1 is the net consisting of transitions in T_1, all of their input and output places, and their connecting arcs.

Fig. 6 is an FCF-component of the net G^* generated by transitions $\{t_1, t_2, t_3, t_4, t_5\}$. It is conflict-free because it does not contain t_6. From this examples, an FCF-component is not always a marked graph and a place in it may have more than one input transition.

Since G^* is an acyclic graph and all its extremity nodes are transitions, any FCF-component of G^* is an acyclic FCF whose extremity nodes are extremity nodes of G^*. Further,

Lemma 1: Suppose that Restrictions 1–3 hold, the Petri net G^* is covered by FCF-components with root nodes.

Proof: Let t^* be any transition of G^*. We construct an FCF-component containing t^*. First, there exists at least one elementary path $t_1p_1t_2p_2 \ldots t_kp_kt^*$ in G^* connecting a source transition to t^*. Set $P^* = \{p_1, p_2, \ldots, p_k\}$ and $T^* = \{t_1, t_2, \ldots, t_k, t^*\}$. Repeat the following recursions until convergence of P^* and T^*:

- For every $t \in T^*$, set $P^* = P^* \cup t\bullet$;
- For every $p \in P^*$ such that $p \bullet \cap T^* = \phi$, set $T^* = T^* \cup \{t\}$ with any $t \in p\bullet$.

The set of places of the subnet N^* generated by transitions in T^* is P^* since any transition in G^* has at most one input place. From the construction, any place in N^* has one outgoing arc and at least one incoming arc. N^* has a unique source transition t_1. Furthermore, any node in N^* can be reached from t_1 and t_1 is a root. □

Let $\mathcal{R}$ be the set of FCF-components with root nodes of G^*. For each of its elements $\hat{N} = (\hat{P}, \hat{T}, \hat{F}, \hat{M}_0)$, let $f(\hat{N})$ be the subnet of G generated by transitions in $\hat{T}$. Clearly, $f(\hat{N})$ can be derived from $\hat{N}$ by adding resource places that are input or output places of transitions in $\hat{T}$.

Theorem 4: Suppose that Restrictions 1–3 hold, a transition t is potentially firable if there exists an element $\hat{N}$ in $\mathcal{R}$ such that t is in $\hat{N}$ and that no siphon of $f(\hat{N})$ can become empty.

Proof: From Lemma 1, there exists an FCF-component $\hat{N}$ containing t. Let us notice that $f(\hat{N})$ is also a net that results from merging a set of nets $\hat{C} = \{\hat{G}_s \mid \hat{G}_s = (\hat{P}_s, \hat{T}_s, \hat{F}_s, \hat{M}_{0s}), \forall s/p_{sr} \in f(\hat{N})\}$. Each $\hat{G}_s$ is the intersection of G_s and $f(\hat{N})$, and has the following properties:

- $\hat{T}_s = \hat{T} \cap T_s, \hat{P}_s = \hat{P} \cap P_s$;
- For any $t \in \hat{T}_s$, since $\hat{N}$ is a subnet of G^* generated by transitions in $\hat{T}$, the input (resp. output) place of t in G_s is a place in $\hat{P}$ if it is an operation place. In any case, both the input and output places of t in G_s belong to $\hat{P}_s$;
- For any $p \in \hat{P} \cap P_s$, all its input and output transitions in $\hat{N}$ are transitions in G_s due to the merging through common transition subnets. Hence, it has one outgoing arc and at least one incoming arc in $\hat{G}_s$.

These properties together with Restriction 3 imply that any node in $\hat{G}_s$ can be reached from p_{sr} and vice versa. Hence, $\hat{G}_s$ is an RCN and $f(\hat{N})$ can be considered as a net that results from merging a set of RCN's through common transition subnets under Restrictions 1–3. According to Theorem 1, $f(\hat{N})$ is reversible if no siphon in it can become empty.

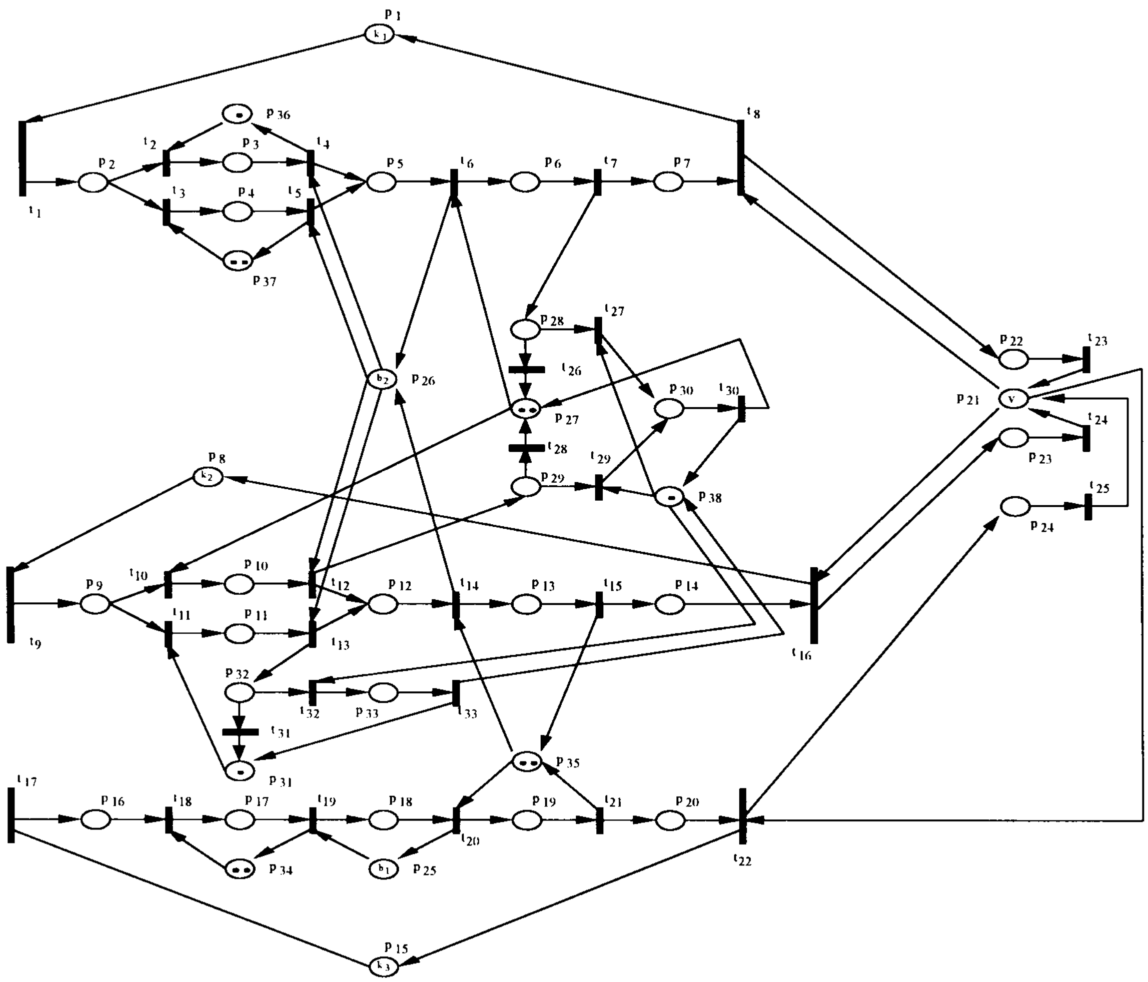

Fig. 10. System model.

Let us show that it is live as well. It is enough to show that every transition t is potentially firable. From the property of $\hat{N}$, there exists a path $t_1p_1t_2p_2\dots t_kp_kt$ in $\hat{N}$ where t_1 is the root transition of $\hat{N}$. Clearly it is firable at M_0. Let M_1 be the marking obtained by firing t_1. Since $f(\hat{N})$ is reversible, there exists a sequence σ of transitions leading M_1 to M_0. Since $\hat{N}$ is an FCF, t_{i+1} is the unique output transition of p_i for $i = 1, \dots, k$ in both $\hat{N}$ and $f(\hat{N})$. Hence, $\sum_{i=1}^{k} M_1(p_i) \le \sum_{i=1}^{k} M_0(p_i) + \bar{\sigma}[t]$. From the definition of M_1 and M_0, $M_1(p_1) = 1, M_1(p_i) = M_0(p_j) = 0, \forall 1 < i \le k, 1 \le j \le k$. Finally, $\bar{\sigma}[t] \ge 1$ which implies that every transition of $f(\hat{N})$ is potentially firable. □

V. Siphons, Circular Structure, and Circular-Wait

The notions of circular structure and circular-wait were introduced in [5], [6] for analyzing liveness and reversibility of RCN-merged nets. Our purpose here is to show relationship between siphons and these notions. First, recall that a circular structure is a set of places $CS = \{p_1, p_{s_1r}, p_2, p_{s_2r}, \dots, p_k, p_{s_kr}\}$ such that $p_1 \in G_{s_kr}, p_{i+1} \in G_{s_ir}, \forall 1 \le i < k$, and p_i and p_{s_ir} share at least one common output transition. A *circular-wait* is defined by a set of circular structures $CS_1, \dots, CS_m$ and a marking $M \in R(M_0)$ such that: $\sum_{p \in P_{ox}} M(p) = M_0(p_{rx}), \forall x \in W$ where W is the set of RCN's whose resource places belong to the circular structures and P_{ox} is the set of operation places of G_x that belong to the circular structures.

Theorem 5: Suppose that Restrictions 1–3 hold. If there exists a siphon S' that can become empty, then:

1. there exist $M \in R(M_0)$ and a circular structure $\{p_1, p_{s_1r}, p_2, p_{s_2r}, \dots, p_k, p_{s_kr}\}$ such that $M(p_i) > 0$ and $M(p_{s_ir}) = 0$, for all i;
2. there exists a circular-wait.

Proof: Let $M_1 \in R(M_0)$ be a marking at which siphon S' is empty. Similar to the proof of Theorem 1, it possible to find a marking M such that any transition $t \notin H$, where H is the set of source transitions of G^*, is not firable at M. Further, the set $R = \{s \mid M(p_{sr}) = 0, s = 1, \dots, n\}$ is not empty.

To prove Claim (1), consider any $s_1 \in R$. Clearly, there exists $p_1 \in P_{s_1}$ with $M(p_1) > 0$. Consider any output transition t_1 of p_1. Since $t_1 \notin H$, it is not firable and has at least one input place that is an empty resource place. Let p_{s_2r} be this place. Repeating the above process leads to a set of places $\{p_{s_1r}, p_1, \dots, p_{s_kr}, p_k, p_{s_{k+1}r}\}$ such that $M(p_{s_ir}) = 0$, $M(p_i) > 0$, $p_i \in G_{s_ir}$ and p_i and $p_{s_{i+1}r}$ share common output transitions. The process stops when $p_l \in P_{s_{k+1}}$ for some $1 \le l \le k$. The circular structure that we are looking for is $\{p_l, p_{s_{l+1}r}, \dots, p_{s_kr}, p_k, p_{s_{k+1}r}\}$.

Consider now Claim (2). We first construct a directed graph (V, A). The set of nodes corresponds to empty resource places and their marked operation places, i.e. $V = V_r \cup V_o$ with $V_r = \{p_{sr} \mid s \in R\}$ and $V_o = \{p \in \bigcup_{s \in R} P_s \mid M(p) > 0\}$. There is an arc from any

TABLE I
MEANINGS OF PLACES IN RCN MODULES.

p_1	Free kanbans associated with products U_1.
p_2	Parts W_1 to be processed.
p_3	Parts W_1 being processed on M_5.
p_4	Parts W_1 being processed on M_6.
p_5	Parts W_1 in B_2.
p_6	Parts W_1 being processed on M_2.
p_7	Parts W_1 having been processed.
p_8	Free kanbans associated with products U_2.
p_9	Parts W_2 to be processed.
p_{10}	Parts W_2 being processed on M_1.
p_{11}	Parts W_2 being processed on M_2.
p_{12}	Parts W_2 in B_2.
p_{13}	Parts W_2 being processed on M_4.
p_{14}	Parts W_2 having been processed.
p_{15}	Free kanbans associated with products U_3.
p_{16}	Parts W_3 to be processed.
p_{17}	Parts W_3 being processed on M_3.
p_{18}	Parts W_3 in B_1.
p_{19}	Parts W_3 being processed on M_4.
p_{20}	Parts W_3 having been processed.
p_{21}	Available AGVs.
p_{22}	Unloading of U_1.
p_{23}	Unloading of U_2.
p_{24}	Unloading of U_3.
p_{25}	The remaining capacity of B_1.
p_{26}	The remaining capacity of B_2.
p_{27}	Available machines in M_1.
p_{28}	M_1 waiting for being available or maintained.
p_{29}	M_1 waiting for being available or maintained.
p_{30}	M_1 being maintained using R.
p_{31}	Available machines in M_2.
p_{32}	M_2 waiting for being available or maintained.
p_{33}	M_2 being maintained using R.
p_{34}	Available machines in M_3.
p_{35}	Available machines in M_4.
p_{36}	M_5 available.
p_{37}	Available machines in M_6.
p_{38}	R available.

resource place p_{sr} to every marked place in G_s. There is also an arc from a place $p \in V_o$ to a resource place p_{sr} if they have a common output transition. Clearly, any node in V_r has at least one outgoing arc and any node in V_o has at least one incoming arc. Further, since any output transition of places in V_o is not a source transition, it is not firable and has an input place that is an empty resource place. As a result, any node in V_o has at least one outgoing arc. Clearly, any circuit in (V, A) corresponds to a circular structure.

We now define another graph (V', A') as follows. Starting from (V, A) and a node $p_{sr} \in V_r$ that is a source node, we remove p_{sr} and nodes $p \in V_o$ having only p_{sr} as predecessor. This process is repeated until no node can be removed. Clearly, the iterative process cannot remove nodes belonging to a circuit in (V, A). The final graph (V', A') is not empty as a consequence of Claim (1). Furthermore, for any node $p_{sr} \in V'$, all marked places in G_s remain in V'.

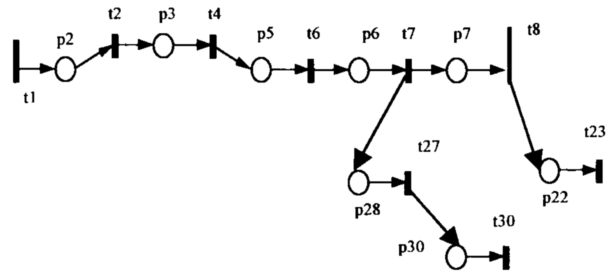

Fig. 11. FCF component of the system net.

Since (V', A') has not source nor sink node, every node belongs to an elementary circuit. Hence, the set of circular structures $CS_1, \ldots, CS_m$ corresponding the set of elementary circuits of (V', A') contains all resource places $p_{sr} \in V'$ and their related marked places. From the property of RCN's, $\sum_{p \in P_{os}} M(p) = M_0(p_{sr})$. $CS_1, \ldots, CS_m$ and M form a circular-wait. □

We notice that Theorem 5 holds as well for Petri nets generated by FCF-components, i.e. $f(\hat{N})$. Furthermore, any circular structure (resp. circular-wait) of $f(\hat{N})$ is a circular structure (resp. circular-wait) of G. As a result,

Corollary 2: Suppose that Restrictions 1–3 hold. The Petri net G is live and reversible if it does not contain any circular structure or there is no circular-wait.

From Theorem 5, the existence of a siphon that can become empty implies the existence of a circular-wait. However, the converse is not true and the existence of a circular-wait does not implies the existence of deadlocked situations. This is true when an operation place has more than one output transition and at least one of them is not involved in the circular-wait. A counter-example is given in Fig. 7. The reachable marking $M = (0, 1, 0, 0, 1, 0)^T$ and the circular structure $\{p_2, p_4, p_5, p_1\}$ form a circular-wait. However, the net is live and reversible. It can be checked using results of this paper. First, Restriction 4 holds, which implies that every transition is potentially firable. There are two minimal siphons $\{p_1, p_2, p_3\}$ and $\{p_4, p_5, p_6\}$. Each of them is a marked trap as well. Hence, by Theorem 2, the RCN-merged net is live and reversible. From the above comments, we conclude that the results of this paper are stronger than those of [5], [6]. Furthermore, siphon-based conditions can be checked using Property 3 or mathematical programming techniques proposed in [3].

VI. AN EXAMPLE

This section presents an example that does not satisfy Restriction 4. Consider a simple manufacturing system that produces three product types, U_1, U_2, and U_3, from three raw part types, W_1, W_2, and W_3. Fig. 8 shows the BOM (Bill of Material) of U_1, U_2, and U_3. There are six machine types $M_i, i = 1, \ldots, 6$ with 2, 1, 2, 2, 1, and 2 identical machines, respectively, two intermediate buffers B_1 and B_2 with capacities of b_1 and b_2, respectively, and v identical AGV's. M_1 and M_2 have a special requirement that after each operation is finished, they may be maintained by a shared robot R for replacing a consumable component (this maintenance process is often found in semiconductor manufacturing). AGV's handle the unloading of U_1, U_2, and U_3. The production of U_1, U_2, and U_3 is controlled by a Kanban system with three types of kanbans, K_1, K_2, and K_3, for U_1, U_2, and U_3, respectively. The numbers of initial free kanbans for K_1, K_2, and K_3 are k_1, k_2, and k_3, respectively. From the manufacturing resources and their respective processes in the BOM, we can construct their RCN modules, as shown in Fig. 9, where dotted areas are common transitions and common transition subnets, denoted with the same labels. Table I shows the meanings of the places while the transitions denote the start and/or end of some operation. After merging the modules, we obtain a system model in Fig. 10.

It can be verified that the system net only has one minimal siphon that does not contain a marked trap: $S1 = \{p_6, p_{12}, p_{26}, p_{27}, p_{28}, p_{29}, p_{30}\}$. From Property 3, the condition of Theorem 1 holds if $F(S1) > 0$. After some algebraic manipulations, it can be shown that $F(S1) > 0$ iff

$$(k_1 < b_2) \vee (k_2 < 2). \tag{4}$$

It can be shown that condition 4 is necessary as well. Thus, the net is reversible iff condition 4 holds. If it does not hold, i.e., $(k_1 \geq b_2)$ and $(k_2 \geq 2)$, then the sequence $t_1^{k_1}(t_2t_4)^{b_2}t_2^a t_3^{\min\{k_1-b_2-a,2\}} t_9^{k_2} t_{10}^2 t_{11}^{\min\{k_2-2,1\}}$, where $a = \min\{k_1 - b_2, 1\}$, leads to a deadlock such that all transitions except those related to U_3 (i.e., t_{17}, t_{18}, t_{19}, t_{20}, t_{21}, t_{22}, and t_{25}) are not firable forever. Since Restriction 4 is not satisfied, we must generate FCF components to check the potential firability of each transition. For example, the net of Fig. 11 is an FCF component N_1 of the resultant net G^* after the resource places are removed from the system net. $f(N_1)$ is the resultant net after the resource places are added to N_1. It is a marked graph and is live iff every resource place is initially marked. As a result, every transition in N_1 is potentially firable. Similarly, it can be shown that transitions not in N_1 are potentially firable as well. Therefore, the system net is live if condition 4 holds.

VII. Concluding Remarks

In this paper, we have analyzed Petri nets resulting from merging Resource Control Nets introduced in [5]. Our analysis has been based on siphons. It has been shown that a RCN-merged net is reversible iff no siphon in it can become empty. Under the reversibility, the liveness of the RCN-merged net reduces to its potential liveness. If each transition in the net does not create parallel processes, the net is proven to be potentially live. Otherwise, the analysis is based on the notion of FCF components. Finally, relations between siphons, circular structure, and circular-wait have formally been established. The results in this paper have been compared with those in [5]. The superiority of the former has been shown.

We notice that the number of siphons to be examined grows exponentially with the number of RCN's involved. Therefore, to apply the results of this paper to synthesizing large resource control systems, one could use the mixed-integer programming approach proposed in [3] for checking, without explicit enumeration of siphons, the existence of potential deadlocks.

References

[1] T. Agerwala and Y. Choed-Amphai, "A synthesis rule for concurrent systems," in *Proc. 15th Design Automat. Conf.*, Las Vegas, NV, 1978, pp. 305–311.

[2] G. Berthelot, "Checking properties of nets using transformation," *Advances In Petri Nets 1985*, G. Rozenberg, Ed. Berlin, Germany: Springer-Verlag, 1985, pp. 19–40.

[3] F. Chu and X. L. Xie, "Deadlock analysis of Petri nets using siphons and mathematical programming," *IEEE Trans. Robot. Automat.*, vol. 13, pp. 793–804, 1997.

[4] J. Ezpeleta, J. M. Colom, and J. Martinez, "A Petri net based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Robot. Automat.*, vol. 11, pp. 173–184, 1995.

[5] M. D. Jeng and F. DiCesare, "Synthesis using resource control nets for modeling shared-resource systems," *IEEE Trans. Robot. Automat.*, vol. 11, pp. 317–327, 1995.

[6] M. D. Jeng, "Petri nets for modeling automated manufacturing systems with error recovery," *IEEE Trans. Robot. Automat.*, vol. 13, pp. 752–760, 1997.

[7] I. Koh and F. DiCesare, "Modular transformation methods for generalized Petri nets and their applications in automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, pp. 963–973, 1991.

[8] B. H. Krogh and C. L. Beck, "Synthesis of place/transition nets for simulation and control of manufacturing systems," in *Proc. 4th IFAC/IFORS Symp. Large Scale Syst.*, Zurich, Switzerland, 1986, pp. 661–666.

[9] K. H. Lee, J. Favrel, and P. Baptiste, "Generalized Petri net reduction method," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-17, pp. 297–303, 1987.

[10] T. Murata, "Petri nets: Properties, analysis and applications," *Proc. IEEE*, vol. 77, pp. 541–580, 1989.

[11] Y. Narahari and N. Viswanadham, "A Petri net approach to the modeling and analysis of flexible manufacturing systems," *Ann. Oper. Res.*, vol. 3, pp. 449–472, 1985.

[12] I. Suzuki and T. Murata, "A method for stepwise refinement and abstraction of Petri nets," *J. Comput. Syst. Sci.*, vol. 27, pp. 51–76, 1983.

[13] R. Valette, "Analysis of Petri nets by stepwise refinement," *J. Comput. Syst. Sci.*, vol. 18, pp. 35–46, 1979.

[14] L. Wang and X. Xie, "Modular modeling using Petri nets," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 800–809, 1996.

[15] M. C. Zhou, K. McDermott, and P. A. Patel, "Petri net synthesis and analysis of a flexible manufacturing system cell," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, pp. 523–531, 1993.

Assessing Determinism of Photo-plethysmographic Signal

Joydeep Bhattacharya and Partha Pratim Kanjilal

***Abstract*— A study was performed to analyze the signal obtained from a noninvasive photo-plethysmographic device from four subjects in different clinical conditions. With the help of the theory of nonlinear dynamics, it is verified that the cardiovascular dynamics is dominated by an underlying chaotic attractor. A new robust and computationally efficient method is presented for the detection of the hidden deterministic structure of a time series. It is shown that the degree of chaos as well as the underlying determinism is directly related to the subject's clinical stability.**

***Index Terms*— Blood pressure waveform, cardiovascular systems, determinism, nonlinear dynamics, singular value decomposition.**

I. Introduction

The analysis of physiological time series can be used to identify hidden dynamical information leading to new insights into the understanding of physiological mechanisms or state. Physiological

Manuscript received August 15, 1998; revised December 24, 1998. This work was supported by a grant from the Department of Electronics, Government of India.

J. Bhattacharya is with the Max Planck Institut fur Physik Komplex System, Dresden D 01187, Germany (e-mail: joy@mpipks-dresden.mpg.de).

P. P. Kanjilal is with the Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology, Kharagpur 721 302, India (e-mail: ppk@ece.iitkgp.ernet.in).

Publisher Item Identifier S 1083-4427(99)03328-7.