A Control Policy of S^3PR Without Siphon Enumeration and Reachability Analysis

Daniel Yuh Chao, Yao-Nan Lien and Fang Yu

Abstract— **Flexible Manufacturing System (FMS) suffers from deadlocks negating the advantages of automation. To prevent deadlocks from occurring, monitors (incurring hardware cost) are often added to siphons for them to be always sufficiently marked. The original Petri net model gets disturbed and some good states are no longer attainable. It has been a hot race to synthesize optimal controllers to be maximally permissive with fewest monitors. Li and Zhou propose simpler Petri net controllers by dividing emptiable siphons into two groups: elementary and dependent and adding monitors for elementary siphons only. In an earlier paper, the author showed that among all 2-dependent siphons (depending on two elementary siphons), only one siphon needs to be controlled by adding a monitor. This greatly simplifies the synthesis as well as minimizes the number of monitors required while making the control net maximally permissive. This paper extends the result to n-dependent siphons with n>2. It shows that in the set *B* of n-dependent siphons, there is only one emptiable (called critical) siphon (analyzed to be the one with the minimal number of tokens in the unmarked set of operations places) needs to be controlled while the rest of siphons are also controlled accordingly. In the worst case, for a k-compound siphon in the uncontrolled net, the total number of member siphons in *B* is k(k-1)(2*k-1)/6.**

I. Introduction

FLEXIBLE Manufacturing System (FMS) is a new type of manufacturing system with a computer-controlled configuration to automatically produce different products. There are three main systems in most FMS: 1. work machines to perform a series of operations; 2. an integrated material transport system with a computer to control the flow of materials, tools, and information throughout the system; 3. auxiliary work stations for loading and unloading, cleaning, inspection, etc.

To effectively utilize the precious resources, they must be shared and carefully coordinated among various competing jobs. This may induce deadlocks, undesirable since it disrupts the automation involved [1].

Ezpeleta *et al.* pioneer a class of Petri nets (PN) called systems of simple sequential processes with resources (S^3PR) [2]. Deadlocks can be avoided by adding a control place —and associated arcs— to each emptiable siphon *S* to prevent it from being unmarked. However, generally too many control places and arcs are required.

Furthermore, the system is overly constrained since to avoid the generation of new siphons, Ezpeleta *et al.* [2] shift all output (called Type-2, or source) arcs of each V_S to the output (called source) transition of the entry (called idle place) of input raw materials to limit their rate into the system, called SMSless approach. Consequently, many reachable states are lost.

Li and Zhou [3,4,21-24] propose the concept of elementary siphons (generally much smaller than the set of all emptiable siphons in large Petri nets) to minimize the new addition of control places. They classify emptiable siphons into two kinds: elementary and dependent. By adding a control place for each elementary siphon S_e, all dependent siphons *S* too are controlled too, thus reducing the number of monitors required rendering the approach suitable for large Petri nets. However, many good states are lost since control arcs end at the source transitions to prevent generating new siphons which disturbs the affected region more than the maximally permissive approach.

In the two-stage approach proposed by Li *et al.* [7], they first find emptiable siphons (and add monitors) using a mixed integer programming (MIP) method to avoid time-consuming complete siphon enumeration (hence more efficient and structurally simpler than existing ones). Then, they rearrange the output arcs of the monitors while preserving liveness to be more permissive.

However, MIP is NP-hard to have exponential time complexity and all dependent siphons may be derived (and need a monitor) before any elementary siphon in the worst case. Also, the number of good states is less than the near optimal one in [7].

Uzam [8] applies the theory of regions to synthesize optimal liveness-enforcing supervisors. Ghaffari *et al.* [9] propose a synthesis methodology for a supervisor by exploring the sufficient and necessary conditions (in terms of plain and popular linear algebra notions) on the existence of an optimal liveness-enforcing supervisor. Although one can always achieve such an optimal supervisor (if existing), it needs complete state enumeration and solving the NP-hard integer linear programming problem. In addition, some monitors are redundant.

Uzam and Zhou [10,11] apply region analysis (RG) to a well-known S^3PR. The benchmark reaches 26750 states, where 21581 are legal, i.e., either good or dangerous states (i.e., boundary states to reach forbidden regions). There are 5299 elements in the set of marking/transition separation instances, denoted by Ω. This implies that 5299 LPPs (Linear Programming Problems) have to be solved to find

Manuscript received August 17, 2011.

D. Y. Chao is with the Department of Management Information Systems, National Chengchi University, Taipei, Taiwan, R.O.C. (phone: 886-2-29387694; fax: 886-2-29393754; e-mail: yuhyaw@ gmail.com).

Y.-N. Lien is with the Department of Computer Science, National Chengchi University, Taipei, Taiwan, R.O.C. (email:lien@cs.nccu.edu.tw)

F. Yu is with the Department of Management Information Systems, National Chengchi University, Taipei, Taiwan, R.O.C. (e-mail: yuf@nccu.edu.tw).

an optimal liveness-enforcing supervisor with 21581 reachable states in the controlled system. However, |Ω| (cardinality of Ω) grows exponentially with respect to the size of a plant model. It is clearly infeasible to solve |Ω| LPPs for either a sizable net or a small net with a sizable initial marking.

To relieve this problem, Uzam *et al.* propose in [10] an iterative way with less computational cost. They divide a reachability graph into deadlock-zone (DZ) and deadlock-free zone (DFZ). A first-met bad marking (FBM) is selected from DZ at each iteration, then a control place is added to prevent the bad marking from being reached by constructing a place invariant (PI) of the Petri net based on the well-established method by Yamalidou *et al.* [12]. Uzam and Zhou further [10] improve the approach in [11] in two aspects: 1) reducing a net model [13] to construct the reachability graph with less computational overhead; and (2) proposing a simplification for the invariant-based control approach.

This method cannot ensure the optimality in general as it reaches 21562 states, 19 fewer than the optimal one representing 0.088%. It is better to avoid reachability graph by using siphon-based controlled model to reach the same 21562 states.

Piroddi *et al*. [5,6] synthesize maximally permissive and minimal redundant supervisors for the well-known S^3PR by controlling a set of selective siphons obtained from solving a set-covering problem during each iteration to forbid all critical markings (under which at least one siphon is empty) and make all uncontrolled siphons controlled. However, as many as n_M inequalities (with n_S variables) need to be solved, where n_M is the number of critical markings.

Furthermore, during each iteration, the method in [14] must be employed to remove redundant monitors with exponential time complexity. Thus, the computational burden remains high.

Also, the controlled net becomes a general Petri net (GPN) since some control arcs are weighted. The traditional MIP method cannot be extended to GPN since the presence of unmarked siphons does not imply deadlocks. Hence, Piroddi *et al*. transform weighted arcs into ordinary ones, which sometimes may cause unnecessary deadlocks as mentioned in [5,6].

In an earlier paper [30], we categorize emptiable siphons into basic, compound, control and mixture siphon. Basic (resp. compound) siphons can be synthesized from elementary (resp. compound) resource circuit. It [15,30] has been shown that basic and compound siphons correspond to elementary and dependent siphons, respectively when the above basic circuits intersect at a single resource place.

We propose in [30] to optimize the number of monitors (good states as well) if one adds monitors in the normal sequence of basic, compound, control, partial and full mixture siphons. We showed that among all 2-dependent siphons (depending on two elementary siphons), only one (called critical) siphon needs to be controlled by adding a monitor. This greatly simplifies the synthesis as well as minimizes the number of monitors required while making the control net maximally permissive.

Furthermore, the computational burden is much less since there is no need to enumerate all minimal siphons, nor to build the reachability graph. It requires neither iterations nor the removal of redundant monitors. In addition, no control arcs are weighted.

Unlike other works, our method finds the above types of siphons efficiently as indicated by Lemma 4 in [20] that it is relatively easy to identify elementary resource (called basic) circuits c_b between two neighboring working processes (WP) and Lemma 6 (all places in any c_b must be resource places) in [20] to simplify the search of c_b. Furthermore, it is easy to find c_b (normally formed among adjacent sharing resource places) when resource places between two adjacent working processes (WP) are arranged in reverse order with no need to search circuits containing far-away resource places.

An earlier paper [31] extended the result to n-dependent siphons with n>2 and showed that in a set of n-compound, n-control and n-mixture siphons, only one monitor is needed. Theory has been developed to identify this critical siphon. For instance, if a control or a compound siphon is emptiable, then it is a critical siphon.

We proposed [31] to find the critical siphon by inferring the token distribution (called unmarked) pattern in an unmarked critical siphon from that of 2-dependent siphons. This has the advantage of avoiding the computation of new siphons derived from monitor places since the unmarked pattern solely determines the controller region (or control arcs) and the initial marking.

This paper further proposes a maximally permissive control policy for a subclass of Petri nets based on the above theory of token distribution pattern of unmarked siphons.

The rest of this paper is organized as follows. Section II presents the basis (Petri nets and S^3PR) to understand the paper. Section III presents the synthesis of different kinds of siphons. Section IV motivates the reader using a 2-dependent system to show that there is a critical siphon and how the critical siphon is identified. The theory is extended to n-dependent siphons with examples in Section V. Section V I concludes the paper.

II. PRELIMINARIES

A Petri net (or Place/Transition net) is a 4–tuple $N = (P, T, F, W)$, where P is the set of *places*, T is the set of *transitions*, $F \subseteq (P \times T) \cup (T \times P)$ is called flow relation of the net, represented by arcs with arrows from places to transitions or from transitions to places, and W: $F \rightarrow Z^+$ (the set of nonnegative integers) is a mapping that assigns a weight to an arc. W(x, y)=0 when no arc exists between two nodes x and y.

M_0: $P \rightarrow Z^+$ is the *initial marking* assigned to each place $p \in P$, $M_0(p)$ *tokens*. (N, M_0) is called a marked net or a net system. In the special case that the flow relation W maps onto {0, 1}; the Petri net is said to be *ordinary* (otherwise, *general*). The incidence matrix of N is a matrix [N]: $P \times T \rightarrow Z$ (the set of integers) indexed by P and T such that $[N](p, t) = W(t, p) - W(p, t)$ where $W(p, t)$ is the weight of the arc from p to t, and $W(t, p)$ is the weight of the arc from t to p.

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. $t^{\bullet}$). Finally, an ordinary PN such that (s.t.) $\forall t\in T$, $|t^{\bullet}|=|^{\bullet}t|=1$, is called a State Machine *(SM)*. A Petri net is strongly connected if "$\forall x$, $x'\in (P \cup T)$, such that $x\neq x'$, there is a directed path from x to x'.

Given a marking M, a transition t is *enabled* if"$p\in{}^{\bullet}t$, $M(p)\geq W(p, t)$, and this is denoted by $M[t>$. Firing an enabled transition t results in a new marking M_1, which is obtained by removing $W(p, t)$ tokens from each place $p\in{}^{\bullet}t$ and placing $W(t, p')$ tokens in each place $p'\in t^{\bullet}$, moving the system state from M_0 to M_1. Repeating this process, it reaches M' by firing a sequence $\sigma=t_1t_2...t_k$ of transitions. M' is said to be reachable from M_0; i.e., $M_0[\sigma>M'$. $R(N, M_0)$ is the set of markings reachable from M_0.

A transition $t\in T$ is live under M_0 *iff* $\forall M\in R(N, M_0)$, $\exists M'\in R(N, M)$, t is enabled under M'. A PN is *live* under M_0 *iff* $\forall t\in T$, t is live under M_0. A Petri net is said to be deadlock-free, if at least one transition is enabled at every reachable marking.

For a Petri net system (N, M_0), a non-empty subset S (resp. Q) of places is called a *siphon* (resp. *trap*) if $^{\bullet}S\subseteq S^{\bullet}$(resp. $Q^{\bullet}\subseteq{}^{\bullet}Q$), i.e., every transition having an output (resp. input) place in S has an input (resp. output) place in S (resp. Q). If $M_0(S)=\sum_{p\in S} M_0(p)=0$, S is called an *empty siphon* at M_0. A *minimal siphon* does not contain a siphon as its proper subset. It is called a *strict minimal siphon* (*SMS*), denoted by S, if it does not contain a trap. An integer vector Y (with components $Y(p)$, $p\in P$) is called a P-invariant *iff* $Y\neq\mathbf{0}$ and $Y^T\bullet[N]=\mathbf{0}$, *where* $[N]$ is the incidence matrix. $||Y||=\{p\in P|Y(p)\neq 0\}$ is the *support* of Y. A *minimal* P-invariant does not contain another P-invariant as its proper subset.

Property 1 [6]: *The linear combination of Y_1 and Y_2 is a P–invariant if both Y_1 and Y_2 are P–invariants. Furthermore, if Y is a P–invariant of N, then given an initial marking M_0, $\forall M \in R(N, M_0)$, $Y^T \bullet M = Y^T \bullet M_0$ or $\omega(M)=\omega(M_0)$, where $\omega(M) = Y^T \bullet M$ (resp. $\omega(M_0) = Y^T \bullet M_0$) is the weighted sum of tokens under M (resp. M_0).*

Thus, the weighted sum of tokens in $||Y||$ is constant under all $M\in R(N, M_0)$. For a net N, control policy d_1 is more permissive than d_2 if $|R(N,M_0)/d_1|> |R(N,M_0)/d_2|$,where $R(N, M_0)/d_x$ means the number of reachable states under policy d_x.

The following definitions are from [7]. The reader can refer to [7] for more details of the S^3PMR model.

Definition 1 [7]: A process net is a strongly connected state machine (SM) (P, T, F, W, M_0) with exactly one initially marked place p^0 (job place), such that every circuit of N contains the place p^0. The other places are called operation places.

Transitions in $p^{0\bullet}$ and $^{\bullet}p^0$ are called *source and sink transitions* of the process net, respectively.

Definition 2 [7]: *An S^3PMR net N is a net that results from adding a set P_R of initially marked places (resource places) to a set of independent process nets. 1) Each resource place r is associated with a set of operation places, $H(r)$. This implies that these operation places require r. 2) For each input transition t of some $p\in H(r)$, there exists an arc from r to t if $^{\bullet}t\cap H(r)=\varnothing$ (empty set, i.e., input places of t but not operation places that use r). 3) For each output transition t of some $p\in H(r)$, there exists an arc from t to r if $t^{\bullet}\cap H(r) =\varnothing$. A resource place is said to be (resp. non) sharing if it is used by holder places in more than (resp. only) one process.*

Note that we have used $H(r)$ instead of $OP(r)$ in [7]. $H(r)$ is also called the set of holder places that use r. Define $H(A)=\{H(r)|\ r\in A\}$ as the set of holder places that use resources in A.

Definition 3 [2]: An S^3PR is defined as the union of a set of nets $N_i=(P_i\cup\{p_i^0\}\cup P_{Ri},T_i,F_i, W_i)$, sharing common places, where the following statements are true.

1) p_i^0 is called the process idle place of N_i. Places in P_i and P_{Ri} are called operation and resource places, respectively.

2) $P_{Ri}\neq\varnothing$; $P_i\neq\varnothing$; $p_i^0\notin P_i$; $(P_i\cup\{p_i^0\})\cap P_{Ri}=\varnothing$;.

i) $\forall p\in P_i$, $\forall t'\in p^{\bullet}$, $\forall t\in{}^{\bullet}p,\exists r_p\in P_{Ri}$, $^{\bullet}t\cap P_{Ri} =t'^{\bullet}\cap P_{Ri} =\{r_p\}$; $\forall r\in P_{Ri}, {}^{\bullet\bullet}r\cap P_i=r^{\bullet\bullet}\cap P_i\neq\varnothing$;

ii) $\forall r\in P_{Ri}$, $^{\bullet}r\cap r^{\bullet}=\varnothing$. $^{\bullet\bullet}(p_i^0)\cap P_{Ri} =(p_i^0)^{\bullet\bullet}\cap P_{Ri} =\varnothing$.

3) N'_i is a strongly connected state machine, where $N'_i=(P_i\cup\{p_i^0\},T_i, F_i, W_i)$ is the resultant net after the places in P_{Ri} and related arcs are removed from N_i.

4) Every circuit of N_i contains the place p_i^0.

5) Any two nets N_i and N_j are composable, denoted by N_i o N_j, if they share a set of common resource places. Every shared place must be resource one.

6) Transitions in $p_i^{0\bullet}$ and $^{\bullet}p_i^0$ are called source and sink transitions, respectively.

Define $\rho(r)=\{r\}\cup H(r)$ to be the support of a minimal P-invariant that contains r. Each operation place in an S^3PR (an S^3PMR) uses a unique resource place, and two consecutive operation places use different resource places. S^3PMR [17] is a generalization of S^3PR by allowing an operation place to use multiple types of resources. Also, when entering the next operation place, it may not release the resource.

In [7], M_0 for each OC place is set to $M_0(p)= M_0(S)$-1; S is said to be limit-controlled since $M_{min}(S)= min_{M\in R(N,\ M)} M(S)=1$.

When all tokens in SMS S of an S^3PMR flow into the *complementary set* of S, denoted by $[S]$, S becomes permanently unmarked. In fact, S and $[S]$ (if $[S]$ is nonmpty) together form the support of a minimal P-invariant. When $[S]$ is part of the support of another P-invariant Y_V, the largest number of tokens in $[S]$, denoted by $M_{max}([S])$, is limited by $M_0(||Y_V||)$. If $M_0(||Y_V||)= M_0(S)-1$, S never becomes unmarked.

In practice, Y_V can be achieved by adding a monitor or control place V, which acts like a resource also having a set of holder places, denoted by $H(V)$, such that $\rho(V)=\{V\}\cup H(V)$ forms the support of a minimal P-invariant Y_V that contains V. Let Y_V *(resp.* Y_S*)* be the minimal P-invariant associated with control place V (resp. *siphon* S) in an S^3PMR. $H(V)=[V]=||Y_V||\backslash\{V\}$ is called the *controller* (or *restrictive*) *region or the set of holder places of V* and $[S]= ||Y_S||\backslash S$ is called the *complementary set* of S.

Definition 4: S is said to reach its limit state when $M([S])=M_{max}([S])$-1; it is limit-controlled iff it is able to reach its limit state but not able to reach empty state.

$[V_S]$ (*resp.* $[S]$) is part of P; the region covered by $[V_S]$ (*resp.* $[S]$) is said to be restrictive since $M_{max}([V_S])$ (resp. $M_{max}([S])$) in the controlled model, denoted by $M_{max}([V_S])_c$ (resp. $M_{max}([S])_c$), is less than that in the original uncontrolled model. This implies that the restrictive region should be as small as possible. Note that the larger the marking, the more states it generates.

To restrict the uncontrolled model as little as possible, $M_{max}([S])_c$ [i.e., $M_{max}([S])$ in the controlled model] should be as close to $M_{max}([S])_o$ [i.e., $M_{max}([S])$ in the original uncontrolled model] as possible while keeping the siphon nonempty. $M_0(V_S)= M_0(S)$-1 satisfies the constraint.

III. Synthesis of SMS

In [18,20], we show that SMS S can be synthesized from resource or core subnets of the minimal strongly connected subnet that contains all resource places in S. The subnet (denoted by I_S) derived from (S, $^{\bullet}S$) is strongly connected (SC) and hence contains an SC *core* subnet N_S with at least two resource (or nonoperation) places; mutual waiting among them deadlocks the system. I_S can be built by attaching directed paths (called *handles*) H to N_S like a handle to a tea pot. H is called a TT- (PP-, TP-, PT-) handle if it goes from a transition (place, transition, place) to a transition (place, place, transition). Note that $S=P(I_S)$ is the place set of I_S and it cannot contain a TT-handle.

A core subnet can be obtained from an elementary circuit, called *core circuit*, by repeatedly adding handles. Details of handles and how an SMS is synthesized from a core subnet are in [18,20] and omitted here to shorten the paper.

Definition 5: *A core circuit c is an elementary circuit from which I_S (I-subnet) of an SMS can be constructed by adding handles upon c. A resource (resp. control) circuit is an elementary circuit where all non-operation places are resource (resp. control) places. A resource (resp. control) siphon is defined similarly. If c contains no control places, it is called a basic circuit; the corresponding synthesized siphon is called a basic siphon.*

Definition 6: A compound circuit $c=c_1$ o c_2 o…c_{n-1} o c_n is a circuit consisting of multiply interconnected elementary circuits c_1, c_2, …, c_n such that *$c_i\cap c_j=\{r_i\}$, $r_i\in R$ (i.e., c_i and c_{i+1} intersects at a resource place r_i) iff $|j-i|=1$. Partial mixture 1 (resp. 2), briefed as p^1-M (resp. p^2-M), is synthesized from the core subnet obtained by adding a TP-handle on c_1 (resp. c_n) side. The resulting siphon is called a full mixture subnet when no TP-handles can be added to synthesize a larger siphon. S_R (resp. S_C) the set of resource (resp. control) places in S. The SMS synthesized from compound circuit c is called an n-compound siphon S_0, denoted by $S_0=S_1$ o S_2 o…S_{n-1} o S_n. S_0 and its associated control and mixtures siphons are said to be n-dependent siphons.*

Definition 7: $[S]^e=\{p|p\in[S], {}^{\bullet\bullet}p\cap[S]= \varnothing\}$ is called the edge set of [S]. The set of siphons with the same $[S]^e$ is said to be in the same set B of n-dependent siphons if each S depends on n basic siphons.

*Definition 8: Siphon S is said to be **more critical** than Siphon S' if S' is controlled after S is controlled but not vice versa. A siphon S in B is said to be **critical** if S is emptiable and the rest of siphons in B are either already controlled or become controlled after a monitor is added to S. A control place for the critical n-dependent is called Level n-1 control place.*

For instance, each of a 2-dependent siphon and its associated control and mixtures siphons are called to be in the same set of 2-dependent siphons associated with the n-compound siphon since they satisfy Def. 7. Denote monitor places for basic siphons as level-1 control places c_1. Similar to resource places, there is a minimal strongly connected subnet N_{c1} (resp. N_{c2}, …, N_{ck}) that contains all c_1 (resp. c_2, …, c_k) places. New SMS, called level-1 control siphons, can be synthesized from any strongly connected subnet of N_{c1} (resp. N_{c2}, …, N_{ck}). Repeating this reasoning, one can construct level-2 (resp. -3, ..., -k) control places, N_{c2} (resp. N_{c3}, …, N_{ck}), as well as level-2 (resp. -3, ...,-k) control siphons.

IV. Motivation

This section motivates the reader using a 2-dependent system by showing a rule that when the critical siphon is unmarked, the tokens in $[S]$ follow a certain pattern. Any siphon not following the rule must not be critical. The n-dependent case will be shown in Section V. To ease the understanding of the theory, we have restricted the nets to the following:

Definition 9: *A k^m-system $K=(N, M_0)$ is a subclass of S^3PR with k resource paces r_1, r_2, …, r_k shared between two processes N_1 and N_2. N_1 (resp. N_2) uses r_1, r_2, …, r_k (resp. r_k, r_{k-1}, …, r_1) in that order. A nonshared resource place r' is used either by N_1or N_2, but not by both and $M_0(r')=1$ for any K. Holder places of r_j in N_1 and N_2 are denoted as p_j and p'_j respectively. A region π in K is a set of resource places that are strongly connected. Let $r\in\pi$. r is called a lower (resp. upper) edge place of π if $(r^{\bullet}\cap T_2)^{\bullet}\cap\pi=\varnothing$ [resp. $^{\bullet}({}^{\bullet}r\cap T_1)$*

Algorithm: Controller Synthesis for A K^m-system
INPUT: An uncontrolled K^m-system.
OUTPUT: Maximally permissive controlled model

1. For each basic siphon, add a monitor V with $M_0(V)= M_0(S)$-1
2. For each n-dependent region (n>2) Θ,
   2.1 For each nonboundary place in Θ,
   Follow Proposition 1 to find its token distribution,
   2.2 For each boundary place in Θ,
   Find all possible exchange operations and corresponding token patterns.
   2.3 Combine token patterns obtained in Steps 2.1 and 2.2. For each such a token pattern, add a monitor with $M_0(V)= M_0(S)$-θ where θ is defined in Proposition 1.
3. Output the resulting controlled model.

Figure 1. The Controller Synthesis Algorithm

$\cap\pi=\emptyset$]. r is called an internal place of π if r is not an edge (lower or upper) place. An internal r is called singular if $M_0(r)=1$. A singular place is called lower (resp. upper) boundary if the place in $(r^\bullet \cap T_2)^\bullet \cap \pi$ (resp. $^\bullet(^\bullet r \cap T_1) \cap \pi$) is neither singular nor edge.

Associated with each region π, there is a siphon S with $S_R=\pi$ since π is strongly connected [20].

As shown in [31], the following rule holds:

Proposition 1: Let (N_0, M_0) be a k^m-system and all lower order (than k^m-system) system has been controlled by adding monitors to critical siphons. S is an emptiable siphon and $\exists M \in R(N, M_0)$ such that $M([S])= M_{max}([S])$, then $\forall r_i$ in S, the token distribution (called unmarked) pattern is as follows:

1) *r_i is a lower edge place:*
   $M(r_i) =0$, $M(p_i) =0$, and $M(p'_i) =M_0(r_i)$.
2) *r_i is an upper edge place:*
   $M(r_i) =0$, $M(p_i) = M_0(r_i)$, and $M(p'_i) =0$.
3) *r_i is an internal place:*
   a) r_i is a shared resource place:
   i) if $M_0(r_i)=1$, then $M(r_i) =1$ and $M(p_i) = M(p'_i) =0$.
   ii) if $M_0(r_i)>1$, then $M(r_i) =0$ and $M(p_i) +M(p'_i) = M_0(r_i)$.
   b) r_i is a non-shared resource place,
   then $M(r_i) =0$ and $M(H(r_i))= M_0(r_i)$.
4) *There is only one emptiable siphon S in the set of k-dependent siphons.*
5) *M_{max}([S])= $M_0(R(S))$ – θ*, where *θ* is the number of internal resource places r_i (i.e., $r\neq r_1$ and ., $r\neq r_k$) with $M_0(r_i)$=1.
6) *The number of emptiable siphons is k(k-1)/2.*

If a siphon in the set of these n-dependent siphons does not follow the above unmarked pattern, one can conclude this siphon is not critical. In an earlier paper [10], we show that if all basic siphons are controlled, some compound siphon is also and may not need a monitor. The converse is not true; even though a compound siphon is controlled; all basic siphons remain uncontrolled and each needs a monitor.

V. CONTROL POLICY

This section develops a maximally permissive control policy based on Proposition 1 or the token pattern for unmarked siphons. Some live states will get lost or the controlled net is not deadlock-free if one just follows the token distribution in Proposition 1 to add monitors and examines some k^m-system to further illustrate the control policy. An example is shown in Fig. 5.a. It reaches 1060 states losing 166 good states by adding a monitor V to each critical siphon with [V]=[S]. This is because there are other token distributions when the siphon is unmarked such as the one obtained by exchange operation mentioned earlier.

Consider token distribution $M'_\Psi =2p_2 + 2(p_4 \oplus p'_4) +2p'_6$ for S with $S_R=\{p_7, p_8, p_9, p_{10}\}$, where $2(p_4 \oplus p'_4)$ indicates that $p_4 + p'_4=2$. Adding a monitor V with [V] = Ψ does not prevent S from becoming unmarked, which can be reached by (exchange operation) moving a token from p_4 to p_3 (associated with singular place p_8) so that $M'_{\Psi'} =2p_2 + p_3+ p'_4 +2 p'_6$ – another forbidden state besides M'_Ψ. Both can be prevented by setting [V]=[S]=$\{p_2, p_3, p_4, p'_4, p'_5, p'_6\}$(since [$S$] should not have holes), but it reaches 1060 states, while there are 1126 live states among all 1432 reachable states.

Note that we cannot move a token from p_2 (on the opposite side of p_3 in contrast to p_4) to p_3 since then the siphon S with $S_R=\{p_8, p_9, p_{10}\}$ becomes unmarked which is impossible since S has been controlled by adding a monitor.

To reach more states, all possible unmarked sets of unmarked operation places must be considered to add a monitor accordingly. They can be identified by finding all possible exchange operations.

Now we give a detailed picture of the controlled model. Note that the above exchange operation cannot be extended to the WP2 side by moving a token from p'_2 to p_3 since $M'_\Psi(p'_2)$=0. Thus, there are only 2 possible exchange operations, which have been listed above.

Similar discussion applies to the singular place p_{10} based on the above controlled model. Consider token distribution $M^*_{\Psi^*}=p_3 + p_4 \oplus p'_4 +2 p'_6$ for $S^*$with $S^*_R=\{p_8, p_9, p_{10}, p_1\}$. Adding a monitor V^* with [V^*] = Ψ^* does not prevent S from becoming unmarked, which can be reached by (exchange operation) moving a token from p'_4 to p'_3 (associated with singular place p_{10}) so that $M^*_{\Psi\wedge} = p_3 \oplus p'_3 + p_4 + 2\ p'_6$ – another forbidden state besides $M^*_{\Psi^*}$. Both can be prevented by setting $[V^*]=[S^*]=\{p_3,\ p_4,\ p'_4\ ,\ p'_5\ ,\ p'_6\}$($[S^*]$ has no holes; S^* is a resource siphon). Note that there is no need to compute S^* to find $[S^*]$. By adding a monitor for each of Ψ^* and $\Psi^\wedge$, the resulting model reaches 1120 states.

Finally, consider the last level monitor with both singular places p_8 and p_{10}. With 2 possible Ψ for each singular place, there are four possible sets of unmarked operation places. The resulting model reaches 1126 states which is maximally permissive. The controller synthesis algorithm is summarized in Figure 1.

VI. CONCLUSION

This paper further proposes a maximally permissive control policy for a subclass of Petri nets based on the theory of token distribution pattern of unmarked siphons in [31]. This has the advantage of avoiding the computation of new siphons derived from monitor places since the unmarked pattern solely determines the controller region (or control arcs) and the initial marking. As a result, this results in fewer monitors and more reachable states. Future work should be extended to more complicated nets.

REFERENCES

[1] N. Visvanadham, Y. Nahari, and T. L. Johnson, "Deadlock prevention and deadlock avoidance in flexible manufacturing systems using petri net models", *IEEE Trans. Robot. Automat.*, vol. 6, no. 6, pp. 713-723, 1990.

[2] 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, Apr. 1995.

[3] Z. Li and M. Zhou, "Elementary Siphons of Petri Nets and Their Application to Deadlock Prevention in Flexible Manufacturing Systems," *IEEE Trans. Syst.*, *Man*, *Cybern. A.*, vol. 34, no. 1, Jan. 2004, pp. 38-51.

[4] Z. W. Li and M. C. Zhou, "Clarifications on the Definitions of Elementary Siphons in Petri Nets," *IEEE Trans. Syst., Man, Cybern., A*: *Systems and Humans*, vol. 36, no. 6, pp. 1227-1229, November, 2006.

[5] Piroddi L., Cordone R., Fumagalli I., "Selective Siphon Control for Deadlock Prevention in Petri Net,s" *IEEE Trans. Syst., Man, Cybern., A.,* vol. 38, no. 6, pp. 1337-1348, 2008.

[6] Piroddi L., Cordone R., Fumagalli I., "Combined Siphon and Marking Generation for Deadlock Prevention in Petri Nets," *IEEE Trans. Syst., Man, Cybern., A.*, vol.39, no. 3, pp. 650-661, MAY 2009.

[7] Li, ZhiWu and Zhou, MengChu, "Two-stage method for synthesizing liveness-enforcing supervisors for flexible manufacturing systems using Petri nets," *IEEE Transactions on Industrial Informatics*, vol. 2, no. 4, pp. 313-325, November, 2006.

[8] M. Uzam, "An optimal deadlock prevention policy for flexible manufacturing systems using Petri net models with resources and the theory of regions. *International Journal of Advanced Manufacturing Technology*, vol. 19, no. 3, pp. 192-208, 2002.

[9] A. Ghaffari, N. Rezg, and X. L. Xie, "Design of a live and maximally permissive Petri net controller using the theory of regions," *IEEE Transactions on Robotics and* 15 *Automation*, vol. 19, no. 1, pp. 137-142, 2003.

[10] Uzam M and Zhou, MengChu, "An improved iterative synthesis approach for liveness enforcing supervisors of flexible manufacturing systems," *Int J Prod Res* vol. 44, no. 10, pp. 1987-2030, 2006.

[11] Uzam M., Zhou M. C., "An iterative synthesis approach to Petri net based deadlock prevention policy for flexible manufacturing systems," IEEE Trans. Syst., Man, Cybern., A., vol. 37, pp. 362-371, 2007.

[12] K. Yamalidou, J. Moody, M. Lemmon, and P. Antsaklis, "Feedback control of Petri nets based on place invariants," *Automatica*, vol. 32, no. 1, pp. 15-28, 1996.

[13] Uzam M., "The use of Petri net reduction approach for an optimal deadlock prevention policy for flexible manufacturing systems," Int. J. Adv. Manuf. Technol., vol. 23, no. 3-4, pp. 204-219, 2004.

[14] Uzam M., Li Z. W., Zhou M. C. "Identification and elimination of redundant control places in Petri net based liveness enforcing supervisors of FMS," Int. J. Adv. Manuf. Technol., vol. 35, pp. 150-168, 2007.

[15] Daniel Yuh Chao, "Computation of Elementary Siphons in Petri Nets For Deadlock Control," Comp. J., (British Computer Society), vol. 49, no. 4, pp. 470–479, 2006.

[16]. F. Chu and X. L. Xie, "Deadlock analysis of Petri nets using siphons and mathematical programming," *IEEE Trans. Robot. Automat.*, vol. 13, pp. 793–840, Dec. 1997.

[17] Y. S. Huang, M.D. Jeng, X.L. Xie and D.H. Chung, "Siphon-based Deadlock Prevention Policy for Flexible Manufacturing Systems," *IEEE Transactions on System, Man, and Cybernetics—PART A: SYSTEMS AND HUMANS*, vol. 36, no. 6, November, 2006.

[18] Daniel Yuh Chao, "Elimination of weighted arcs of a deadlock prevention policy for flexible manufacturing systems ," submitted.

[19] D. Y. Chao, "Reducing MIP Iterations for Deadlock Prevention of Flexible Manufacturing Systems," Int J Adv Manuf Technol., vol. 41, no. 3, pp. 343-346**.** 2009, doi: 10.1007/s00170-008-1473-x.

[20] D. Y. Chao, "An incremental approach to Extract Minimal Bad Siphons," Journal of Information Science and Engineering, vol. 23, no. 1, pp. 203-214, Jan. 2007.

[21] Li, Z.W. and Zhou, M.C., "Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach," Springer-Verlag, London, 2009.

[22] Li, Z.W. and Zhou, M.C., "Control of Elementary and Dependent Siphons in Petri Nets and their Application," IEEE Trans. Syst., Man, Cybern., A., vol. 38, no. 1, pp. 133-148, 2008.

[23] Li, Z.W. and Zhou, M.C., "On Controllability of Dependent Siphons for Deadlock Prevention in Generalized Petri Nets," IEEE Trans. Syst., Man, Cybern., A., vol. 38, no. 2, pp. 369-384, 2008.

[24] Li, Z.W., Zhang, J. and Zhao, M., "Liveness-enforcing Supervisor Design for a Class of Generalized Petri Net Models of Flexible Manufacturing Systems.," IEE Proceedings Control Theory & Applications, vol. 1, no. 4, pp. 955-967, 2007.

[25] Li, Z. W. and Zhou, M. C., "On Siphon Computation for Deadlock Control in a Class of Petri Nets," IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Human, vol. 38, no. 3, pp. 667-679, 2008.

[26] P. H. Starke, *INA: Integrated Net Analyzer*, 1992. Handbuch.

[27] E. R. Boer and Tadao Murata, "Generating basis siphons and traps of Petri nets using the sign incidence matrix," IEEE Trans. on Circuits and Systems, I--Fundamental Theory and Applications, vol. 41, no. 3, pp. 266-271, 1994.

[28] Z. W. Li, M. C. Zhou, and M. D. Jeng, "A Maximally Permissive Deadlock Prevention Policy for FMS based on Petri Net Siphon Control and the Theory of Regions," IEEE Transactions on Automation Science and Engineering, vol. 5, no. 1, pp. 182-188, 2008.

[29] Z. W. Li and M. C. Zhou, A Survey and Comparison of Petri Net-based Deadlock Prevention Policy for Flexible Manufacturing Systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Review, vol. 38, no. 2, pp. 172-188, 2008.

[30] Shih Shih, Y. Y. and D.Y. Chao "Sequence of Control in S3PMR," Computer Journal, doi:10.1093/comjnl/bxp081, 2010, vol. 53, no. 10, pp. 1691-1703.

[31] D.Y. Chao,"n-dependent Critical Siphon in An S^3PMR," manuscript, submitted.