Decentralized Supervision of Petri Nets with a Coordinator

Jianhong Ye, Zhiwu Li, *Senior Member, IEEE* and Alessandro Giua, *Senior Member, IEEE*

Abstract

This paper develops a decentralized supervision policy for a Petri net through collaboration between a coordinator and subnet controllers. The coordinator is chosen from the subnet controllers by solving an integer linear programming problem. An optimal objective function is used to minimize the communication cost between the subnet controllers and the coordinator. Furthermore, a protocol to reach an agreement on the firing conditions of common transitions among the subnet controllers is proposed. Observation agreement and control agreement can be achieved by the 'and' operator in logic algebra. Control agreement is used to decide the firing conditions of common transitions in the next step. The firing of common transitions, which will lead to a new marking that violates the given constraints, will be forbidden by the control agreement. A feasibility analysis of the proposed decentralized control framework is discussed. Finally, four examples are presented to illustrate the proposed approach.

Index Terms

Petri nets, Decentralized supervision, Agreement, Coordinator, Discrete event systems.

Published as: J.H. Ye, Z.W. Li, A. Giua, "Decentralized Supervision of Petri Nets with a Coordinator,"
IEEE Trans. on Systems Man & Cybernetics, Systems, Vol. 45, No. 6, pp. 955-966, 2015.
DOI: 10.1109/TSMC.2014.2373316.

J. Ye is with the School of Computer Science and Technology, Huaqiao University, Xia'men and also with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China (e-mail: leafever@hqu.edu.cn).

Z. Li (corresponding author) is with the Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia, and also with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China (e-mail: zhwli@xidian.edu.cn).

A. Giua is with Aix Marseille Université, CNRS, ENSAM, Université de Toulon, LSIS UMR 7296, 13397, Marseille, France and also with the Department of Electrical and Electronic Engineering, University of Cagliari, Cagliari 09124, Italy (email: alessandro.giua@lsis.org).

Nomenclature

N	A Petri net $N = (P, T, F, W)$
N'	A decentralized system split from N
ℓ	The number of constraints
n	The number of subnets in N'
$\mathbb{N}$	The set of non-negative integers, $\mathbb{N} = \{0, 1, 2, \ldots\}$
S_i	A subnet of N', $i \in \{1, 2, \ldots, n\}$
c_k	A supervisor (subnet controller), $k \in \{1, 2, \ldots, \ell\}$
T_s	A set of common transitions, $\|T_s\| = m \in \mathbb{N}$
$T_{c,i}$	A transition subset of T controlled by a supervisor c_i, $i \in \{1, 2, \ldots, \ell\}$
$T_{o,i}$	A transition subset of T observed by a supervisor c_i, $i \in \{1, 2, \ldots, \ell\}$
P_i^k	The place set in subnet S_i associated with the kth constraint
T_i^k	The transition set in subnet S_i controlled and observed by a supervisor generated by the kth constraint
M	The marking of a whole plant
M_{τ,T_s}	The marking of the preset of T_s at time τ
λ_τ	An observation agreement vector at time τ
κ_τ	A control agreement vector at time τ
$\delta_{i,j}^k$	$\delta_{i,j}^k = 1$ if transition t_j in subnet S_i participates in the control decision for the kth constraint, otherwise $\delta_{i,j}^k = 0$
$c_{i,l}$	The delay between the subnet controllers of subnets S_i and S_l by propagating a bit
L	The distance between two subnet controllers
R	The propagation speed of signals going through a wire or a fiber
B	The rate of transmission
$\Im_{i,j}$	The preset of a common transition t_j in subnet S_i
$\Re_{i,j}$	The postset of a common transition t_j in subnet S_i
τ_c	A maximum agreement convergence time
Δ_t	A time unit for a subnet controller to start a communication repair service
Δ_o	A time unit for the coordinator to start a communication repair service
Δ_r	A time unit for the coordinator to send periodically a live message to subnet controllers

I. Introduction

A discrete event system (DES) is a dynamic system that evolves in accordance with the abrupt occurrence of physical events. Such systems arise in a variety of contexts ranging from computer operating systems to the control of complex multimode processes. The supervisory control theory introduced by Ramadge and Wonham [1–3, 32] provides strategies to restrict the behavior of a plant by synthesizing supervisors which ensure that most of the given constraints are satisfied.

Decentralized control has received a great deal of attention in the DES area over the past decade. Usually, decentralized control problems in DES have been studied by means of formal languages and automata. An early study on decentralized DES is proposed in [4]. In this work, multiple supervisors are modeled by automata, which are responsible for controlling a plant to ensure that only the desirable event sequences can occur. Moreover, the decentralization of the supervision under partial observation is studied in [5], which is an extension of the work reported in [4]. The control of DES with partial observation has been addressed by using coalgebra and coinduction in [6]. These results are generalized to the decentralized and modular supervisory control. In modular control of DES, the overall system is obtained as a parallel composition of local systems. The existing work mentioned above does not consider the communication among supervisors. In [7], a novel information structure model is presented to deal with this problem. Each supervisor utilizes a combination of direct observations (obtained from sensors reading available to another supervisor), since communication may be costly. A strategy to minimize communication costs between different sites is developed in [8, 9].

Petri nets are one of the mathematical tools to deal with modeling and control of a DES [10, 11, 48–50] and find extensive applications in manufacturing, particularly for deadlock analysis and control based on either structural analysis [12, 17–20, 33, 35, 37, 39, 40] or reachability graph analysis [13–16, 34]. They use mathematical and graphical representations to describe the events and conditions of the behavior in terms of the causal relationships between events. In recent years, many studies used ordinary and modified Petri nets to model and control DESs. Petri nets have a compact system representation and are potentially helpful in reducing the complexity of decentralized supervisory control problems. However, Petri nets have not received much attention compared to the contributions dealing with the decentralized control by automata. There are mainly two strategies to investigate decentralized supervisory control by using Petri nets: supervision with communication and supervision with no communication. We mention [21, 22] for the decentralized control with no communication, and [23, 24] for decentralized control with communication. It is worth noting that a special class of modified Petri nets, namely distributed agent oriented Petri nets (DAOPNs) [25–27], is developed to model DES. DAOPNs inherit the definitions and properties of Petri nets. In compliance with the basics of Petri nets, the proposed DAOPNs add agent oriented modeling components to increase the model's agency and autonomy. Resolution rules are defined to control the firing of conflicting transitions and the release of competing tokens. Although DAOPNs are more suitable to model DES, they are benefited from adding a lot of attributes such as properties of time, color sets, communication, database accessing, and status sharing with ordinary Petri nets. These attributes will affect the analysis of behavior. The work in this paper is a prelude to a systematic study on the agreement control. In order to pave the way for future work, we only consider ordinary Petri nets in this research.

Iordache and Antsaklis [24] introduced the concept of decentralized admissibility (d-admissibility) that is an extension to the decentralized system of the centralized admissibility in Petri nets. In this case, communication is available with constraints and a supervisor can communicate with another supervisor. However, the basic notion of d-admissibility in a decentralized system requires each local supervisor to observe all observable transitions. Even if the supervisor proposed in [24] is distributed, a global observation is still required for each supervisor. Another approach for constraint decomposition is proposed in [21, 22]. The advantage of the approach is that the communication among distributed systems is not necessary. However, this leads to overly restrictive control laws.

Gasparri et al. [23] describe a general framework for a decentralized control law with collaboration among supervisors. This approach is said to be consensus-based since it uses collaborative approaches for estimation and control (see [28] for a general introduction to consensus). Communication is available and limited within one-hop neighbors. A sufficient condition to achieve decentralized admissibility is provided, which focuses on the communication topology of the network of supervisors. However, the consensus must be synchronized through a global clock, which is hard to be implemented in an asynchronous DES. Furthermore, the protocol in [23] to reach a consensus is motivated by communication theory [28]. The consensus always reaches common knowledge about the initial state. However, the states in a Petri net are changed by the firing of transitions, and hence the consensus should be considered from the current state.

To overcome a drawback of the global clock and update the consensus to be suitable for Petri nets, this paper proposes a general framework for a truly decentralized control law with communication that could be effectively implemented. The communications among subnet controllers are allowed and the constraints are enforced in a centralized environment. Then, a coordinator is chosen from subnet controllers by using integer linear programming (ILP), which reduces the communication cost, thus maintaining the permissiveness of the centralized solution. Furthermore, a protocol to reach an agreement on the firing conditions of common transitions among the subnet controllers and a mechanism to ensure dependable communication connection are proposed. Finally, feasibility analysis of the proposed decentralized control framework is discussed. The agreement algorithm is extended from the work in [23]. The original contributions of this paper include the optimization procedure to choose the coordinator and the dependable communication connection to fault repair.

Decentralized Petri nets have found applications in different real world systems, such as mission control and task sequencing for a team of autonomous vehicles. For instance, in [29, 30], Petri nets are used to manage mutual exclusion, ordering and synchronization for missions defined on each vehicle. The solution guarantees a deadlock-free centralized Petri net.

The rest of the paper is organized as follows. Section II presents the concepts of Petri nets and their centralized/decentralized supervision. A motivating example is described in Section III. In Section IV, the proposed agreement-based decentralized supervision of Petri nets is described. In Section V, convergence time and feasibility analysis are reported. Four examples to demonstrate the theoretical analysis are presented in Section VI. Conclusions and future work are discussed in Section VII.

II. Theoretical Background

This section provides the basics of the theoretical background involved in the paper. For more details, we refer the reader to [31, 36, 38]. A Petri net N is a four-tuple (P, T, F, W), where P and T are finite, non-empty, and disjoint sets. P is the set of places and T is the set of transitions. $F \subseteq (P \times T) \cup (T \times P)$ is called a flow relation of the net, represented by arcs with arrows from places to transitions or from transitions to places. $W : (P \times T) \cup (T \times P) \to \mathbb{N}$ is a mapping that assigns a weight to an arc: $W(x, y) > 0$ if $(x, y) \in F$, and $W(x, y) = 0$ otherwise, where $x, y \in P \cup T$ and $\mathbb{N} = \{0, 1, 2, \cdots\}$ is a set of non-negative integers. A net is self-loop free (pure) if $\nexists x, y \in P \cup T, f(x, y) \in F \wedge f(y, x) \in F$. $N = (P, T, F, W)$ is called an ordinary net, denoted as $N = (P, T, F)$, if $\forall f \in F, W(f) = 1$.

A marking M of N is a mapping from P to $\mathbb{N}$. $M(p)$ denotes the number of tokens contained in place p. A place p is marked at marking M if $M(p) > 0$. (N, M_0) is called a net system or marked net and M_0 is called an initial marking of N. Markings and vectors are usually described as multisets or formal sum notations. As a result, a marking M can be denoted by $\sum_{p \in P} M(p)p$. For example, a marking $M = (3, 1, 0, 4, 0, 2)^T$ in a net with six places can be denoted by $M = 3p_1 + p_2 + 4p_4 + 2p_6$. Let $x \in P \cup T$ be a node in a net $N = (P,\ T,\ F,\ W)$. The preset of x is defined as ${}^\bullet x = \{y \in P \cup T | (y, x) \in F\}$. While the postset of x is defined as $x^\bullet = \{y \in P \cup T | (x, y) \in F\}$. Let X be a set of nodes with $X \subseteq P \cup T$. We have ${}^\bullet X = \cup_{x \in X} {}^\bullet x$ and $X^\bullet = \cup_{x \in X} x^\bullet$. A Petri net $N = (P, T, F, W)$ can be represented by its incidence matrix D, where D is a $|P| \times |T|$ integer matrix with $D(p, t) = W(t, p) - W(p, t)$. For a place p (transition t), its incidence vector, a row (column) in D, is denoted by $D(p, \cdot)$ $(D(\cdot, t))$.

A transition $t \in T$ is enabled at marking M if $\forall p \in {}^\bullet t$, $M(p) \geq W(p, t)$, which is denoted as $M[t\rangle$. If t is enabled, it can fire. Its firing yields another marking M' such that $\forall p \in P, M'(p) = M(p) - W(p, t) + W(t, p)$, which is denoted by $M[t\rangle M'$. Marking M' is said to be reachable from M if there exist a transition sequence $\sigma = t_1 t_2 \ldots t_n$ and markings $M_1, M_2, \ldots,$

and M_{n-1} such that $M[t_1\rangle M_1[t_2\rangle M_2 \ldots M_{n-1}[t_n\rangle M'$. This is denoted by $M[\sigma\rangle M'$. The set of markings reachable from M in N defines the reachability set of (N, M), denoted as $R(M)$. Let $N = (P, T, F, W)$ be a net and σ be a finite sequence of transitions. The Parikh vector of σ is $\overrightarrow{\sigma}$: $T \rightarrow \mathbb{N}$, mapping t in T into the number of occurrences of t in σ. Define $\overrightarrow{t_1} = \{1, 0, \ldots, 0\}^T$, $\overrightarrow{t_2} = \{0, 1, \ldots, 0\}^T$, ..., and $\overrightarrow{t_k} = \{0, 0, \ldots, 0, 1\}^T$, assuming $k = |T|$.

A plant transition is said to be uncontrollable if its firing cannot be inhibited by an external action. It is said to be unobservable if its firing cannot be directly detected or measured. A GMEC (generalized mutual exclusion constraint) [41–44] is defined as a condition that limits a weighted sum of tokens contained in a subset of places and includes both sequential and parallel mutual exclusions. Many constraints that deal with exclusions between states and events can be transformed into GMEC. Given a Petri net N, a constraint is denoted by $lM \leq c$ with $l \in \mathbb{Z}^{1\times|P|}$ and $c \in \mathbb{Z}$, while a set of constraints is denoted by $LM \leq H$ with $L \in \mathbb{Z}^{\ell\times|P|}$, $H \in \mathbb{Z}^{\ell}$, and $\ell \geq 1$, where $\mathbb{Z}$ is a set of integers, ℓ is the number of constraints, and M is a reachable marking of N. The GMEC provides a monitor solution which takes the form of a set of control places D_c:

$$D_c = -LD \tag{1}$$

$$M_{0,c} = H - LM_0 \tag{2}$$

where D is the incidence matrix of the plant N, D_c is the incidence matrix of the supervisor, $M_{0,c}$ is the initial marking of the supervisor, and M_0 is the initial marking of N. A supervisor is admissible if it only controls controllable transitions and only detects observable transitions. The constraints $LM \leq H$ are admissible if the supervisor defined by (1) and (2) is admissible, where $H \in \mathbb{Z}^{\ell}$ and M is the marking vector of N. When inadmissible, the constraints $LM \leq H$ are generally transformed into the admissible form $L_aM \leq H_a$ such that $L_aM \leq H_a \Rightarrow LM \leq H$. Then, the supervisor enforcing $L_aM \leq H_a$ is admissible, and enforces $LM \leq H$ as well. A plant N with the sets of controllable and observable transitions T_c and T_o, respectively, at the initial marking M_0 is denoted by (N, M_0, T_c, T_o).

Let $N' = (P', T', F')$ be a decentralized net that is split from $N = (P, T, F)$. N' consists of a set of subnets $S_1 = (P_1, T_1, F_1), S_2 = (P_2, T_2, F_2), \ldots,$ and $S_n = (P_n, T_n, F_n)$ such that $P' = P_1 \cup P_2 \cup \ldots \cup P_n$, $T' = T_1 \cup T_2 \cup \ldots \cup T_n$, and $F' = F_1 \cup F_2 \cup \ldots \cup F_n$. A transition $t \in T'$ is a common transition if t belongs to S_i and S_j in N', $i, j \in \{1, 2, \ldots, n\}$, and $i \neq j$. Let $T_s \subseteq T'$ denote a set of common transitions of N'. Two new notations are used in this paper: subnet controllers and a coordinator. A subnet controller is a supervisor that is chosen from several supervisors in a decentralized net and a coordinator is selected from the different subnet controllers. Let us now introduce the concept of d-admissibility originally proposed in [24] for a decentralized scenario. Suppose that there is a decentralized system N' split from $N = (P, T, F, W)$, ℓ is the number of constraints in N, c_k is a supervisor generated from the kth constraint in N, $k \in \{1, 2, \ldots, \ell\}$, $T_{c,k}$ is a transition subset of T controlled by c_k in N, and $T_{o,k}$ is a transition subset of T observed by c_k in N. The kth constraint is d-admissible if c_k can still be controlled and observed by $T_{c,k}$ and $T_{o,k}$ in N', respectively, which means that the kth constraint is still valid in N'. A set of constraints is said to be d-admissible if each constraint in it is d-admissible.

An observation agreement λ_τ of N' is a mapping from the common transition set T_s to $\{0, 1\}$. The value of $\lambda_\tau(\alpha(t))$ denotes whether the common transition t is allowed to fire without considering constraints at $\tau + 1$ time. Moreover, a control agreement κ_τ of N' is also a mapping from the common transition set T_s to $\{0, 1\}$ and $\kappa_\tau(\alpha(t))$ denotes whether the common transition t is allowed to fire by considering constraints at $\tau + 1$ time. For convenience of discussion, all the transitions in this paper are assumed to be observable and controllable.

III. A MOTIVATING EXAMPLE

Let us consider the net system shown in Figure 1, which is taken from [23] (ignoring the dashed places and arcs). The example illustrates the manufacturing of a product composed of two different types of parts. In the first phase, two different types of parts (p_{a1} and p_{b1}) are produced in parallel by machines MA_{1a} and MA_{2a}, respectively. Each part is moved to a common area by robot MA_3. In the second phase, the semi-finished products are assembled by using machine MA_4 to obtain the final product (p_{ab}) that leaves the manufacturing cell. Three buffers (B_1, B_2 and B_3) are introduced, where B_1 and B_2 are used to decouple the production of semi-finished products from their transportation and B_3 is used for the storage of the assembled products.

When the supervisor of the manufacturing cell is concerned, the following control requirements should be satisfied for net model shown in Figure 1. If buffer B_1 or B_2 is full, the entrance of parts in machine MA_{1a} or MA_{2a} has to be denied even if the machine is idle. Suppose that both the buffers have the same capacity 4, then the requirements can be written, for the buffers B_1 and B_2, respectively, as:

$$M(p_2) + M(p_4) \leq 4 \tag{3}$$

and

$$M(p_8) + M(p_{10}) \leq 4 \tag{4}$$

Clearly, p_2 and p_4 are associated with constraint (3). At the same time, a supervisor c_1 is generated from constraint (3). The transitions set $\{t_1, t_3\}$ is controlled and observed by supervisor c_1.

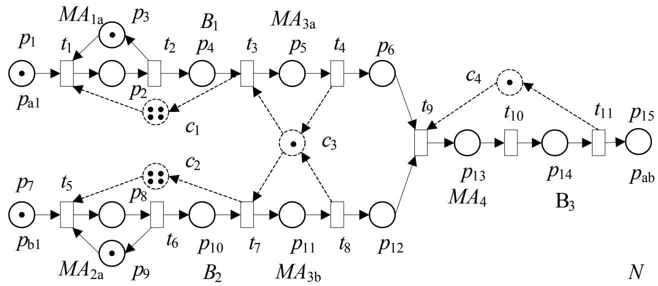

Figure 1. A Petri net model of a manufacturing work cell.

Another requirement is that the mutual exclusion of the robot MA_3 should be guaranteed. The corresponding constraint is

$$M(p_5) + M(p_{11}) \leq 1 \tag{5}$$

Finally, the constraint in the assembly phase is

$$M(p_{13}) + M(p_{14}) \leq 1 \tag{6}$$

Note that the marking M in Eqs. (3)–(6) is any marking in $R(M_0)$. The supervisors whose places and arcs are represented by the dashed lines in Figure 1 can be obtained by using the GMEC.

In the real world, a system is usually partitioned into multiple decentralized sub-systems located in different geographical situations. In order to enforce constraints for a distributed model and benefit from the existing constraint control knowledge, we first consider a distributed network as a whole and obtain supervisors of the whole Petri net, as shown in Figure 1. Then, we split the whole net into a decentralized one according to the actual situation, as shown in Figure 2. Some transitions constitute the interfaces between the different subnets and are called common transitions. At the same time, some monitor places will be split into several places which belong to different subnets. In order to avoid overly restrictive control laws as in [21, 22], we assume that every split monitor place contains the same initial tokens with the original monitor place. However, under this assumption, there are some markings that violate the constraints. As shown in Figure 2, subnets are S_1, S_2, S_3, S_4 and S_5, the set of common transitions is $T_s = \{t_3, t_4, t_7, t_8\}$, and the monitor place c_3 is decentralized into two monitor places c_{3a} and c_{3b}, which holds one token in c_{3a} and c_{3b}, respectively. Note that the transitions t_3 and t_7 in the subnets S_3 and S_4 can fire at the same time. By this concurrent firing, we have $M(p_5) + M(p_{11}) = 2$, which violates constraint condition (5). Thus, the additional control law needs to be built to prevent the case from occurring and keep the state space of the controlled system in the set of legal markings.

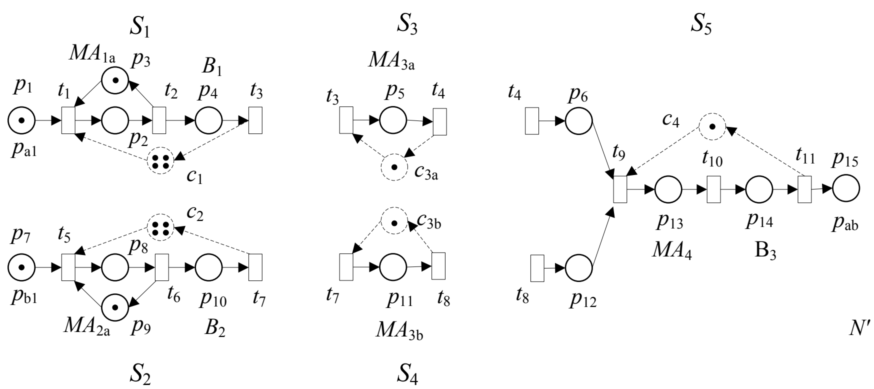

Figure 2. A decentralized Petri net of a manufacturing cell.

IV. Decentralized Supervision of Petri nets with a coordinator

The previous section shows that the design of a new control law for a decentralized system is necessary. This paper develops a communication mechanism between the coordinator and the subnet controllers to issue a proper control action. For reaching an agreement, we first need to choose a coordinator with the details shown in Section IV-A. In some cases, if there is no supervisor in one subnet, we can choose a place from the preset of the common transitions in this subnet as a controller. The responsibility of the coordinator is to reach an agreement and broadcast this decision value to all other subnet controllers. We can see that the communication is necessary between the coordinator and the subnet controllers. Let us now introduce the assumptions made in the communication mechanism for the proposed scenario:

- Subnet controllers are directly connected to the coordinator by using a Star LAN [45] as an underlying architecture. The LAN is private, in other words, there do not exist other messages passing on the network except those generated by the considered subnets.
- Every subnet controller communicates with the coordinator by exchanging messages through a network UDP/IP protocol [45].
- Every subnet controller has an unique id and can be identified by the coordinator on the Star LAN.
- Let $c_{i,l}$ be a variable that defines the delay between the subnet controllers of the subnets S_i and S_l by propogating a bit, i.e., $c_{i,l}$ is the propagation delay that can be computed by L/R, where L is the distance between the two subnets and R is the propagation speed of the signals going through a wire or a fiber, and is generally two thirds of the speed of light ($200,000$ km/sec).
- In a network based on packet switching, transmission delay is the required time to push all of the packet's bits into the wire. In other words, this is the delay caused by the data-rate of the link. This delay is proportional to the packet's length in bits. It is given by the number of bits divided by the rate of transmission (represented symbolically by B, which is usually sent in bits per second).
- The number of tokens in one place is considered as a standard packet.

The flowchart of the control algorithm is shown in Figure 3 and includes *'phase 1'* to *'phase 4'* (modeling the corresponding phases described in Section IV-B), the communication failure suspicion (modeling the dependable communication connection module in Section IV-C), the broadcast sub-system (modeling the reliable UDP/IP protocol and Star LAN), and the propositions (modeling the constraints).

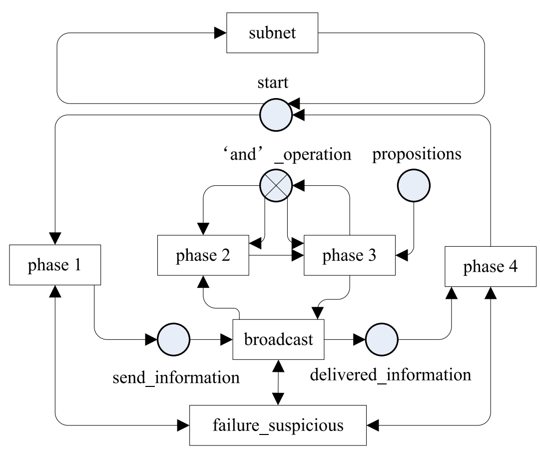

Figure 3. Abstract structure of an agreement system.

If there is a token in the place *'start'*, two sequences *'subnet'* and *'phase1'* can be processed. The fact depends on whether the firing of the subnet transitions involves common transitions. If the token flows in a subnet system, it will go through the *subnet system* and back to the place *'start'* (the upper part of Figure 3). Otherwise, the token will go through the *'phase 1'*, *'broadcast'*, *'phase 2'*, *'phase 3'*, *'broadcast'* again, *'phase 4'* and back to the place *'start'*, sequentially. The token flow through the Petri net simulates the behavior of the agreement protocol implemented on Algorithms 1, 2, and the method of dependable communication connection.

A. Coordinator Choice

Algorithm 1 can be used to choose an optimal coordinator via a cost function and an ILP problem. First, suppose that t_j is a common transition in subnet S_i, where $1 \leq j \leq m$, $1 \leq i \leq n$, m is the number of common transitions, and n is the number of subnets. Then, t_j is controlled or observed by at least a supervisor, i.e., $\delta^k_{i,j} = 1$ if and only if $\{t_j\} \cap (\underset{1\leq k\leq \ell}{\cup} T^k_i) \neq \emptyset$. We can say that transition t_j in subnet S_i participates in the control decision for the kth constraint.

Second, let us introduce two notations to be used. $\Im_{i,j}$ denotes the preset of a common transition t_j in subnet S_i and $\Re_{i,j}$ denotes the postset of t_j in subnet S_i, where $i \in \{1, \cdots, n\}$ and $j \in \{1, \cdots, m\}$. Assume now that there are two sets $\Im_{i,j}$ and $\Re_{i,j}$; places in $\Im_{i,j}$ and $\Re_{i,j}$ participate in the observable or controllable decision making for the kth constraint, and then the following relationships hold:

$$|\Im_{i,j} \cap (\bigcup_{1\leq k\leq \ell} P_i^k)| \geq 0 \tag{7}$$

$$|\Re_{i,j} \cap (\bigcup_{1\leq k\leq \ell} P_i^k)| \geq 0 \tag{8}$$

where P_i^k denotes the place set in subnet S_i associated with the kth constraint and ℓ is the number of constraints. Furthermore, at least one common transition enforces the kth constraint, i.e.,

$$\exists j \in \{1, \cdots, m\}, |\{t_j\} \cap (\bigcup_{\substack{1\leq i\leq n,\\ 1\leq k\leq \ell}} T_i^k)| \geq 1 \tag{9}$$

where T_i^k is the transition set in subnet S_i controlled and observed by a supervisor that is generated from the kth constraint. Finally, for transition $t \in T_i^k$, $i \in \{1, \cdots, n\}, k \in \{1, \cdots, \ell\}$, we obtain

$$\sum_{j=1}^{m} \delta_{i,j}^k \geq 0 \tag{10}$$

$\forall r \in \{1, \cdots, n\}$, we can use an ILP problem to minimize the cost, as seen in Algorithm 1, i.e.,

$$\min \quad \sum_{l=1}^{n} (2 \cdot c_{r,l} + \sum_{j=1}^{m} (|\Im_{l,j}| + |\Re_{l,j}| \cdot \delta_{l,j})/B) \tag{11}$$

Algorithm 1 Choice of an optimal coordinator

Input: N', constraints (7)-(10), L, R, and B, where L is the distance between two subnet controllers, R is the propagation speed of signals going through a wire or a fiber, and B is the rate of transmission.
Output: An optimal coordinator from subnet controllers.
Step 1: Solve (11) subject to (7)–(10).
Step 2: Choose the minimal solution c_i in (11) as the coordinator, where $i \in \{1, \cdots, n\}$.

B. The agreement algorithm

In subnets, the transitions except common transitions can fire concurrently since they do not affect the agreement. The agreement algorithm only pays attention to the common transitions and is divided into 4 phases that are sequentially executed.

Phase 1 In each subnet, a subnet controller is responsible to upload the markings of the places in the preset or postset of common transitions in its subnet to the coordinator. This phase is called *estimates*.

Phase 2 The coordinator receives all the markings of the places in the preset of common transitions, and takes 'and' operations with these markings to give an observation agreement. This phase is called *observation agreement*.

Phase 3 A control agreement will be further obtained after the coordinator gathers the markings of the places in the postset of common transitions and performs the 'and' operations for the constraints again. The control agreement indicates the firing conditions of the common transitions in the next step. This phase is called *control agreement*.

Phase 4 The decision broadcasts to the corresponding subnet controllers. The plant evolves consequently. This phase is called *acknowledgements*.

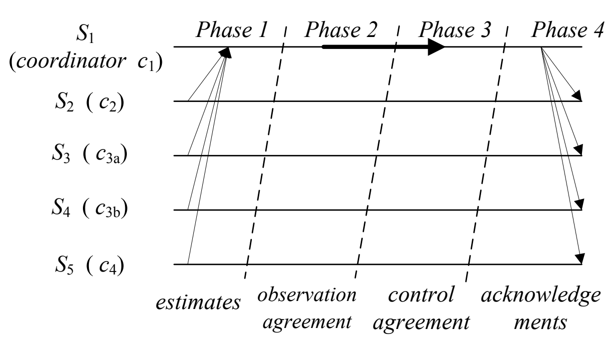

Figure 4. A simple communication model of a decentralized manufacturing cell.

Figure 4 illustrates that the manufacturing of a product is composed of five subnet controllers. Assume that supervisor c_1 is chosen as a coordinator. The termination time of the agreement is defined as the time elapsed from the first message sent by a subnet controller in *Phase 1* to all agreement information correctly received in *Phase 4*.

Let M be the current marking of a net N', and $t_1, t_2, \ldots$, and t_m be the common transitions. Suppose that the number of constraints is ℓ. Let us introduce two notations P_i^k and T_i^k before we use them. P_i^k denotes the set of the places in subnet S_i, which are associated with the kth constraint, and transition set $T_i^k \subseteq T_{c,k} \cup T_{o,k}$ denotes the transitions in S_i that are controlled and observed by a supervisor generated by the kth constraint, where $i \in \{1, 2, \ldots, n\}$ and $k \in \{1, 2, \ldots, \ell\}$. For instance, as shown in Figure 2, we have the place set $P_1^1 = \{p_2, p_4\}$ and the transition set $T_1^1 = \{t_1, t_3\}$, where p_2 and p_4 come from the places that are associated with constraint (3) in subnet S_1 and t_1 and t_3 come from the transitions controlled and observed by supervisor c_1 generated from constraint (3).

Algorithm 2 The agreement computation

Input: (N', M), $S_i \subseteq N'$, $1 \leq i \leq n$, $|T_s| = m$, $m \in \mathbb{N}$
Output: A decentralized control law
Step 1: Apply Algorithm 1 to choose an optimal coordinator.
Step 2: Assume that the marking M_{τ,T_s} at time τ is $M({}^\bullet(T_s))$, where $T_s = \cup_{1 \leq i,j \leq n}(S_i \cap S_j)$.
Step 3: For any $t_y \in T_s$, $1 \leq y \leq m$, the observation agreement λ_τ is designed as an integer vector $(\alpha(t_1), \alpha(t_2), \cdots, \alpha(t_m))$ describing the firing state of the common transitions without considering constraints at time $\tau + 1$, where $\lambda_\tau(\alpha(t_y)) = \wedge_{\substack{1 \leq x \leq n \\ 1 \leq y \leq m}} M_{\tau,T_s}(\Im_{x,y})$, M_{τ,T_s} is the marking of the preset of T_s at time τ, $\Im_{x,y}$ is the preset of t_y in subnet S_x, and the notation $\wedge$ denotes the 'and' operation in logic algebra. The expression $\wedge M_{\tau,T_s}(\Im_{x,y})$ represents the 'and' operations between the token values in every place in the set $\Im_{x,y}$. The nonzero value can be regarded as logical constant '1'. The function $\alpha(t)$ maps t in T_s into the number of $0's$ or $1's$, where $\alpha(t_i) = 1$ means that the transition t_i can fire at time $\tau + 1$, otherwise $\alpha(t_i) = 0$.
Step 4: For any $t_y \in T_s$, $1 \leq y \leq m$, design the integer vector $\kappa_\tau = (\alpha(t_1), \alpha(t_2), \cdots, \alpha(t_m))$ as the control agreement describing the firing policy for all the common transitions by considering the constraints. If there is a kth constraint $M(p_c) + M(p_g) \leq h$ with $h \in \mathbb{N}$, p_c, $p_g \in P_i^k$, and $i \in \{1, 2, \ldots, n\}$, then there exists $x \in \{1, 2, \ldots, n\}$ such that $P_x^k \cap \Re_{x,y} \neq \emptyset$ and $\kappa_\tau(\alpha(t_y)) = \lambda_\tau(\alpha(t_y)) \wedge (h - M(P_x^k))$, otherwise $\kappa_\tau(\alpha(t_y)) = \lambda_\tau(\alpha(t_y))$.
Step 5: The control agreement κ_τ is broadcasted to the subnet controllers by the coordinator.

Assume that in Figure 2 the current time is τ. Then we have $T_s = \{t_3, t_4, t_7, t_8\}$, the marking $M = p_3 + p_4 + p_9 + p_{10} + 3c_1 + 3c_2 + c_{3a} + c_{3b} + c_4$ for the whole plant, and the marking $M_{\tau,T_s} = p_4 + c_{3a} + p_{10} + c_{3b}$ for the places in the preset of the common transitions. Algorithm 2 calculates two vectors: the observation agreement and the control agreement. In our article, the observation agreement determines the firing condition of common transitions at time $\tau + 1$ without considering the constraints. When we consider the validity of constraints, we can furthermore calculate the control agreement to determine the firing condition of common transitions at time $\tau + 1$. We only spread the control agreement between the coordinator and the subnet controllers in the network. If the decentralized system in Figure 2 is considered by Algorithm 2, a solution of observation agreement and control agreement can be obtained with $\lambda_\tau = (\overset{t_3}{1}, \overset{t_7}{1}, \overset{t_4}{0}, \overset{t_8}{0})$ and $\kappa_\tau = (\overset{t_3}{1}, \overset{t_7}{1}, \overset{t_4}{0}, \overset{t_8}{0})$, respectively, implying that transitions t_3 and t_7 can fire at time $\tau + 1$. Transition t_3 is assumed to fire, and the coordinator ensures that there are no other common transitions that fire between the firing of transition t_3 in subnets S_1 and S_3. After transition t_3 fires, we can obtain the observation agreement $\lambda_{\tau+1} = (\overset{t_3}{0}, \overset{t_7}{1}, \overset{t_4}{1}, \overset{t_8}{0})$ and the control agreement $\kappa_{\tau+1} = (\overset{t_3}{0}, \overset{t_7}{0}, \overset{t_4}{1}, \overset{t_8}{0})$. Transitions t_4 and t_7 are allowed to fire at $\lambda_{\tau+1}$. However, the firing of transition t_7 violates constraint (5) and to be blocked in $\kappa_{\tau+1}$. In this case, transition t_4 becomes the only firable common transition at time $\tau + 2$.

Algorithm 2 is summarized in order to be easily understood by the reader. The entire agreement progress can be described as follows: subnet controllers upload the marking of the places in the preset and postset of common transitions to the coordinator, the observation agreement and the control agreement are calculated by the coordinator, and then the control agreement is spread to the subnet controllers as an acknowledgement to determine the firing condition of common transitions at time $\tau + 1$. In Algorithm 2, the agreement is calculated by the coordinator and every subnet controller only communicates with the coordinator. We do not design a mechanism in detail to show how the common transitions in subnets are correctly controlled by subnet controllers according the control agreement, while we only assume that the subnet controllers and coordinator have this capacity intuitively.

C. Ensuring a Dependable Communication Connection

The coordinator has a local suspected failure module, ensuring that the faulty subnet controllers can be found in time. The method consists of the following rules.

1) Initial: A system (N', M), $|T_s| = m \in \mathbb{N}$; c_i, a coordinator in subnet S_i, $i \in \{1, 2, \ldots, n\}$; c_j, a subnet controller in subnet S_j, $j \in \{1, 2, \ldots, n\} \backslash \{i\}$.
2) Subnet controller c_j sends marking $M(S)$ with $S = ({}^\bullet(T_s) \cup (T_s)^\bullet) \cap P_j$ to c_i and starts a timer at the same time in *Phase 1*. If c_j has not received any acknowledgement from c_i after Δ_t time, c_j will start the communication repair service to coordinator c_i.
3) Coordinator c_i starts a timer when the first marking message from subnet controllers arrives. As soon as coordinator c_i receives markings from one subnet controller, it removes the subnet controller name from the suspected list. If the suspected list is non-empty after Δ_o time, coordinator c_i will start the communication repair with the subnet controller that remains in the suspected list.

4) The coordinator c_i calculates observation agreement and control agreement after it has received all markings involving common transitions from subnet controllers and replies the agreement (acknowledgement) to them.
5) Each subnet controller c_j replies with an answer message to coordinator c_i whenever it receives the agreement and closes its timer meanwhile.

Note that there are two kinds of information: process tokens and communication messages. Process tokens carry the information that fully describes the marking of subnet places. A communication message is designed to establish a dependable network connection. The coordinator always contains a communication message that records any data including the state of the subnet controllers during the execution of agreement, the list of suspected subnet controllers, whether a subnet has failed, whether a subnet has reached a decision, a timer, and a broadcast related to the data and so on. This gives us a global external view of the system. Furthermore, it is easy to see that the termination time of the agreement is strongly related to the network traffic and the end to end delays. This will be demonstrated in Section V.

V. Convergence Time and Feasibility Analysis

Any subnet S_i in Figure 2 is represented by a node $node_i$, $1 \leq i \leq n$, in Figures 5 and 6. Let T_i and T_j be the transition sets belonging to subnets S_i and S_j, respectively. If $T_i \cap T_j \neq \emptyset$, then a solid line will be drawn to connect $node_i$ and $node_j$, where $1 \leq i, j \leq n$. In a distributed system, c_i represents a subnet controller in S_i, $1 \leq i \leq n$. Subnet controllers in different subnets are directly connected to the coordinator by using a Star LAN as an underlying architecture and communicate with the coordinator by exchanging messages through a network protocol. As shown in Figure 5(a) with dashed lines, we assume that c_1 is chosen as the coordinator. The symbol κ_{τ_c} beside c_i, $i \in \{1, 2, 3a, 3b, 4\}$, shown in Figure 5(b), implies that an agreement on the control agreement κ_{τ_c} is reached by the coordinator and subnet controllers after τ_c time.

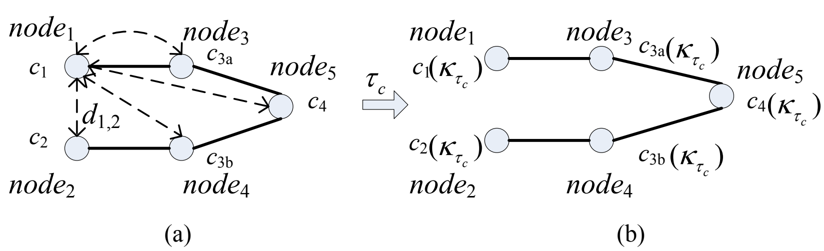

Figure 5. Abstract model of the manufacturing cell.

Theorem 1: For a set of subnets S_i, $i \in \{1, 2, \ldots, n\}$, described by an undirected graph $\partial = \{V, E\}$ with $|V| = n$, assume that a coordinator c_i is derived from Algorithm 1. Then, the agreement is reached at time

$$\tau_c = \max \quad (2 \cdot c_{i,l} + \sum_{j=1}^{m} (|\Im_{l,j}| + |\Re_{l,j}| \cdot \delta_{l,j})/B) + (m + \ell)/Q \tag{12}$$

where $l \in \{1, \ldots, n\}$, m is the number of common transitions, ℓ is the number of constraints, and Q is the speed of CPU.

Proof 1: The convergence time can be divided into three parts: the time of estimates, the time of observation agreement and control agreement, and the time of acknowledgements. According to Algorithm 2, the convergence of the observation and control agreement is reached in the coordinator after $m + \ell$ times of 'and' operations. The time of estimates and acknowledgements depends upon the communication delay between the coordinator and the subnet controllers, which is described by the value of $(\max \quad (2 \cdot c_{i,l} + \sum_{j=1}^{m} (|\Im_{l,j}| + |\Re_{l,j}| \cdot \delta_{l,j})/B))$. As it is already known, R, B and Q are constants. Obviously, the convergence time τ_c is mainly determined by the network distance (propagation delay) and station loads (transmission delay) in the network.

Here Δ_o denotes the delay after which the coordinator starts a communication repair service. The relationship between the time Δ_o and the convergence time τ_c satisfies:

$$\Delta_o \geq \tau_c \tag{13}$$

The transmission cost in the computation of the termination time is considered as all the delays that are required to perform the estimates and acknowledgements only. In other words, the time $(m+\ell)/Q$ can be ignored since the cost of 'and' operations in the process of generating the observation and control agreement is less than 0.1% of the cost of sending estimates or receiving acknowledgements in a communication delay.

VI. Experimental results

IBM System X3100 is chosen as a laboratory platform in the experiment. Its CPU clock speed is 3.1GHz and the memory capacity of the computer is 4GB. We use MATLAB 2013a and SIMULINK [46, 47] to simulate network models. The parameters are setup as follows: $B = 100$Mbps, R=200,000km/sec and a packet size for UDP/IP is 64 bytes. The ILP problem is solved by Lindo 14.0 [51]. As mentioned above, there is less time consumed in the agreement calculation than that in the communication. Hence, the following experiments focus on the communication time delay only.

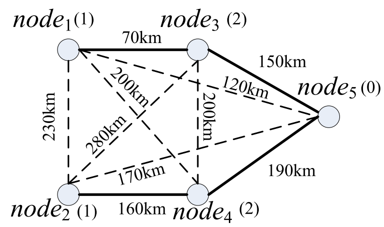

Figure 6. The network system.

Example 1: This example is used to illustrate how to choose a coordinator according to the method proposed in Algorithm 1. The input parameters include the distances between the subnet controllers and the number of packets that are needed to be uploaded for the agreement from the subnet controllers.

Figure 6 shows the values of L corresponding to each path distance between different subnets shown in Figure 2. The value (contained in a pair of braces) beside the symbol $node_i$ ($i \in \{1, 2, \ldots, 5\}$) in Figure 6 represents the number of packets needed to be uploaded from the subnet controller to the coordinator. As shown in Figure 2, the marking of p_4 in subnet S_1 only takes part in the observation agreement of the common transition t_3 and the marking of $t_3^\bullet = \{c_1\}$ does not affect the control agreement. There is one packet to be uploaded. Hence, it is '1' in the braces on the side of $node_1$. The markings of ${}^\bullet t_3$ and $t_3^\bullet$ in $node_3$ are needed in the observation and control agreement, respectively. Therefore, the number of their packets is '2'. The numbers of other nodes are the same. If c_1 is chosen as the coordinator, the agreement results can be obtained after receiving 5 packets from all subnet controllers, where the transmission delay is $\tau_{1\_t} = 5 * 64/B$, the propagation delay is $\tau_{1\_p} = (230 + 70 + 200 + 120)/R$, and the total simulation agreement time is $\tau_1 = \tau_{1\_t} + 2 * \tau_{1\_p}$ =9.4ms. By solving the ILP problem in Algorithm 1, the minimal simulation agreement time τ_4 =9.06ms can be obtained. Therefore, c_{3b} is the best choice as the coordinator in Figure 2.

Example 2: This example is used to compare the convergence time and the messages exchanged to reach agreement between Algorithm 2 in this research and those in [23] and [24]. A system runs once for each different methods.

The results of the experiments can be seen in Table 1. In [24], a common transition fires in a subnet only after receiving the marking of places that are connected with the common transition in the centralized model and then update the states subsequently. For example, $node_1$ needs to determine the marking of places p_5 and c_{3a} for transition t_3 to be fired, and transmits feedback to places p_5 and c_{3a}. There are four packets with respect to communication in this step. Assume that there is a transition sequence $\sigma = t_3t_7t_4t_8$ in process, and the transmission delay and the propagation delay are $\tau'_t = (2{+}2{+}1{+}1){*}64/B$ and $\tau'_p = 2 * (70 + 160 + 150 + 190)/R$, respectively. The total agreement time is $\tau' = \tau'_t + 2 * \tau'_p$.

On the other hand, Gasparri et al. [23] propose the decentralized supervision of Petri nets through the collaboration among supervisors without any coordinator. Communication is assumed to be available but limited within one-hop neighbors. An agreement is reached under the firable transitions information that is exchanged in the communication topology network. As shown in the solid lines in Figure 5(a), if we want to know the firing condition of t_3 in $node_1$, the enabled conditions of t_1, t_3, t_4, t_5, t_7, and t_8 as observation agreement conditions and the enabled conditions of t_1, t_3, t_5 and t_7 as control agreement conditions should be known. The control agreement will be successively spread to $node_3$, $node_5$, $node_4$ and $node_2$ through the solid lines in Figure 5(b). It will spend delay time $\tau''_{t_3} = 4 * 64/B + 2 * (70 + 150 + 190 + 160)/R$ =8.26ms to reach the agreement condition for transition t_3. The total agreement delay time is $\tau'' = \tau''_{t_3} + \tau''_{t_7} + \tau''_{t_4} + \tau''_{t_8}$ if a transitions sequence σ fires.

Note that the agreement convergence in this paper runs up to 3.64 times faster than the method in [23] mainly thanks to the optimal coordinator by using the ILP problem in Algorithm 1. Although an ILP problem is in theory NP-complete, when the size of a considered problem is finite, an optimal solution of the ILP problem can be found within limited time. We can give the optimal coordinator with time complexity of $O(n^2 * \ell^m)$, where n is the number of subnets in the decentralized system, ℓ is the number of constraints, and m is that of common transitions.

By enforcing (12), we can calculate an agreement maximum time $\tau_c = \tau_{4\_3} = 2 * 64/B + 2 * 200/R$ =3.28ms if c_{3b} is chosen as the coordinator. We define that Δ_t is the time unit for a subnet controller to start a communication repair service. In general, $\Delta_t > \Delta_o$ holds. In this case, we take $\Delta_t = 6ms > \Delta_o$=4ms $> \tau_c$.

Example 3: This example is used to illustrate the dependable network connection for the subnet controllers and the coordinator by the method in Section IV-C. We design a network environment in which there exists an occurrence probability

Table I
THE COMPARISON RESULTS OF ALGORITHM 2 AND THE ALGORITHMS IN [24] AND [23]

	Agreement convergence time in one process	The messages for the agreement
$Algorithm\ 2\ in\ this\ paper$	9.06ms	4 packets
$Iordache\ and\ Antsaklis$ [24]	9.54ms	6 packets
$Gasparri\ et\ al.$[23]	33.04ms	16 packets

Table II
THE NETWORK LATENCY AFTER THE PROCESS RUNNING 100 TIMES WITH 10% CONNECTION FAILURE RATE

	The delay for the reliable communication connection	Messages for keeping the reliable communication connection
$The\ Method\ in\ Section$ IV-C	$40 * 4$ ms	$40 * 4$ packets
$Periodical\ communication$	0ms	$<$ 400 packets

of connection failures. Then we check the error correcting time and how many messages are needed to reestablish dependable connection.

Suppose that the system in Figure 2 runs 100 times. Each time the system needs four agreement calculations. The probability of connection failure is set to 10. After a time unit Δ_o, the coordinator establishes dependable connection to the subnet controllers in its suspected list if the list has not been cleared. There is a Δ_o time delay to detect the connection failure and repair it. The other idea is that the coordinator sends a live message [45] periodically to each subnet controller with a delay Δ_r (abbreviated periodical communication). In this case, if Δ_r is 2ms that is less than τ_c, the connection failure can be found immediately and the latency of the dependable connection is minimized. However, the generation of additional live messages will lead to a big increase on the network load. The results are shown in Table 2.

Example 4: This example is used to illustrate the efficiency of the agreement algorithm in a real-world network environment. We design a more complicated network than Example 3, in which there are more parameters, such as packet loss rates, bit error rates, and noise [45]. In this experiment, we make the decentralized system run 5 times in the simulating network. Every time, we put different numbers of tokens in the system as the initial state. They are 10, 100, 1000, 5000 and 10000 tokens, respectively. The tokens in places p_{a1} and p_{b1} are kept equal, for example, in the first time, $M_0(p_{a1}) = M_0(p_{b1}) = 10$. M_0 is the initial marking. We compare the theoretical and experimental results on the consuming time which starts from the tokens putting in p_{a1} and p_{b1} to the final products leaving the manufacturing cell.

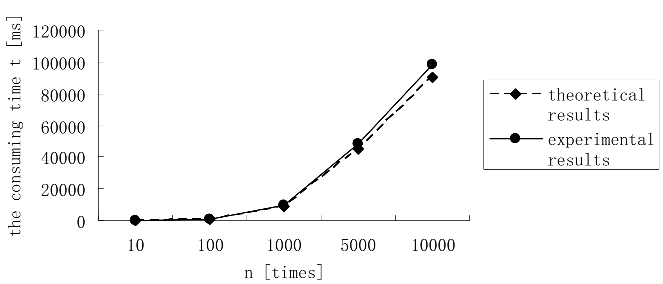

Figure 7. The theoretical results vs. the experimental results.

For the sake of clarity, we now explain packet loss rates, bit error rates, and noise. Packet loss rates mean that one or more packets of data travelling across a computer network fail to reach their destination. We set the packet loss rate to be 0.1% (1 lost packet in every 1000 packets), which is the tolerable upper bound. The bit error rate (BER) is the number of bit errors divided by the total number of transferred bits during a studied time interval. The BER is set to be 10^{-6}. To make it closer to a real situation scenario, Gaussian white noise is introduced into the model. The connection failure proportion is generated from a random function rather than a constant 10%. Other parameters are the same as above. Figure 7 shows the fitting relationship between the two results. With the increasing tokens, agreement information exchange in different subnets is more and more frequent, which gives rise to the occurrence of network jam phenomena and extends the completion time of the whole task. However, the deviation between the two curves keeps less than 6.03%. The simulation results enlighten the possibility to obtain good performances according to the agreement algorithm in this paper.

VII. Conclusion and future work

This paper focuses on the problem of decentralized supervision of Petri nets. In order to enforce constraints for a distributed model and benefit from the existing GMEC control results, we first consider a distributed network as a whole and design supervisors for the whole net. Then, we split the whole net into a set of subsystems according to the actual situation. The addressed problem is how to ensure that the constraints remain valid in this decentralized system.

Different from the methods in [21, 22] where the subnets do not to communicate with each other, we assume that the communication between the subnet controllers is allowed. Thus, the constraints that are not d-admissible can be enforced by first solving a centralized design problem and then decentralizing the solution. A coordinator is chosen from the subnet controllers, and a protocol to reach an agreement on the firing conditions of common transitions is proposed. The theoretical analysis and experimental results show the feasibility and effectiveness of the algorithms. Four experimental studies of the efficiency of the algorithms are conducted. In the proposed approach, thanks to the presence of a coordinator, the agreement can be obtained faster with respect to the methods in [23, 24].

As a future study, we will study the performance of such a decentralized control policy in the framework of timed nets, with the objective of maximizing the average firing rate of some transitions. Finally, we also plan to address for timed nets, the problem of reaching a given target state under control. We will derive a decentralized algorithm to reach such a target marking in minimal time while ensuring that the safety constraints are always satisfied.

References

[1] P. Ramadge and W. M. Wonham, “Modular feedback logic for discrete event systems”, *SIAM Journal on Control and Optimization*, vol. 25, no. 5, pp. 1202–1218, 1987.

[2] P. Ramadge and W. Wonham, “The control of discrete event systems”, *Proceedings of the IEEE*, vol. 77, no. 1, pp. 81–98, 1989.

[3] W. M. Wonham and P. Ramadge, “Modular supervisory control of discrete event systems”, *Mathematics of Control, Signals and Systems*, vol. 1, no. 1, pp. 13–30, 1988.

[4] F. Lin and W. M. Wonham, “Decentralized control and coordination of discrete event systems with partial observation”, *IEEE Transition on Automatic Control*, vol. 35, no. 12, pp. 1330–1337, 1990.

[5] K. Rudie and W. M. Wonham, “Think globally, act locally: Decentralized supervisory control”, *IEEE Transition on Automatic Control*, vol. 37, no. 11, pp. 1692–1708, 1992.

[6] J. Komenda and J. H. van Schuppen, “Control of discrete event systemss with partial observation using coalgebra and coinduction”, *Discrete Event Dynamic Systems*, vol. 15, no. 3, pp. 257–315, 2005.

[7] K. Rudie, S. Lafortune and F. Lin, “Minimal communication in a distributed discrete-event system”, *IEEE Transitions on Automatic Control*, vol. 48, no. 6, pp. 957–975, 2003.

[8] K. Rudie, S. Lafortune and F. Lin, “Minimal communication in a distributed discrete event control system”, in *Proceedings of American Control Conference*, San Diego, CA, pp. 1965–1970, 1999.

[9] Y. F. Chen, Z. W. Li and K. Barkaoui, “New Petri net structure and its application to optimal supervisory control: Interval inhibitor arcs”, *IEEE Transactions on Systems, Man and Cybernetics: Systems*, vol. 44, no. 10, pp. 1384-1400, 2014.

[10] Z. W. Li and M. C. Zhou, “Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems”, *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 34, no. 1, pp. 38–51, 2004.

[11] Z. W. Li and M. C. Zhou, “Clarifications on the definitions of elementary siphons in Petri nets”, *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol.36, no.6, pp.1227-1229, Nov. 2006.

[12] Z. W. Li and M. Zhao, “On controllability of dependent siphons for deadlock prevention in generalized Petri nets”, *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 38, no. 2, pp. 369–384, 2008.

[13] J. F. Zhang, M. Khalgui, Z. W. Li, G. Frey, O. Mosbahi, and H. B. Salah, “Reconfigurable coordination of distributed discrete event control systems”, *IEEE Transactions on Control Systems Technology*, DOI: 10.1109/TCST.2014.2313352, 2014.

[14] Y. F. Chen and Z. W. Li, “Design of a maximally permissive liveness-enforcing supervisor with a compressed supervisory structure for flexible manufacturing systems”, *Automatica*, vol. 47, no. 5, pp. 1028–1034, 2011.

[15] Y. F. Chen, Z. W. Li, M. Khalgui, “Design of a maximally permissive liveness-enforcing Petri net supervisor for flexible manufacturing systems”, *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 2, pp. 374–393, 2011.

[16] Y. F. Chen, Z. W. Li, and M. C. Zhou, “Optimal supervisory control of flexible manufacturing systems by Petri nets: A set classification approach”, *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 2, pp. 549–563, 2014.

[17] Z. W. Li and M. C. Zhou, “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, 2006.

[18] Z. W. Li, H. S. Hu, and A. R. Wang, “Design of liveness-enforcing supervisors for flexible manufacturing systems using Petri nets”, *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 37, no. 4, pp. 517–526, 2007.

[19] Z. W. Li and M. C. Zhou, “Control of elementary and dependent siphons in Petri nets and their application”, *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 38, no. 1, pp. 133–148, 2008.

[20] Z. W. Li, M. C. Zhou, and N. Q. Wu, “A survey and comparison of Petri net-based deadlock prevention policies for flexible manufacturing systems”, *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 38, no. 2, pp. 173–188, 2008.

[21] F. Basile, A. Giua and C. Seatzu, “Supervisory control of Petri nets with decentralized monitor places”, in *Proceedings of American Control Conference*, New York City, USA, vol. 1, pp. 4957–4962, 2007.

[22] F. Basile, A. Giua. and C. Seatzu, “Some new results on supervisory control of Petri nets with decentralized monitor places”, in *Proceedings of World Congress of the International Federation of Automatic Control*, Seoul, Korea, vol. 1, pp. 531–536, 2008.

[23] A. Gasparri, A. Giua, D. Di Paola, G. Ulivi and D. Naso, “Consensus based decentralized supervision of Petri nets”, in *Proceedings of IEEE International Conference on Decision and Control*, Orlando, Florida, USA, vol. 2011, pp. 1128–1135, 2011.
[24] M. V. Iordache and P. Antsaklis, “Decentralized supervision of Petri nets”, *IEEE Transactions on Automatic Control*, vol. 51, no. 2, pp. 376–381, 2006.
[25] C. Kuo, C. Wang and K. Huang, “Behavior modeling and control of 300 mm fab intrabays using distributed agent oriented Petri net”, *IEEE Transition on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 33, no. 5, pp. 641–648, 2003.
[26] A. Timofeev, F. Kolushev, and A. Bogdanov, “Hybrid algorithms of multi-agent control of mobile robots”, in *Proceedings of International Joint Conference on Neural Networks*, Washington, DC, USA, vol. 6, pp. 4115–4118, 1999.
[27] N. Q. Wu, M. Zhou, and Z. W. Li, “Resource-oriented Petri net for deadlock avoidance in flexible assembly systems”, *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 38, no. 1, pp. 56–69, 2008.
[28] R. O. Saber, J. A. Fax and R. M. Murray, “Consensus and cooperation in networked multi-agent systems”, *Proceedings of IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
[29] F. Lu, W. Wu, H. Su and J. Chu, “ Non-blocking decentralized control of discrete event systems based on Petri nets”, *Asian Journal of Control*, vol. 12, no. 3, pp. 323–335, 2003.
[30] N. Palomeras, P. Ridao, C. Silvestre and A. El-Fakdi, “Multiple vehicles mission coordination using Petri nets”, in *Proceedings of IEEE International Conference on Robotics and Automation*, Anchorage, AK, vol. 1, pp. 3531–3536, 2010.
[31] T. Murata, “Petri nets: Properties, analysis and application”, *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.
[32] K. Cai and W. M. Wonham, “Supervisor localization for large discrete-event systems: Case study production cell”, *International Journal of Advanced Manufacturing Technology*, vol. 50, no. 9, pp. 1189–1202, 2010.
[33] J. Ezpeleta, J. M. Colom, and J. Martinez, “A Petri net based deadlock prevention policy for flexible manufacturing systems”, *IEEE Transactions on Robotics and Automation*, vol. 11, no. 2, pp. 173–184, 1995.
[34] 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.
[35] Z. W. Li, G. Y. Liu, M. H. Hanisch, and M. C. Zhou, “Deadlock prevention based on structure reuse of Petri net supervisors for flexible manufacturing systems”, *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol.42, no.1, pp. 178–191, 2012.
[36] J. L. Peterson, *Petri Net Theory and the Modeling of Systems*, New Jersey: Prentice-Hall, 1981.
[37] Z. W. Li, N. Q. Wu, and M. C. Zhou, “Deadlock control of automated manufacturing systems based on Petri nets–A literature review”, *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol.42, no.4, pp.437–462, 2012.
[38] Z. W. Li and M. C. Zhou, *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach*. London: Springer-Verlag, 2009.
[39] 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.
[40] Z. W. Li and M. C. Zhou, “On siphon computation for deadlock control in a class of Petri nets”, *IEEE Transactions on Systems, Man and Cybernetics, Part A-Systems and Humans*, vol. 38, no. 3, pp. 667–679, 2008.
[41] A. Giua, F. DiCesare, and M. Silva, “Generalized mutual exclusion constraints on nets with uncontrollable transitions”, in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Chicago, IL, U.S.A, pp. 974–979, 1992.
[42] A. Ghaffari, N. Rezg and X. Xiaolan, “Feedback control logic for forbidden-state problems of marked graphs: application to a real manufacturing system”, *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 18–29, 2003.
[43] M. P. Fanti, A. Giua and C. Seatzu, “Generalized mutual exclusion constraints and monitors for colored Petri nets”, in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Washington, D.C., USA, vol. 2, pp. 1860–1865, 2003.
[44] F. Basile, P. Chiacchio and A. Giua, “Suboptimal supervisory control of Petri nets in presence of uncontrollable transitions via monitor places”, *Automatica*, vol. 42, no. 6, pp. 995–1004, 2006.
[45] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks (5th Edition)*, New Jersey: Prentice-Hall, 2011.
[46] O. H. Ekkehard, *Environmental Modeling: Using MATLAB*, Berlin Heidelberg, Germany: Springer, 2012.
[47] Simulink, Multidomain Simulation and Design Environment, http://www.mathworks.com/products/simulink, accessed August, 2013.
[48] Y. F. Chen and Z. W. Li, *Optimal Supervisory Control of Automated Manufacturing Systems*, CRC Press, Taylor & Francis Group, January 2013.
[49] D. Liu, Z. W. Li, and M. C. Zhou, A parameterized liveness and ratio-enforcing supervisor for a class of generalized Petri nets, *Automatica*, vol.49, no.11, pp. 3167–3179, Nov 2013.
[50] H. S. Hu, M. C, Zhou, and Z. W. Li, “Deadlock-free Control of Automated Manufacturing Systems with Flexible Routes and Assembly Operations Using Petri Nets,” *IEEE Transactions on Industrial Informatics*, vol.9, no.1, pp. 109-121, 2013.
[51] Lingo 14.0, Premier Optimization Modeling Tools, http://www.lindo.com, accessed August, 2013.

Jianhong Ye received his B.S. degree in computer engineering from Fuzhou University, Fu’zhou, China, in 2002, his M.S. degree in computer software and theory from Xihua University, Cheng’du, China, in 2006, and Ph.D. degree in computer software and theory from University of Electronic Science and Technology of China, Cheng’du, China, in 2009. He is currently an associate Professor of network engineering, School of Computer Science and Technology, Huaqiao University. He is also a post-doctor in control science and engineering, School of Electro-Mechanical Engineering, Xidian University, Xi’an, China. His research interests include manufacturing system modeling and control, discrete event systems, Petri net theory and applications, and process mining in big data. Dr. Ye is a member of the China Computer Federation. He is the author or coauthor of more than 10 journal papers published in International Journals.

Zhiwu Li (M'06-SM'07) received his B.S., M.S., and Ph.D. degrees in mechanical engineering, automatic control, and manufacturing engineering, respectively, all from Xidian University, Xi'an, China, in 1989, 1992, and 1995, respectively. He joined Xidian University in 1992 and now he is also with the Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau. From June 2002 to July 2003, he was a Visiting Professor at the Systems Control Group, Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada. From February 2007 to February 2008, he was a Visiting Scientist at the Laboratory for Computer-Aided Design (CAD) and Lifecycle Engineering, Department of Mechanical Engineering, Technion-Israel Institute of Technology, Technion City, Haifa, Israel. From November 2008 to October 2010, he was a Visiting Professor in Automation Technology Laboratory, Institute of Computer Science, Martin-Luther University of Halle-Wittenburg, Halle (Saale), Germany. He was a senior visiting scientist in Conservatoire National des Arts et Mtiers (CNAM), Paris, France, supported by the program Research in Paris in 2010, by Cai Yuanpei Program in 2012, and by Visiting Professor Program due to Cnam in 2014. He was an invited advisory professor in Meliksah Universitesi, Kayseri, Turkey, supported by the Research Grant of the Scientific and Technological Research Council of Turkey (2012-2014). He serves as a host professor of Research Fellowship for International Young Scientists, National Natural Science of Foundation of China.

He is a coauthor with MengChu Zhou, Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach, (Springer, 2009) and Modeling, Analysis and Deadlock Control in Automated Manufacturing Systems (Beijing, 2009, in Chinese), and with YuFeng Chen, Optimal Supervisory Control of Automated Manufacturing Systems (CRC Press, Taylor and Francis Group, 2013). He is a Co-editor with Professor Abdulrahman Al-Ahmari of a book Formal Methods in Manufacturing: Recent Advances (IGI Global 2013). His current research interests include Petri net theory and application, supervisory control of discrete event systems, workflow modeling and analysis, system reconfiguration, and data and process mining.

He is the General Co-Chair of the IEEE International Conference on Automation Science and Engineering, August 23-26, Washington, DC, 2008. He is a financial Co-Chair of the IEEE International Conference on Networking, Sensing, and Control, March 26-29, 2009, a member of International Advisory Committee, 10th International Conference on Automation Technology, June 27-29, 2009, a Co-Chair of the program committee of the IEEE International Conference on Mechatronics and Automation, August 24-27, 2010, and members of the program committees of many international conferences. He serves an Associate Editor of the IEEE Transactions on Automation Science and Engineering from 2008 to 2013, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans from 2011 to 2012, IEEE Transactions on Systems, Man, and Cybernetics: Systems from 2013, Information Sciences from 2014, and International Journal of Discrete Event Control Systems. He is a Guest Editor of Special Issue on "Petri Nets for System Control Automation" in Asian Journal of Control, Special Issue on "Petri Nets and Agile Manufacturing" in Transactions of the Institute of Measurement and Control, and Special Issue on Modeling and Verification of Discrete Event Systems in ACM Transactions on Embedded Computing Systems.

He is a member of Discrete Event Systems Technical Committee of the IEEE Systems, Man, and Cybernetics Society, and a member of IFAC Technical Committee on Discrete Event and Hybrid Systems from 2011 to 2014. He serves as a frequent reviewer for 40+ international journals including Automatica and a number of the IEEE Transactions as well as many international conferences. He is listed in Marquis Who's Who in the world, 27th Edition, 2010. Dr. Li is a recipient of an Alexander von Humboldt Research Grant, Alexander von Humboldt Foundation, Germany. He is a senior member of IEEE and is the founding chair of Xi'an Chapter of IEEE Systems, Man, and Cybernetics Society.

Alessandro Giua is professor of Automatic Control at the Department of Electrical and Electronic Engineering (DIEE) of the University of Cagliari, Italy and at the Information and Systems Sciences Laboratory (LSIS) of Aix-Marseille University, France. He received a Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, New York in 1992. He has also held visiting positions in several institutions worldwide, including Xidian University, Xi'an, China.

His research interests include discrete event systems, hybrid systems, networked control systems, Petri nets and failure diagnosis. On these topics he has published extensively and has managed several research projects.

He is the Editor in Chief of the IFAC journal Nonlinear Analysis: Hybrid Systems, and a Department Editor of Discrete Event Dynamic Systems. He has previously served in the editorial board of several journals including: IEEE Trans. on Automatic Control, IEEE Trans. on Control Systems Technology and European Journal of Control.

He has been General Chair or Program Chair for international conferences such as WODES 1998, IFAC ADHS 2006 and IEEE CASE 2011 and has served in the steering committees of these series. He has also been chair of the IFAC Technical Committee 1.3 on Discrete Event and Hybrid Systems from 2008 to 2014 and has served as a member of the program committee for several international events in this area.

Currently he is serving in the IEEE Control Systems Society as member of the Board of Governors (2013-2015) and is working at the organization of the conferences IEEE CDC 2016 as General Chair and WODES 2016 as a Program Chair.