2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Consensus-Based Decentralized Supervision of Petri nets

Andrea Gasparri, Donato Di Paola, Alessandro Giua, Giovanni Ulivi and David Naso

Abstract— **In this work, by starting from the theoretical framework proposed in [1], the problem of decentralized supervision of a Petri net through collaboration among supervisors is addressed. Communication is assumed to be available but limited to one-hop neighbors, i.e., supervisors reachable from one another with respect to a certain communication radius r. A sufficient condition to achieve decentralized admissibility by focusing the attention on the communication topology of the network of supervisors is provided. Furthermore, under the assumption of control under concurrent firing, a protocol to reach an agreement on the control input among the supervisors is proposed. Finally, a feasibility analysis for the implementation of the proposed decentralized control framework is discussed.**

I. INTRODUCTION

A discrete event system (DES) is a dynamic system that evolves in accordance with the abrupt occurrence, at possibly unknown irregular intervals, 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 [2], is a method for automatically synthesizing supervisors that restrict the behavior of a plant such that as much as possible of the given specifications are fulfilled. The plant is assumed to spontaneously generate events. The events are in either one of the following two categories controllable or uncontrollable. The supervisor observes the string of events generated by the plant and might prevent the plant from generating a subset of the controllable events. However, the supervisor has no means of forcing the plant to generate an event.

Although decentralized control of discrete event systems using automata is a well explored topic, relatively few contributions deal with the decentralized control of Petri nets. Here we are concerned with the problem of enforcing a special class of state specification, called Generalized Mutual Exclusion Constraints, using monitor places [3], [4]. A monitor place p_s controls a transition t if there exists an arc (p_s, t), and it detects a transition t' if there exists either an arc (p_s, t') or an arc (t', p_s).

In [1] Iordache and Antsaklis introduced the concept of decentralized admissibility (d-admissibility) as an extension to the decentralized setting of the centralized admissibility concept for Petri nets. We believe this decentralized setting, although inspirational for the present work, contains some implicit assumptions that make it not immediately applicable in real cases. These implicit assumptions are two. On one hand, the basic notion of d-admissibility requires that each local supervisor should observe all detectable transitions, i.e., all transitions that a centralized monitor should observe to solve the given problem. Thus, while the supervisor proposed in [1] is distributed, global observation is still required. On the other hand, the solution proposed to go beyond the previously mentioned issue requires that a local site that can observe a detectable transition, should transmit this information to all other sites that cannot observe it. However, the control scheme used in [1] is correct only under the assumption that this information is immediately available at all sites, which is not realistic.

Note that to overcome this problem in [5], [6] a different approach for constraint decomposition was proposed. The advantage of this approach is that no communication among distributed sites is necessary. However, this leads to overly restrictive control laws.

The objective of this paper is proposing a general framework for a truly decentralized control law with communications that can be effectively implemented.

The rest of the paper is organized as follows. In Section II a example to introduce the problems related to the time-delay of the communication channel under the assumption of concurrent firing is described. In Section III related work is presented. In Section IV the algebraic structure of bounded lattices along with the consensus algorithm are introduced. In Section V concepts regarding Petri nets along with its centralized/decentralized supervision are given. In Section VI the proposed consensus-based decentralized supervision of Petri nets is described. In Section VII an example to corroborate the theoretical analysis is proposed. Finally, in Section VIII conclusions are drawn and future work is discussed.

II. A MOTIVATING EXAMPLE

Let us first discuss the communication protocol used in this paper. We assume that the local supervisors are synchronized through a global clock, while no assumption is made on the firing of the plant transitions, that may occur asynchronously.

During each control cycle the following steps are sequentially executed.

a) Each supervisor blocks all its controlled transitions.
b) Each supervisor determines the status of the detectable transitions it can observe, i.e., it determines if they have fired since the previous observation.

A. Gasparri and G. Ulivi are with the Department of Computer Science and Automation (DIA), University of "Roma Tre", 00146 Roma Italy {gasparri, ulivi}@dia.uniroma3.it
A. Giua is with the Department of of Electrical and Electronic Engineering (DIEE) University of Cagliari, Italy giua@diee.unica.it
D. Di Paola is with the Institute of Intelligent Systems for Automation (ISSIA), National Research Council (CNR), 70126 Bari, Italy dipaola@ba.issia.cnr.it
D. Naso is with the Department of Electrical and Electronic Engineering (DEE), Polytechnic of Bari, Bari, Italy naso@poliba.it

978-1-61284-799-3/11/$26.00 ©2011 IEEE

c) The information on the detected transitions spreads in the network. We call this step *observation consensus*.
d) The supervisors agree on a control input that satisfies the specification. We call this step *control consensus*.
e) Each supervisor sends to its controlled transitions the updated control input valid for the rest of the cycle, i.e., specifies how many times the transition is allowed to fire during the period.
f) The supervisors stay idle for the rest of the cycle while the plant evolves.

Three issues deserve some further comments:

First we observe that between steps b) and e) detectable but uncontrolled transitions may fire asynchronously. However, the firing of these transitions will not affect the admissibility of the control law determined at step d).

Secondly, it should be noted that during steps a) to e) the controllable transitions are blocked, while the new control input is being computed. The total length of these steps should be small with respect to the length of step f), when enabled controlled transitions are left free to fire.

Finally, the control law computed in step e) must be admissible even if the firing of controlled transitions during step f) is not immediately observed and used to update the control. Such a control law has already been discussed in the literature [7] and is known as *control under concurrent firing*. Here we briefly discuss it.

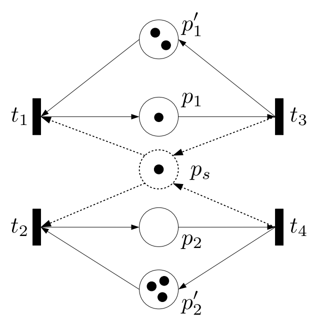

Fig. 1. A simple Petri net with a monitor place.

Consider the net in Figure 1 (ignoring the dashed place and arcs) and assume that the objective of a supervisor is to enforce the constraint $\mu_1 + \mu_2 \leq 2$, i.e., the sum of the tokens in places p_1 and p_2 should not exceed 2. Assuming transition t_1 and t_2 are controllable, the monitor places p_s that enforces this solution is represented in the figure as well (in dashed line). The monitor works as follows. Initially place p_1 contains a single token and p_2 is empty. The monitor has a single initial token to denote that one more token can be added to place p_1 or p_2, and both transitions t_1 and t_2 are initially control enabled. As soon as one of them fires, putting a token into its output place, the monitor looses its token and both t_1 and t_2 are control disabled.

In a real implementation, however, the supervisor needs some time to detect the firing of one transition and consequently to update the control law. In a centralized system it may be reasonable to assume that this time is negligible and that for all practical purposes only one transition can fire at a time. In a decentralized system, however, due to communication delays this time cannot be neglected and a concurrent firing of two enabled transitions, or even a multiple firing of a single transition, may occur.

Under this assumption we need to change the control law. In the case just discussed, under concurrent firing we have two maximally permissive control laws: either the monitor should assign the token to transition t_1 and thus disable t_2, or assign the token to transition t_2 and thus disable t_1. Note that when p_1 and p_2 are both empty, the monitor shall contain two tokens and may assign k_1 tokens to t_1 and k_2 tokens to t_2 (with $k_1 + k_2 = 2$) to denote that during next cycle t_1 (resp., t_2) can fire up to k_1 (resp., k_2) times. In step d) of the control cycle we assume that a law of this type, valid for concurrent firing, is computed.

Finally, we also note that the firing of any transition other than t_1 and t_2 (the controlled transitions) will not make the control law computed by the monitor unfeasible: in fact, the firing of all other transitions will never increase the token content in p_1 and p_2. On the contrary, each time the firing of t_3 and t_4 is detected the number of tokens in the monitor will increase and more permissive control laws can be computed.

In our framework, communication among supervisors is assumed to be available but limited to one-hop neighbors, i.e., supervisors reachable by another one with respect to a certain communication radius r. Our first contribution is to provide a systematic approach to the problem of achieving d-admissibility by focusing the attention on the communication topology of the network of supervisors. As a result, a sufficient condition to achieve d-admissibility by means of collaboration is proposed. Our second contribution is to provide a decentralized way of achieving an agreement among the supervisors on the control input under the assumption of concurrent firing. Finally, a feasibility analysis for the implementation of the proposed decentralized supervision technique is discussed.

III. Related Work

Several works concerning the decentralization of the supervisory control of Discrete Event Systems (DES) have been proposed in literature.

An early work on decentralized DES has been proposed in [8]. In that reference, which is an extension of the work proposed in [9], the decentralization of the supervision under the partial observation has been studied. The DES are modeled using controlled automata, diminishing the design complexity by means of distribution of supervisors. Following this trend, in [10], the control of discrete-event systems with partial observations has been addressed using coalgebra and coinduction. The results of this paper is 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.

As mentioned above, in [1], an extension of the centralized admissibility concept to the decentralized setting of a Petri Net is proposed. Based on decentralization concept, authors propose two methods to design decentralized supervisors. In both methods, the solution can be distributed by means of communication, and the formalization of an Integer Linear Programming (ILP) problem can be used to minimize the amount of communication required by the solution.

Dealing with distributed discrete-event systems, the problem of communication among supervisors has to be addressed. In [11], a novel information structure model is presented to deal with this problem. Existence results are given for the cases in which supervisors do and do not anticipate future communications, and a synthesis procedure is given for the case when supervisors do not anticipate communications. In [12], each agent (supervisor) uses a combination of direct observation (obtained from sensor readings available to that agent) and communicated information (obtained from sensor readings available to another agent). Since communication may be costly, a strategy to minimize communication between sites is developed.

In [13], a hybrid approach to deal with the decentralized control of DES is proposed. This work describes a solution for achieving non-blocking decentralized supervisory control of DES. In particular, the proposed approach gives a graphical way of designing coordinators to keep the non-blocking property of the closed-loop system with decentralized supervisors. Furthermore, the proposed approach guarantees the closed-loop system to be maximally permissive.

Decentralized Petri nets have been also used in different real-world applications, such as mission control and task sequencing for team of autonomous vehicles. For instance, in [14], PNs have been used to manage mutual exclusion, ordering and synchronization for missions defined on each vehicle. The proposed solution guarantees a deadlock-free centralized PN, which is distributed over the team.

IV. Theoretical Background

In this section some concepts concerning the algebraic structure of bounded lattices along with the consensus algorithm for multi-agent systems are introduced.

A. Bounded Lattice

An algebraic structure $\{\mathbb{L}, \oplus, \otimes\}$, consisting of a set $\mathbb{L}$ and two binary operations "join" $\oplus$, and "meet" $\otimes$, on $\mathbb{L}$ is a lattice if the following axioms hold for all elements $a, b, c \in \mathbb{L}$: associativity, commutativity, absorption and idempotence.

A bounded lattice is an algebraic structure of the form $\mathcal{L} = \{\mathbb{L}, \oplus, \otimes, \perp, \top\}$ such that $\{\mathbb{L}, \oplus, \otimes\}$ is a lattice, $\perp$ (the lattice's bottom) is the identity element for the join operator $\oplus$, and $\top$ (the lattice's top) is the identity element for the meet operator $\otimes$.

In the rest of the paper, the attention will be restricted to a bounded lattice $\mathcal{L}$ built over a subset $\mathbb{L}_m \subset \mathbb{N}$ including the zero 0, where m denotes the upper bound of the subset.

B. Multi-Agent System and Consensus Algorithm

A multi-agent system (MAS) is a system composed of multiple interacting intelligent agents [15]. Multi-agent systems provide a useful abstraction to model interaction among several entities. A multi-agent system is commonly described by an undirected graph $\mathcal{G} = \{V, E\}$, where $V = \{v_i : i = 1, ..., n\}$ is the set of nodes (agents) and $E = \{e_{ij} = (v_i, v_j)\}$ is the set of edges (connectivity) representing the point-to-point communication channel availability. This abstraction will be used to model the network of supervisors $\mathcal{S}$ focusing mainly on their interaction.

The consensus problem forms the foundation of the field of distributed computing [16]. Distributed computation over networks has a tradition in systems and control theory starting with the pioneering work of Borkar and Varaiya [17] and Tsitsiklis and Athens [18] on asynchronous asymptotic agreement problem for distributed decision-making systems. In networks of agents (or dynamic systems), "consensus" means to reach an agreement regarding a certain quantity of interest that depends on the state of all agents. A "consensus algorithm" (or protocol) is an interaction rule that specifies the information exchange between an agent and all of its neighbors on the network. For an overview of Consensus and Cooperation in Networked Multi-Agent Systems the reader is referred to [19].

***Lemma** 1 (Simple Consensus over a Bounded Lattice):* Let us consider a multi-agent system described by its communication graph $\mathcal{G} = \{V, E\}$, with $|V| = n$. Let us assume each agent has an internal state, that is $\mathbf{x}_i = [x_1^{(i)}, \ldots, x_q^{(i)}]^T \in \mathbb{L}_m^q$. Let us consider an interaction rule $\mathcal{R}: \mathbb{B}_m^q \times \mathbb{L}_m^q \to \mathbb{B}_m^q$ described by the binary operator $\oslash$ such that the following holds:

- $a \oslash b = b \oslash a$ (commutativity)
- $a \oslash (b \oslash c) = (a \oslash b) \oslash c$ (associativity)
- $a \oslash a = a$ (idempotence)

If the communication graph $\mathcal{G} = \{V, E\}$ remains connected over time then the multi agent system reaches a common steady state in a finite time $\bar{k}$, that is:

$$x_i(\bar{k}) = \bar{x} \qquad \forall i \in V \tag{1}$$

where

$$\bar{x} = x_1(k_0) \oslash x_2(k_0) \oslash \ldots \oslash x_n(k_0)$$

is the combination of all the internal agents state at time $k = k_0$, and $\bar{k} \leq d(\mathcal{G})$, with $d(\mathcal{G})$ the diameter of the graph, i.e., the greatest distance between any pair of vertices.

Proof: The proof follows the same argument used in [20] to prove the abstract convergence of consensus algorithms. ■

In the following, for sake of clarity, we will refer to the consensus algorithm over a bounded lattice as "lattice consensus".

V. Distributed supervision and d-admissibility

In this work, Petri nets of the form $\mathcal{N} = (P, T, F, W)$ where P is the set of places, T the set of transitions, F the set of arcs, and W the weight function are considered [21].

Let us first introduce the definition of decentralized supervisor as follows:

***Definition** 1 (Decentralized Supervisor):* A decentralized supervisor $\mathcal{S}$ consists of a set of supervisors

$\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n\}$ operating in parallel, such that a given specification is satisfied. In this work, specifications of the form $L\,\mu \leq b$ where $L \in \mathbb{Z}^{n_c \times |P|}$, $b \in \mathbb{Z}^{n_c}$ with n_c the number of specifications, and μ is the marking of $\mathcal{N}$ will be considered.

Let us now introduce the concept of supervision based on monitor places following [3].

***Definition** 2 (Supervision Based on Monitor Places):* The supervision based on monitor places provides a supervisor in the form of a set of monitor places D_s :

$$D_s = -L\,D \quad (2)$$

$$\mu_{0,s} = b - L\,\mu_0 \quad (3)$$

where D is the incidence matrix of the plant $\mathcal{N}$, D_s is the incidence matrix of the supervisor, $\mu_{0,s}$ the initial marking of the supervisor, and μ_0 is the initial marking of $\mathcal{N}$. The supervised system, that is the closed-loop system, is a PN of incidence matrix $D_c = [D^T(-L\,D)^T]^T$.

Let μ_c be the marking of the closed-loop, and let $\mu_c|\mathcal{N}$ denote μ_c restricted to the plant $\mathcal{N}$. $t \in T$ is closed-loop enabled if μ_c enables t; t is plant-enabled, if $\mu_c|\mathcal{N}$ enables t in $\mathcal{N}$.

Let us now introduce the concept of supervisor admissibility for a centralized scenario, i.e., c-admissibility.

***Definition** 3 (Supervisor Admissibility):* A supervisor is admissible, if it only controls controllable transitions and it only detects observable transitions. The constraints $L\mu \leq b$ are admissible if the supervisor defined by (2) - (3) is admissible. When inadmissible, the constraints $L\mu \leq b$ are transformed (if possible) to an admissible form $L_a\mu \leq b_a$ such that $L_a\mu \leq b_a \Rightarrow L\mu \leq b$. Then, the supervisor enforcing $L_a\mu \leq b_a$ is admissible, and enforces $L\mu \leq b$ as well. A plant $\mathcal{N}$ with a set of q_c controllable transitions T_c and q_o observable transitions T_o will be denoted as $(\mathcal{N}, T_c, T_o)$.

Let us now introduce the concept of d-admissibility proposed in [1] for a decentralized scenario.

***Definition** 4 (Decentralized Admissibility):* Given $(\mathcal{N}, \mu_0, T_{c,1}, \ldots, T_{c,n}, T_{o,1}, \ldots, T_{o,n})$ a constraint is d-admissible if there is a set $\mathcal{C} \subseteq \{1, 2, \ldots, n\}$, $\mathcal{C} \neq \emptyset$, such that the constraint is c-admissible with respect to $(\mathcal{N}, \mu_0, T_c, T_o)$, where $T_c = \cup_{i\in\mathcal{C}} T_{c,i}$ and $T_o = \cap_{i\in\mathcal{C}} T_{o,i}$. A set of constraints is d-admissible if each of its constraints is d-admissible.

***Remark** 1:* As pointed out in [1], even though the definition of d-admissibility can take advantage of situations in which there are sets $T_{o,i}$ that are not disjoint, the sets $T_{o,i}$ do not necessarily need to share common transitions, as the set $\mathcal{C}$ may be a singleton. However, in a distributed scenario, where $\mathcal{C}$ is not expected to be a singleton, the fulfillment of the T_o condition is significantly constrained by the scale of the system, since an intersection of the observable transitions for each supervisor will likely tend to an empty set with respect to an increasing size of the system.

VI. Consensus-based Decentralized Supervision of Petri-Nets

In this section a consensus-based approach to achieve the d-admissibility in a distributed fashion by allowing one-hop collaboration among supervisors is described. The idea is to develop a communication mechanism by which first the supervisors can retrieve the set of observable transitions and successively can issue a proper control action. To this end, two different consensus algorithms, namely an *observation* consensus and a *control* consensus, will be introduced.

Let us now summarize the assumptions made in the introduction for the proposed scenario:

***Assumptions** 1:*

a) Control under concurrent firing,
b) A global clock with sampling time Δ_p is available for the supervisors,
c) Controllable transitions can be fired according to the sampling time Δ_p,
d) Detectable but uncontrollable transitions may fire asynchronously,
e) The network of supervisors $\mathcal{S}$ is described by an undirected graph $\mathcal{G}$,
f) The communication range is limited by a maximum communication radius $r \in \mathbb{R}$,
g) A unique identifier (ID) is associated to each transition t,
h) A state t_i is associated to each transition i, and it denotes the number of times the transition has been fired.

A. Observation Consensus

An observation consensus is a collaboration mechanism by which supervisors can reach a common knowledge about status of all the controllable transitions of a Petri Net.

Let us indicate with h the time at which a controlled transition is fired according to the sampling time Δ_p associated to the global clock previously introduced. Furthermore, let us introduce an additional sampling time Δ_c for the consensus algorithm and let us indicate with k the k-th step of this algorithm. Obviously, this sampling time Δ_c must be such that the convergence of the consensus algorithm is always reached before another transition is fired. As it will be shown in Section VI-C, this turns out to be a design parameter related to the diameter $d(\mathcal{G})$ of the graph $\mathcal{G}$ describing the interconnection among the supervisors.

At this point, by neglecting the temporal index h for the sake of clarity, let us define the local state $T_o^{[i]}(k)$ of a supervisor i at step k as the integer vector $\left[t_1^{(i)}(k), \ldots, t_{q_o}^{(i)}(k)\right]$ describing the state of the observable transitions available to this supervisor at step k, where $t_j^{(i)}(k) = v$ if the transition j has been fired v times in the last interval of time Δ_p and the i-th supervisor is aware of it at the step k of the observation consensus, $t_j^{(i)}(k) = \perp$, with $\perp = 0$, otherwise. Note that, the local knowledge $t_j^{(i)}(k)$ might differ from the real state t_j of a transition j at time h as this information might not be available yet to the supervisor i at step k of the observation consensus. Indeed, the goal of this consensus mechanism is to let each supervisor have a local knowledge which is consistent with the real state of all the observable transitions of a Petri net.

Let us now describe the collaborative technique by which each supervisor can achieve this common knowledge about the state of all the observable transitions.

***Lemma** 2 (Observability via Lattice Consensus):* Let us consider a Petri Net described as $(\mathcal{N}, \mu_0, T_{c,1}, \ldots, T_{c,n}, T_{o,1}, \ldots, T_{o,n})$. Furthermore, let us consider a set $\mathcal{S}$ of supervisors described by an undirected graph $\mathcal{G} = \{V, E\}$ with $|V| = n$. If the graph $\mathcal{G} = \{V, E\}$ is connected, then by applying the lattice consensus given in Lemma 1 with respect to the "join" operator ($\oplus$), the following holds:

$$T_o^{[i]}(k) = \bar{T}_o \qquad \forall\, k \geq k_o \tag{4}$$

where $\bar{T}_o = \bigoplus_{i \in V} T_{o,i}$ is the set of observable transition, $T_o^{[i]}(k)$ is the local (partial) knowledge of set $\bar{T}_o$ of observable transition for the i-th supervisor a the step k and k_o is the number of steps required to reach the convergence.

Proof: The proof of this lemma can be simply derived from Lemma 1 by assuming the vector $T_o^{[i]}(k)$ as the current state $x^{(i)}(k)$ for each agent and the vector $\bar{x} = T_o^{[1]}(0) \oplus T_o^{[2]}(0) \oplus \ldots \oplus T_o^{[n]}(0)$ as the combination of all the internal supervisors state at time $k = 0$. ■

In the following, a characterization of the convergence time of the lattice consensus described in Lemma 2 is provided. Indeed, it can be noticed that in the more general case where no assumption is made on the locality of the transition observability for each supervisor, this protocol is optimal as it requires a minimal exchange of information.

***Lemma** 3 (Convergence Time):* Let us consider a Petri Net described as $(\mathcal{N}, \mu_0, T_{c,1}, \ldots, T_{c,n}, T_{o,1}, \ldots, T_{o,n})$ and let us consider a set $\mathcal{S}$ of supervisors (multi-agents system) described by an undirected graph $\mathcal{G} = \{V, E\}$ with $|V| = n$. If the graph $\mathcal{G} = \{V, E\}$ is connected, then the consensus given in Lemma 2 reaches the convergence in time:

$$k_o = d(\mathcal{G}) \tag{5}$$

where $d(\mathcal{G})$ is the diameter of the graph, i.e., the greatest distance between any pair of vertices.

Proof: In order to prove this lemma, let us consider two supervisors S_i, S_j described respectively by the vertexes v_i, v_j for which a path of length $\bar{d} = d(\mathcal{G})$ exists. According to the Assumptions 1-e),1-f) at each time step k, the state of the two supervisors will be updated as follows:

$$\begin{aligned} T_o^{[i]}(k) &= \oplus_{\, h \in \mathcal{V}_i^\star(k)} T_o^{[h]}(0) \\ T_o^{[j]}(k) &= \oplus_{\, h \in \mathcal{V}_j^\star(k)} T_o^{[h]}(0) \end{aligned}$$

with $\mathcal{V}_i^\star(k) = \mathcal{V}_i(k) \cup \{i\}$ where $\mathcal{V}_i(k)$ is the set of k-hops neighbors for the i-th supervisor. In particular, being $\bar{d}$ the longest path among two supervisors, after $\bar{d}$ steps the neighborhoods $\mathcal{V}_i^\star(\bar{d}) = \mathcal{V}_j^\star(\bar{d}) = V$. Therefore, the current state of both supervisors will be:

$$T_o^{[i]}(\bar{d}) = T_o^{[j]}(\bar{d}) = \oplus_{\, h \in V} T_o^{[h]}(0) = \bar{T}_o$$

and so will it be for any other supervisor S_h. ■

Let us now introduce a theorem that details a set of conditions under which d-admissibility can be achieved through collaboration among supervisors.

***Theorem** 1 (Decentralized Collaborative Admissibility):* Let us consider a Petri Net described as $(\mathcal{N}, \mu_0, T_{c,1}, \ldots, T_{c,n}, T_{o,1}, \ldots, T_{o,n})$ and let us assume a network of supervisors $\mathcal{S}$ described by an undirected graph $\mathcal{G} = \{V, E\}$ with $|V| = n$. Furthermore, let us assume the set of constraints to be c-admissible with respect to $(\mathcal{N}, \mu_0, T_c, \bar{T}_o)$ where $T_c = \cup_{i \in V} T_{c,i}$ and $\bar{T}_o = \cup_{i \in V} T_{o,i}$. If the network of supervisor $\mathcal{S}$ applies the consensus algorithm described in Lemma 2, a sufficient condition for its d-admissibility with $\mathcal{C} = V$ is that the graph $\mathcal{G}$ is connected.

Proof: The theorem can be proven by applying the Definition 4 to the result given by Lemma 2. Indeed, the consensus algorithm described in Lemma 2 allows each supervisor $\mathcal{S}_i$ to reach $T_o^{[i]}(d) = \bar{T}_o$ after d steps. Therefore, according to Definition 4, the d-admissibility is a consequence of the fact that:

$$\begin{aligned} T_o &= \cap_{i \in V} T_{o,i} = \cap_{i \in V} T_o^{[i]}(d) \\ &= \cap_{i \in V} \bar{T}_o = \cup_{i \in V} \bar{T}_o \\ &= \bar{T}_o \end{aligned}$$

which implies $(\mathcal{N}, \mu_0, T_c, T_o) = (\mathcal{N}, \mu_0, T_c, \bar{T}_o)$. ■

***Remark** 2:* It should be noticed that connectivity condition guarantees the feasibility of a decentralized implementation. Indeed, by assuming the graph $\mathcal{G} = \{V, E\}$ to be connected, the set of supervisors can always set-up an observation consensus by which an agreement toward $\bar{T}_o$ can be reached.

B. Control Consensus

A control consensus is a collaboration mechanism by which supervisors can reach an agreement on a control input that satisfied the specifications.

In particular, once the consensus towards the set of observable transitions has been reached, supervisors are required to issue a control action so that the plant is left free to evolve. Nevertheless, with the assumption of control under concurrent firing, a decentralized mechanism for the coordination among the supervisors is required . This is due to the fact that, a situation of mutual exclusion, as for the concurrent firing of transitions, implies a nondeterministic behavior of the Petri net. Therefore, the supervisors must be aware of the control action taken by each other in order to satisfy the specifications.

As in the case of the observation consensus, let us consider a sampling time Δ_c for the consensus algorithm (the same as in the observation consensus case for sake of simplicity) and let us drop the global time index h for the sake of clarity. Let us now denote with $C^{[i]}(k) = \left[c_1^{(i)}(k), \ldots, c_{q_c}^{(i)}(k)\right]$ the integer vector describing the firing policy for all the controllable transitions of which the i-th supervisor is aware of up to the k-th step of the control consensus. Note that, while at step $k = 0$ such a vector describes exactly the control policy adopted by the i-th supervisor, for any $k > 0$, this vector integrates the control policy of all the supervisors it has (directly or indirectly) collaborated with, while at step $k = d(\mathcal{G})$ it describes exactly the control policy of all the supervisors $\{\mathcal{S}_1, \ldots, \mathcal{S}_n\}$ which define the decentralized supervisor $\mathcal{S}$. In particular, $c_j^{(i)}(k) = v$, describes the fact that the j-th transition can fire at most v times according o all supervisor of which the i-th supervisor is aware of (including itself) up to the step k of the control consensus, while $c_j^{(i)}(k) = m$, with $\top = m$, describes the fact that the supervisor i does not have any information concerning the control policy of the j-th transition up to step k.

A particular attention should be paid to the case of concurrent firing described in the introduction (Fig. 1) for which a single centralized monitor place is distributed into two or more monitor places assigned to different supervisors in the decentralized scenario. In the proposed control framework, we assume that only one of those supervisors takes a decision which will be communicated to the others by means of the control consensus. Note that, this assumption is not so restrictive as both supervisors will be aware of this situation each time it happens due to the information provided by the observation consensus previously performed.

Let us now describe the collaborative technique by which each supervisor can reach a complete knowledge about the control policy adopted by the other supervisors. Clearly, this implies that, even though the behavior of the Petri net is not deterministic, control actions can be taken so that specifications are satisfied.

***Lemma** 4 (Controllability via Lattice Consensus):* Let us consider a set $\mathcal{S}$ of supervisors described by an undirected graph $\mathcal{G} = \{V, E\}$ with $|V| = n$. If the graph $\mathcal{G} = \{V, E\}$ is connected, then by applying the lattice consensus given in Lemma 1 with respect to the "meet" operator ($\otimes$), the following holds:

$$C^{[i]}(k) = \bar{C} \qquad \forall\, k \geq k_c \tag{6}$$

where $\bar{C} = \bigotimes_{i \in V} C_i(0)$ is the global result of the set of local policy adopted by each supervisor $\mathcal{S}_i$ and k_c is the time required to reach the convergence.

Proof: The proof of this lemma can be simply derived from Lemma 1 by assuming the vector $C^{[i]}(k)$ as the current state $x^{(i)}(k)$ for each agent and the vector $\bar{x} = C^{[1]}(0) \otimes C^{[2]}(0) \otimes \ldots \otimes C^{[n]}(0)$ as the combination of all the internal supervisors state at time $k = 0$. ∎

By following the analysis proposed for the observation consensus, let us now provide a characterization of the convergence time of the lattice consensus described in Lemma 4.

***Lemma** 5 (Control Consensus Convergence Time):* Let us consider a set $\mathcal{S}$ of supervisors (multi-agents system) described by an undirected graph $\mathcal{G} = \{V, E\}$ with $|V| = n$. If the graph $\mathcal{G} = \{V, E\}$ is connected, then the consensus given in Lemma 4 reaches the convergence in time:

$$k_c = d(\mathcal{G}) \tag{7}$$

where $d(\mathcal{G})$ is the diameter of the graph, i.e., the greatest distance between any pair of vertices.

Proof: The proof follows the same argument used to prove in Lemma 3 the convergence time of the observation consensus. ∎

C. Feasibility Analysis

In this section, an analysis concerning the feasibility of the proposed approach is provided. Indeed, as described in the introduction, a particular attention should be paid to the relationship between the sampling time Δ_p of the Petri net evolution and the sampling time Δ_c of the two consensus algorithms for the implementation of such a control framework. In particular, by assuming the sampling time Δ_p for the evolution of the Petri to be given, the sampling time Δ_c for the two consensus algorithms should be chosen according to the following inequality:

$$k_c\, \Delta_c + k_o\, \Delta_c \leq \Delta_p \tag{8}$$

which simply states that the time required for the convergence of the two consensus algorithms must be less or equal to the time between two consecutive steps of the Petri net evolution. In particular, by exploiting the result provided respectively by Lemma 3 for the observation consensus and Lemma 5 for the control consensus, the inequality described by eq. (8) can be fulfilled assuming:

$$\Delta_c \leq \frac{\Delta_p}{2\, d(\mathcal{G})}. \tag{9}$$

This implies that the choice of the sampling time S_a for the two consensus algorithms is strictly related to the particular network topology adopted for the collaboration among the supervisors, thus it can be considered as a parameter design.

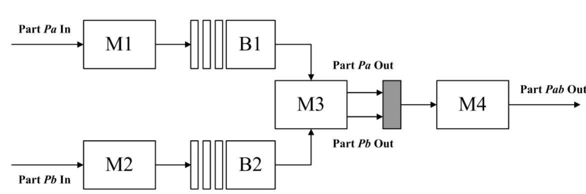

Fig. 2. Abstract model of the manufacturing workcell.

VII. An Example

In this section, a manufacturing example is presented to corroborate the theoretical analysis and the feasibility of the proposed approach. Fig. 2 describes the abstract model of the considered manufacturing work-cell. In detail, this example illustrates the the manufacturing of a product composed of

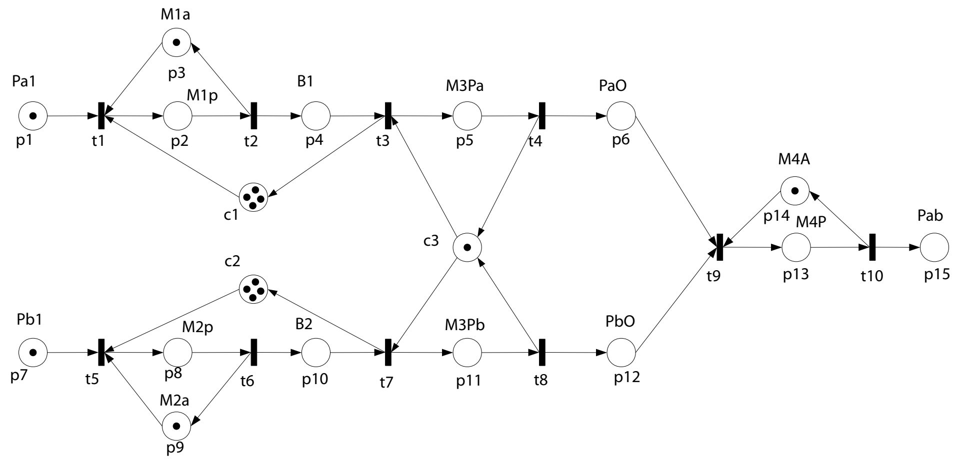

Fig. 3. PN of the manufacturing workcell with the centralized supervisor.

two different types of parts. In a first phase, two different types of parts (Pa and Pb) are produced in parallel by using machines M1 and M2, each product is moved to a common area by the robot M3. In a second phase, the resulting products are assembled by using machine M4 to obtain the final product (Pab) that leaves the manufacturing cell. Note that, in order to decouple the part production from their transportation, two buffers (B1 and B2), one for each machine, are introduced.

The proposed manufacturing work-cell can be modeled by means of a PN as depicted in Fig. 3. The machines could be either available (M1A $\{p3\}$, M2A $\{p9\}$, M4A $\{p14\}$) or processing a part (M1P $\{p2\}$, M2P $\{p8\}$, M3Pa $\{p5\}$, M3Pb $\{p11\}$, M4P $\{p13\}$). As parts enter the work-cells (PaI $\{p1\}$, PbI $\{p7\}$), they are placed in two buffers (B1 $\{p4\}$, B2 $\{p10\}$) respectively, and finally they are moved to the common assembly phase (PaO $\{p6\}$, PbO $\{p12\}$) to obtain final products (Pab $\{p15\}$).

Therefore, the whole process can be divided into three main phases:

- the manufacturing of part Pa, described by the sequence of transitions $\{t1, t2, t3, t4\}$;
- the manufacturing of part Pb, described by the sequence of transitions $\{t5, t6, t7, t8\}$;
- the assembly phase described by the sequence of transitions $\{t9, t10\}$.

As far as the supervision of the manufacturing work-cell is concerned, the following supervisory requirements have to be satisfied for the aforementioned PN $\mathcal{N}$. If the buffer is full (B1 or B2), the entrance of parts in the machine (M1 or M2) has to be denied. By assuming the two buffers to have the same capacity h, the requirements can be written, for buffer B1 and B2 respectively, as:

$$\mu_2 + \mu_4 \leq h, \tag{10}$$

$$\mu_8 + \mu_{10} \leq h. \tag{11}$$

Another requirement is that the mutual exclusion of the robot (M3) have to be guaranteed; the constraint is:

$$\mu_5 + \mu_{11} \leq 1. \tag{12}$$

It should be noticed that, according to Fig. 3, a c-admissible supervisor can be obtained by assuming $T_o = \{t_1, t_3, t_4, t_5, t_7, t_8\}$ and $T_c = \{t_1, t_3, t_5, t_7\}$. Let us recall that, according to the definition provided in Section V, a supervisor is admissible if it only controls controllable transitions and only detects observable transitions.

At this point, in order to enforce these constraints in a distributed way, we attempt to obtain a d-admissible representation of the PN. To this end, we assume to have the following decentralized supervisor:

- $\mathcal{S}_1$ with $T_{c,1} = \{t_1\}$ and $T_{o,1} = \{t_1, t_3\}$;
- $\mathcal{S}_2$ with $T_{c,2} = \{t_5\}$ and $T_{o,2} = \{t_5, t_7\}$;
- $\mathcal{S}_{3a}$ with $T_{c,3a} = \{t_3\}$ and $T_{o,3a} = \{t_3, t_4\}$;
- $\mathcal{S}_{3b}$ with $T_{c,3b} = \{t_7\}$ and $T_{o,3b} = \{t_7, t_8\}$.

Thus, enforcing (10), (11) for $h = 4$, and the constraint (12), results in the control places C_1, C_2, C_{3a}, and C_{3b} as shown in Fig. 4, where detectable transitions that are not observable for a supervisor are drawn empty. Note that, the monitor place C_3 has been decentralized into two monitor places C_{3a} and C_{3b}. Furthermore, we assume that the control law related to these two monitors is taken by the supervisor C_{3a}. This implies that, the supervisor C_{3b}, according to Section VI-B, always takes $\top$ as a control decision for t_3 and t_7.

Nevertheless, the system turns out to be dc-admissible if communication is allowed among the supervisors.

In particular, let us assume the communication topology of the network of supervisors to be described by the following undirected graph $\mathcal{G} = \{V, E\}$ with $V = \{1, 2, 3, 4\}$ and $E = \{(1, 2), (1, 3), (3, 4)\}$. According to Theorem 1, since the set of constraints is c-admissible with respect to ($\mathcal{N}$, μ_0, T_c, $\bar{T}_o$) where $T_c = \cup_{i \in V} T_{c,i}$ and $\bar{T}_o = \cup_{i \in V} T_{o,i}$ and the graph $\mathcal{G}$ is connected, the d-amissibility can be enforced by applying the lattice consensus given in Lemma 2.

Furthermore, according to the assumption of control under concurrent firing, the transitions t_3 and t_7 cannot be fired at the same time. This implies that a coordination among the supervisors is required to avoid the two enabled transitions

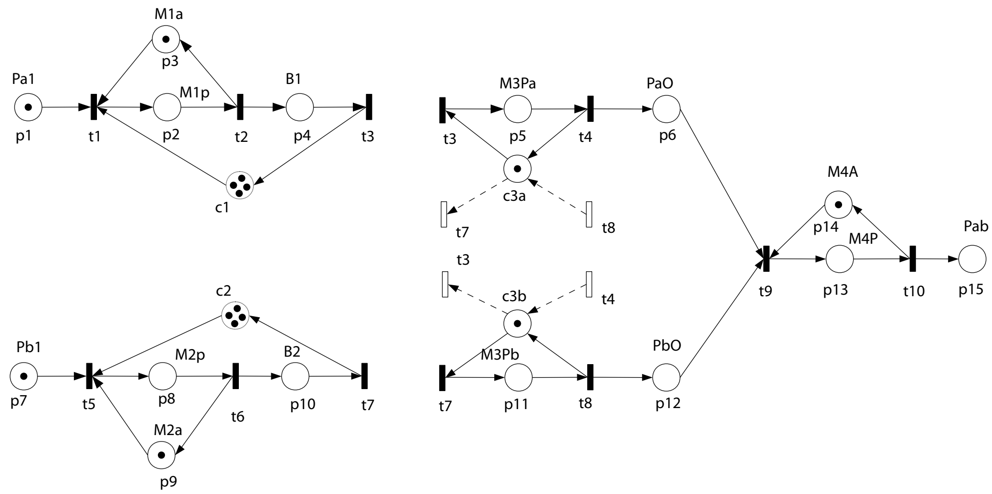

Fig. 4. Decentralized PN of the manufacturing workcell.

to be fired at the same time. Indeed, according to Lemma 4 this can be obtained by exploiting the proposed control consensus.

Finally, according to Lemmas 3 and 5, the convergence time for both consensus algorithms is $d = 3$ steps. Regarding the optimality of the communication, i.e., the minimum number of messages to be exchanged in order to reach the consensus, it should be noticed that, generally speaking, the optimality is strictly related to the diameter of the graph. Therefore, an algorithm which does not require a number of messages higher than this threshold can be considered to be effective. Indeed, this is the case for the proposed algorithms which do require exactly the same number of messages as the diameter of the network.

VIII. Conclusions

In this work the problem of decentralized supervision of a Petri net has been addressed. Starting from the theoretical framework introduced in [1] by Iordache and Antsaklis, a general framework for the design of a truly decentralized control law which can be effectively implemented has been proposed. In particular, by assuming the communication among supervisors to be limited to one-hop neighbors, a sufficient condition for achieving d-admissibility focusing the attention on the communication topology of the network of supervisors has been proposed. Furthermore, under the assumption of control under concurrent firing, a protocol to reach an agreement on the control input among the supervisors has been proposed. Finally, a feasibility analysis of the proposed decentralized control framework has been discussed.

References

[1] M. Iordache and P. Antsaklis, "Decentralized supervision of Petri nets," *Automatic Control, IEEE Transactions on*, vol. 51, no. 2, pp. 376 – 381, feb. 2006.

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

[3] A. Giua, F. DiCesare, and M. Silva, "Generalized mutual exclusion constraints on nets with uncontrollable transitions," in *IEEE Int. Conf. on Systems, Man, and Cybernetics*, Chicago, Illinois, 1992, p. 974979.

[4] J. O. Moody and P. J. Antsaklis, *Supervisory Control of Discrete Event Systems Using Petri Nets*. Kluwer, 1998.

[5] F. Basile, A. Giua, and C. Seatzu, "Supervisory control of Petri nets with decentralized monitor places," in *26th American Control Conference*, New York, USA, July 2007, pp. 4957 – 4962.

[6] ——, "Some new results on supervisory control of Petri nets with decentralized monitor places," in *17th IFAC World Congress*, Seoul, Korea, July 2008.

[7] L. E. Holloway, B. H. Krogh, and A. Giua, "A survey of Petri net methods for controlled discrete event systems," *Discrete Event Dynamic Systems*, vol. 7, pp. 151–190, 1997.

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

[9] ——, "Decentralized supervisory control of discrete-event systems," *Information Sciences*, vol. 44, no. 3, pp. 199–224, 1988.

[10] J. Komenda and J. H. Schuppen, "Control of discrete-event systems with partial observations using coalgebra and coinduction," *Discrete Event Dynamic Systems*, vol. 15, no. 3, pp. 257–315, 2005.

[11] G. Barrett and S. Lafortune, "Decentralized control with communicating controllers," *IEEE Transactions on Automatic Control*, vol. 45, no. 9, pp. 1620–1638, 2000.

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

[13] 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, 2010.

[14] N. Palomeras, P. Ridao, C. Silvestre, and A. El-Fakdi, "Multiple vehicles mission coordination using Petri nets," in *Proc. IEEE International Conference on Robotics and Automation, ICRA 2010*, 2010.

[15] M. Wooldridge, *An Introduction to MultiAgent Systems*. Wiley Publishing, 2009.

[16] N. A. Lynch, *Distributed Algorithms*. Morgan Kaufmann Publishers, Inc., 1997.

[17] V. Borkar and P. Varaiya, "Asymptotic agreement in distributed estimation," *Automatic Control, IEEE Transactions on*, vol. 27, pp. 650 – 655, 1982.

[18] J. N. Tsitsiklis and M. Athens, "Convergence and asymptotic agreement in distributed decision problems," *Automatic Control, IEEE Transactions on*, vol. 29, no. 8, pp. 690 – 696, 1984.

[19] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215 –233, jan. 2007.

[20] A. Fagiolini, M. Pellinacci, M. Valenti, G., G. Dini, and A. Bicchi, "Consensus-based distributed intrusion detection for multi-robot systems," in *Proc. IEEE International Conference on Robotics and Automation*, 2008.

[21] T. Murata, "Petri nets: Properties, analysis and applications." *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, apr 1989.