DECENTRALIZED SUPERVISION OF MOBILE SENSOR NETWORKS USING PETRI NET

Fatemeh Jafarinejad¹ and Ali A. Pouyan²

¹Department of Computer & IT Engineering, Shahrood University of Technology, Shahrood, Iran ²Department of Computer & IT Engineering, Shahrood University of Technology, Shahrood, Iran

ABSTRACT

In semiautonomous mobile sensor networks, since human operators may be involved in the control loop, particular improper actions may cause accidents and result in catastrophes. For such systems, this paper proposes a decentralized supervisory control system to accept or reject the human-issued commands so that undesirable executions never be performed. In the present approach, Petri nets are used to model the operated behaviors and to synthesize the decentralized supervisory system. The presented technique could be applied to large-scale and complicated wireless mobile sensor networks.

KEYWORDS

District Event System, Petri net, Sensor Network, Supervisory Control.

1. INTRODUCTION

Nowadays, sensor networks (SNs) have been used in different areas such as networking, embedded systems, pervasive computing, and multi agent systems due to its wide array of realworld applications [1]. In particular, wireless sensor networks (WSNs) with the ability of sensing, storing and processing data can detect and monitor any different physical conditions such as temperature, pressure, sound, etc. Moreover, WSNs can be deployed in extremely harsh environments and hostile regions (ocean floor, active volcanoes, mines, forests) [2,3]. Furthermore, they are used in wide variety of fields such as control systems, health monitoring, bio-medical applications, detect pollution, detect smoke to fire alarm, military (battlefield surveillance and troop movements), burglary and so on [4].

Because of time consumption and hardship of configuration of WSNs, mobile sensor networks (MSNs) are used to support self-configuration, adaptability, scalability, and optimal performance. These features, achieved by changing network topology, can react to the events of environment or change the mission planning [5].

In some of MSN systems human operators use semiautonomous robots for charging the static sensors, repairing replacing or removing the static sensors, maintaining network coverage for both sensing and communication, and investigating condition of launching an alert by several static sensors [6] (Fig. 1). In such cases, human errors in sending a command to robots have a significant influence on system. Therefore, the use of a controller to control and filter the commands received from the human is a good idea to manage these "human-in-the-loop" errors and it improves the overall reliability of the system. This kind of controller is called supervisor

and in such environment with some robots, designing a Supervisor is necessary to restrict action of robots to preserve the mutual exclusions of the multi-agent system.

In this paper a Petri net decentralized supervisor to control the agents of an MSN system is proposed. In section 2, we explain a brief summary of the related work. Section 3 brings an introduction to Petri nets and its usage in supervisory control. In section 4 the proposed method is described by an example. Finally the conclusion is mentioned in section 5.



Figure 1. Human – involved MSNEs

2. RELATED WORK

Autonomous robotic sensor agents are used in complex environments for active investigation. In [7], the networks of such sensor agents for these circumstances have been studied.

An important issue in these applications is control of robots. Some papers use the Supervisory control theory to provide a suitable framework for controlling the states in a MSN system.

Lee and Hsu in [8] propose (for the first time using Petri nets) a technique to design supervisory agents to prevent abnormal human operations from being carried out. This supervisory approach was also applied to human-computer interactive systems [6].

In [9] Lee and Chung proposed a PN-based localization scheme on a discrete event control framework for indoor service robots.

In 2008, Lee [6] also proposes a command filtering framework to accept or reject the humanissued commands such that undesirable executions are never performed. He uses Petri nets to model the operated behaviors and to synthesize the command filters for supervision.

In [5], Low describes a distributed layered architecture for resource-constrained multi robot cooperation, which is utilized in autonomic mobile sensor network coverage. In the upper layer, a dynamic task allocation scheme self-organizes the robot coalitions to track efficiently across regions. It uses concepts of ant behavior to self-regulate the regional distributions of robots in proportion to that of the moving targets to be tracked in a non-stationary environment. As a result, the adverse effects of task interference between robots are minimized and network coverage is improved. In the lower task execution layer, the robots use self-organizing neural networks to coordinate their target tracking within a region. Both layers employ self-organization techniques,

which exhibit autonomic properties such as self-configuring, self-optimizing, self-healing, and self-protecting.

In 2010, Moldoveanu [1], continued Low work and proposed a command filtering framework to accept or reject the human-issued commands so that undesirable executions are never performed. He also used Petri nets to model the operated behaviors and to synthesize the command filters for supervision.

3. PETRI NET AND IDEA OF SUPERVISORY CONTROL

We can think of a human-in-the-loop system as a discrete-event system (DES) because the progress in the system and state changes in it are driven by occurrences of individual events of human action. Supervisory control theory (SCT) provides a suitable framework for synthesizing DES [10]. Primarily, SCT was studied by automaton based models but nowadays an increasing interest is given to Petri net based models [11]. Petri net is a graphical and mathematical modeling tool which can be used as a visual communication aid. Basically, Petri net is a bipartite graph consisting of two types of nodes, places and transitions, connected by arcs.

A Petri net is a 5-tuple [12], Petri net = (P, T, F, W, M_0) where:

 $P = \{p_1, p_2, \dots, p_m\} \text{ is a finite set of places,} \\T = \{t_1, t_2, \dots, t_n\} \text{ is a finite set of transitions,} \\F \subseteq (P \times T) \cup (T \times P) \text{ is a set of arcs (flow relation),} \\W: F \rightarrow \{1, 2, 3, \dots\} \text{ is a weight function,} \\M_0: P \rightarrow \{1, 2, 3, \dots\} \text{ is the initial marking,} \end{cases}$

 $P \cap T = \emptyset \text{ and } P \cup T \neq \emptyset. \tag{1}$

Places and transitions are called nodes and denote states and events in the DES. In Petri net with m places and n transitions, the incidence matrix A is $n \times m$ matrix whose elements are:

$$A_{ij} = w(t_i, p_i) - w(t_i, p_j)$$
 (2)

Where w (t, p) is the weight of the arc between p and t. If all arcs in Petri net have weights equal to 1, it should be noted that:

 $A = O - I \tag{3}$

The matrices I (input matrix) and O (output matrix) provide a complete description of the structure of PN. If there are no self loops, the structure may be described only by A. The incidence matrix allows an algebraic description of the evolution of the marking of PN. The marking of Petri net changes from marking m_k to marking m_{k+1} :

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \mathbf{W}^{\mathrm{I}}.\mathbf{v} \tag{4}$$

Where v is a transition vector composed of non-negative integers that correspond to the number of times a particular transition has been fired between markings m_k and m_{k+1} . [13]

The behavior and state changes of a DES system can be modeled by a Petri net model. In some cases we want to control the qualitative behavior of a system. Given an uncontrolled system, there exist supervisors which restrict the behavior of the system to the desired behavior by dynamically disallowing some of controllable events [14].

In Petri net model of the system, any marking of Petri net represent a state of system and transitions represent the state change or behavior of system. Undesired behaviors of system can be expressed as states in Petri net that are unwanted to occur, called forbidden states. If these states are not related to each other, called arbitrary forbidden state, they can be controlled by reachability analysis of Petri net (e. g. work of Dideban and Alla in [15]).

But if the constraints are related to each other and sets of forbidden states can be described in a linear equation, the problem of restricting the behavior, called generalized mutual exclusion problem, can be solved by Place-invariant analysis of Petri net model. This analysis will impose some extra places, called supervisor places, in net. Finally by solving an LP equation, the initial marking and incidence matrix of supervisor places are:

$$M_{0,s} = b - LM_0$$
 (5)
 $D_s = -LD$ (6)

In which the b and L are elements of constraint equation: LM < b. L is the weight of each place in this constraint, and b is the upper bound of linear composition of marking of places.

In large scale systems, because of reasons such as security and cost of sending commands, a control architecture without central coordination and with communication between local supervisors is desired, the idea which is called "thinking globally and acting locally" by Rudie and Wonham in [16].

A decentralized supervisor S consists of a set of supervisors $S_1, S_2, ..., S_n$ operating in parallel, such that a given specification is satisfied. This idea has been proposed for various applications such as manufacturing, failure detection, and communication protocol [16].

In [17] an algorithm to achieve the decentralized supervisor, given the centralized one is addressed. The idea is suitable in conditions where controllability and observability constraints of transitions to the specified supervisor are satisfied and therefore the admissibility and deadmissibility of constraints are satisfied. The Algorithm is proposed as follow in [17]:

Algorithm 1: Supervisory Design of a Deadmissible Constraint:

- 1) Let M₀ be the initial marking of the system N, and S be control places of the centralized supervisor, enforcing L.M<b. The constraints are admissible for disjoint sites of network, called de-admissible constraints.
- For all i∈ S, let x_i∈ N be a state variable of S_i: Define S_i for all i∈ S, by the fallowing rules:
- Initialize $x_i = b LM_0$
- If $t \in T_{c,i}$, $t \in C^{\bullet}$ and $x_i < W_s(C, t)$, then S_i disables t.
- If t fires and $t \in T_{o,i}$, $t \in \bullet C$, then $x_i = x_{i+1} W_s(C, t)$.
- If t fires and $t \in T_{o,i}$, $t \in C^{\bullet}$, then $x_i = x_i \cdot W_s(C, t)$.
- In this algorithm, C• is set of transitions belonging to output transitions of a control place. •C is the set of input transitions of a control place. $T_{c, i}$ is set of transitions that are controllable from the supervisor S_i . $T_{o, i}$ is set of transitions that are observable from the supervisor S_i . $W_s(C, t)$ is the weight of the arc between place C and transition t in supervisor S.

4. DECENTRALIZED SUPERVISION OF ROBOTS IN MSN

One idea to overcome the errors in human-in the-loop condition of a semiautonomous mobile sensor is to construct a system to filter the commands received by human. Set of admissible states of a command filtering systems in MSN are does that preserve the collision avoiding movements of robots and deadlock-free operation of them.

Supervisor system of MSNs can have different architecture to receive and send information of robots to each other. In [1], a different idea for designing architecture of supervision of agents is discussed. It addresses works of the other papers to design a command filtering framework and proposes a P2P communication framework as an architecture for MSNs (Fig. 2).



Figure 1. Command filtering framework for MSNs

We also use P2P architecture, But we use a decentralized command filter system where each robot knows just useful information of the others and according to these information decide if the command received from human is erroneous or not. Thus, the Petri net model of our system should be in a way that it can be divided into models in robots or agents of the system.

PN model of [1] is a centralized model so that the state of all robots and state of environment are controlled locally. It can be difficult to use this model in cases where the environment is more complex and where we have different constraint for each robot.

It is preferred to construct the Petri net model of each robot just according to the environment and finally, the overall system model is the composition of Petri net model of individual robots. One Advantage of this model is that if the environment is more complex the overall Petri net model is not very hard to be drived.

4.1 Example

To illustrate and explain the concept of decentralized supervision of MSN, we use the example that is presented in [1] (Fig. 3). Therefore, the proposed Petri net model is easily comparable with the Petri net model presented in [1].

The example consists of 8 rooms. Sensors and 5 robots which repair them are scattered in the rooms. Rooms can be reached from any other rooms. The goal is avoiding collision of robots in rooms. Initially all robots are in room 8. There is two constraints:

- The capacity of rooms1, 2, 5 and 6 is just one robot
- The capacity of rooms 3 and 4 is only two robots.



Figure3. Mobile wireless surveillance system with 5 robots

In [1], the Petri net model of system is based on the environment and agents. But because we want to model the overall system and then divide it in Petri net model of each robot, we develop another model of the system.

We make a Petri net model for each robot and the combination of Petri net model of all robots constructs the overall system model. In this example, because we don't have any constraint on an especial robot, the Petri net models of all robots are the same and we just show the model of one of them (Fig. 4)

State of each robot is as its existence in one room and absence of it from other rooms. Behavior or state change of robots is related to their exchanging room, so the state change of robot is related to the environment and the way between rooms. If two rooms have a way to each other, the robot can change state of itself to be in the adjacent room from the other. Hence, the Petri net model is derived very simply. Figure 4 shows the Petri net model of robot 1. In this figure, the places are named as RiPj which means that robot i is in palace j, and transitions are named as iPjPk which means that robot i moves from room j.



Figure 4. Petri net Model of Robot 1.

The constraints of such system are:

a1p1+a2p1+a3p1+a4p1+a5p1+a6p1+a7p1+a8p1 <=1

 $a1p2+a2p2+a3p2+a4p2+a5p2+a6p2+a7p2+a8p2 <= 1\\a1p5+a2p5+a3p5+a4p5+a5p5+a6p5+a7p5+a8p5 <= 1\\a1p6+a2p6+a3p6+a4p6+a5p6+a6p6+a7p6+a8p6 <= 1\\a1p3+a2p3+a3p3+a4p3+a5p3+a6p3+a7p3+a8p3 <= 2\\a1p4+a2p4+a3p4+a4p4+a5p4+a6p4+a7p4+a8p4 <= 2\\$

By solving the above constraint equations for system, the net model with control places will be achieved. The system with its supervisor can be seen in figure 5. In this figure the Petri net model of all the robots other than robot 1 are presented as robot i just for simplicity of the model shape.

Appling the decentralized approach of [17] we can disjoint the supervisor of each robot agent and construct a decentralized supervision for this MSN system. The key concept in this work is to construct the overall Petri net system in a way which can be decomposed into some disjoint supervisors. In the case of MSN systems it is important to construct the Petri net model in view point of robot agents to be able to make a decentralized supervisory system. The supervisor of robot 1 is presented in figure 6.



Figure 5. Centralized Supervisor of the MSN Example.



Figure 6. Decentralized Supervisor of Robot 1.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we suggest use of a decentralized supervision to control the robots in an MSN system. In this multi agent system, each robot has only access to the state of itself and only knows necessary information about others. In this system the security and information encapsulation are as important as controlling procedure. So, the overall system is finally decomposed and divided in supervisors of each of the robots and the overall system is modeled in robots point of view. Another advantage of dividing Petri net model of the system into Petri net model of individual robots is that it is easier to acquire the Petri net model of the entire system which is less sensitive to environmental changes.

For the future work there are two main directions of research. The basic idea of this paper can be applied to different supervisory applications to see their results and compare them to existing approaches. Moreover, acquiring a general architecture for this category of problems will be a required work.

REFERENCES

- F. Moldoveanu, D. Floroian and D. Puiu, (2010) "Petri Nets and Agents to Supervisory Control of Complex Environment", Bulletin of the Transilvania University of Brasov, Vol. 3, No. 52, pp 267-276.
- [2] F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, (2002) "A survey on sensor networks," IEEE communication Magazine, pp 102-114.
- [3] G. Werner-Allen, (2006) "Deploying a Wireless Sensor Network on an Active Volcano," IEEE Trans. Internet Computing, vol. 10, no. 2, pp 18–25.
- [4] A. A. Pouyan, S. M. M. Salehi and F. Jafarinejad, (2013) "A Markov-Based Model for Fault Tolerant and Reliable Wireless Sensor Networks," proceeding of 11th Iranian Conference on Intelligent Systems.
- [5] Low, K.H., Leow, W.K., et al., (2006) "Autonomic Mobile Sensor Network with Self-coordinated Task Allocation," IEEE Trans. Syst., Man, Cybern. C, No. 3, p. 315-327.
- [6] J. S. Lee, (2008) "A Petri Net Design of Command Filters for Semiautonomous Mobile Sensor Networks," IEEE Trans. Ind. Electron., No. 4, p. 1835-1841.
- [7] E. Petriu and T. Whalen, (2004) "Robotic Sensor Agents: A New Generation of Intelligent Agents for Complex Environment Monitoring," IEEE Instrumentation & Measurement Magazine, vol. 7, No. 3, p. 46-51.
- [8] J. S. Lee, P. L. Hsu, (2003) "Remote Supervisory Control of the Human-in-the-loop System by Using Petri Nets and Java," IEEE Trans. Ind. Electron. Vol. 50, No. 3, p. 431-439.
- D. Lee, W. Chung, (2006) "Discrete Status Based Localization for Indoor Service Robots," IEEE Trans. on Industrial Electronics. Vol. 53, No. 5, p. 1737-1746.
- [10] R. J. Ramadge and W. M. Wonham, (1987) "Supervisory control of a class of discrete event processes," SIAM Journal on Control and Optimization, vol. 25, No. 1, pp 206-230.
- [11] A. Ghaffari, N. Rezg and X. L. Xie, (2003) "Design of Live and Maximally Permissive Petri Net Controller Using Theory of Regions", IEEE Trans. On Robotics and Automation, vol. 19, no. 1.
- [12] T. Murata, (1989) "Petri nets: Properties, analysis and applications," Proceedings of the IEEE, vol. 77, No. 4, pp 541-580.
- [13] A. Gudelj, D. Kezić and S. Vidačić, (2012) "Marine Traffic Optimization Using Petri Net and Genetic Algorithm," Scientific Journal on Traffic and Transportation Research, Vol. 24, No. 6.
- [14] R. Kumar and L. E. Holloway, (1996) "Supervisory Control of Deterministic Petri Nets with Regular Specification Languages," IEEE Trans. on Automatic Control, Vol. 41, No. 2, pp 245- 249.
- [15] A. Dideban and H. Alla, (2009) "Controller Synthesis By Petri Nets Modeling," Third International Workshop on Verification and Evaluation of Computer and Communication Systems (VECoS 2009).
- [16] K. Rudie and W. M. Wonham, (2002) "think globally, act locally: decentralized supervisory control," IEEE Transaction on Automatic Control, Vol. 37, No. 11, pp 1692 – 1708.M. V. Iordache and P. J. Antsaklis, (2006) "decentralized supervision of Petri Net," IEEE Trans. on Automatic Control, Vol. 51, No. 2, pp 376-381.