SUPERVISORY CONTROL OF SEMIAUTONOMOUS MOBILE SENSOR NETWORKS

Florin D. MOLDOVEANU, Dan I. FLOROIAN

Department of Automation, Transilvania University of Brasov; moldof@unitbv.ro

***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 command filtering framework to accept or reject the human-issued commands so that undesirable executions are never performed. In the present approach, Petri nets are used to model the operated behaviors and to synthesize the command filters for supervision. An application to a mobile wireless surveillance system is provided to show the feasibility of the developed approach. It is believed that the technique presented in this paper could be further applied to large-scale wireless mobile sensor networks.**

Keywords*:* *Command Filters, Mobile Robots, Mobile Sensor Networks, Petri Nets, Wireless Sensor Networks.*

1. INTRODUCTION

Recently, there has been an increasing emphasis on developing wide-area distributed wireless sensor networks (WSNs) with self-organization capabilities to cope with sensor failures, changing environmental conditions, and different environmental sensing applications [1, 2]. In particular, mobile sensor networks (MSNs) hold out hope to support self-configuration mechanisms, guaranteeing adaptability, scalability, and optimal performance, since the best network configuration is usually time varying and context dependent. Mobile sensors can physically change the network topology, reacting to the events of the environment or to changes in mission planning. References [3, 4] points out the advantages of sensor mobility and several algorithms for network self-organization after the occurrence of predetermined events are also proposed. However, most of the MSN literature focuses on a sensory system of fully-autonomous mobile robots without human intervention.

In real applications, human operators may use semiautonomous robots, as shown in Figure 1 (a), to 1) further investigate conditions if several static sensors launch an alert, 2) maintain network coverage for both sensing and communication, 3) charge the static sensors, or 4) repair, replace, or remove the static sensors. For such human-in-the-loop systems, human errors have a significant influence on system reliability, at times more than technological failures.

Much research has been conducted to classify human errors and develop a mechanism for their reduction. Lee and Hsu [5] proposed for the first time using Petri nets (PNs) a technique to design supervisory agents for preventing abnormal human operations from being carried out. This supervisory approach was also applied to human-computer interactive systems [4]. PN has been developed into a powerful tool for modeling, analysis, control, optimization, and implementation of various engineering systems [6, 7]. Lee and Chung [4] proposed a PN-based localization scheme on a discrete event control framework for indoor service robots.

Figure 1(b) adopts the supervisory framework [5], to an MSN system composed of several static sensors and semiautonomous mobile robots regulated by human operators through a wireless network. According to the status feedback from both the sensors and robots, the supervisors provide permitted commands for human operators by disabling the actions which violate specifications. The human operator can then trigger only limited commands based on the observed status. However, the supervision is from an active viewpoint to enable or disable the commands in advance and leads to limited human actions. In addition, the MSN system requires a fast sampling rate with low-latency communication to provide supervisors with an up-to-date status to make the decision. Furthermore, each supervisor and the MSN system is based upon a client/server architecture with centralized communication, which is not an ideal topology for distributed sensor network systems.

In this paper, instead of using a client-server architecture, distributed peer-to-peer (P2P) communication between mobile robots is applied. The advantages of P2P include increased scalability (capacity scales with popularity), robustness (no single point of failure), fault tolerance,

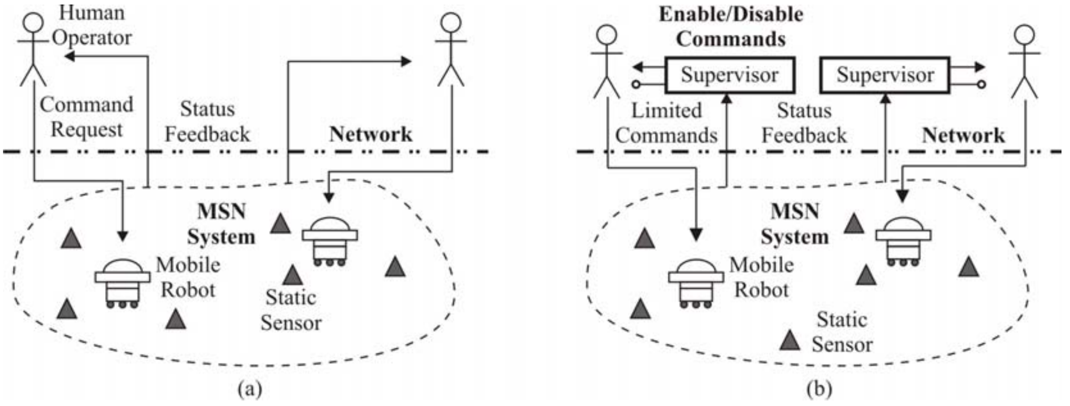

Figure1. (a) Human-involved MSNs. (b) Applied supervisory framework for such MSNs

resilience to attack, and better support and management in distributed cooperative environments. Moreover, from a passive point of view, a command filter is proposed to avoid improper control actions from being carried out as the robot receives the human commands.

As shown in Figure 2, the human operator sends command requests to the mobile robot through a wireless network. Inside the robotic computer, the command filter acquires the system status via distributed P2P communication and makes the decision to accept or reject the commands so as to meet the specifications, e.g., the collision avoidance among robots. The role of a command filter is to interact with the human operator and the mobile robot so that the closed human-in-the-loop system satisfies the requirements and guarantees that undesirable executions never occur.

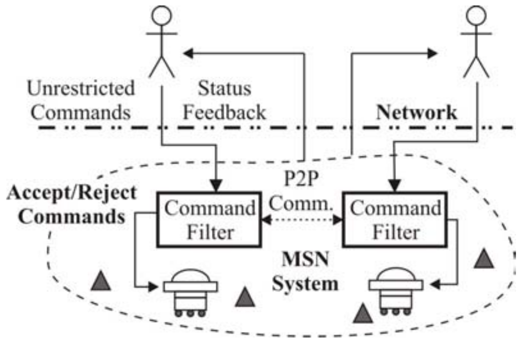

Figure 2: Proposed command filtering framework

In this paper, PNs are used in designing the command filters, yielding a compact and graphical model for the MSN. Basically, the PN design of the filters is identical to the design of the supervisors in [6], except for the implementation framework as shown in Figures 1(b) and 2. To demonstrate the feasibility of the proposed filtering framework, an application to a mobile wire less surveillance system is illustrated in this paper. During system operation, our approach ensures that remote commands from the human operator meet the given collision-avoidance requirements. Note that the research work presented in this paper is conducted in an office-like environment.

2. PETRI NETS – BASED MODELING

This section first introduces the basic PN concept, and then shows the modeling of human-involved MSNs.

2.1 Basic PN Concepts

A PN is identified as a particular kind of bipartite directed graph populated by three types of objects. They are places, transitions, and directed arcs connecting places and transitions. Formally, a PN can be defined as [8]:

$$G=(P,T,I,O) \tag{1}$$

where: $P=\{p_1,p_2,..,p_m\}$ finite set of places, where $m>0$; $T=\{t_1,t_2,\ldots,t_n\}$ finite set of transitions with $P\bigcup T\neq\varnothing$ and, where $n>0$; $I:P\times T\rightarrow N$ input function that defines a set of directed arcs from *P* to *T*, where $N=\{0,1,2,..\}$; $O:T\times P\rightarrow N$ output function that defines a set of directed arcs from *T* to *P*.

A marked PN is denoted as (G,M_0), where $M_0:P\rightarrow N$ is the initial marking. A transition *t* is enabled if each input place *p* of *t* contains at least the number of tokens equal to the weight of the directed arc connecting *p* to *t*. When an enabled transition fires, it removes tokens from its input places and deposits them on its output places. PN models are suitable for representing systems that exhibit concurrence, conflict, and synchronization.

Some important PN properties include a boundness (no capacity overflow), liveness (freedom from deadlock), conservativeness (conservation of nonconsumable resources), and reversibility (cyclic behavior). The concept of liveness is closely related to the complete absence of deadlocks. A PN is said to be live if, no matter what marking has been reached from the initial marking, it is possible to ultimately fire any transition of the net by progressing through further firing sequences. This means that a live PN guarantees deadlock-free operation regardless of the firing sequence. Validation methods of these properties include reachability analysis, invariant analysis, reduction method, siphons/traps-based approach, and simulation [8].

2.2 Modeling of semiautonomous MSNs

In semiautonomous MSNs, human behavior can be modeled using the command/ response concept. As shown in Figure 3, each human operation is modeled as a task with a start transition, end transition, progressive place and completed place. Transitions drawn with dark symbols are events controllable by the remotely located human through the network. Note that the start transition is a controllable event as "command" input, while the end transition is an uncontrollable event as "response" output. Nonhuman actions can be simply modeled as a single event transition.

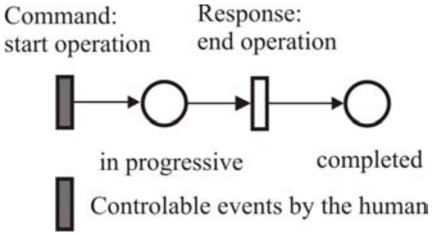

Figure 3: Modeling of human behavior using the command / response concept

3. COMMAND FILTER DESIGN

3.1. Specification Type

The objective of a command filter is to ensure the reaction of human-issued commands contained within the set of admissible states, called the specification. In this paper, two main types of specifications are considered:

1) Collision-avoidance movements: This specification presents the physical constraints of the limited resources, such as rooms and hallways. Each room limits the number of mobile robots that enter or stay avoid collisions.

2) Deadlock-free operations: This specification ensures that a given command will not lead the system to a deadlock at which no further action is possible.

The liveness of a PN is closely related to the complete absence of deadlocks. A PN is said to be live if, no matter what marking has been reached from the initial marking, it is possible to ultimately fire any transition of the net by progressing through further firing sequences. This means that a live PN guarantees deadlock-free operation regardless of the firing sequence.

During the system operation, the proposed command filter enforces these specifications by accepting or rejecting human-issued commands.

3.2 Synthesis of Command Filters

Definition 3.1: Considering two nets $G_1 = (P_1, T_1, I_1, O_1)$ and $G_2 = (P_2, T_2, I_2, O_2)$ with initial marking M_{01} and M_{02}, respectively. The synchronous composition of G_1 and G_2 is a net $G = (P, T, I, O)$ with initial marking M_0:

$$G = G_1 \parallel G_2 \tag{2}$$

where: $P = P_1 \cup P_2$; $T = T_1 \cup T_2$;
$I(p,t) = I_i(p,t)$,
if $(\exists i \in \{1,2\})$, $[p \in P_i \wedge t \in T_i]$,
else $I(p,t) = 0$;
$O(p,t) = O_i(p,t)$,
if $(\exists i \in \{1,2\})$, $[p \in P_i \wedge t \in T_i]$,
else $O(p,t) = 0$;
$M_0(p) = M_{01}(p)$
if $p \in P_i$, else $M_0(p) = M_{02}(p)$.

In this paper, an agent that specifies which events are to be accepted or rejected when the system is in a given state is called a command filter. For a system with plant model G and specification model H, the filter can be obtained by synchronous composition of the plant and specification models:

$$F = G \parallel H \tag{3}$$

where the transitions of H are a subset of the transitions of G, i.e., $T_H \subset T_G$. Note that F obtained through the above construction, in the general case, does not represent a proper filter, since it may contain deadlock states from which a final state cannot be reached. Thus, the behavior of F should be further refined and restricted by PN analysis. The design procedure of PN-based command filters consists of the following steps:

Step 1)Construct the PN model of the human commands and system responses.
Step 2)Model the required specifications.
Step 3)Compose the system and specification models to synthesize the preliminary command filter.
Step 4)Analyze and verify the properties of the composed model.
Step 5)Refine the model to obtain a deadlock-free, bounded, and reversible model according to the defined specifications.

3.3 Implementation Using Agent Technology

Agent technology is a new and important technique in recent novel researches of artificial intelligence. Using agent technology leads to a number of advantages such as scalability, event-driven actions, task-orientation, and adaptivity [9]. The concept of an agent as a computing entity is very dependent on the application domain in which it operates.

From a software technology point of view, agents are similar to software objects, although these run upon a call by other higher level objects in a hierarchical structure. On the contrary, in the narrow sense, agents must run continuosly and autonomously. The distributed multiagent coordination system is defined as the agents that share the desired tasks in a cooperative point of view and are autonomously executing at different sites.

For our purposes, we have adopted the description of an agent as a software program with the capabilities of sensing, computing, and networking associated with the specific function of command filtering for the MSN systems. A filtering agent is implemented to aquire the system status by autonomously sensing and the P2P networking abilities, after which computing is performed to accept or reject the associated commands so that desired specifications are satisfied.

From a practical point of view, the filtering agent of each robot would run the developed PN model as a state machine and have capabilities with position location (room number in our case). Each room is equipped with a sensing and communication device, such as an RFID reader or a ZigBee module, to provide the vacancy information to the robot which would enter a particular room. On the other hand, the robot may also communicate with other robots to obtain their locations.

4. EXAMPLE: A MOBILE WIRELESS SURVEILLANCE SYSTEM

4.1. System Description

The semiautonomous MSN system in Figure 2 can be applied to a mobile wireless surveillance system, which is composed of many static sensors and several human-controlled mobile robots. In this example, three mobile robots are placed on a floor with five rooms, as shown in Figure 4. Each robot can move to each room according to indicated directions. To avoid possible collisions, robots are not allowed simultaneously in the same room (R1, R2, or R3) except R4 and R5, which admits two robots during the surveillance period. The initial states of the robots are in R1, R3, and R5, respectively.

4.2 PN Modeling

By applying the command/response concept and based on the system description, the PN model of the human-controlled mobile robots is constructed as shown in Figure 5. It consists of 19 places and 28 transitions, respectively. Corresponding notation of the PN model is described in Table 1.

4.3 Command Filter Design

The five rooms represent the resources shared by the three mobile robots. Since more than one robot may require access to the same room, R1-R3 and R4, R5 can only be allowed to have one and two robots at a time, respectively, and collisions and deadlocks may thus occur. Hence, the objective is to design a command filter to ensure the whole system against these undesired situations. The required three specifications are:

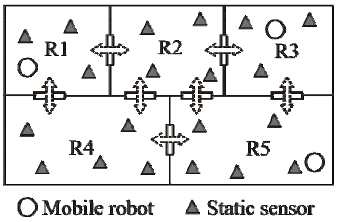

Figure 4: Mobile wireless surveillance system with three robots

Table 1: Notation of the PN in Figure 5

Place	Description	Trans.	Description
p1	Robot in R2	t1	Cmd: start moving to R1
p2	Moving to R1	t2	Re: end moving to R1
p3	Moving to R2	t3	Cmd: start moving to R2
p4	Robot in R1	t4	Re: end moving to R2
p5	Moving to R4	t5	Cmd: start moving to R4
p6	Moving to R1	t6	Re: end moving to R4
p7	Robot in R4	t7	Cmd: start moving to R1
p8	Moving to R2	t8	Re: end moving to R1
p9	Moving to R4	t9	Cmd: start moving to R2
p10	Moving to R3	t10	Re: end moving to R2
p11	Moving to R2	t11	Cmd: start moving to R4
p12	Robot in R3	t12	Re: end moving to R4
p13	Moving to R5	t13	Cmd: start moving to R3
p14	Moving to R3	t14	Re: end moving to R3
p15	Robot in R5	t15	Cmd: start moving to R2
p16	Moving to R2	t16	Re: end moving to R2
p17	Moving to R5	t17	Cmd: start moving to R5
p18	Moving to R5	t18	Re: end moving to R5
p19	Moving to R4	t19	Cmd: start moving to R3
		t20	Re: end moving to R3
		t21	Cmd: start moving to R2
		t22	Re: end moving to R2
		t23	Cmd: start moving to R5
		t24	Re: end moving to R5
		t25	Cmd: start moving to R5
		t26	Re: end moving to R5
		t27	Cmd: start moving to R4
		t28	Re: end moving to R4

1) Robot moving to Room i is allowed only when Room i is empty, where i = 1, 2, 3. Thus, we have three subspecifications denoted as Spec-1.1 to Spec-1.3.
2) Robot moving to Room i is allowed only when Room i has no more than one robot, where i = 4, 5. Thus, we have two subspecifications denotes as Spec-2.1 to Spec-2.2.
3) Liveness, i.e., no deadlock states, must be enforced throughout system operation.

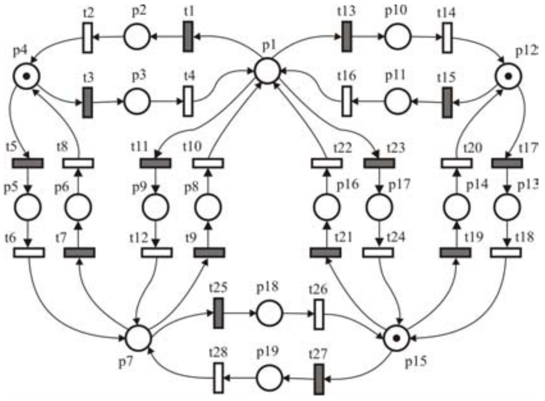

Figure 5: PN model of the human-controlled mobile robots

In the specification model, Spec-1 and Spec-2 are enforced by using the mutual exclusion theory with limited tokens. The composed PN model of both the systems are specifications is shown in Figure 6.

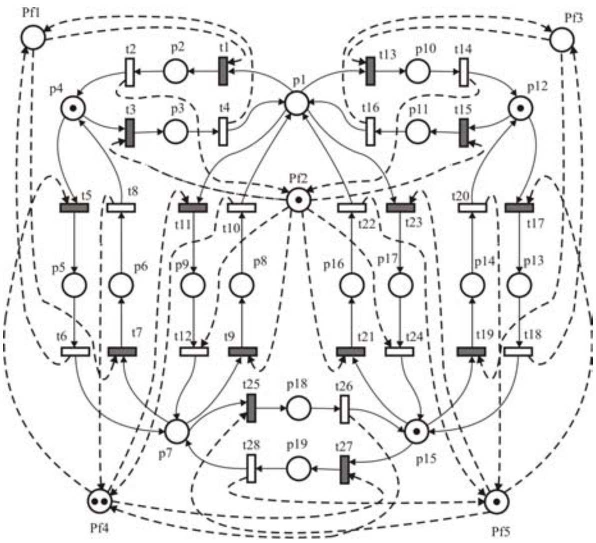

Figure 6: PN model with command filtering functions

The filtering place Pf1-Pf5 are drawn thicker and the filtering arcs are shown with dashed lines. A filtering place is modeled as an input place of the transitions that need such a resource, and as an output place of those transitions that release this resource. Take an example of Pf1 that physically means Room 1 being available. It makes two transitions t1 and t7 mutually exclusive. Intuitively, performance of t1 is only allowed if Room 1 is available and t7 has not yet been fired. If t7 has been fired, t1 cannot be executed until t6 is given to signal that Room 1 is available again. Thus, only one robot is allowed to be in Room 1 at any time, thereby avoiding any collision.

The filtering places Pf1 to Pf5 (for Spec-1 and Spec-2) are used to prevent undesired human operations that lead to resource conflicts on the part of the system.

The corresponding notation for the filtering places is described in Table 2. Note that the Pf1-Pf3 are one-bounded places and the Pf4-Pf5 are two-bounded, indicating the number of admitted robots.

Table 2: Notation of the filtering places

Place	Description
Pf1	Spec-1.1: R1 admits one robot.
Pf2	Spec-1.2: R2 admits one robot.
Pf3	Spec-1.3: R3 admits one robot.
Pf4 (2-bound)	Spec-2.1: R4 admits two robots.
Pf5 (2-bound)	Spec-2.2: R5 admits two robots.

4.4 Analysis and Verification

At this stage, due to its ease of manipulation, support for graphics import, and ability to perform structural and performance analysis, the software package MATLAB PN Toolbox [7] was chosen to verify the behavioral properties of the composed PN model using reachability analysis. The validation results reveal that the present PN model is deadlock-free, bounded, and reversible. The deadlock-free property means that the system can be executed properly without deadlocks (Spec-3), boundedness indicates that the system can be executed with limited resources, and reversibility implies that the initial system configuration is always reachable. For the human-controlled robot in the proposed command filtering framework, the human commands are accepted or rejected to satisfy the specifications so that collisions are avoided during the surveillance period. As shown in Table 3, without command filtering, the state space is 1330 with undesirable collision states. By using the filtering approach, the state space is reduced to 331. Over 75% of states would be avoided during the surveillance period, i.e., improper actions that violate Spec-1 to Spec-3 and lead to these undesired states would be successfully filtered.

In this approach, the command filter consists only of places and arcs, and its size is proportional to the number of specifications that must be satisfied. Thus, it is believed that the presented technique could be further applied to large-scale wireless MSNs. However, the specifications designed in this paper lead to limitations for office-like structured environments. New specifications for applications to unstructured environments and large-scale networks should be investigated in the future.

Table 3: Comparison

Petri net models	Places	Transitions	Arcs	State space
Unfiltered syst.	19	28	56	1330
Filtered syst.	24	28	84	331

5. CONCLUSIONS

In this paper, a framework to develop a command filter for semiautonomous MSNs with the human-in-the-loop has been presented. The command filter is systematically designed and implemented using PN modeling and agent technology. To demonstrate the practicability of the proposed approach, an application to the mobile wireless surveillance system is illustrated. According to state acquisition via distributed P2P communication, the developed command filter ensures the specifications by accepting or rejecting human-issued commands.

Compared with previous work [4, 5] which, from an active viewpoint, sought to enable or disable actions leading to limited human commands, this paper has proposed another scheme, from a passive viewpoint, to accept or reject the actions as the robot receives the human commands. Hence, in the proposed filtering framework, human operators could request unrestricted commands, and the command filters would make real-time decisions based on an event-trigger and on-demand P2P networking.

References

[1] Culler, D., Estrin, D., et al.: *Guest Editors's Introduction: Overview of Sensor Networks*. In *IEEE Computer* (2003), Vol. 37, No. 8, pp.41-49.

[2] Lee, J.S., Huang, Y.C.: *ITRI ZBnode: A ZigBee/IEEE 802.15.4 Platform for Wireless Sensor Networks*. In *Proc IEEE Int. Conf. Syst., Man, Cybern.* (2006) Taipei (Taiwan), pp.1462-1467

[3] Giordano, V., Ballal, P., et al.: *Supervisory Control of Mobile Sensor Networks: Math Formulation, Simulation and Implementation*. In *IEEE Trans. Syst., Man., Cybern. B, Cybern.,* (2006) Vol. 36, No.4, pp.806-816.

[4] Low, K.H., Leow, W.K., et al.: *Autonomic Mobile Sensor Network with Self-coordinated Task Allocation*. In *IEEE Trans. Syst., Man, Cybern., C, Appl. Rev.*, (2006) Vol. 36, No.3, pp. 315-327.

[5] Lee, J.S. , Hsu, P.L.: *Remote Supervisory Control of the Human-in-the-loop System by Using Petri Nets and Java*. In *IEEE Trans. Ind. Electron.* (2003) Vol. 50, No. 3, pp.431-439.

[6] Lee, J.S.: *A Command Filtering Framework to Collision Avoidance for Mobile Sensory Robots*. In *Proc IEEE Int. Symp. Ind. Electron.* (2007), Vigo (Spain), pp.136-141.

[7] Matlab: *Petri Nets Toolbox* (2008).

[8] Pastravanu, Oc., Matcovschi, et al.: *Applications of Petri Nets in Studying Discrete Event Systems*. Iasi (Romania), Ed. Gh. Asachi (In Romanian), 2002.

[9] Floroian, D.: *Multiagent Systems*. Ed. Albastra, Cluj-Napoca, (In Romanian) 2009

[10] Lee, D., Chung, W.: *Discrete-status-based Localization for Indoor Service Robots*. In *IEEE Trans. Ind. Electron.* (2006) Vol. 53, No. 5, pp.1737-1746.