Proceedings of the 18th World Congress
The International Federation of Automatic Control
Milano (Italy) August 28 - September 2, 2011

Decentralized Coordination System for Multiple AGVs in a Structured Environment ⋆

Sergio Manca * **Adriano Fagiolini** ** **Lucia Pallottino** **

* *Sofidel S.p.A., Lucca, Italy.*
** *Centro "E. Piaggio", University of Pisa, Italy.*

Abstract: In this paper, we propose a decentralized coordination algorithm for safe and efficient management of a group of mobile robots following predefined paths in a dynamic industrial environment. The proposed algorithm is based on shared resources and proved to guarantee ordered traffic flows avoiding collisions and deadlocks. In consistency with the model of distributed robotic systems (DRS), no centralized mechanism, synchronized clock, shared memory or ground support is needed. A local inter-robot communication is required among a small number of spatially adjacent robotic units.

Keywords: Distributed control, autonomous vehicles, collision avoidance.

1. INTRODUCTION

In the existing literature, multi-vehicle coordination has been largely studied in the operational context of Autonomous Guided Vehicles (AGVs), that are typically used for industrial transportation. In autonomous decentralized manufacturing system (ADMS), the logistic system (based on AGVs) plays a central role. However, after deployment of a team of AGVs, management and coordination problems, such as the deadlock and collision avoidance, conflict resolution and shared resource negotiation arise naturally.

A team of robots can be coordinated using either a centralized or a decentralized control architecture, see e.g. Pallottino et al. (2007), LaValle (2006), Alami et al. (1998) and Lygeros et al. (1998). In the majority of industrial applications, a centralized architecture is currently in use, where a unique decision maker is responsible for solving e.g. motion planning and coordination problems. While all these approaches have the advantage to be complete, they are characterized by a significant computational burden and thus their use is limited to simple problem settings involving few vehicles. To improve a system's performance in terms of safety and efficiency, a continuous monitoring of environmental changes and generation of modified paths are inevitable. In Wu and Zhou (2007) and Fanti (2002) a technique, based on a Petri net, that avoids deadlocks through rerouting is presented. A method using the notion of composite robot is presented in Svestka and Overmars (1995). Another centralized approach, using *master-slave* control, is proposed in Yuta and Premvuti (1992). An approach based on the so called coordination diagram is proposed in O'Donnell and Lozano-Periz (1989) and Olmi et al. (2008). In LaValle and Hutchinson (1998) a coordination algorithm, which can be considered in between centralized and decoupled planning, are presented. In Guo and Parker (2002) a distributed route planning method for multiple mobile robots is proposed, that uses so-called Lagrangian decomposition technique. A framework for decentralized and parallel coordination system, based on dynamic assignment of robot motion priorities, is developed in Azarm and Schmidt (1997), but only the collision avoidance problem has been addressed. In Kato et al. (1992), a decentralized approach based on traffic rules has been proposed. In Wang and Premvuti (1995), the workspace is decomposed into discrete spatial resources and robots move on preplanned paths applying the concept of distributed mutual exclusion Lamport (1986 I and II) to coordinate their motions. The algorithms proposed in those works, however, require a communication and coordination data exchange among all robots and a significant computational effort.

Our goal is to provide a decentralized coordination algorithm for safe and efficient coordination of a multi-vehicle systems in an industrial automation environment. Given a group of AGVs with pre-assigned paths, we develop a fully decentralized coordination algorithm that, when executed by every robot, collectively allows multiple autonomous mobile robots to travel through a discrete traffic network. The network is composed of passage segments, intersection, and terminals that are considered as shared, discrete resources. The algorithm is proved to guarantee ordered traffic flow, in particular, the limited capacity of resources is always respected, no collision occurs at any intersection and deadlocks are avoided. A multi-vehicle system running this algorithm operates under hypothesis of no centralized mechanism, such as a centralized CPU, shared memory, or a synchronized clock is assumed. No ground support (such as an arbiter at each intersection) is employed.

The paper is organized as follows. The decentralized coordination problem for a group of AGVs moving within structured environments is presented in Section 2. The proposed solution is described in Sec. 3, while proofs of the properties of collision avoidance and deadlock freeness are given in Sec. 4. Finally, the effectiveness of the approach is shown in Sec. 5.

2. OVERVIEW OF THE PROBLEM

An industrial application often involves a number of AGVs delivering goods and material among workstations and storage pipes. The environment is completely known and structured, e.g. AGV are allowed to move along fixed routes.

The fact that the workspace is shared among AGVs and humans makes safety a prior requirement. Hence, to limit the AGVs workspace they usually follow predetermined, physical, or virtual guided paths, divided into a series of segments, i.e. single elements of the shuttles ways (see Fig. 1) that connect workstations to storages and vice-versa and that may intersect. To gain efficiency and flexibility on the factory floor, bidirectional paths can also be adopted. The choice of bidirectional paths allows an increased routing flexibility and space utilization with decreased delivery costs.

⋆ This work has been partially supported by CONET, the Cooperating Objects Network of Excellence, EC contract FP7-2007-2-224053, and CHAT - EC contract IST 224428, and PLANET, EC contract ICT-2009-2130.

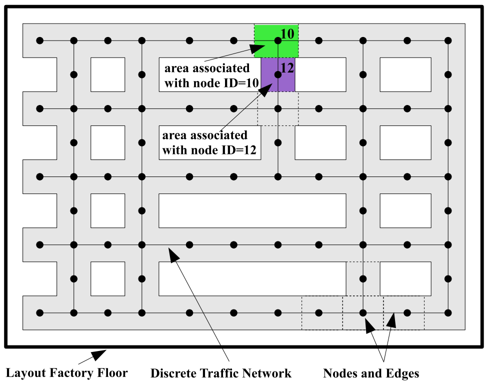


Fig. 1. Example of a layout of a factory floor

A natural way to represent such an environment makes use of a *path graph* $G_p = (H_p, E_p)$, where $H_p = \{\eta_1, \ldots, \eta_\varepsilon\}$ is a set of nodes representing segments and $E_p \subseteq H_p \times H_p$ is a set of edges representing adjacency relations between the segments. Hence, in such traffic networks, nodes can be considered as *resources* that must be shared and managed to ensure safety of the overall system. To ensure that collisions are avoided, the graph G_p is assumed to be such that the distance between any pair of nodes must be larger than the maximum size of agents working in the system.

For a generic AGV A_i, the desired path p_{A_i} is an ordered sequence of nodes that corresponds to a path of G_p, e.g. $p_{A_i} = \{\eta_{i,s}, \ldots, \eta_{i,f}\} \subset H_p$. Let $l_{A_i} = \{\eta_{i,c}, \ldots, \eta_{i,r}\} \subset p_{A_i}$, the configuration vector of A_i that represents the current node $\eta_{i,c}$ and the following desired nodes. The configuration vector consists of a variable number of nodes and it will be described in detail later in this section.

In a traffic network there are typically three typologies of encounters between pairs of vehicles characterized by the resources that needed to be shared (see Fig. 2), Olmi et al. (2008).

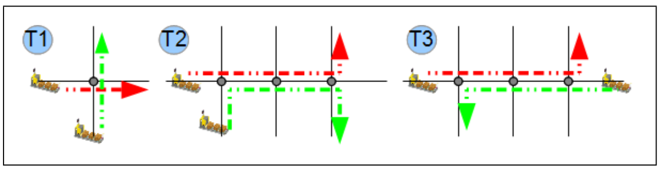


Fig. 2. Types of encounters: (T1) crossroad, (T2) follower, (T3) frontal

Let η be a sequence of ordered nodes, e.g. $\eta = \{\eta_1, \ldots, \eta_n\}$. We denote with $\overline{\eta} = \{\eta_n, \ldots, \eta_1\}$ the reverse sequence. Furthermore, let $\#\eta = n$ be the number of nodes in η.

The three encounter typologies (or sub-sequences of nodes) between A_i and A_j are characterized as follows:

(T1) *CROSSROAD*: if the only sub-sequence of nodes in both l_{A_i} and l_{A_j} consists in a single node;

(T2) *FOLLOWER*: if the sub-sequences of nodes in both l_{A_i} and l_{A_j} consists in a sequence of at least two nodes;

(T3) *FRONTAL*: if the sub-sequences of nodes in both l_{A_i} and $\overline{l}_{A_j}$ consists in a sequence of at least two nodes.

Notice that more complicated encounters consist of a combination of above mentioned typologies.

On one side, resources can be managed at node level, as in the work Wang and Premvuti (1995) where only the first desired node (*Micro Resource*) is contended between agents. This kind of approach is clearly not efficient in terms of optimizing the traffic flow and not able to avoid system deadlocks. On the other, resources can be managed as sets of nodes (*Macro Resources*), as e.g. in Olmi et al. (2008), where a set of desired nodes is contended and managed in order to avoid deadlocks a priori.

In our approach, the coordination system of each active agent within the logistic area will manage the access to shared resources at two levels:

- *MACRO LEVEL*: each agent competes for obtaining the right to access to its macro resource;
- *MICRO LEVEL*: once it has obtained access to a macro resource, each agent competes to use its individual parts, i.e. the micro resources.

The resources management at macro level will be used to coordinate agents in an efficient way and to ensure the absence of system deadlocks, whereas the resources management at the micro level will be used to avoid collisions.

More formally, every single node of A_i's path is a micro resource. Hence, for each agent, the set of micro resources represents the set of path nodes:

$$mR_{A_i} = p_{A_i} = \{\eta_{i,s}, \ldots, \eta_{i,f}\} \quad (1)$$

Definition 1. The micro resource $\eta_v \in mR_{A_i}$ of A_i is a *shared micro resource* with A_j if $\eta_v \in mR_{A_i} \cap mR_{A_j}$.

Definition 2. A *macro resource* $MR_i = \{\eta_{i,1}, \eta_{i,2}, \ldots, \eta_{i,d}\} \subset p_{A_i}$, of agent A_i, is an ordered sequence of future consecutive shared micro resources of its path.

Let $MR_{A_i} = \{MR_{i,1}, \ldots, MR_{i,R_i}\}$ be the set of R_i macro resources of agent A_i.

In a structured environment a macro resource is typically a corridor or a narrow passage.

Definition 3. A macro resource $MR_{i,k}$ with $k \in \{1, \cdots, R_i\}$ of A_i is *shared* with A_j if there exist $h \in \{1, \cdots, R_j\}$ and a micro resource $\eta_v \in MR_{i,k} \cap MR_{j,h}$.

Notice that a single macro resource can be shared (also partially) among multiple agents (Fig. 3).

Definition 4. Given the current and the next desired nodes $\eta_{i,c}$ and $\eta_{i,c+1}$, the configuration is $l_{A_i} = \{\eta_{i,c}, MR_{i,\kappa}\}$, if $\eta_{i,c+1}$ belongs to a shared resource $MR_{i,\kappa}$ in MR_{A_i} otherwise $l_{A_i} = \{\eta_{i,c}, \eta_{i,c+1}\}$.

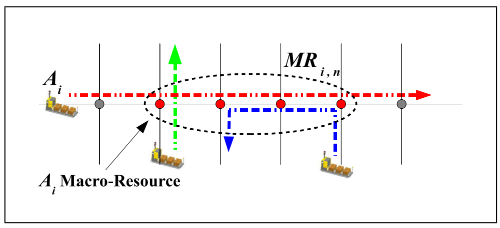


Fig. 3. Example of a macro resource shared partially with multiple AGV

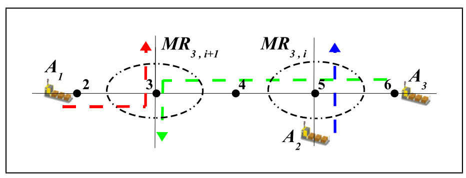


Fig. 4. Example of micro and macro resources shared with multiple AGV

An example of micro and macro resources and vector configuration is reported in Fig. 4. In this case, for agent A_3 we have:

$$\begin{cases} p_{A_3} = mR_{A_3} = \{\eta_{3,s},\ldots,6,5,4,3,9,\ldots,\eta_{3,f}\} \\ MR_{A_3} = \{\ldots,MR_{3,i},MR_{3,i+1},\ldots\} \rightarrow \begin{cases} MR_{3,i} = \{5\} \\ MR_{3,i+1} = \{3\} \end{cases} \\ l_{A_3} = \{\eta_{3,c},MR_{3,i}\} = \{6,5\} \end{cases}$$

In this framework, the AGV dynamics is not taken into account, and we assume that any AGV has an a priori preassigned path. Hence, in the proposed approach, any AGV knows the path graph G_p, its own path, and its set MR_{A_i} of shared Macro resources.

The AGVs system operates under the model of DRS. Assuming no centralized decision maker, inter-robot communication for agents coordination is only required among neighboring robotic units. Inter-robot communication is based on the model of *sign-board* (SB), Wang (1994) and Wang and Premvuti (1994). A sign-board is a conceptual displaying device on-board each robot. A message can be posted on a sign-board only by the robot itself, but can be read by all robots in its neighborhood whenever it is needed. The model of the sign-board has been chosen because is a fully distributed model.

A message displayed on the sign-board represents the current agent's state and consists of static (e.g. AGV identification number) and dynamic fields (e.g. AGV currently occupied node) needed for cooperation.

The current implementation of the sign-board, used for the developed coordination system, consists of 20 fields (see following table). A larger number of fields can be used to take into account of a larger number of nodes in the Macro resources.

FIELDS	DATA
F1	ID: identification AGV number
F2	Priority: number representing the AGV level of priority
F3 and F4	Current node: position of the current node in the workspace
F5	Agent speed: the current value of the AGV velocity
F6	Macro-state: label identifying the state of request/ownership of the Macro resource intended to be used
F7	Node ID: identification number of AGV current node
F8	Macro resource size: number of nodes in the Macro resource that the agent intend to use
F9,...,F18	Macro resource ID: identification number of nodes of the Macro resource intended to be used
F19	Micro-state: label identifying the state of request/ownership of Micro resource intended to be used
F20	Micro-Resource ID: identification number of Micro resource intended to be used

3. THE PROPOSED COORDINATION SYSTEM

The behavior of the coordination system, on each agent, can be divided into several cyclic steps: Check for shared nodes path, Communication with neighbours (by reading the neighbours' sign-boards), Priority group creation, Check Macro/Micro resources shared with neighbours, Competition for resources, Access and speed management, Use of resources and update of the sign-board.

A detailed and exhaustive description of the algorithm is not possible for space limitations. In this section, we will provide a qualitative description of main parts of the coordination algorithm. Initializing the system, each AGV sets the configuration vector to the current state, i.e. $l_{A_i} = \{\eta_{\mathrm{i,c}}\}$.

Check for shared nodes path

Given the current AGV position $\eta_{i,c}$ and l_{A_i}, the coordination system simply monitors nodes in $l_{A_i} \setminus \{\eta_{i,c}\}$, and specifically whether they belong to an AGV Macro resource or not, that is if they are shared with some other agent. If this is the case, fields from F8 to F18 are updated while field F20 is updated in any case. If $\eta_{i,c+1}$ belongs to a Macro resource, the agent updates fields F6 and F19 showing its intention to use the resource. For example, referring to Fig. 4 AGV 1 updates F8= 1, F9= 3 and F20= 3. F6 is then updated to *request* while F19 is not updated since the Micro resource will be requested only when the Macro resource it belongs to has been owned.

Priority group creation

The competition algorithm is priority-based. We use a dynamic priority scheme based on three values:

- The size of Macro resource that the generic agent wants to use or it is already using (dMR =F8 and it is zero if there are no Macro resources);
- The maximum speed v_{max} of agent;
- The identification number of the agent ID=F1.

For any AGV, the priority scheme generates two groups $P_{i,high}$ and $P_{i,low}$ of agents at higher and lower priorities with respect to A_i priority value F2, respectively. Whenever A_i read A_j sign-board, the scheme sets higher priority to the agent that is going to use its Macro resource for the shortest time. Indeed, let $P_i = v_{i,max}/dMR_i$ this value is stored in F2 and updated when the dimension of the desired Macro resources is changed. We set $P_{i,high} = \{A_j | P_j > P_i\} \cup \{A_j | P_j = P_i \wedge ID_j < ID_i\}$ and $P_{i,low} = \{A_j | P_j < P_i\} \cup \{A_j | P_j = P_i \wedge ID_j > ID_i\}$.

Taking into account other information as the mission priority level, AGVs battery level, etc. it is possible to extend the priority scheme to more complex behaviours.

Check Macro/Micro resources shared with neighbours

If nodes in F9-F18 and F20, are shared with some neighbours, the type of encounter (i.e, CROSSROAD, FOLLOWER or FRONTAL) is checked for Macro resources.

The check output γ, for both Macro and Micro levels, is a variable whose value depends on the desired resource structure and it specifies the need of competition between two agents. For agents A_i and A_j, $\gamma_{M_{i,j}} = 0$ if the Macro resource is not shared, $\gamma_{M_{i,j}} = 1$ if the Macro resource is shared but the competition can be solved at Micro level (such as between A_1 and A_3 in fig.4) and $\gamma_{M_{i,j}} = 2$ when the competition is at Macro level (the entire resources will be assigned to single agent, e.g. FRONTAL encounter in fig.2).

At Micro level, $\gamma_{m_{i,j}} = 2$ if the Micro resource has already been won by an agent, $\gamma_{m_{i,j}} = 1$ if the Micro resource is shared and $\gamma_{m_{i,j}} = 0$ if the Micro resource is not shared.

Competition for resources

The competition for resources at the Micro level only happens after the Macro resource has been owned (see fig 5). However for safety reason, agents compete for all singular nodes (Micro resources) of their path also if they are not shared.

Recall that the goal of coordination is to manage access to shared resources avoiding collisions and deadlock. This can be achieved defining appropriate competition rules common to all agents and based on agents states, based on the approach proposed in **?** for other cooperative systems.

Let Macro and Micro-state of agents (reported in the sign-board in F6 and F19) be as follows:

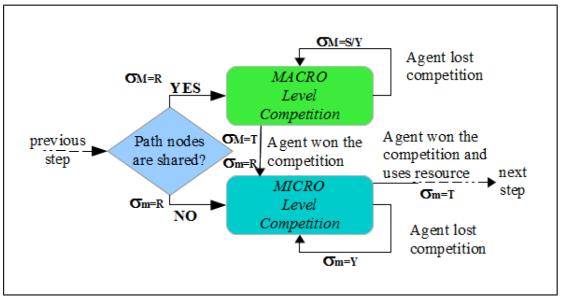


Fig. 5. Competition for resources scheme

F6 $= \sigma_{i,M} \in \Sigma_M \equiv$ {NONE , STOP , YIELD , REQUEST , TAKEN}
F19 $= \sigma_{i,m} \in \Sigma_m \equiv$ {NONE , YIELD , REQUEST , TAKEN}

where $\sigma_i = N$ (NONE) when A_i is not interested in the resource, $\sigma_i = S$ or Y (STOP/YIELD) when A_i waits for the desired resource to become available, $\sigma_i = R$ (REQUEST) when A_i is interested in accessing the desired resource, $\sigma_i = T$ (TAKEN) when A_i has right to access to (hence own) the desired resource.

The abovementioned rules describe how events based on the states of neighbouring AGVs let the vehicle change its own state. In particular, the generic event can be assigned with a logical variable $e_{i,l} \in \{\text{TRUE,FALSE}\}$ depending on the vector configurations (l_{A_i}) and states $\sigma_{i,M}, \sigma_{i,m}$ of neighbouring AGVs involved in the competition.

The competition sub-systems (Fig. 5) can be seen as Discrete Event Systems (DES), and indeed Finite State Machines (FSMs) describe the dynamic evolution of the agent state variables and of the competition at Macro and Micro level.

Access and speed management

This module, at both Macro and Micro levels, manages the speed of the agents. The states of agents concerning competition for Macro and Micro resources are translated into low level control law for the agent speed.

Speed in the states NONE and REQUEST is not modified, in TAKEN is set to maximum whereas in the states YIELD and STOP agents slow down in order to allow other agents, which are using or owning the resources, to transit and release them. Thus, agent speed v_{A_i} is a function of distance $D_{\eta_{i,c+1}}$ from next node and time T_r required by other agent to use the shared resource.

Use of resources and update of the sign-board

Each agent, once gained access to a resource, releases the previous resource making it available to other agents (Fig. 6).

4. COLLISION AVOIDANCE AND DEADLOCK FREE PROPERTIES

Distributed Mutual Exclusion

The mutual exclusion access to resources is a fundamental problem both at micro and macro level.

Consider a scenario with n agents competing for a shared micro resource. We must guarantee that agents, based on exchanged information (*Sign-Board*), access to the resource in a mutual exclusion way, so that based on node spatial separation assumption, collisions are avoided. At any time and for the same micro resource, no more than one agent can have $\sigma_m = \text{T}$.

To prove the *mutual exclusion access*, consider for simplicity two agents, A_i and A_j, that compete for the same free micro resource. Hence, suppose that $\exists \eta_v \in mR_{A_i} \cap mR_{A_j}$ s.t.:

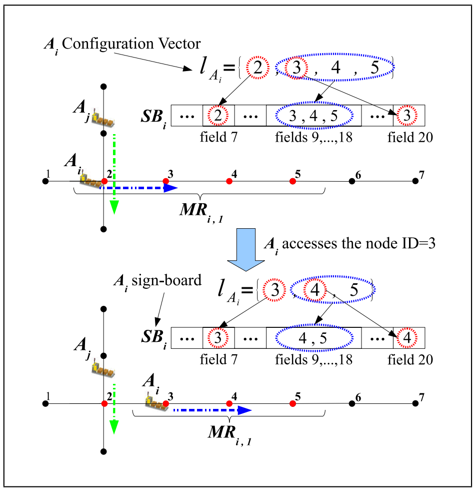


Fig. 6. Access to new resource, release of the used one and sign-board update

- $\eta_v = \eta_{i,c+1} = \eta_{j,c+1}$ (the resource is the next node for both AGVs),
- $\gamma_{m_{i,j}} = 1$ (competition for the resource is needed),
- $\nexists A_k \,|\, \eta_v = \eta_{k,c} \vee (\eta_v = \eta_{k,c+1} \wedge \sigma_{k,m} = \text{T})$ (the resource is not occupied or owned by other AGVs).

Based on the competition rules, for each agent, the transitions of σ_m to TAKEN, at generic time t_k, is possible only if $\sigma_m = \text{R}$ and the condition $e = \text{TRUE}$ (that represent the condition that the micro resource is free and either there is no other agents interested in the resource or all agents have already granted the permission to use the resource) at previous time t_{k-1}, i.e.

$$\forall A_i, \sigma_{i,m}(t_k) = \text{T} \Rightarrow \exists t_{k-1} < t_k | e_i = \text{TRUE} \wedge \sigma_{i,m}(t_{k-1}) = \text{R}. \qquad (2)$$

Furthermore, if at generic time t_k, a specific resource is already occupied or taken by an agent, then no other agent can access to the resource (i.e. the competition does not start), i.e.

$$\forall A_i\, A_j, \sigma_{i,m}(t_k) = \text{T} \wedge \sigma_{j,m}(t_k) \neq \text{T} \Rightarrow \nexists t > t_k | \sigma_{i,m}(t) = \sigma_{j,m}(t) = \text{T}. \qquad (3)$$

Therefore, to ensure the mutual exclusion access, it is sufficient to prove that is not possible that two agents win the resource access competition simultaneously (Fig. 7(a)). From Eq. (2), the following logical implication holds

$$\exists t_k | \sigma_{i,m}(t_k) = \sigma_{j,m}(t_k) = \text{T} \Rightarrow \exists t_{k-1} | \sigma_{i,m}(t_{k-1}) = \sigma_{j,m}(t_{k-1}) = \text{R} \wedge e_i(t_{k-1}) = e_j(t_{k-1}) = \text{TRUE}.$$

Hence, to prove mutual exclusion access, it is sufficient to prove that the latter logical condition is false.

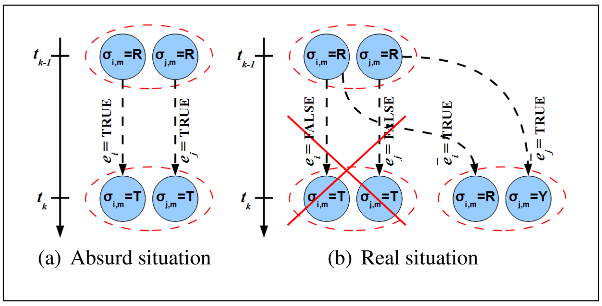


Fig. 7. Dynamic evolution of micro state σ_m during a competition among A_i, A_j for access, in a mutual exclusion way, to shared micro resource

Observing the competition from the A_i's standpoint, if $e_i(t_{k-1}) =$ TRUE and $\sigma_{i,m}(t_{k-1}) = \text{R}$ from the competition rules two cases are possible:

$$A_j \in P_{i,low} \wedge e_i(t_{k-1}) = \text{TRUE} \Rightarrow \sigma_{j,m}(t_{k-1}) = \text{Y} \vee \text{N} \neq \text{R}$$
$$A_j \in P_{i,high} \wedge e_i(t_{k-1}) = \text{TRUE} \Rightarrow \sigma_{j,m}(t_{k-1}) = \text{N} \neq \text{R}.$$

The same holds for A_j. Hence:

$$\sigma_{i,m}(t_{k-1}) = \sigma_{j,m}(t_{k-1}) = \text{R} \Rightarrow e_i(t_{k-1}) = e_j(t_{k-1}) = \text{FALSE}$$

and the simultaneous access to the shared resource is impossible. A possible real situation is depicted in Fig. 7(b).

The same reasoning can be easily extended to n agents $A = \{A_1, \dots, A_n\}$ competing for the same micro resource. In fact assumptions (2) and (3) are still valid, thus the following logical implication still holds:

$$\exists t_k | \sigma_{1,m}(t_k) = \dots = \sigma_{n,m}(t_k) = \text{T}$$
$$\Rightarrow \exists t_{k-1} | \sigma_{1,m}(t_{k-1}) = \dots = \sigma_{n,m}(t_{k-1}) = \text{R}$$
$$\wedge e_1(t_{k-1}) = \dots = e_n(t_{k-1}) = \text{TRUE}.$$

However, analyzing the competition from each agent standpoint, it is possible to prove that the former logical condition is false and hence the mutual exclusion access is guaranteed.

Let A_i, A_j be in competition for a FRONTAL type shared macro resource (Fig. 8). The proof of mutual exclusion access at macro level follows straightforwardly from the proof of micro level access.

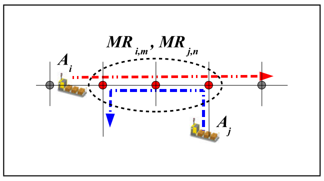


Fig. 8. Example of Deadlock with FRONTAL type macro resource

Deadlock Avoidance

A *deadlock* is a situation wherein two or more competing actions are waiting for the other to finish, and thus neither ever does. In our specific case, it is a situation wherein an agent group form a circular chain, where each agent waits for a resource that the next agent in the chain holds.

The information necessary to identify deadlocks can be obtained directly from the paths graph. From the undirected graph G_p, we can extract a directed graph $G_o = (H_o, E_o)$ (Motion Graph) with $H_o = \bigcup_i \{\eta_{i,c}, \eta_{i,c+1}\}$ and $E_o = \bigcup_i \{(\eta_{i,c}, \eta_{i,c+1})\}$ based on current and next node of any agent.

Thus, from the analysis of G_o we can conclude that the system is deadlock–free if any directed sub-graph of G_o is cycle-free, see Fig. 9.

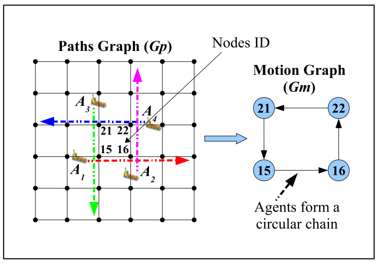


Fig. 9. Example of Deadlock

In our system, the deadlock is managed at Macro level and avoided with the use of the following constraint.

Consider a generic agent A_i that is interested in a macro resource $MR_{i,k}$ consisting of at least two nodes, i.e.

$$MR_{i,k} | \#MR_{i,k} \geq 2. \quad (4)$$

If there exist at least two agents A_j, A_r with multiple access shared resources with A_i and s.t. also their macro resources consist of at least two nodes, i.e.

$$\begin{array}{l} \exists A_j \mid \#MR_{j,s} \geq 2 \wedge MR_{j,s} \cap MR_{i,k} \neq \emptyset \wedge \gamma_{M_{i,j}} = 1, \\ \exists A_r \mid \#MR_{r,h} \geq 2 \wedge MR_{r,h} \cap MR_{i,k} \neq \emptyset \wedge \gamma_{M_{i,r}} = 1, \end{array} \quad (5)$$

then A_i can have access to the resource if at most one agent takes the resource and the other agents give to A_i the permission to use it.

This constraint prevents from forming directed cycle sub-graph of G_o and thus deadlocks. De facto if n agents, with $n > 2$, form a circular chain there exists at least one agent for which at previous time, constraint conditions (4) and (5) hold but it did not respected them.

5. SIMULATIONS

The proposed algorithm has been tested on a large number of agents (n=100), for the sake of space and clarity we report an example of coordination with only 4 AGVs.

However, to give an idea, we will show the execution time of the algorithm considering up to 100 agents involved in a competition. The execution time considered is, of course, only an indication, depending itself from the hardware and software available during the simulation. Nevertheless, for n=4, 60 and 100 agents that compete for the same resource, the Fig. 10 shows that the algorithm execution time does not increase significantly with the number of AGVs.

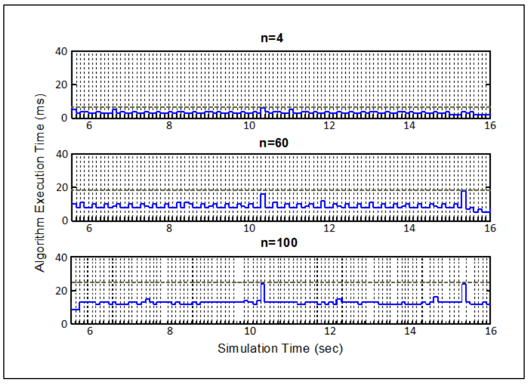


Fig. 10. Execution time of competition algorithm for $n = 4, 60, 100$

Fig. 11 and Fig. 12 illustrate an example of system coordination behavior. Simulation parameters such as AGV speed (2m/s) and dimension and nodes distances (10m) come from the real industrial environment of Sofidel S.p.A. plant in Germany.

In Fig. 11 the paths for the 4-AGVs system are reported. Macro, Micro-States and the AGV speeds during the coordination system algorithm are reported in Fig. 12.

For the example considered, during the competition phase, the AGVs priorities are A_1, A_2, A_3 and A_4.

6. CONCLUSIONS AND FUTURE WORK

A coordination algorithm for a multiple AGVs system that guarantees deadlock, blocking and collision avoidance has been developed. The coordination system is decentralized, hence inter-robot communication is only required among spatially adjacent robotic units. Thus the proposed coordination system

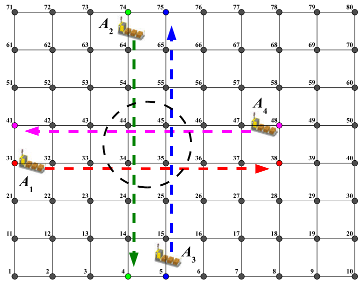
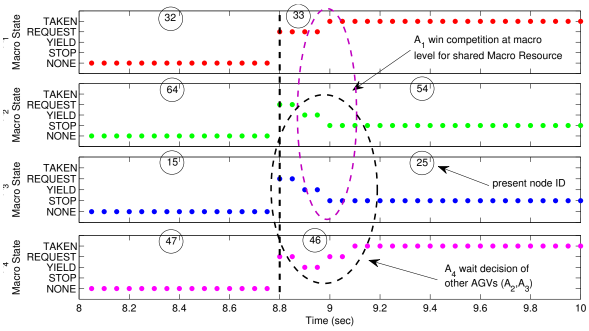
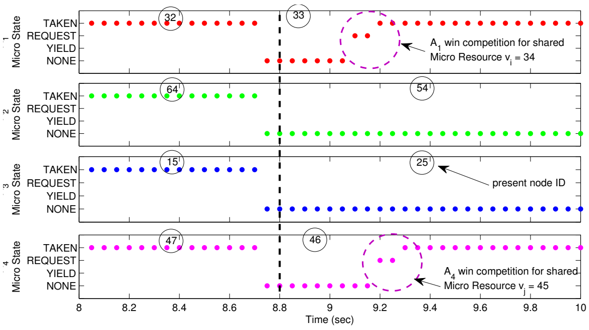


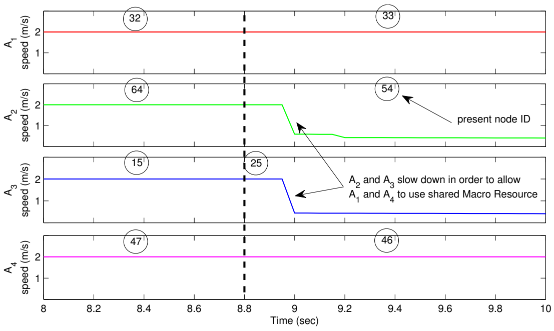
Fig. 11. Example of paths with shared critical resources



(a) Macro-State



(b) Micro-State



(c) AGVs' speeds

Fig. 12. Macro/Micro state and AGVs' speeds during competition. Every agent require access to the macro resources simultaneously, but they are finally accessed exclusively.

is scalable to a large number of AGVs. Future work will concern the integration of the proposed coordination system with a decentralized motion planning algorithm, that increases performance and maintains the safety of the overall system.

7. ACKNOWLEDGMENTS

Authors wish to thank Fabrizio Lapucci and Leonardo Rocchi working with Sofidel S.p.A. for data provided for simulations.

REFERENCES

Alami, R., Fleury, S., Herrb, M., Ingrand, F., and Robert, F. (1998). Multi-robot cooperation in the martha project. *Robotics Automation Magazine, IEEE*, 5(1), 36–47.

Azarm, K. and Schmidt, G. (1997). Conflict-free motion of multiple mobile robots based on decentralized motion planning and negotiation. *Proceedings of IEEE International Conference on Robotics and Automation*, 4, 3526–3533.

Fanti, M. (2002). A deadlock avoidance strategy for agv systems modelled by coloured petri nets. *Proc. of 6th International Workshop on Discrete Event Systems*, 61–66.

Guo, Y. and Parker, L. (2002). A distributed and optimal motion planning approach for multiple mobile robots. *Proceedings of IEEE International Conference on Robotics and Automation*, 3, 2612–2619.

Kato, S., Nishiyama, S., and Takeno, J. (1992). Coordinating mobile robots by applying traffic rules. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3, 1535–1541.

Lamport, L. (1986 I and II). The mutual exclusion problem: part i—a theory of interprocess communication and partii—statement and solutions. *J. ACM*, 33(2), 313–348.

LaValle, S.M. (2006). *Planning Algorithms*. Cambridge University Press, Cambridge, U.K.

LaValle, S. and Hutchinson, S. (1998). Optimal motion planning for multiple robots having independent goals. *IEEE Transactions on Robotics and Automation*, 14(6), 912–925.

Lygeros, J., Godbole, D., and Sastry, S. (1998). Verified hybrid controllers for automated vehicles. *IEEE Transactions on Automatic Control*, 43(4), 522–539.

O'Donnell, P. and Lozano-Periz, T. (1989). Deadlock-free and collision-free coordination of two robot manipulators. *Proceedings of IEEE International Conference on Robotics and Automation*, 1, 484–489.

Olmi, R., Secchi, C., and Fantuzzi, C. (2008). Coordination of multiple agvs in an industrial application. *Proceedings of IEEE International Conference on Robotics and Automation*, 1916–1921.

Pallottino, L., Scordio, V.G., Frazzoli, E., and Bicchi, A. (2007). Decentralized cooperative policy for conflict resolution in multi-vehicle systems. *IEEE Transaction on Robotics*, 23(6), 1170–1183.

Svestka, P. and Overmars, M. (1995). Coordinated motion planning for multiple car-like robots using probabilistic roadmaps. *Proceedings of IEEE International Conference on Robotics and Automation*, 2, 1631–1636.

Wang, J. (1994). On sign-board based inter-robot communication in distributed robotic systems. *Proc. of IEEE Intl. Conference on Robotics and Automation*, 2, 1045–1050.

Wang, J. and Premvuti, S. (1994). Resource sharing in distributed robotic systems based on a wireless medium access protocol (csma/cd-w). *Proc. of the IEEE/RSJ/GI Intl. Conference on Intelligent Robots and Systems*, 2, 784–791.

Wang, J. and Premvuti, S. (1995). Distributed traffic regulation and control for multiple autonomous mobile robots operating in discrete space. *Proceedings of IEEE International Conference on Robotics and Automation*, 2, 1619–1624.

Wu, N. and Zhou, M. (2007). Shortest routing of bidirectional automated guided vehicles avoiding deadlock and blocking. *IEEE/ASME Transactions on Mechatronics*, 12(1), 63–72.

Yuta, S. and Premvuti, S. (1992). Coordinating autonomous and centralized decision making to achieve cooperative behaviors between multiple mobile robots. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3, 1566–1574.